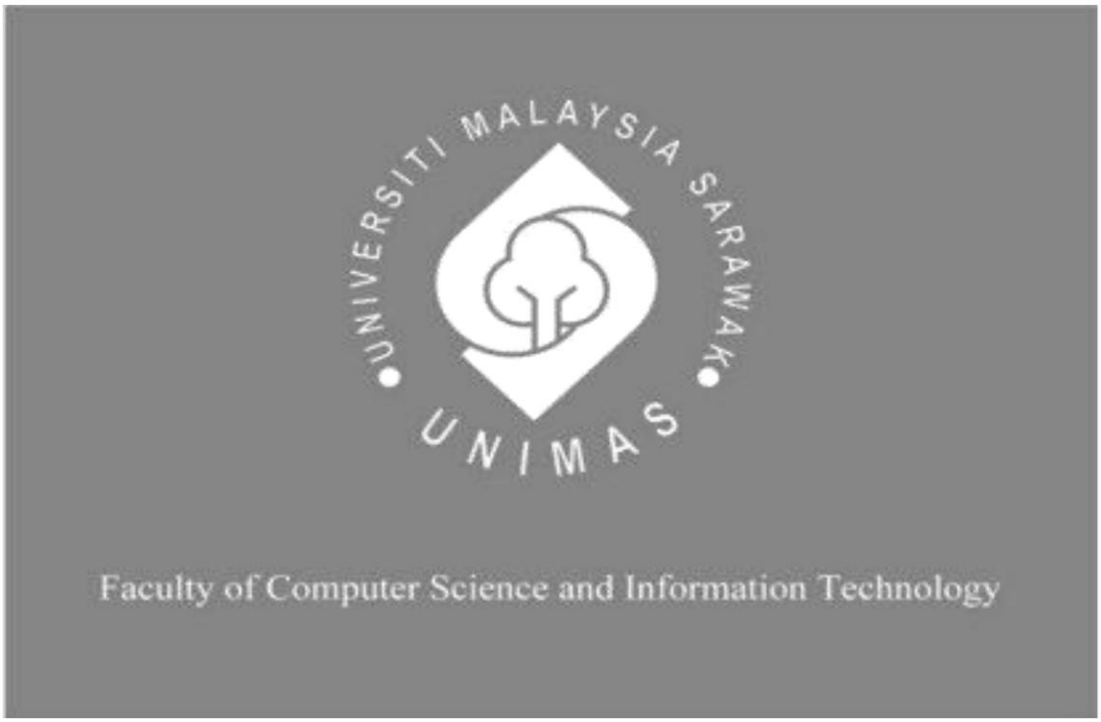



# *FindIt: Simplifying Campus Item Recovery with AI-Driven Lost and Found App*

Ngu Keh Cong

Bachelor of Software Engineering with Honours

2025

# FindIt: Simplifying Campus Item Recovery with AI-Driven Lost and Found App

Ngu Keh Cong

This project is submitted in partial fulfilment of the requirements for the degree of Bachelor of Software Engineering with Honours

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2025

# FindIt: Mempermudah Pemulihan Barang Hilang di Kampus dengan Aplikasi Kehilangan dan Penemuan Dipacu AI

Ngu Keh Cong

Projek ini merupakan salah satu keperluan untuk Ijazah Sarjana Muda
Kejuruteraan Perisian dengan Kepujian

Fakulti Sains Komputer dan Teknologi Maklumat

UNIVERSITI MALAYSIA SARAWAK

2025

# DECLARATION

I hereby declare that the project is my original work. I have not copied from any other student's work or from any other sources except due to reference or acknowledgement is not made explicitly in the text, nor has any part had been written for me by another person.

_____

(Ngu Keh Cong, 80369) 17th January 2025

Faculty of Computer Science and

Information Technology

Universiti Malaysia Sarawak

## ACKNOWLEDGEMENT

I would like to express my deepest gratitude to everyone who supported and contributed to the successful completion of this Final Year Project (FYP). Firstly, my heartfelt thanks go to my beloved supervisor, Dr. Wang Hui Hui for giving me the opportunity to work under her supervision. Her invaluable guidance and insightful feedback had been giving me constant support throughout the FYP. I would also like to thank my FYP examiner, Dr. Sze San Nah for her constructive suggestions on my FYP, which greatly increase the project's quality. Furthermore, a special thanks go to FYP coordinator, Prof. Wang Yin Chai for his patience in giving guidelines of FYP, and answering students' questions to ensure a smooth process throughout the FYP journey. Next, I am sincerely grateful to my academic institution, Universiti Malaysia Sarawak (UNIMAS) because it provides the necessary resources, platform, and conducive environment for the students to complete the project. I would also like to acknowledge all the lecturers who have contributed to my academic growth. Furthermore, I am also grateful to my family and friends for their continuous support and tolerance throughout the project. I have found the source of strength and motivation from your moral support. Lastly, I would like to thank everyone who has indirectly contributed during this period. Your support has been invaluable and deeply appreciated.

# ABSTRACT

This project presents "FindIt," an AI-driven mobile and web-based application designed to address inefficiencies in item recovery within university campuses. The system automates the matching of lost and found items by leveraging image matching technology, which can significantly reduce manual effort of both admin and students, and improving recovery rates. Waterfall model is applied in this project to guide development, and incorporating systematic phases such as requirements gathering, design, implementation, testing, and maintenance. To understand the critical issues with existing informal channels such as fragmented reporting processes and lack of timely update, and the suitable features to be included in the project, user feedback is collected via surveys in Google Form. FindIt is designed to integrate a robust feature set tailored for university students, likes AI-based image matching, real-time notifications, and status tracking of reported items are included in the mobile application. An administrator-focused web interface is also designed to ensure an efficient management of claims and user accounts. The system's effectiveness was evaluated through functional and usability testing to highlights significant improvements in item recovery efficiency, user satisfaction, and system reliability. This project contributes to the application of AI in practical, particularly in improving campus experiences.

# ABSTRAK

Projek ini memperkenalkan "FindIt," sebuah aplikasi mudah alih dan berasaskan web yang dipacu oleh AI, direka untuk menangani ketidakefisienan dalam pemulihan barang hilang di kampus universiti. Sistem ini mengautomatikkan padanan barang hilang dan ditemui dengan memanfaatkan teknologi pemadanan imej, yang dapat mengurangkan usaha manual oleh pentadbir dan pelajar, serta meningkatkan kadar pemulihan. Model Air Terjun digunakan dalam projek ini untuk membimbing pembangunan, dan menggabungkan fasa sistematik seperti pengumpulan keperluan, reka bentuk, pelaksanaan, ujian, dan penyelenggaraan. Untuk memahami isu kritikal dengan saluran tidak rasmi sedia ada seperti proses pelaporan yang berpecah dan kekurangan kemas kini tepat pada masanya, maklum balas pengguna dikumpulkan melalui tinjauan di Google Form. FindIt direka untuk mengintegrasikan set ciri yang kukuh disesuaikan untuk pelajar universiti, seperti padanan imej berasaskan AI, pemberitahuan masa nyata, dan penjejakan status barang yang dilaporkan dalam aplikasi mudah alih. Antara muka web yang berfokuskan pentadbir juga direka untuk memastikan pengurusan tuntutan dan akaun pengguna yang cekap. Keberkesanan sistem ini dinilai melalui ujian fungsian dan kebolehgunaan untuk menekankan peningkatan ketara dalam kecekapan pemulihan barang, kepuasan pengguna, dan kebolehpercayaan sistem. Projek ini menyumbang kepada aplikasi AI dalam amalan, terutamanya dalam meningkatkan pengalaman kampus.

# TABLE OF CONTENTS

DECLARATION ..... i
ACKNOWLEDGEMENT ..... ii
ABSTRACT ..... iii
ABSTRAK ..... iv
TABLE OF CONTENTS ..... v
LIST OF TABLES ..... viii
LIST OF FIGURES ..... x
Chapter 1: Introduction ..... 1
1.1 Introduction ..... 1
1.2 Problem Statement ..... 2
1.3 Project Scope ..... 2
1.4 Aims and Objectives ..... 4
1.5 Brief Methodology ..... 4
1.6 Significance of project ..... 6
1.7 Project Schedule ..... 7
1.8 Expected outcome/Project outcome ..... 7
1.9 Project Outline ..... 7
Chapter 2: Literature Review ..... 9
2.1 Introduction ..... 9
2.2 Review on Similar Existing Systems ..... 9
2.2.1 Overview ..... 9
2.2.2 Lost and Found ..... 9
2.2.3 Lost'NFound ..... 16
2.2.4 Lost and Found Network ..... 22
2.3 Comparison of the Features in Existing Systems and Proposed System 28
2.4 Review on Technologies and Tools used in Development ..... 31
2.4.1 Laravel ..... 31
2.4.2 Bootstrap ..... 31
2.4.3 Flutter ..... 31
2.4.4 Laragon ..... 32
2.4.5 Visual Studio Code ..... 32
2.4.6 PyTorch ..... 33
2.4.7 FastAPI ..... 33
2.4.8 Android Studio ..... 34
2.5 Review on AI Models and Image Similarity Technologies ..... 34
2.5.1 Justifications for Integrating DINOv2 Model for Image Similarity Task ..... 34
2.6 Conclusion ..... 35
Chapter 3: System Analysis and Design ..... 37
3.1 Introduction ..... 37
3.2 Requirement Analysis Phase ..... 38
3.2.1 Data Analysis ..... 38
3.2.1.1 Analysis of Demographic ..... 39
3.2.1.2 Analysis on Reasons of Developing the Proposed System 41

3.2.1.3 Analysis on Proposed System's Functionalities and Feedback 47
3.2.2 System Requirement Analysis ..... 50
3.2.2.1 Functional Requirements ..... 51
3.2.2.1.1 Use Case Diagram ..... 51
3.2.2.1.2 Use Case Specification ..... 51
3.2.2.1.3 Functional Requirements ..... 63
3.3 System Design Phase ..... 65
3.3.1 Overall System Architectural Design ..... 65
3.3.2 Activity Diagram ..... 67
3.3.3 Class Diagram ..... 81
3.3.4 Wireframe ..... 86
3.3.4.1 Student User Interface ..... 86
3.3.4.2 Admin's User Interface ..... 114
3.4 Implementation Phase ..... 121
3.5 Verification Phase ..... 122
3.6 Maintenance Phase ..... 122
3.7 Conclusion ..... 122
Chapter 4: Implementation ..... 124
4.1 Introduction ..... 124
4.2 Development Tools ..... 124
4.2.1 Laravel ..... 124
4.2.2 Bootstrap ..... 124
4.2.3 Flutter ..... 125
4.2.4 Laragon ..... 125
4.2.5 Visual Studio Code ..... 126
4.2.6 PyTorch ..... 126
4.2.7 FastAPI ..... 127
4.2.8 Android Studio ..... 127
4.2.9 Hugging Face ..... 128
4.3 System Implementation ..... 128
4.3.1 Student ..... 129
4.3.1.1 Student Registration ..... 129
4.3.1.2 Email Verification ..... 131
4.3.1.3 Student Login ..... 135
4.3.1.4 Student Logout ..... 136
4.3.1.5 Forget Password ..... 137
4.3.1.6 Change Password ..... 141
4.3.1.7 Report Lost/ Found Item ..... 142
4.3.1.8 Manage Reported Item ..... 144
4.3.1.9 Search and Filter Item ..... 146
4.3.1.10 Claim Items without Report Item Lost ..... 150
4.3.1.11 Claims Possible Matches for the Lost Item Reported ..... 151
4.3.1.12 Track Claim Status ..... 154
4.3.1.13 Notifications ..... 156
4.3.2 Admin ..... 157
4.3.2.1 Admin Login ..... 158

4.3.2.2 Reset Password ..... 159
4.3.2.3 Admin Logout ..... 163
4.3.2.4 Side Menu ..... 163
4.3.2.5 Dashboard ..... 164
4.3.2.6 View Student Information ..... 165
4.3.2.7 Manage Item Characteristics ..... 166
4.3.2.8 Process Claim Request ..... 168
Chapter 5: Testing ..... 174
5.1 Introduction ..... 174
5.2 Functional Testing ..... 174
5.2.1 Authentication Module ..... 175
5.2.2 Item Management Module ..... 189
5.2.3 Item Search and Discovery Module ..... 204
5.2.4 Claim Management Module ..... 210
5.2.5 Student Profile Management Module ..... 222
5.2.6 Admin Dashboard Module ..... 224
5.3 Usability Testing ..... 224
5.3.1 Student Questionnaire Results ..... 225
5.3.2 Admin Questionnaire Results ..... 230
Chapter 6: Conclusion ..... 238
6.1 Introduction ..... 238
6.2 Project Achievements ..... 238
6.3 Limitations and Constraints ..... 239
6.4 Conclusion ..... 240
6.5 Future Work ..... 241
References ..... 242
Appendix A ..... 246
Appendix B ..... 249

## LIST OF TABLES

Table 2.1: Comparison of the Features of Three Existing Systems and Proposed System. ....28
Table 3.1: Use Case Specification for Register. .... 51
Table 3.2: Use Case Specification for Login. ....52
Table 3.3: Use Case Specification for Logout. ....53
Table 3.4: Use Case Specification for Managing Student Profile (Student). ..... 54
Table 3.5: Use Case Specification for Managing Student Profile (Admin). ..... 54
Table 3.6: Use Case Specification for Report Lost/ Found Item. ....54
Table 3.7: Use Case Specification for Receiving Notifications on Potential Matches between Lost and Found Items .... 55
Table 3.8: Use Case Specification for Viewing Reported Items. .... 56
Table 3.9: Use Case Specification for Editing Reported Items. .... 57
Table 3.10: Use Case Specification for Deleting Reported Items .... 57
Table 3.11: Use Case Specification for Browsing and Searching Items. .... 58
Table 3.12: Use Case Specification for Submitting Claim Request. .... 59
Table 3.13: Use Case Specification for Tracking Claim Status. .... 60
Table 3.14: Use Case Specification for Processing Claim Request. .... 61
Table 3.15: Use Case Specification for Viewing Claim Approval History. ....62
Table 3.16: Use Case Specification for Managing Item Characteristics. .... 62
Table 3.17: Functional Requirements. ....63
Table 5.1 Student register .... 175
Table 5.2 Student login with verified email .... 177
Table 5.3 Student verify email address ....178
Table 5.4 Student check verification status to login ....179
Table 5.5 Student request new verification email .... 180
Table 5.6 Student logout ....181
Table 5.7 Student receive password reset email ....181
Table 5.8 Student and Admin reset password before login successfully .... 182
Table 5.9 Logged-in student reset password .... 184
Table 5.10 Admin login ....187
Table 5.11 Admin receive password reset email .... 188
Table 5.12 Student report new lost / found item .... 189
Table 5.13 Student view own reported lost / found items details ....192
Table 5.14 Student edit own reported lost / found item details .... 193
Table 5.15 Student delete own reported lost / found item details .... 195
Table 5.16 Admin create new item characteristics instance ....196
Table 5.17 Admin view item characteristics .... 199
Table 5.18 Admin edit item characteristics .... 201
Table 5.19 Admin delete item characterstics ....203
Table 5.20 Student search items .... 204
Table 5.21 Student clear filter / search keywords ....206
Table 5.22 Student view item details ....207
Table 5.23 Student submits claim request for a found item without reporting item as lost .... 210
Table 5.24 Student submits claim request for a matched found item related to own reported lost item .... 211
Table 5.25 Student track claim request status of a matched found item .... 213
Table 5.26 Student track status of all claim requests ....214

Table 5.27 Admin compare items' information in a claim request ..... 215
Table 5.28 Admin process claim request .....217
Table 5.29 Admin manage previously processed claims .....221
Table 5.30 Student manage profile .....222
Table 5.31 Admin view student profile information ..... 223
Table 5.32 Admin view charts in dashboard ..... 224
Table 6.1 Project objectives and achievements ..... 238

## LIST OF FIGURES

Figure 1.1: Phases in Waterfall Model (Hoory & Bottorff, 2024). ..... 5
Figure 2.1: Registration. ..... 10
Figure 2.2: Login. ..... 10
Figure 2.3: Browse item by category (lost items & found items). ..... 11
Figure 2.4: Filter option. ..... 11
Figure 2.5: Browse item. ..... 12
Figure 2.6: View item details. ..... 13
Figure 2.7: Browse item lost nearby and recently. ..... 13
Figure 2.8: Publish lost or found item information. ..... 14
Figure 2.9: View my item. ..... 14
Figure 2.10: Edit and delete item information. ..... 15
Figure 2.11: Browse lost and found item. ..... 17
Figure 2.12: Filter by location. ..... 18
Figure 2.13: Search and filter. ..... 18
Figure 2.14: Message finder or owner. ..... 19
Figure 2.15: Publish lost or found item information. ..... 20
Figure 2.16: View my items. ..... 20
Figure 2.17: Delete item. ..... 21
Figure 2.18: Register account. ..... 23
Figure 2.19: Login. ..... 23
Figure 2.20: Main Page. ..... 24
Figure 2.21: Filtered by location and date posted. ..... 25
Figure 2.22: Social Media Page. ..... 25
Figure 2.23: Create new listing. ..... 26
Figure 2.24: My listings. ..... 26
Figure 2.25: Contact advertiser. ..... 27
Figure 3.1: Waterfall Methodology (RemoteScout, 2023). ..... 38
Figure 3.2: Analysis of Respondent's Age. ..... 39
Figure 3.3: Analysis of Respondent's Gender. ..... 39
Figure 3.4: Respondent Faculty Analysis. ..... 40
Figure 3.5: Analysis of the prevalence of misplaced, lost, or found items among university students. ..... 41
Figure 3.6: Analysis on Efficiency In Recovering Lost Items. ..... 42
Figure 3.7: Analysis on Accessibility of the Lost and Found Information. ..... 42
Figure 3.8: Analysis on the Level of Frustration of Searching Item Information Manually. ..... 43
Figure 3.9 Analysis on Necessity of a Centralized Lost and Found Management System. ..... 43
Figure 3.10: Analysis on the Level of Concern regarding the Inability to Track the Status of Lost Items. ..... 44
Figure 3.11: Analysis on the Satisfaction with the Speed of Lost and Found Item Updates. ..... 45
Figure 3.12: Analysis on the Inefficient Process of Requesting Updates on the Lost Items. ..... 45
Figure 3.13: Analysis on the Understanding of the Current Procedures for Reporting Lost or Found Items. ..... 46
Figure 3.14: Analysis on Prior Usage of Lost and Found System. ..... 47
Figure 3.15: Analysis on Most Desirable Feature for the Proposed System. ..... 47

Figure 3.16: Analysis on User Preference for Automatic Matching and Notification Features. ..... 48
Figure 3.17: Analysis on User Concerns about a Lost and Found System. .....49
Figure 3.18: Use Case Diagram. .....51
Figure 3.19: Overall System Architecture of Proposed System (FindIt). .....65
Figure 3.20: Activity Diagram for Register. .....67
Figure 3.21: Activity Diagram for Login. ..... 68
Figure 3.22: Activity Diagram for Logout. ..... 69
Figure 3.23: Browse and Search Item. .....70
Figure 3.24: Activity Diagram for Submit Claim Request. .....71
Figure 3.25: Activity Diagram for Tracking Item Claim Status. .....72
Figure 3.26: Activity Diagram for Report Lost/ Found Item. ..... 73
Figure 3.27: Activity Diagram for Editing Lost/ Found Item Information. ..... 74
Figure 3.28: Activity Diagram for Delete Lost/ Found Item. .....75
Figure 3.29: Activity Diagram for Student Manage Profile. .....76
Figure 3.30: Activity Diagram for Admin View Claim Approval History and Confirm Item Collection. .....77
Figure 3.31: Activity Diagram for Admin Manage Users .....78
Figure 3.32: Activity Diagram for Admin Manage Item Categories, Colours and Location. ..... 79
Figure 3.33: Activity Diagram for Admin Process Claim Request from Students ..... 80
Figure 3.34: Class Diagram of Proposed System. ..... 81
Figure 3.35: Student Register. ..... 86
Figure 3.35 shows a simple user registration screen for a mobile app. The screen includes fields likes name, email, and password, a button to submit registration request, and a link for users who already have an account to "Sign in.". .....86
Figure 3.36: Student Login. .....87
Figure 3.37: Homepage/ Item Listing Page. .....88
Figure 3.38: More Option. .....89
Figure 3.39: Profile Menu .....90
Figure 3.40: Confirm to logout. .....91
Figure 3.41: Profile Details. .....92
Figure 3.42: Additional Filter .....93
Figure 3.43: Item Details of a Found Item. ..... 94
Figure 3.44: Claim Justification. ..... 95
Figure 3.45: Item Details of a Lost Item. .....96
Figure 3.46: My Item. .....97
Figure 3.47: Item Details of ‘My Lost Item’. ..... 98
Figure 3.48: Error Message for Edit Item Details. ..... 99
Figure 3.49: Edit Item page. ..... 100
Figure 3.50: Confirm to Edit Item Details. .....101
Figure 3.51: Confirm to Delete Item. ..... 102
Figure 3.52: Error to Delete Item. ..... 103
Figure 3.53: Potential Matches (My Item). ..... 104
Figure 3.54: Accept Potential Matches and Claim Item. .....105
Figure 3.55: Claim Justification. ..... 106
Figure 3.56: My Claims (My Item). ..... 107
Figure 3.57: Claim Details (My Claims). ..... 108

Figure 3.58: My Found Item (Item Details). ..... 109
Figure 3.59: Report Lost/ Found Item. ..... 110
Figure 3.60: Error Message of Reporting Item. ..... 111
Figure 3.61: Notifications ..... 112
Figure 3.62: My Claims. ..... 113
Figure 3.63: Admin Login. ..... 114
Figure 3.63 illustrates a login webpage for admin. Admin need to provide email address and password to login. ..... 114
Figure 3.64: Admin Dashboard. ..... 114
Figure 3.65: Admin Profile Menu. ..... 115
Figure 3.66: Manage Students. ..... 115
Figure 3.67: Confirm Delete Student's Account. ..... 116
Figure 3.68: Review Claim Request. ..... 116
Figure 3.69: Claim Requests. ..... 117
Figure 3.70: Confirm Approve Claim Request. ..... 118
Figure 3.71: Confirm Reject Claim Request. ..... 118
Figure 3.72: Manage Item Category. ..... 119
Figure 3.73: Manage Item Color. ..... 119
Figure 3.74: Manage Item Location. ..... 120
Figure 3.75: Claim Approval History Overview. ..... 120
Figure 3.76: Claim Approval Details. ..... 121
Figure 4.1 Laragon ..... 125
Figure 4.2 Visual Studio Code ..... 126
Figure 4.3 Android Studio ..... 127
Figure 4.4 Hugging Face ..... 128
Figure 4.5 Student Registration ..... 129
Figure 4.6 Error Handling of Empty Fields ..... 129
Figure 4.7 Error Handling of Invalid Email and Password ..... 130
Figure 4.8 Email Verification ..... 131
Figure 4.9 Email Resend Countdown ..... 131
Figure 4.10 Verification Email Received ..... 132
Figure 4.11 Verification Passed ..... 133
Figure 4.12 Verification Failed ..... 133
Figure 4.13 Check Verification Status before Emal Verified ..... 134
Figure 4.14 Login Page ..... 135
Figure 4.15 Login Failed ..... 135
Figure 4.16 Logout button ..... 136
Figure 4.17 Confirm logout ..... 136
Figure 4.18 Enter email address to receive password reset link ..... 137
Figure 4.19 Email not found ..... 137
Figure 4.20 Reset link sent ..... 137
Figure 4.21 Password Reset Email ..... 138
Figure 4.22 Password reset form ..... 139
Figure 4.23 Password does not match ..... 139
Figure 4.24 Password reset successfully ..... 140
Figure 4.25 Password reset link invalid ..... 140
Figure 4.26 Change password screen ..... 141
Figure 4.27 Current password incorrect ..... 141
Figure 4.28 New password invalid ..... 141
Figure 4.29 Add a new lost/found item ..... 142

Figure 4.30 Insert photo ..... 142
Figure 4.31 Compress large image ..... 143
Figure 4.32 My reported item ..... 143
Figure 4.33 Manage Reported Item ..... 144
Figure 4.34 Edit item information ..... 144
Figure 4.35 Delete reported item ..... 144
Figure 4.36 Edit item information failed ..... 145
Figure 4.37 Delete failed ..... 145
Figure 4.38 Filter options ..... 146
Figure 4.39 Filter applied (Homepage) ..... 146
Figure 4.40 Search item ..... 146
Figure 4.41 Lost item information ..... 148
Figure 4.42 Found item information ..... 148
Figure 4.43 Recovered item information (matched lost item) ..... 149
Figure 4.44 Recovered item information (no matched lost item) ..... 149
Figure 4.45 Submit claim request with justification ..... 150
Figure 4.46 List of matched found items ..... 151
Figure 4.47 Claim matched found item ..... 151
Figure 4.48 Already claimed ..... 153
Figure 4.49 Claimed by others / other claim ongoing ..... 153
Figure 4.50 All claims for the same reported lost item ..... 154
Figure 4.51 Claim details (involves matched found item) ..... 154
Figure 4.52 More Options ..... 155
Figure 4.53 All submitted claims ..... 155
Figure 4.54 Claim details ..... 155
Figure 4.55 Match notification ..... 156
Figure 4.56 Notifications ..... 157
Figure 4.57 Admin login screen ..... 158
Figure 4.58 Admin login failed ..... 158
Figure 4.59 Send password reset link screen ..... 159
Figure 4.60 Not existing email address ..... 159
Figure 4.61 Password reset email for admin ..... 160
Figure 4.62 Reset email password for admin ..... 160
Figure 4.63 Password not matched ..... 161
Figure 4.64 New password requirements ..... 161
Figure 4.65 Password reset successfully ..... 162
Figure 4.66 Invalid password reset link ..... 162
Figure 4.67 Admin logout ..... 163
Figure 4.68 Side menu ..... 163
Figure 4.69 Dashboard ..... 164
Figure 4.70 Students information ..... 165
Figure 4.71 Category Management ..... 166
Figure 4.72 Add new category ..... 167
Figure 4.73 Edit existing category ..... 167
Figure 4.74 Category exists ..... 167
Figure 4.75 Update fail (category name is referenced by items) ..... 167
Figure 4.76 Delete fail (category name is referenced by items) ..... 167
Figure 4.77 Claim review interface with found item list ..... 168
Figure 4.78 Item comparison interface for matching lost and found items ..... 169

Figure 4.79 Item comparison screen showing matched item with similarity score ..... 169
Figure 4.80 Approve claim request ..... 171
Figure 4.81 Reject claim request ..... 171
Figure 4.82 Claim Approval History Interface with Status Tracking ..... 172
Figure 4.83 Confirm item collection ..... 172
Figure 4.84 Claim details ..... 173
Figure 4.85 Claim approved message ..... 173
Figure 4.86 Item collected message ..... 173
Figure 5.1 Satisfaction Level on Student Interface ..... 225
Figure 5.2 Satisfaction Level on Authentication and Student Profile ..... 226
Figure 5.3 Satisfaction Level on Item Management by Student ..... 227
Figure 5.4 Satisfaction Level on Search and Discovery by Student ..... 228
Figure 5.5 Satisfaction Level on Claim Management by Student ..... 229
Figure 5.6 Satisfaction Level on Student Notifications ..... 230
Figure 5.7 Satisfaction Level on Admin Interface ..... 231
Figure 5.8 Satisfaction Level on Admin Authentication ..... 232
Figure 5.9 Satisfaction Level on Admin Dashboard ..... 233
Figure 5.10 Satisfaction Level on Student Management by Admin ..... 234
Figure 5.11 Satisfaction Level on Item Characteristics Management by Admin ..... 235
Figure 5.12 Satisfaction Level on Claim Processing by Admin ..... 236
Figure 5.13 Satisfaction Level on Claim History Management by Admin ..... 237
Figure A.1 Project Gantt Chart for FYP1. ..... 246
Figure A.2 Project Gantt Chart for FYP1. ..... 247
Figure A.3 Project Gantt Chart for FYP2. ..... 248
Figure B.1 Survey Form Description. ..... 249
Figure B.2 Section I for Demographics. ..... 250
Figure B.3 Section II for Current Experience and Challenges Faced. ..... 251
Figure B.4 Section II for Current Experience and Challenges Faced. ..... 252
Figure B.5 Section III for Desired Features and Feedback. ..... 253
Figure C.1 FindIt App User Experience Survey Form Description ..... 254
Figure C.2 Student's Satisfaction Level on User Interface ..... 255
Figure C.3 Student's Satisfaction Level on Functionalities ..... 256
Figure C.4 Student's Satisfaction Level on Functionalities ..... 257
Figure C.5 FindIt Admin Interface User Experience Survey Form Description 258
Figure C.6 Admin's Satisfaction Level on User Interface ..... 259
Figure C.7 Admin's Satisfaction Level on Functionalities ..... 260
Figure C.8 Admin's Satisfaction Level on Functionalities ..... 261

# Chapter 1: Introduction

## 1.1 Introduction

According to a study by Pixie Technology Inc., an average of 2.5 days per year is spent on searching for the lost items (Habersham, 2021), which wastes quite a vast amount of time, and the scenario is worse if the items are not found on time. Obviously, the study shows that the people are careless in handling personal items, and no doubt this scenario applies to the students too.

Universities are huge and the students often have to travel to more than one location within the campus or between campuses during the course of one day. It can sometimes result in students misplace their personal belongings or lose their valuable items and don't know where they lost them. The current process to report and recover lost items is through word-of-mouth or social media groups only which is not efficient and reliable. These methods will cause a delay or failure of item retrieval. Hence, it is essential to develop a system to meet the needs of the university's students.

The system utilizes a mobile application for the owner to upload images and item details to enable the AI-driven image matching model matching lost item images with the list of found items uploaded. This will help automate the matching of lost and found items by handover the job of comparing and analyzing the image to AI model, reducing the manual effort on matching the possible lost and found items. If the image is unavailable when reporting the lost items, system will not generate potential matches for the item.

Since timely notifications plays a significant role to enhance user experience and satisfaction, the notification system will notify users when there is any update regarding the status of the reported items, for instances, when there is new potential matches or approval of claim from the admin. The notifications will help reduce user's anxiety and frustration and feel more secure regarding the recovery process. On the other side, the admin plays the role

to use web-based system to manage the items and approve the claim of the items. By implementing this lost and found system, there will be an improvement and increased efficiency of the university's lost and found operations.

## 1.2 Problem Statement

In UNIMAS, currently there is no centralized platform for the students to report and recover lost items. Existing methods often relied heavily on word-of-mouth, notification in students' social media groups, or asking the staff about found items. This fragmented approach will delay the recovery process or even worse the item will not be recovered permanently (Anas et al., 2023) because it limits the visibility and accessibility for the owner outside these informal channels. Another problem is that the students can only report the lost items physically during office hours and ask for help of other students in the social media groups, leading to anxiety when they wait for updates. Consequently, the owner can't get real-time notifications when the similar item is found.

## 1.3 Project Scope

The project involves developing both a mobile application and web-based system for the users who are students and the admin respectively to simplify the item recovery process in university by integrating an image matching model. The system does not focus on verification for the students' convenience, the mobile application, built with Dart and Flutter, is designed exclusively for them to access the services anytime and anywhere. Registration and login features are required in order to use the system. The mobile app will allow the students to upload photos if available and item descriptions in order to report the lost items or found items. After this, the AI model have to take over the job to compare the uploaded photos with the existing photos in the database to display the potential matches results to both students and the admins if there is any. Item characteristics will also be considered in

matching items. Next, the app will provide real-time notifications to inform the owner whenever there is new potential match or the item is approved for claiming. The student is allowed to request claiming the items not in the match result too.

For the web-based system, which built with PHP and Laravel, is designed for the administrator to manage the lost and found items. For instances, the admin will have the rights to view, create, edit and delete the item characteristics of the lost and found items. These item characteristics are the predefined options that the students must select from when reporting an item to ensure consistency and accuracy of the reporting process. The system will also enable the admin to review the potential matches given by the AI model and approve the claims. Apart from that, the system will give admin the ability to manage user accounts, like view, or delete the user account.

Finally, an API is developed as a central communication layer to ensure the seamless data exchange between the mobile app, web-based system and the database. The data stored in the database should include user information, images and details of item, and the matching result.

The uploaded photos will be compared with the existing photos in the database, and if there is any potential matches between the lost and found items, the students and the admin will be notified. The owner can request to claim the item from the list of potential matches or list of found items manually.

However, the system is limited to be operated within UNIMAS compound only because the system uses a predefined set of fixed locations that users can select when reporting lost or found items. sThese locations are specifically tailored to areas within the UNIMAS campus. As a result, the system cannot be used to report items lost or found outside the UNIMAS campus, as it does not include external locations in its database. Another

challenge is the accuracy of potential matches. The accuracy of the potential matches generated is based on the model chosen and the quality of the image.

## 1.4 Aims and Objectives

- To design and develop the FindIt: Simplifying Campus Item Recovery with AI-Driven Lost and Found app
- To integrate AI-driven features for automating item matching processes within the app
- To evaluate the usability and functionality of the lost and found App.

## 1.5 Brief Methodology

Waterfall is chosen as the methodology to manage this project. It is a widely used project management method characterized by a sequential and linear process (Hoory & Bottorff, 2024). The methodology consists of 5 phases of management, where each phase requires a deliverable from the previous phase before proceed to the next phases (Hoory & Bottorff, 2024). The methodology emphasizes comprehensive documentation for each phase which helps in maintaining the clarity throughout the software development lifecycle (SDLC). Even though the effort of documenting the project is cumbersome, but the detailed documentation helps tracking the project's progress easier and simplifies software maintenance and further improvement and modernization on the system (Nikitin, 2024). This is one of the reason waterfall methodologies is suitable for the project. Another reason is the methodology suitable for the project with well-defined requirements (Nikitin, 2024) and requires upfront planning which aligns well with academic requirements for project report.

Referring to figure 1.1, waterfall methodology consists of 5 phases, requirements, design, implementation, verification, and maintenance.

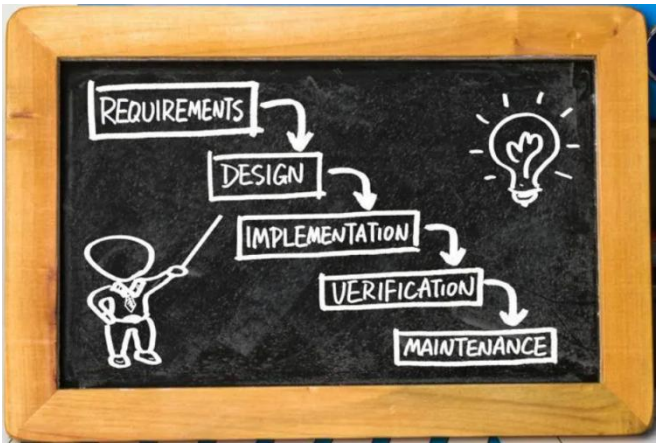



*Figure 1.1: Phases in Waterfall Model (Hoory & Bottorff, 2024).*

Phase 1: Requirements

This is the initial phase where all the project requirements are gathered from the students in the university. The students will be surveyed using questionnaire in Google Form to collect their insights and relevant data. The requirements collected will be analyzed and they should encompass the main functionalities of the system developed (GeeksforGeeks, 2023). This phase is important to ensure the system developed will be useful and meet the needs of the students.

Phase 2: Design

After the requirements is understood comprehensively, the next phase is to design the system or solutions to meet the requirements. The system design includes overall system architecture explaining the operation of system at a macro level. The interaction between users with the system will be illustrated using the use case diagrams, and then each identified use case will be further developed to provide the details descriptions in the use case specifications. Furthermore, activity diagrams will be built to define the system's workflows and processes, and class diagrams will outline the structure and relationships of the system's components.

Then, the wireframes for the mobile application and mobile system will be designed. This phase serves as the guides for the implementation phase (Laoyan, 2024).

Phase 3: Implementation

Based on the requirements collected and solutions in the previous 2 phase, the system is developed in this implementation phase. Each system features will be developed one by one sequentially until forming a complete system. To identify and resolve the issues or bugs which arise throughout the development process, continuous testing are performed.

Phase 4: Verification

The system undergoes quality assurance during the verification phase (Atlassian, n.d.), where functionality and usability testing are conducted to ensure all the specified requirements and functions are met correctly. Any identified issues and bugs are documented and addressed in this phase.

Phase 5: Maintenance

The final phase is system maintenance. Continuous support for the software, such as fixing bugs, implementing updates or improvements based on the user feedback is critical to meet the evolving needs over time. This phase emphasizes maintaining system stability and enhancing functionality to keep the system reliable and effective.

## 1.6 Significance of project

The project plays a crucial role to address the current challenges faced by the students to recover the lost items conveniently and efficiently. Currently there is no standard procedure or guideline for the students to report the lost and found items, so the project helps in ensuring organized and not scattered lost and found information by providing a centralized

platform to locate all of them in one platform. The students will no longer need to search for information across different sources like social media, messaging application, or word-of-mouth. Instead, the system will allow students to report and search items in one place, avoiding any key information of the lost item is overlooked.

Apart from that, the project important in reducing the time and effort of the users to find a list of potential matches from a large amount of found items. With AI-driven image matching, visual characteristics of the item in the image will be compared automatically, bypassing the need for users to find the lost items by manually sifting through every found items, but instead focusing on the items which has been identified as potentially matched. This can greatly speed up the process of recovering the item. The redundancy of checking if the same item matched with the new report can be minimized since the system will take the roles actively monitoring the new reports, generating the potential matches automatically. This approach will keep the users away from committing extra effort since the system will "do the searching and matching".

## 1.7 Project Schedule

Gantt chart for the project schedule is attached in APPENDIX A.

## 1.8 Expected outcome/Project outcome

- A fully functional mobile application and web-based system which provide a formal channel to students in the entire campus to report and recover lost belongings.
- Improved lost items recovering efficiency to reduce delays or failed collections.
- The expected outcome is a fully functional AI-driven lost and found app that simplifies item recovery on campus, enhancing efficiency and user satisfaction.

## 1.9 Project Outline

This Final Year Project Report consists of 6 chapters, which are introduction, literature review, requirement analysis and design, implementation, testing and finally conclusion and future work.

Chapter 1 outlines the brief introduction and overview of the project. The chapter includes the project background, problem statement, project scope, objectives, methodology, significance of project, project schedule, expected outcome and the project outline.

Chapter 2 focuses on literature review of the project. The features of the similar system existing in the market will be analyzed and compared with the proposed system. Technologies and tools used in the project will be covered in this chapter.

Chapter 3 involves requirements and analysis of the project. The requirement gathered is analyzed in this chapter. All the necessary system design like system architecture design, use case diagram, use case specifications, activity diagram, class diagram and user interface is developed in this chapter.

Chapter 4 depicts the implementation of the system. The system design is transformed to a functional system through the system development and implementation process.

Chapter 5 covered testing, which focuses on the test plan, developing the test cases, and carried out functional and usability testing.

Chapter 6 describes about the conclusion and future works. This chapter concludes the project and provides recommendations for possible future work to make improvements.

# Chapter 2: Literature Review

## 2.1 Introduction

In this chapter, a detailed analysis on the existing systems which are similar to the system proposed in this project will be carried out to identify their strengths and limitations. Through reviewing these systems, valuable insights on how the system should be developed and designed will be gained. The conclusion drawn from the analysis will help in forming a system addressing the shortcomings identified in existing solutions.

## 2.2 Review on Similar Existing Systems

### 2.2.1 Overview

There will be 3 similar existing systems analyzed in this section, they are "Lost and Found", "Lost'NFound" and "Lost and Found Network". Their features will be discussed in the following section.

### 2.2.2 Lost and Found

Lost and Found is a mobile application offered by Techju, designed as a community driven project to help users in recovering their misplaced or lost belongings via community support. (*Lost and Found - Apps on Google Play*, 2024). The application aims to improve trouble-free lost and found process by streamlining the process of recovering the lost items. In addition to retrieve the lost items, the platform promotes the communal awareness and a sense of responsibility for the people using it (*Lost and Found - Apps on Google Play*, 2024).

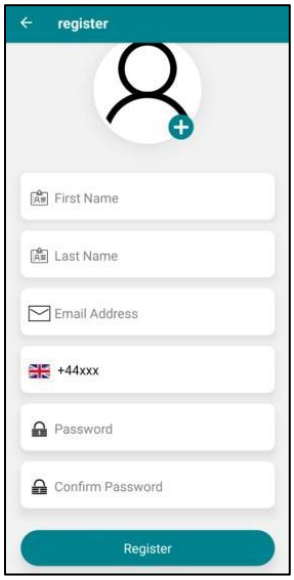



*Figure 2.1: Registration.*

Before the users can start using all the functions provided by the platform, registration is necessary. A non-registered user is not allowed to report and post the lost and found information in the system. Referring Figure 2.1, the name, email address, phone number, and password need to be provided to complete the registration process.

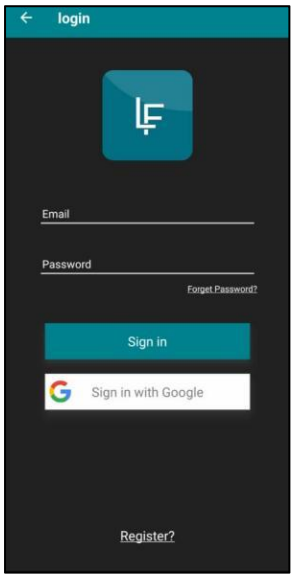



*Figure 2.2: Login.*

As shown in Figure 2.2, the registered user can then login before proceed to use the system. User can reset the password in case they forget the password by clicking on the "Forget

Password" option. The user can prefer to sign in with google to save time by reducing the hassle of creating and managing new account.

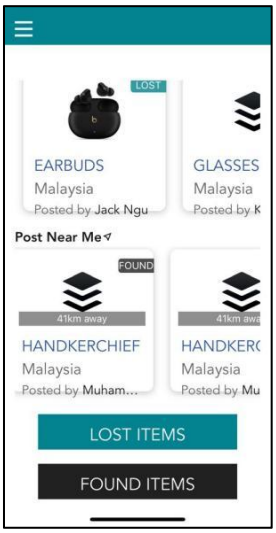



*Figure 2.3: Browse item by category (lost items & found items).*

From Figure 2.3, all the items reported in the system are listed under 2 main categories which are "lost items" and "found items". This can help users to locate the desired lost and found information quickly and effectively. Users can navigate to the "lost items" category to put effort helping others find their belongings, or navigate to the "found items" category to check if the lost item was found by others.

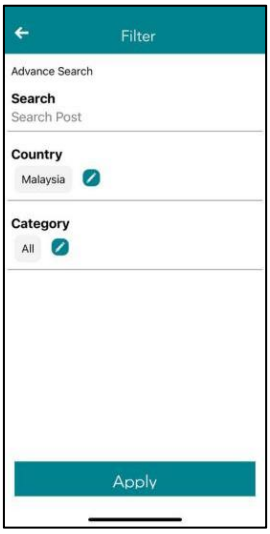



*Figure 2.4: Filter option.*

Filter can be applied to a search to refine the search results, returning the possible match of items efficiently. For example, users are allowed to filter the search results by specify the country, and category as depicted in Figure 2.4. “Country” filter will ensure the lost or found items searched and returned are within a specific country while ‘category’ filter specifying the type of items searched to exclude irrelevant results.

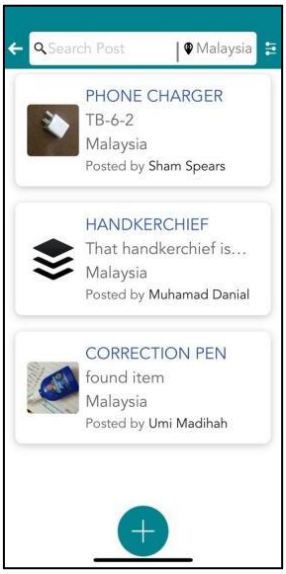



*Figure 2.5: Browse item.*

Figure 2.5 shows that the user can browse the brief item’s metadata such as the item image, name, item description, country, and the username. The user can prefer to report the lost or found item here by clicking on the “+” button so that they can quickly post the item information here if the desired lost or found item is not identified.

*Figure 2.6: View item details.*

Figure 2.6 illustrates all the item details displayed to the users, such as location where the item is found, contact information of the person who upload the item, item category and offered reward. This information is useful to match the item with the rightful owners.

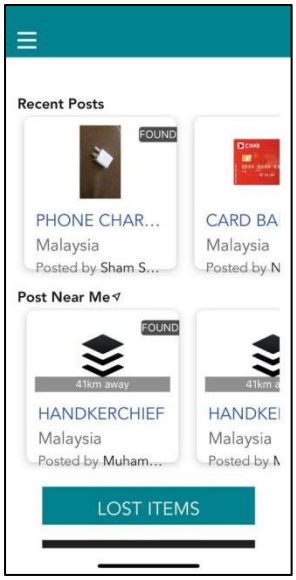



*Figure 2.7: Browse item lost nearby and recently.*

Figure 2.7 depicts that the system offers features for the users to browse the latest lost and found item. It can save the time and effort by reducing the possibility of searching through

the outdated entries or records. Apart from that, the system enable the users to browse the lost and found items in their vicinity, providing convenience to the users.

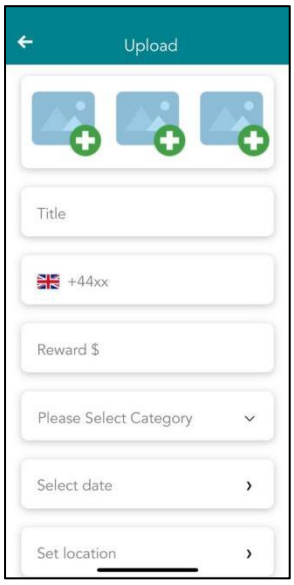



*Figure 2.8: Publish lost or found item information.*

From Figure 2.8, users are required to provide the item image, item name, phone number, reward to the person who find it if any, item category, date, location and item description when publishing a lost or found item on the platform. The comprehensive information provided will maximize the chance of reuniting the items with the owner. The platform allows users to pin a location in the map to show a more accurate reference point for others.

*Figure 2.9: View my item.*

After publishing the lost or found item information in the platform, the user can navigate to review the all the item reported here as shown in Figure 2.9.

*Figure 2.10: Edit and delete item information.*

Figure 2.10 shows that the user can edit the post by clicking on the "Edit" button, and "delete" to remove the post after the item is found and returned.

**Strengths of Lost and Found**

- Sign in using Google

  The "Sign in with Google" functionality help simplify the process of registration, as the user no longer need to provide their name, email address, phone number, password and more personal details. The feature also simplifies the login process by eliminating the needs to remember separate credentials for the mobile application, allowing users to login with the same credentials as their Google account. Apart from that, this feature applies Google Authenticate to adds an additional layer of security when the user login.

**Weaknesses of Lost and Found**

- No email verification upon account registration (excluding “sign in with google” is inherently secure)

  After the user had registered an account, no authentication code is sent to their email for verification purpose which may lead to duplicate accounts with invalid email addresses, affecting the platform’s reliability. A valid email is important for the user to receive notifications like item match alerts and password reset. This could compromise the integrity of the lost and found process, preventing the users from registering an account since the malicious users had register fake accounts using the legitimate email addresses.

### 2.2.3 Lost’NFound

The Lost’NFound mobile application connect together the item owner who lost the item and the finder. GPS technology is utilized in this mobile application to assist the users to find the lost and found item more conveniently by showing all the items reported lost or found nearby to the user location (*Lost and Found - Apps on Google Play*, 2023). The app emphasizes on the privacy of the users since the user don’t have to register to use the mobile apps (*Lost and Found - Apps on Google Play*, 2023). Hence, communication take place without personal information revealed to others.

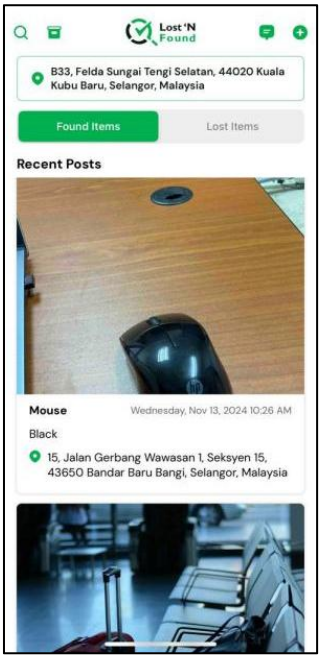



*Figure 2.11: Browse lost and found item.*

Figure 2.11 showcases homepage of the application where all the item reported lost or found near the user location are displayed. The items are divided into 2 main sections, which are "found items" and "lost items". If the user would like to view the lost and found item not in their current location, they can choose a different location rather than their current location by clicking on the address shown in the top part of Figure 2.11.

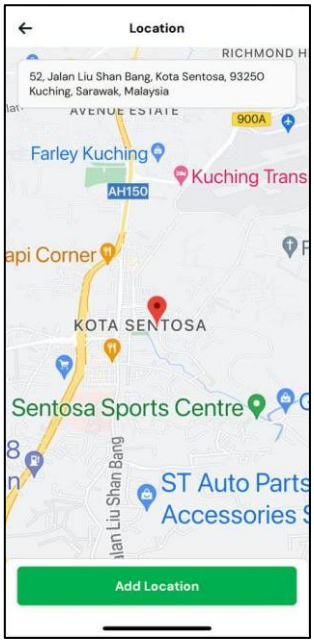



*Figure 2.12: Filter by location.*

Figure 2.12 clearly illustrates that the user can switch their current location by pinpointing the desired location in the map. The red pin in the map can be dragged and moved to any location. The item lost and found near the latest location pinned in the map will be returned in the search result after the "add location" is clicked.

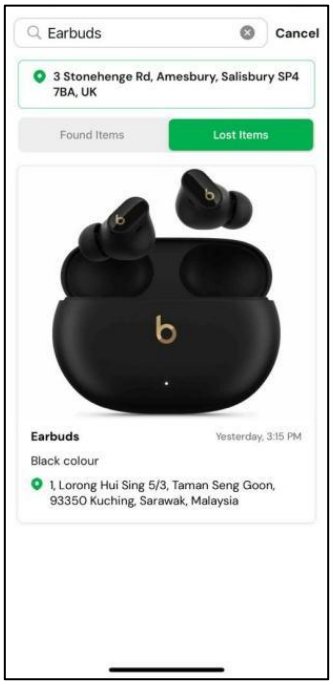



*Figure 2.13: Search and filter.*

As depicted in Figure 2.13, Lost'NFound mobile application provides a feature to search specific items with an accurate location as the filter. Normally other lost and found system only allow to apply filter of broader categories such as country and city, but not accurate up to an exact location. This functionality is achieved by allowing the users to pin the location on the map.

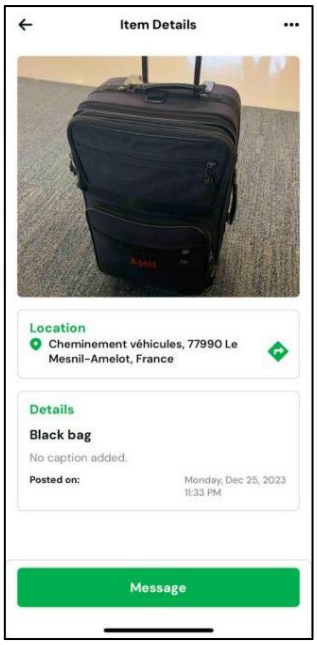



*Figure 2.14: Message finder or owner.*

As shown in Figure 2.14, the "Message" button is available for the users to click on to start communicating with the finder or owner of the items. The communication happens in the app, and no personal information such as the phone number and email addresses will be shared or leaked to the other party in this process.

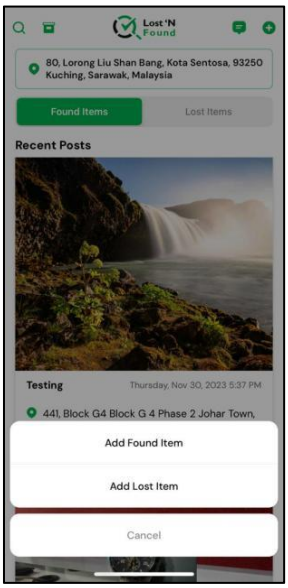



*Figure 2.15: Publish lost or found item information.*

Similar to other lost and found system, the platform allow user to post their lost and found items as shown in Figure 2.15.

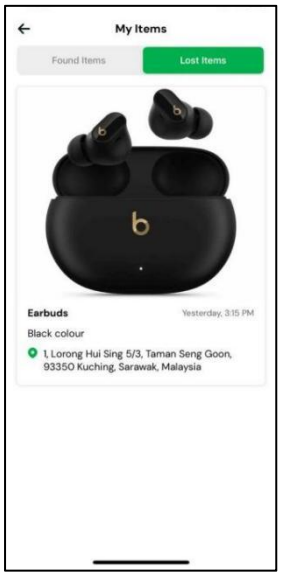



*Figure 2.16: View my items.*

Figure 2.16 depicts that the user can review all the items they have reported in the past. All the published item information is listed under 2 main sections which are “Found Items” and “Lost Items”.

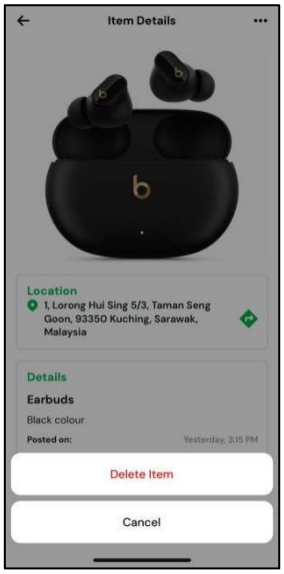



*Figure 2.17: Delete item.*

Figure 2.17 showcases that the user can delete the item they have posted in the platform after a successful retrieval process. There is no option to edit the item information and the only choice is to delete the post and create a new one.

**Strengths of Lost'NFound**

- Pin location to filter search results

  The geolocation feature enables the users to pin a specific location on the map as a filter for the search results. It has improved the system usability and effectiveness because this is an intuitive way for users to visualize and select locations with greater precision. Normally other systems only offer filter option of country, state, or city, not accurate up to specific area. An accurate location filter ensures a high accuracy search, which aids in narrowing down the search results to those closest to the pinned location and hence saves time to identify desired posts of lost or found item information.

- No exposure of private contact information

  There are no user's contact details like email addresses and phone number displayed in the lost or found item information to keep their private information secure. To achieve this, in-app chat feature is provided to ensure secure communication between

the users. Thus, any potential risks of harassment and spam will be avoided, encouraging the users to interact with each other without any concern of exposing any sensitive data.

**Weaknesses of Lost'NFound**

- Too few filter option available for searching items

  The system only offers one filtering options for searching, which can hinder the user from getting the best out of the search. For instance, there is no option to filter by category, meaning users cannot opt to see results by item type. This limitation contributes to a longer search time to filter through unrelated results, negatively affecting the overall efficiency and user experience with the system.

- Item information cannot be edited, only option to delete is available

  The system does not allow users to update or modify the information of an item once it has been posted. The only option available is to delete the post and create a new post entirely if there is any errors or needs to update the item details ensuring most up-to-date information. This shortcoming reduces the system usability and user satisfaction.

### 2.2.4 Lost and Found Network

Lost and Found Network is a global search engine for lost and found information developed by Greenitco Technologies Pvt Ltd (About Us, 2024). The system can be accessed in both form of website and mobile application, catering to a broader audience. This dual-platform availability improves the accessibility significantly, allowing seamless interaction with the system and providing convenience for users with different preferences (Jones, 2024). It has even provided a Facebook page where the lost and found information is shared to increase the exposure and reach broader audiences and those who are hard-to-reach (Petkovic et al., 2021).

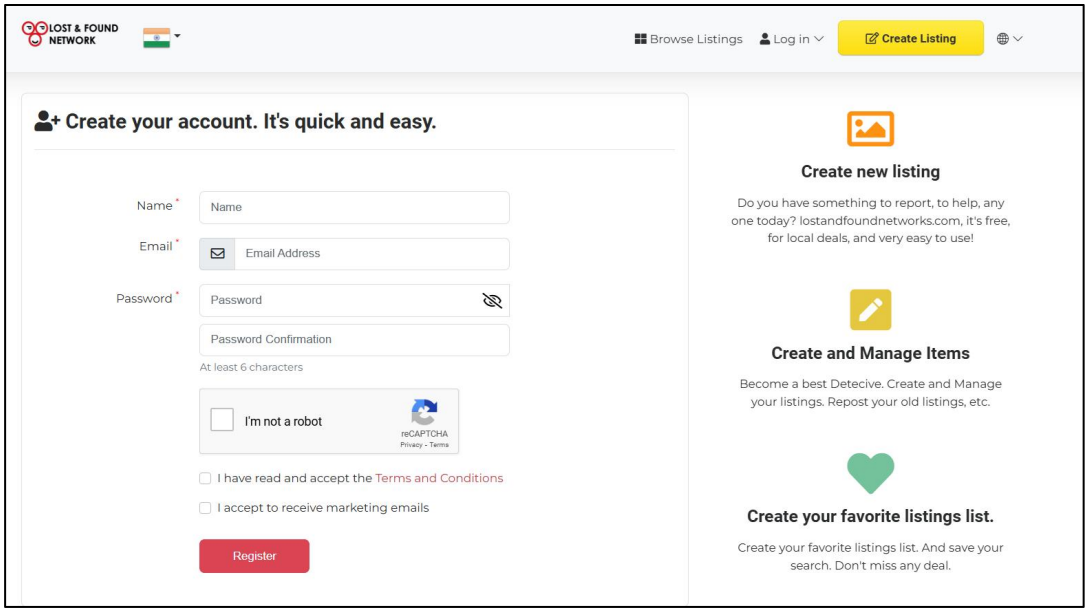



*Figure 2.18: Register account.*

Figure 2.18 depicts that pre-registration is necessary before the user can access to functions like listing the lost and found information in the system.

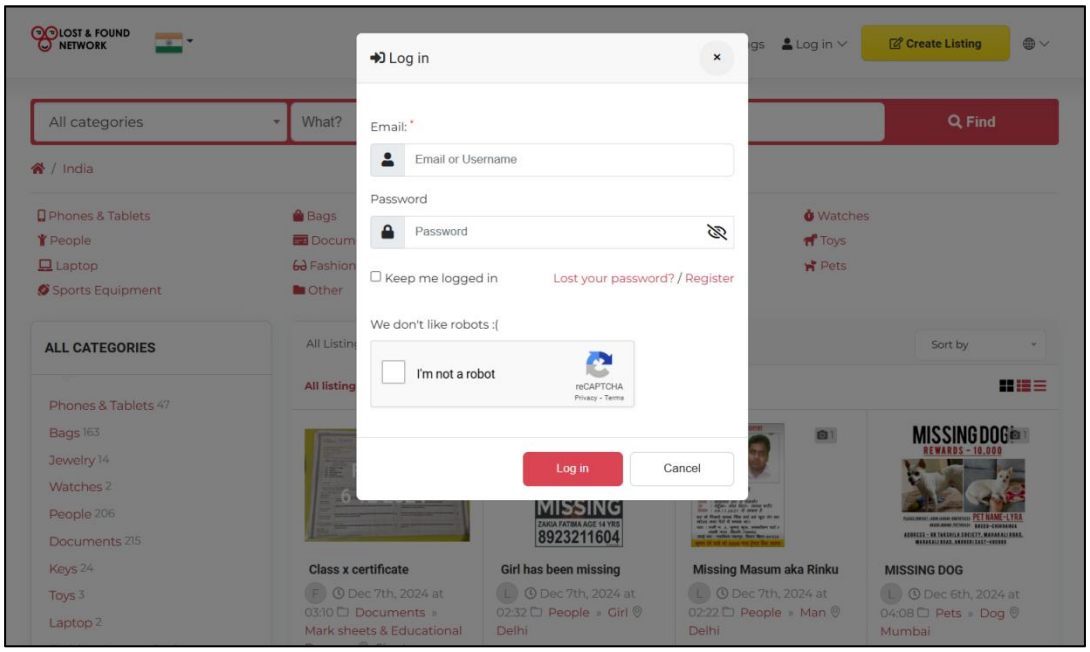



*Figure 2.19: Login.*

The users must provide valid credentials such as email address, password, alongside passing the reCAPTCHA verification to login successfully as shown in Figure 2.19.

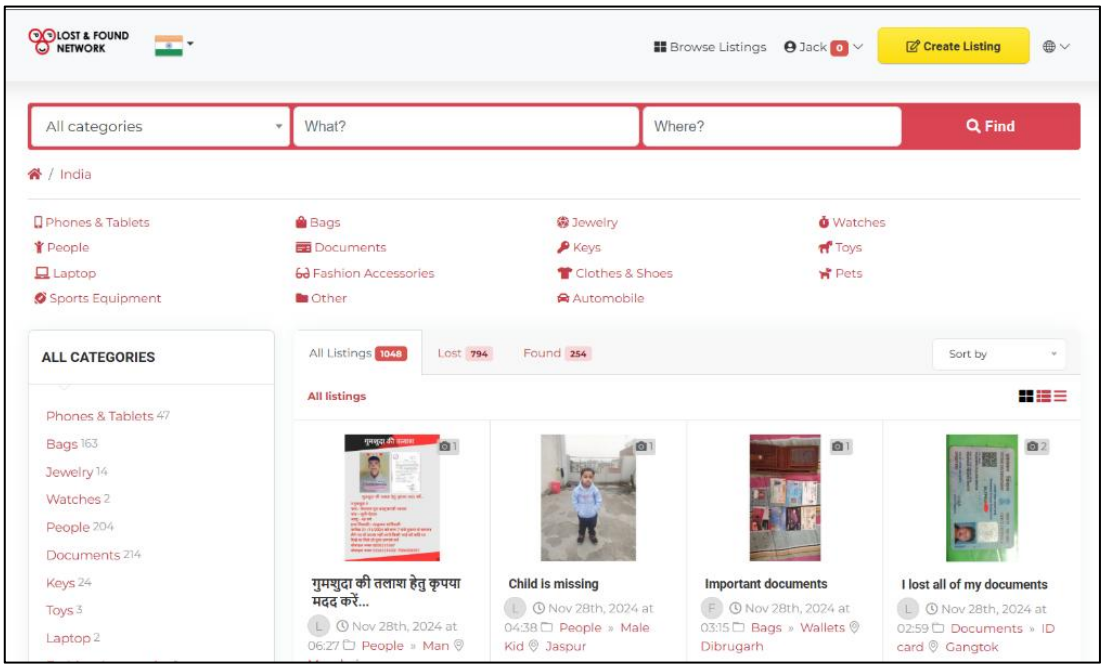



*Figure 2.20: Main Page.*

Figure 2.20 illustrates the main page of the website where the users can browse all the item listings here. The user can search for specific lost or found item by inputting the item's name or descriptions in the search bar alongside with filter option which includes the item categories and location. The returned search results are based on the search keywords and the filters applied. If the user wants to display the item based on the filter alone, the user can click on the category name to view all the items related to the category. Each listing includes a photo, description, location, and date of the incident.

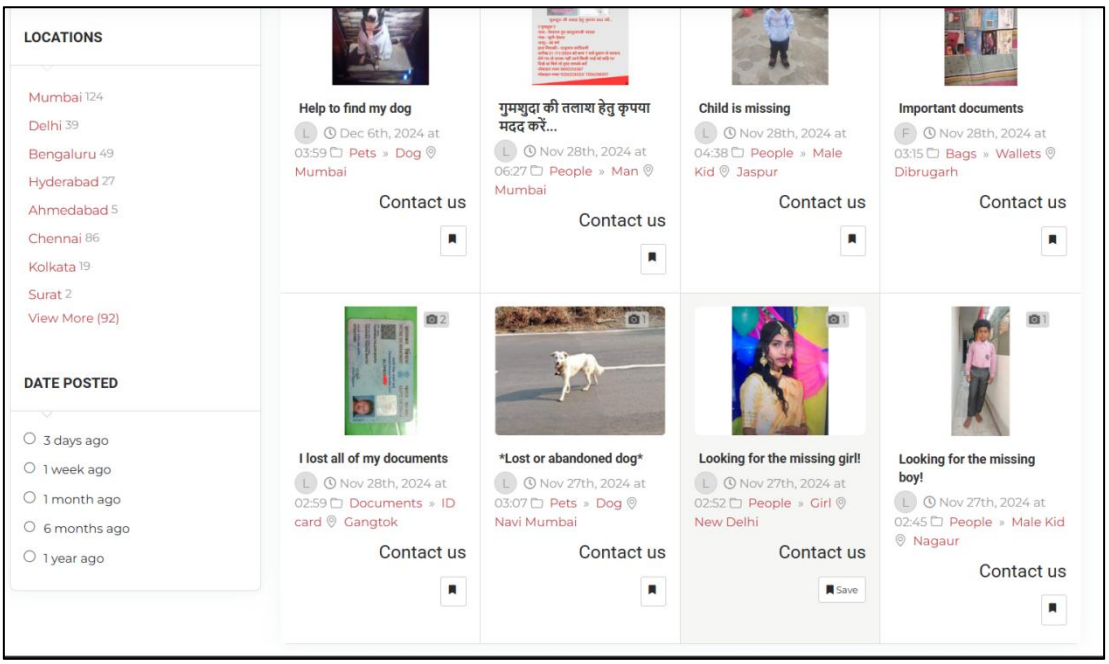



*Figure 2.21: Filtered by location and date posted.*

Additionally, the system allows the users to solely filter the result based on the location or date posted as shown in Figure 2.21.

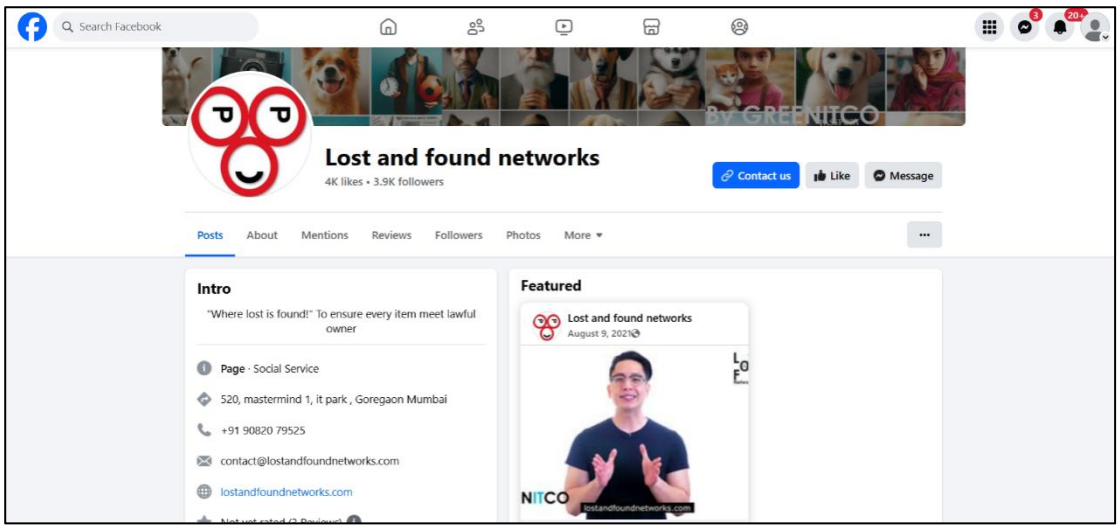



*Figure 2.22: Social Media Page.*

As shown in Figure 2.22, the lost and found item information will be shared in the social media, Facebook page here to reach broader audience increasing the possibility of recovering the lost items.

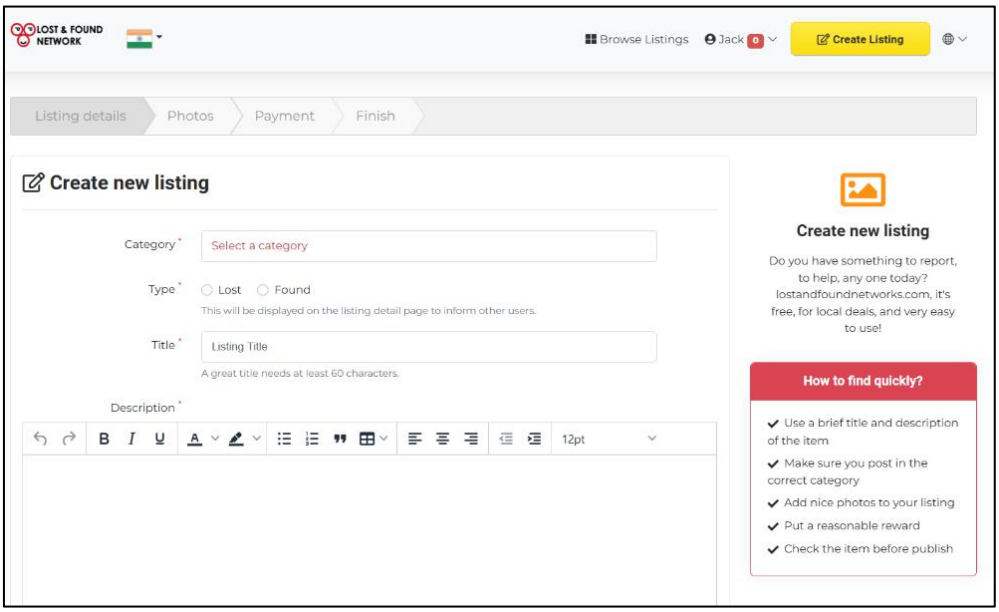



*Figure 2.23: Create new listing.*

As illustrated in Figure 2.23, Every registered user is allowed to create a new listing for lost or found items by providing the relevant item details such as category, item type (lost or found), name, description, any offered rewards, city, contact information (email and phone number) of reporter, and photo if available.

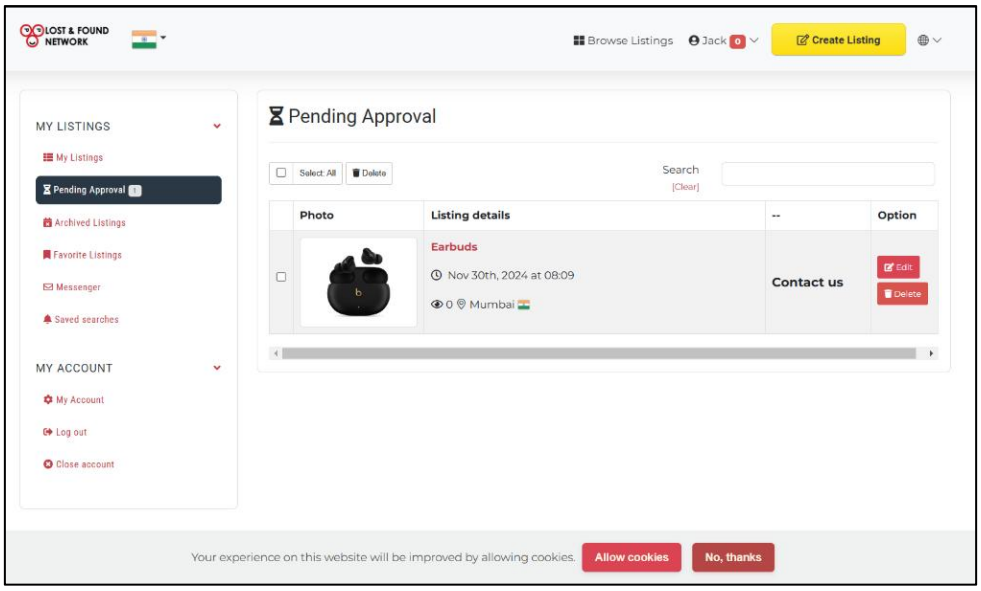



*Figure 2.24: My listings.*

As shown in Figure 2.24, the user can view their own listing of items in the system. After the items is reported to be lost or found in the system, the items will not be listed right away and publicly visible to other users. The listings must first receive approval from the system admin in advance. Apart from that, the user are free to edit or delete the item details before or after the item is listed successfully in the system anytime to ensure flexibility in case of errors.

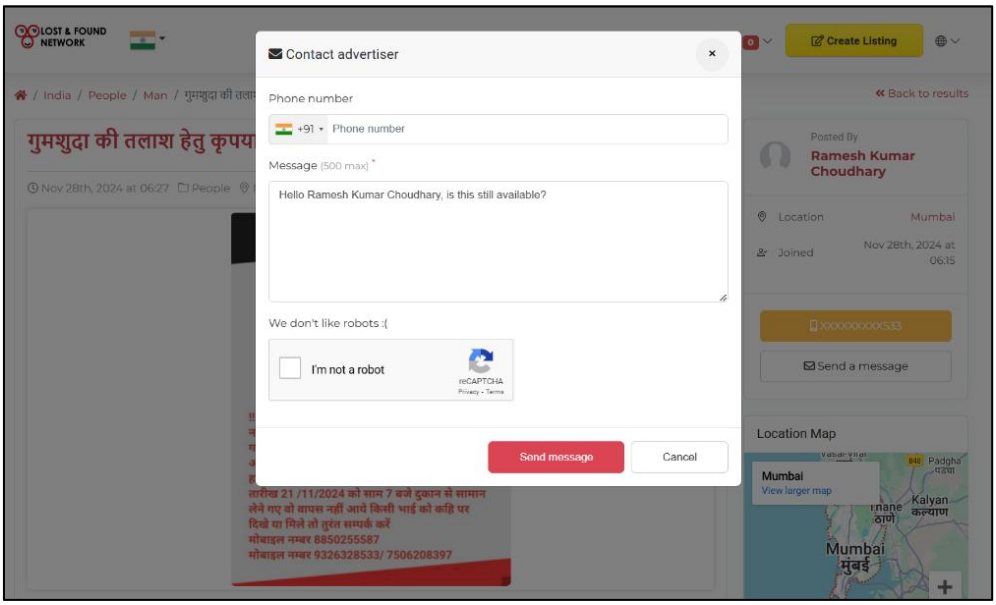



*Figure 2.25: Contact advertiser.*

The system allows the user to contact the advertiser (owner or the person who found an item) using in-app chat function to claim or return items as shown in Figure 2.25. To initiate the chat, the user simply needs to enter the phone number and messages. Alternatively, the user can call the advertiser directly if the situation is emergency since the phone number is provided in the item listing details.

**Strengths of Lost & Found Network**

- Provide official social media page for higher exposure of information

  The system owns a social media page where the lost and found information will be posted on it to help increase the visibility, ensure reaching the audience not actively

use the application itself. The social media posts can be shared easily and circulated among a wider network, positively increasing the chance of recovering the lost items.

- More filter option (i.e. by the number of days since an item was posted)

  Lost & Found Network adds usability by allowing people to filter the items based on the days since the items were posted. This helps users find what they need faster without having to wade through potential item information in a large database.

**Weaknesses of Lost & Found Network**

- Wait for approval before the item details publicly visible

  The system requests the users to go through an approval process to make the item details visible. Although this step may help ensure the integrity and authenticity of the item information shared on the system, it can lead to delays, especially in contexts that are urgent where people need immediate visibility of their lost or found items. The waiting period might affect the overall user experience and satisfaction for the students who are in a urgent situation.

## 2.3 Comparison of the Features in Existing Systems and Proposed System

*Table 2.1: Comparison of the Features of Three Existing Systems and Proposed System.*

| Systems / Features | Existing Systems | | | Proposed System |
|---|---|---|---|---|
| | Lost and Found | Lost'NFound | Lost and Found Network | FindIt |
| Platform | iOS and Android | iOS and Android | iOS, Android and Website | Android |
| Pre-registration | Yes | No | Yes | Yes |
| Target user | Public | Public | Public | Students |
| Browse lost and found | Yes | Yes | Yes | Yes |

| | | | | |
|---|---|---|---|---|
| information | | | | |
| Search option | Yes | Yes | Yes | Yes |
| Filter option | Yes | Yes | Yes | Yes |
| Report lost and found item | Yes | Yes | Yes | Yes |
| Edit lost and found information | Yes | No | Yes | Yes |
| Delete lost and found information | Yes | Yes | Yes | Yes |
| Pin location | Yes | Yes | No | No |
| Built-in chat system | No | Yes | Yes | No |
| Image matching between lost and found items | No | No | No | Yes |
| Device notifications | No | Yes | Yes | Yes |
| Email notifications | No | No | Yes | Yes |
| Browse recovered item information | No | No | No | Yes |
| Approval for listing item | No | No | Yes | No |
| Track the status of reported lost and found item | No | No | No | Yes |
| Submit claim request | No | No | No | Yes |

Based on Table 2.1, ‘Lost and Found Network’ has the most features among the existing systems while ‘Lost and Found’ and ‘Lost’NFound; have the least features among the existing system. All the existing systems have provided all the basic features required in a lost and found system like manage, search, filter and browse the lost and found information. ‘Lost and Found network’ has some special feature that other 2 existing systems does not offer. One of them is receiving approval to list the item and make it publicly visible to all the users. Another feature is that the ability for the user to receive email notifications from the system.

The proposed system, ‘FindIt’ supports Android platform only and is specifically designed for university students, whereas the existing systems focus on public and supports platforms, including Android, iOS, and websites. By focusing on university students as the target audience, the system will be more tailored to their needs. The system provides most of the main features offered by the existing systems and offers a few improvements over the existing system as shown in the Table 2.1. Firstly, ‘FindIt’ introduces a special feature not found in the existing system, etc. image matching between the lost and found items. The feature will help improve the efficiency in item recovery thanks to the automation in matching the items using images. Moreover, ‘FindIt’ allows the students to track the status of the reported items and request for a claim, ensuring the lost and found process is transparent and structured. Unlike some of the existing systems, ‘FindIt’ does not provided features to send message to other users in the system. This is because no necessity for communication between the students since all the found item will be returned to the admin temporarily before the true owner is identified. Additionally, ‘FindIt’ does not request the students to receive approval for item listings because this will consume more time for the recovery process, affecting the efficiency of the systems badly. Overall, the proposed system is a competitive and innovative solution to be implemented in university campus for the students.

## 2.4 Review on Technologies and Tools used in Development

### 2.4.1 Laravel

Laravel is an open-source and powerful PHP framework designed to streamline the process of developing the web applications. The framework follows the model-view-controller (MVC) architecture which separate the development process into three interconnected components: model, view and controller (Fastfwd, 2022). This approach will help to produce organized and maintainable code. With its eloquent Object-Relational Mapping system, the developer can interact with the database using PHP syntax, avoid using the complex SQL queries (Fastfwd, 2022). It can save a lot of time to access and query the database. Laravel also includes the feature which is the blade templating engine which allow the developers to create lightweight template for the web pages easily (Fastfwd, 2022). It is well-suited to develop the website of the proposed system: FindIt since Laravel provide an extensive list of powerful features.

### 2.4.2 Bootstrap

Bootstrap is a CSS framework aims to simplify web design by providing a collection of easy-to-use tools. Responsive grid system, pre-designed components and customized utilities are provided to help developers quickly creating web design that can adapt to device with different size seamlessly. There are also some JavaScript plugins provided in the framework (GeeksforGeeks, 2024b). With these amazing features provided, quick development of professional-looking web design is possible since the developers no longer need to design the webpage from scratch and instead use the pre-designed components which ensure consistency design across different webpage (GeeksforGeeks, 2024b). Therefore, more effort can be assigned to content creation and logic implementation.

### 2.4.3 Flutter

Flutter is a framework built on Dart, designed for building platform-independent mobile apps. It supports cross-platform development where the developers can design single code producing apps that can operate on different platforms, like Android, iOS, desktop and web. Flutter includes integrated tools for testing and debugging, allowing developers to quickly identify and address any problems during development process. Besides, key features of Flutter encompass the accessibility to native features, and customized widgets and tools which are all important factors of building a visually appealing and highly functional mobile apps (Onipe, 2023).

### 2.4.4 Laragon

Laragon is a simple, modern and flexible Windows development server which integrates a few programming languages, database management tools, DNS services, web servers and terminal emulators to ease the software development process for PHP and Node JS applications. For example, PHP, Apache, Nginx, MySQL, Node JS are provided. Laragon provides a user-friendly interface where the developers can control all the software it manages such as turn on or off the server processes. Manual efforts are greatly reduced since Laragon is responsible to configure the necessary files and updates system paths automatically once switching between different versions of the extensions happened (Karunaratne, 2022). This tool is definitely a good option to run and debug the system in the development server.

### 2.4.5 Visual Studio Code

Visual Studio Code is a code editor that offers a vast array of features in the form of plugins which the developers can install to their development environment according to their demands and requirements. Its cross-platform compatibility supports installation on different platforms, including macOS, Windows, and Linux. Visual Studio Code ships with Node.js,

Typescript, Javascript in default, while its extension ecosystem supports for different programming languages such as Python, C++, Java, C#, PHP and etc (*Visual Studio Vs Visual Studio Code - What's Best in 2024?*, 2024). It is suitable for this project due to its ease of use, extensibility, and flexibility.

### 2.4.6 PyTorch

PyTorch is an open-source machine learning framework that allow developers to build and train deep learning models using Python, which speeds up the process between research prototyping and deployment (Yasar & Lewis, 2022). Specifically, Furthermore, PyTorch provides a vast ecosystem of tools and libraries, such as Hugging Face Transformers library which allows developers to load and deploy state-of-the-art models with minimal code (Hoffman, 2024). Additionally, PyTorch is widely used as the foundation for many influential AI models, including those developed by Meta AI, OpenAI, and Google DeepMind, making it a reliable and interoperable choice for integrating third-party AI models (Stewart, 2025).

### 2.4.7 FastAPI

FastAPI is a high-performance web framework enabling developers to build APIs with Python. As one of the fastest web frameworks available, it can process multiple requests at the same time. Its Python 3.7++ capabilities allow the creation of reliable APIs with little code. It has offered powerful features like generating documentation automatically for APIs using Swagger UI and ReDoc based on the type hints and docstrings in the code, which reduces development time and increases API comprehension. Additionally, it also helps in data validation as it automatically validates request and input data, ensuring consistency and reducing errors (Simplilearn, 2024). For these reasons, it is chosen to build the API for the AI model, enabling seamless communication between the system and AI model.

### 2.4.8 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for developing android mobile apps (*Download and Install Android Studio | Android Developers*, 2024). The emulator that comes with Android Studio enables developers to test their applications in a simulated environment that mirrors real Android devices. It is available in many screen sizes, resolutions and Android versions for testing purposes (*Run Apps on the Android Emulator*, 2024). Besides its powerful emulator, Android Studio also provides a unified environment where developers can build, test and debug applications for all Android devices (*Meet Android Studio*, n.d.). This combination caters to every aspect of the development process which minimizes the use of multiple tools.

## 2.5 Review on AI Models and Image Similarity Technologies

### 2.5.1 Justifications for Integrating DINOv2 Model for Image Similarity Task

To find the potential matches between the lost item and found items or perform image similarity analysis, a visual representation model which can extract meaningful representations/features from the images is essential. For the FindIt project, DINOv2 (Self-Distilled Image Pretraining with No Labels version 2) model, a self-supervised vision transformer developed by Meta AI (Shizuya, 2025) for computer vision task is chosen to be used. It is trained on a massive dataset of 142 million images without requiring any labels or annotations. Since the free CPU hardware resource provided in HuggingFaces is limited, with only 2 vCPU and 16 GB RAM (*Hugging Face Free Tier*, n.d.), the smaller distilled model, ViT-Base is integrated into the system in this case instead of using full size of the model which need more powerful hardware for running inference. The justifications of choosing DINOv2 model for image similarity task are explained in the following paragraphs.

DINOv2 is a good choice for the project because the pretrained model can be integrated to the system without further training or fine-tuning. The models are already

pretrained to generate general-purpose visual features that can work across image distributions and different domains and hence not require task-specific or domain-specific fine-tuning (Oquab et al., 2024). As a result, this remarkable generalization capabilities across different domains and object types are suitable for lost and found system which need to handle diverse item categories, and the implementation process can be simplified by immediate integration.

Apart from that, DINOv2 produces strong performance in image retrieval and similarity. DINOv2 model demonstrates competitive performance or even surpasses other state-of-the-art weakly supervised models like CLIP and OpenCLIP across a wide range of benchmarks (Oquab et al., 2024). Notably, DINOv2's robust and high-performance visual features significantly outperform both self-supervised and weakly-supervised models in terms of instance-level recognition (Oquab et al., 2024), which is important in finding and retrieving similar item images from the database. The robust image features can be extracted thanks to the optimization of both high-level (overall scene understanding) and low-level (detailed image patches) features simultaneously (Caron et al., 2021), which is crucial for comparing and matching images of lost and found items.

Lastly, DINOv2 can represent an entire image as a single vector. This single vector representation is ideal for tasks like image search and clustering, enabling efficient similarity comparisons between items (*Dinov2 Base · Models · Dataloop*, 2025). When a new query image is input, its feature vector/embedding will be generated and compared with other image embeddings stored in database previously by using cosine similarity as a distance measure to determine the similarity between 2 item images (Oquab et al., 2024). For this project, considering only when the similarity between images> 0.5, they will be marked as similar.

## 2.6 Conclusion

In summary, a comprehensive review of three existing systems along with the proposed system are carried out in this chapter. The features, strengths and limitations of the existing systems are being discussed and analyzed. Then, the features of the existing systems and proposed system are compared. Additionally, the software tools used to develop the proposed lost and found system, including Laravel, Bootstrap, Flutter, Laragon, Visual Studio Code, TensorFlow, FastAPI, Android Studio, are reviewed.

# Chapter 3: System Analysis and Design

## 3.1 Introduction

This chapter discusses in detail the phases involved in the Waterfall Model software development methodology. The selection of methodology is crucial to define the systematic procedures to deliver a functional system within the specified timeframe. Figure 3.1 presents the phases of the Waterfall Model. The focus of this chapter will be placed on the first two phases, namely requirement analysis and system design, while the overview of the subsequent phases like implementation, verification and maintenance will be provided in this chapter.

The requirement analysis phase has been divided into two sub-works, namely: Data Analysis and System Requirements Analysis. Data Analysis would involve questionnaire distributions to gather information whereby the data extracted will be subjected to detailed analysis for meaningful findings and identification of system requirements. In the system requirement analysis phase of software development, use case diagram is used to visualize the functional requirements of a system by illustrating the interactions between users (actors) and the system, ensuring all functional requirements are accounted for. For the system design phase, the focus is to develop comprehensive diagrams and documentation of the proposed system, such as a system architecture diagram, activity diagram, class diagram and wireframe.

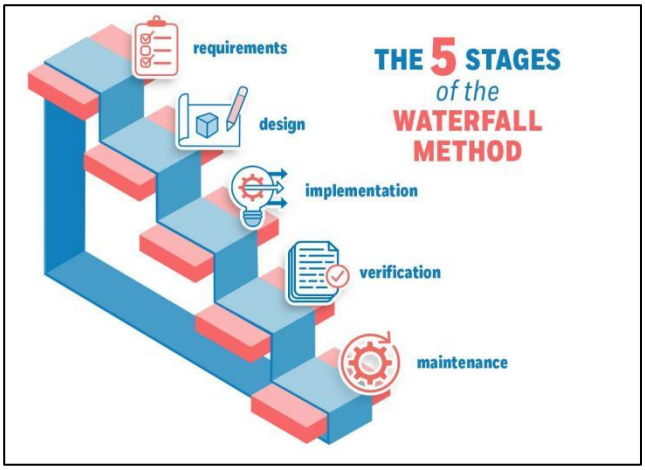



*Figure 3.1: Waterfall Methodology (RemoteScout, 2023).*

## 3.2 Requirement Analysis Phase

Data analysis and system requirement analysis will be conducted via questionnaire and use case diagram respectively in this phase to ensure that all the requirements are captured accurately.

### 3.2.1 Data Analysis

Questionnaire is distributed to the respondents through Google Form to collect relevant data for analysis. The questionnaire is attached in APPENDIX B for reference. The target audience of this questionnaire is any undergraduate students in UNIMAS. The questionnaire was divided into three sections: Section I Demographics, Section II Current Experience and Challenges Faced, and Section III Desired Features and Feedback. Section I aims to collect the demographic details of the respondent, Section II aims to understand reasons of developing the proposed system while Section III aims to identify the necessary requirements to be included in the system and collect feedback from the respondents regarding the proposed system. A total of 33 respondents had answered the questionnaire and the data collected are presented in chart form for better analysis.

#### 3.2.1.1 Analysis of Demographic

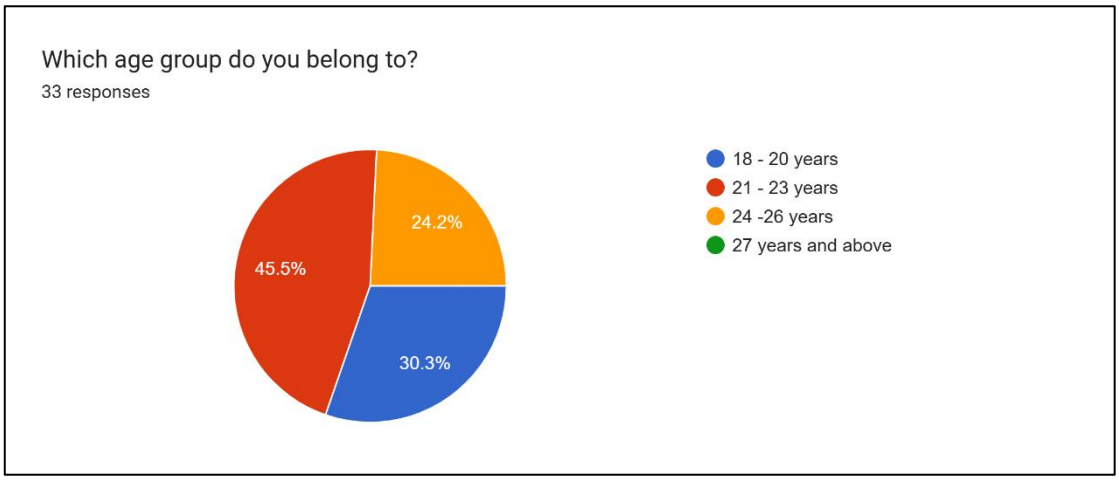



*Figure 3.2: Analysis of Respondent's Age.*

As shown in Figure 3.2, most of the respondents (45.5%) are between 21 to 23 years old, followed by 30.3% of respondents with age group 18-20 years old and 24.2% of the respondents are between 24 to 26 years old. There is no respondent aged 27 and above.

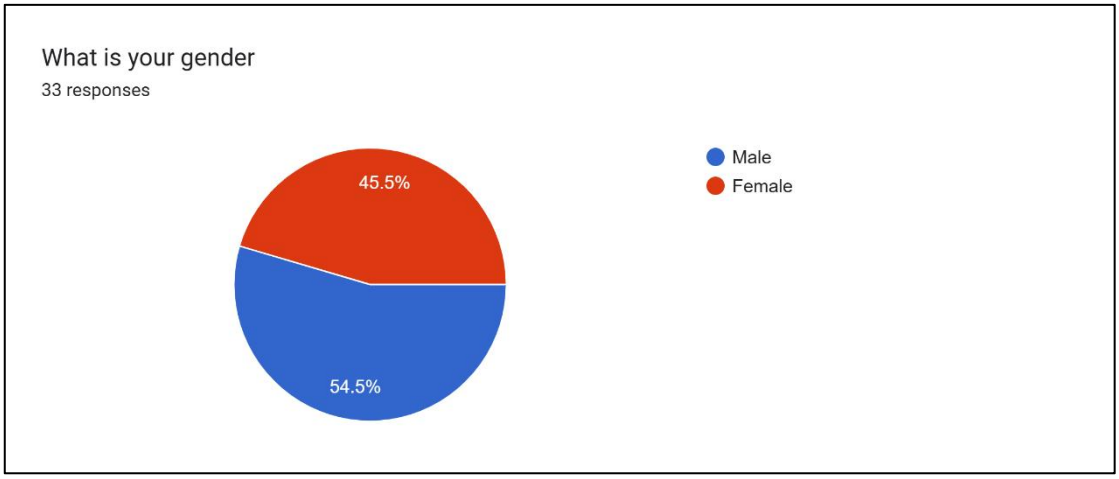



*Figure 3.3: Analysis of Respondent's Gender.*

There are 54.5% of male respondents and 45.5% of respondents participated in the survey as depicted in Figure 3.3.

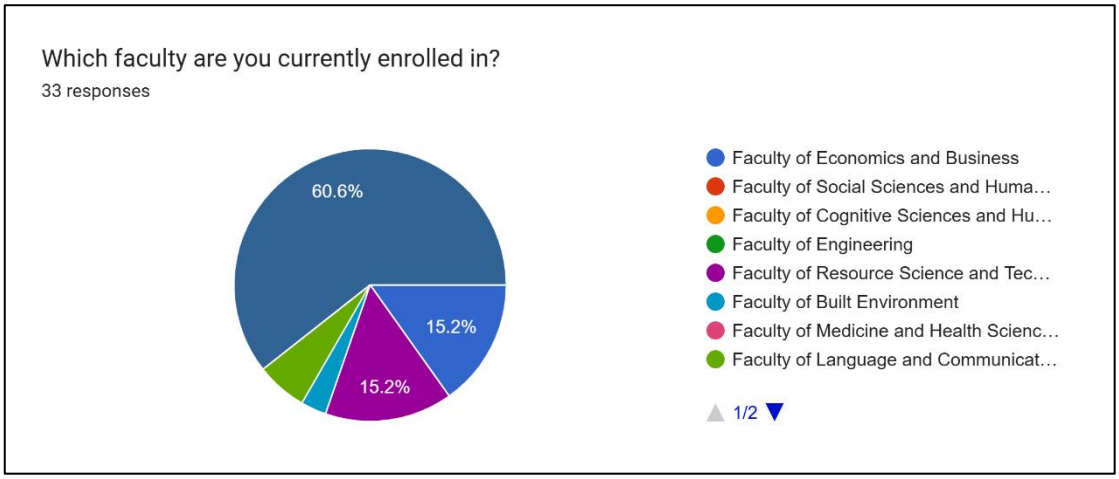



*Figure 3.4: Respondent Faculty Analysis.*

Figure 3.4 shows that Faculty of Economics and Business dominates with 60.6% of respondents. Faculty of Social Sciences and Humanities and Faculty of Cognitive Sciences and Humanities each account for 15.2% of respondents. Meanwhile, Faculty of Engineering has 6.1%, and the Faculty of Resource Science and Technology has 3%. The remaining faculties have no respondents taken part in this survey.

### 3.2.1.2 Analysis on Reasons of Developing the Proposed System

Section II of the Questionnaire focus on analysis of the reasons developing the proposed system. The following charts are the results of the questionnaire for Section II and will be analyzed in detail. From Figure B.3 in APPENDIX B, “1” always represents “Strongly Disagree” while “5” always represents “Strongly Agree” in the linear scale from 1 to 5 except for Figure 3.5.

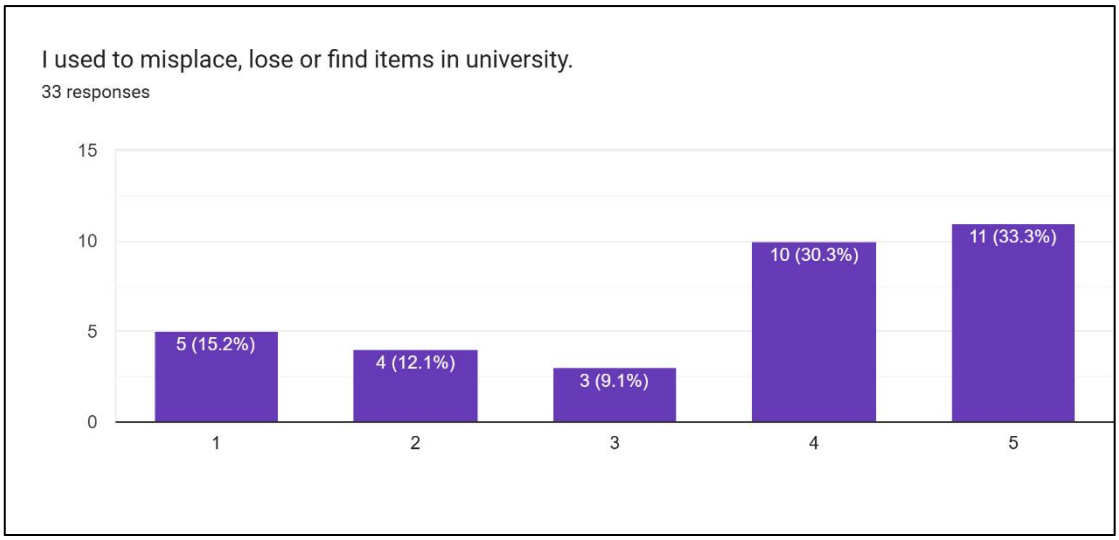



*Figure 3.5: Analysis of the prevalence of misplaced, lost, or found items among university students.*

For the question shown in Figure 3.5, “1” represents “Never” while “5” represents “Always” in the linear scale from 1 to 5. The majority of respondents (33.3%) admits that they experienced frequent misplacement, loss, or finding of items in university. A considerable number of students (30.3%) has moderate frequency with the issue, while fewer students had less frequencies on the issue (15.2% for "1," 12.1% for "2," and 9.1% for "3"). Overall, this chart indicates that most of the students faced the problem of misplacing, losing, or finding items during university years.

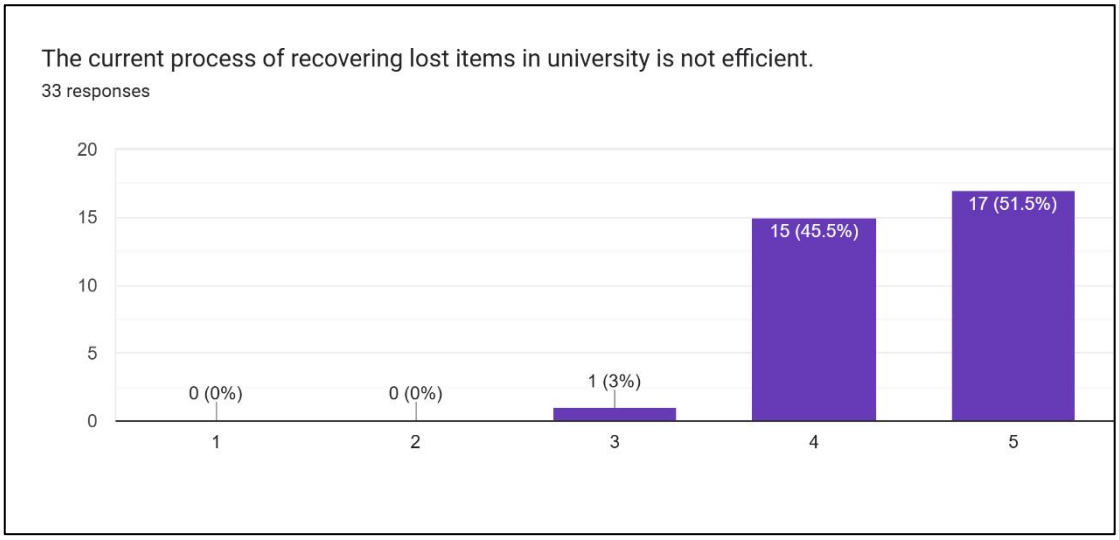



*Figure 3.6: Analysis on Efficiency In Recovering Lost Items.*

Figure 3.6 shows that most of the respondents (61.5%) strongly agree that the current process of recovering lost items in university is not efficient, followed by 45.5% agree with the statement. Only 3% of the respondent stays in neutral position, while nobody disagreed with the statement.

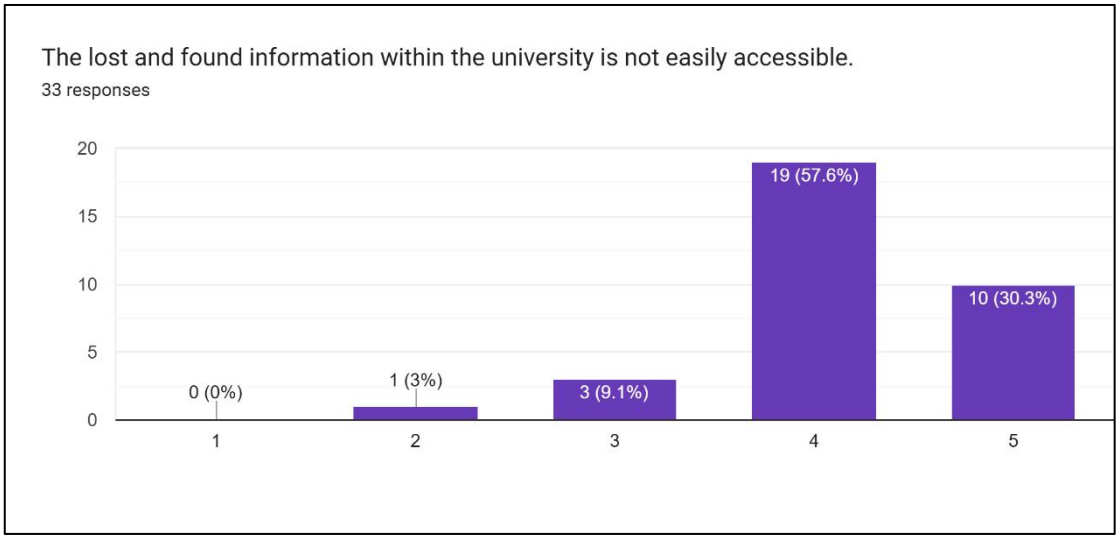



*Figure 3.7: Analysis on Accessibility of the Lost and Found Information.*

Figure 3.7 shows that most of the respondents (57.6%) agree that the lost and found information within the university is not easily accessible, while 30.3% strongly agree to the

statement. Only a small minority remained in a neutral position and disagree (9.1% and 3% respectively).

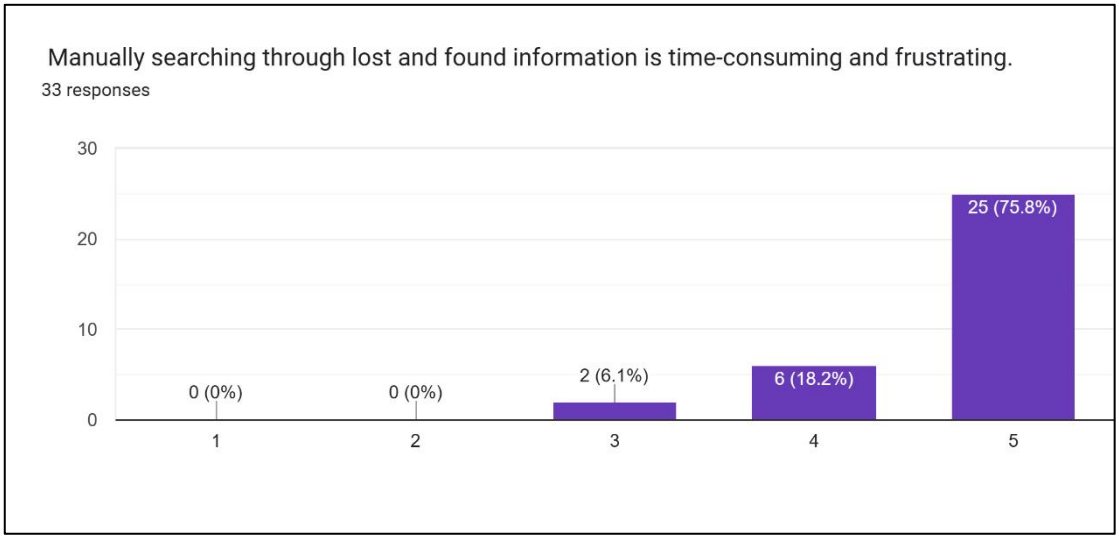



*Figure 3.8: Analysis on the Level of Frustration of Searching Item Information Manually.*

Figure 3.8 shows that most of the respondents (75.8%) strongly agree that it is really frustrated to manually search through lost and found information, while 18.2% agree to the statement. Only a small minority remained in a neutral position (6.1%).

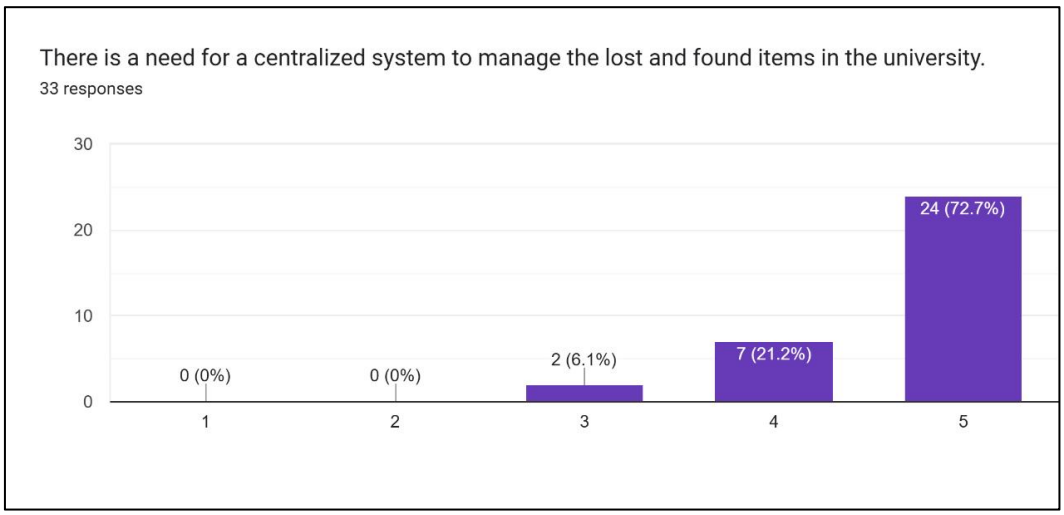



*Figure 3.9 Analysis on Necessity of a Centralized Lost and Found Management System.*

Figure 3.9 illustrates that most of the respondents (72.7%) strongly agree that it is really frustrated to manually search through lost and found information, while 18.2% agree to the statement. Only a small minority remained in a neutral position (6.1%).

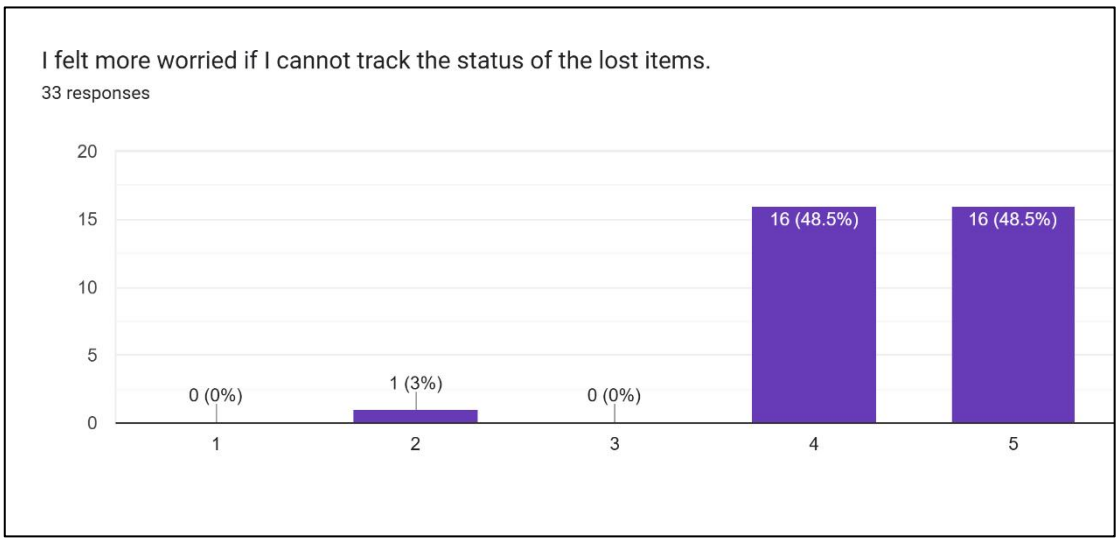



*Figure 3.10: Analysis on the Level of Concern regarding the Inability to Track the Status of Lost Items.*

The majority of respondents (48.5%) strongly agree that they may feel worried if unable to track the status of the lost items as shown in Figure 3.10. Same amounts of respondents (45.8%) agree to the statement, while 3% of the respondents disagree to the statement.

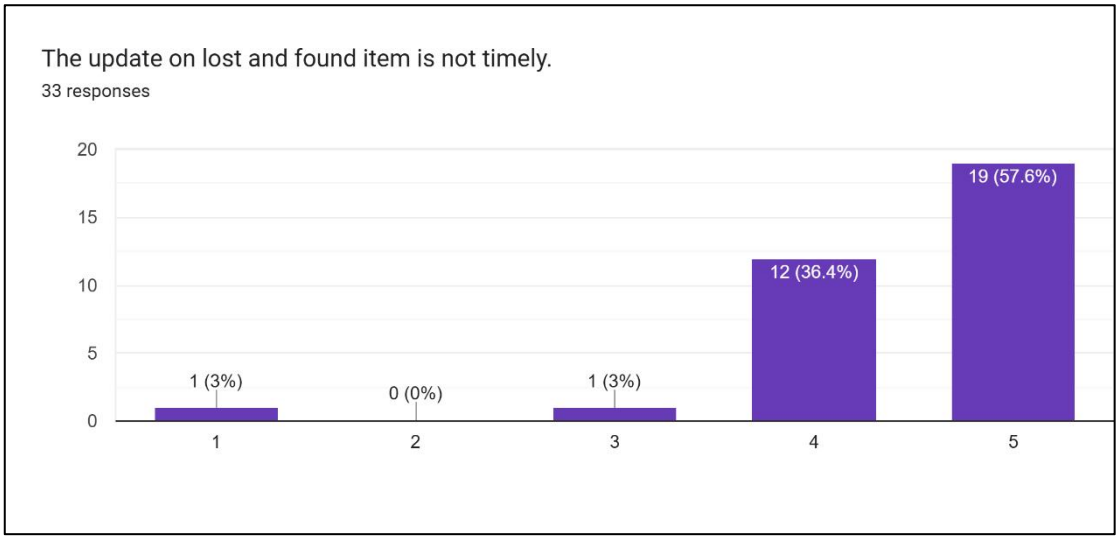



*Figure 3.11: Analysis on the Satisfaction with the Speed of Lost and Found Item Updates.*

Figure 3.11 depicts that most of the respondents (57.6%) strongly agree that the updates on lost and found item is not timely, followed by 36.4% of the respondents agree to the statement. Only 3% of the respondents remains neutral for this statement and an equal percentages of respondents strongly disagree with the statement.

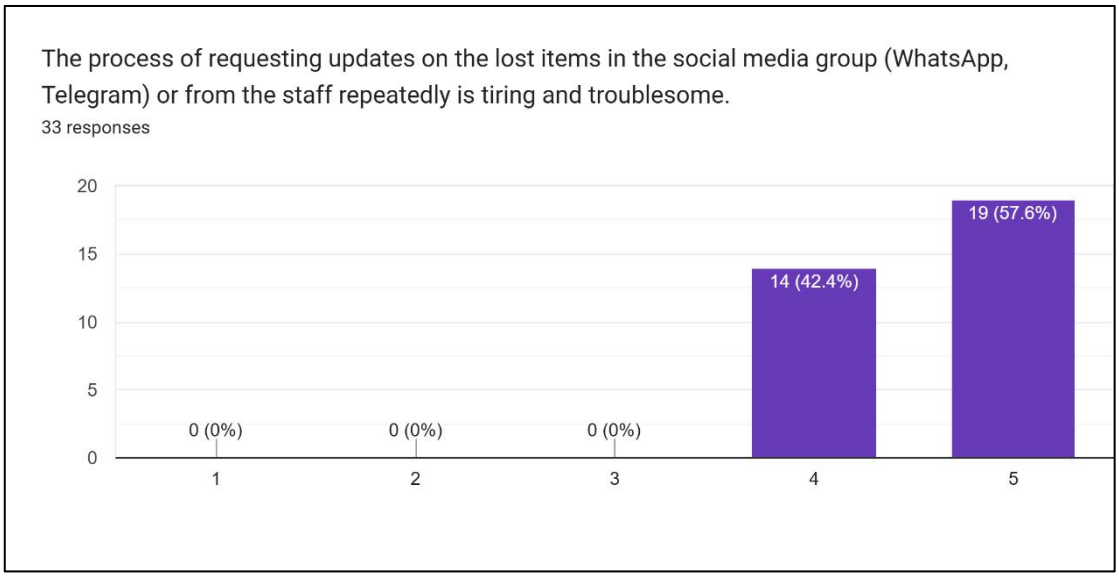



*Figure 3.12: Analysis on the Inefficient Process of Requesting Updates on the Lost Items.*

Figure 3.12 illustrates that most of the respondents (57.6%) strongly agree that the current method of requesting updates is inefficient, while 42.4% agree to the statement. None of the respondents disagree to the statement.

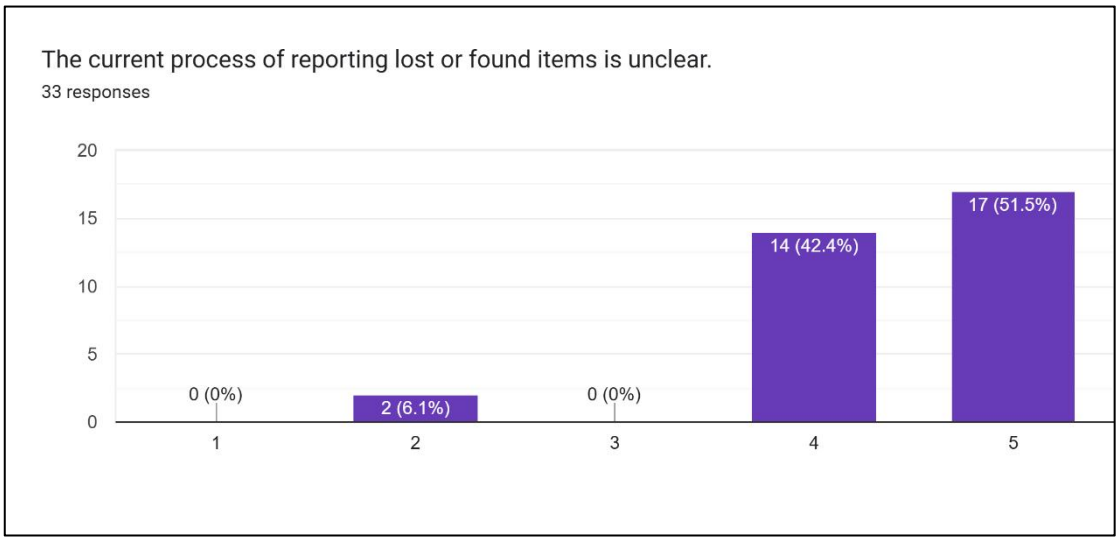



*Figure 3.13: Analysis on the Understanding of the Current Procedures for Reporting Lost or Found Items.*

Most of the respondents (51.5%) strongly agree that they are unclear about the current procedures for reporting lost or found items, while 42.4% agree to the statement as shown in Figure 3.13. Only a small minority disagree to the statement (6.1%).

### 3.2.1.3 Analysis on Proposed System's Functionalities and Feedback

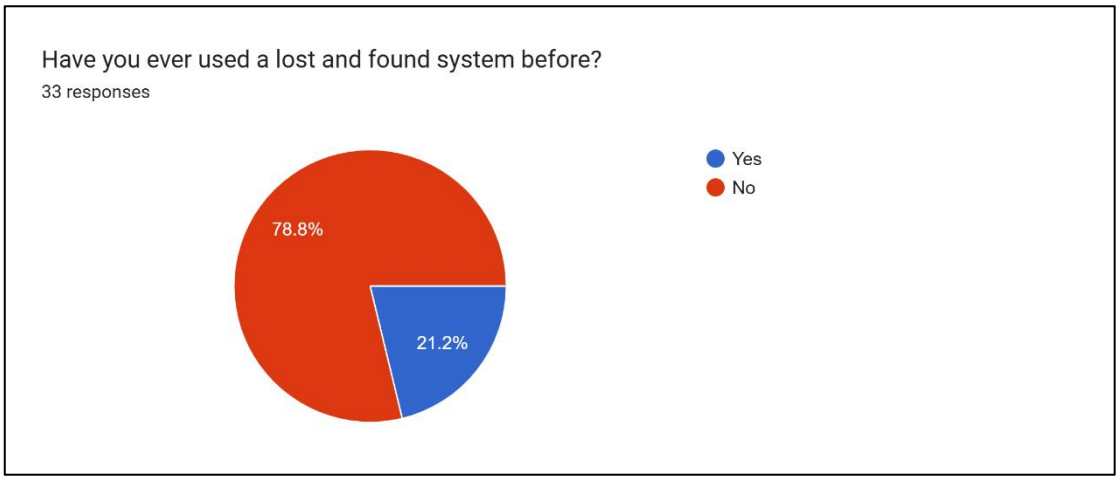



*Figure 3.14: Analysis on Prior Usage of Lost and Found System.*

Referred to Figure 3.14, of the 33 responses, majority of the respondents (78.8%) have no prior experience in using a lost and found system. Only 21.2% of the respondents have previously used a lost and found system.

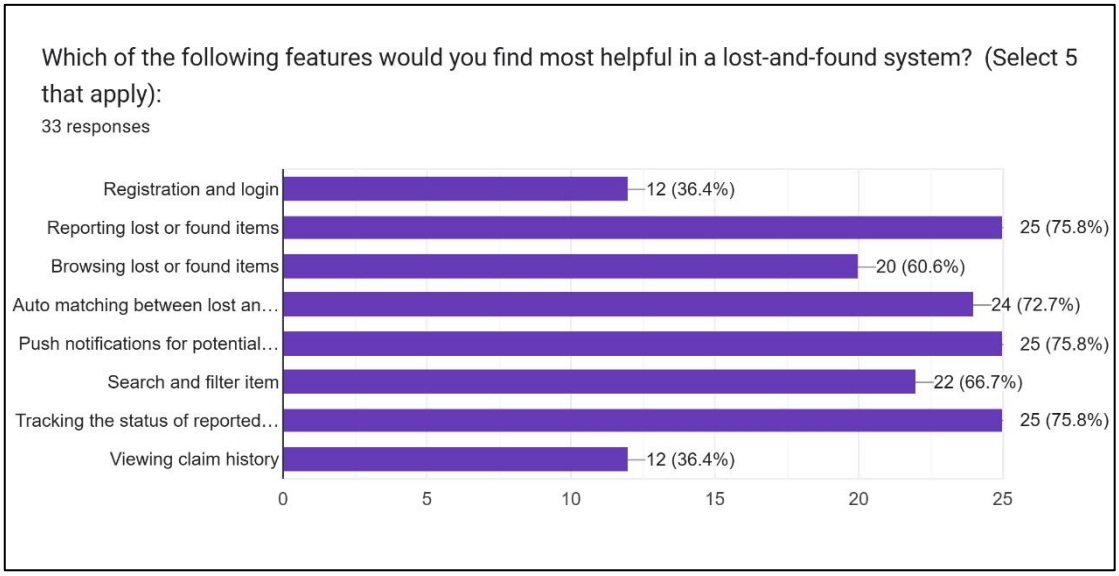



*Figure 3.15: Analysis on Most Desirable Feature for the Proposed System.*

The chart represents the results asking respondents choosing the 5 features they think will be the most helpful in a lost and found system. Figure 3.15 depicts that most popular features among the respondents (over 70%) are reporting lost and found items, auto matching between

lost and found items, push notifications for potential matches between lost and found items, and tracking the status of reported items. The moderately popular features chosen from the respondents (60-70%) are browsing lost or found items, and search and filter item. Lastly, the less popular features include registration and login, and viewing claim history.

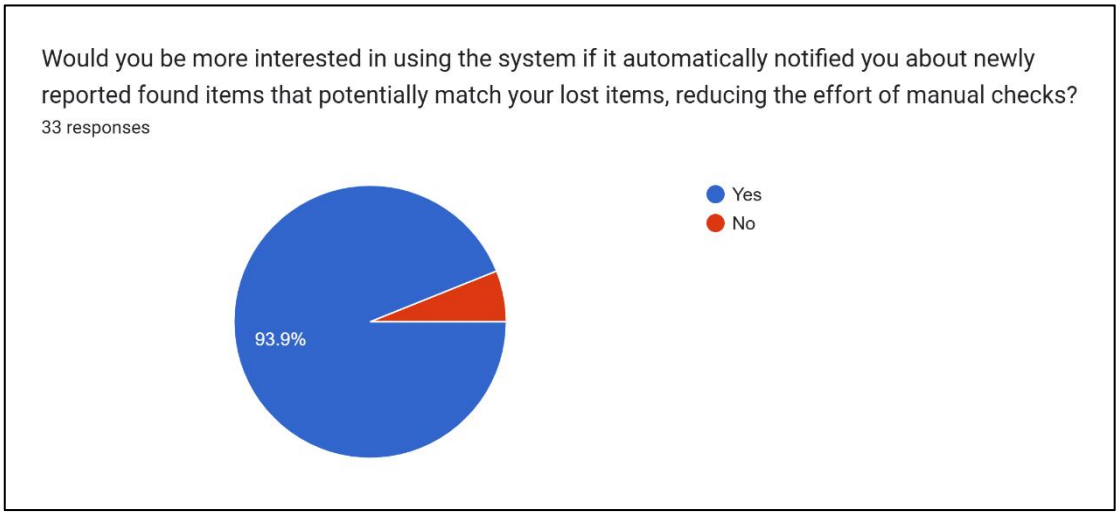



*Figure 3.16: Analysis on User Preference for Automatic Matching and Notification Features.*

Almost all the respondents (93.9%) would be more interested in using the proposed lost and found system if automatic matching between lost and found items and notifications features are provided in the system as shown in Figure 3.16, showing a strong preference for these features.

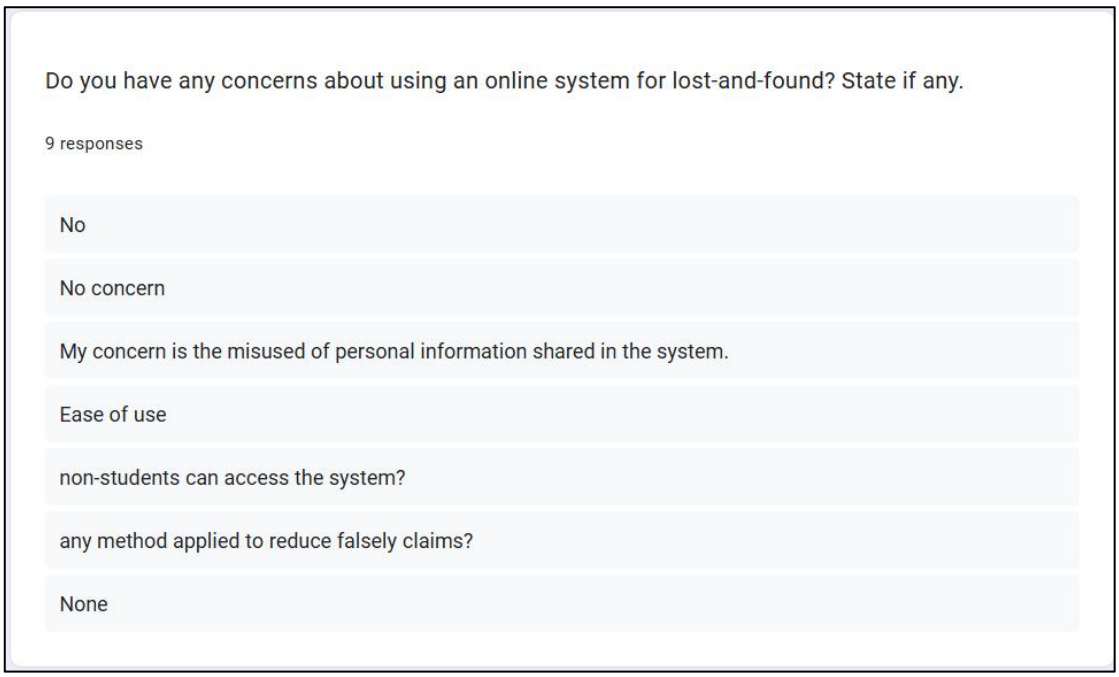



*Figure 3.17: Analysis on User Concerns about a Lost and Found System.*

As shown in Figure 3.17, there are a total of 8 responses given by the respondents for this open-ended question. 3 of the respondents show no concern over a lost and found system. One of the respondents expresses concern about the potential misuse of personal information shared within the system and whether non-students can access the system. This highlights the importance of preventing unregistered users to access the system. Furthermore, concerns are raised about the usability of the system. Hence, it is vital to ensure that the user interface is user-friendly. Another respondent asks about methods to reduce false claims. These concerns point to the need for a role takes part in approve the claim of an item.

The data collected from the questionnaire during the system requirement analysis phase has given rich information that shall be helpful in developing the lost and found system. A number of key features are identified from the analysis, which should form part of the design of the system. The system should focus on the implementation of a user-friendly interface so as to ensure access and ease of use by the students. The questionnaire responses reveal a high demand for efficient reporting and tracking mechanisms regarding lost and

found items. Moreover, the majority of respondents strongly preferred automated features such as the capability of item matching and push notification system in order to notify users about the potential match.

Security aspects emerged as a significant concern, so it is necessary to restrict system access to registered users only. Additionally, an approval mechanism should be integrated to validate item claims, maintaining the system's integrity by preventing fraudulent claims.

The system has, therefore, been designed keeping in mind such basic and popular features as browsing, searching, and filtering. However, additional functionalities can be provided: user registration, login system, and tracking of previous claims. The implementation of these features derived from the questionnaire analysis will ensure the system effectively addresses user requirements, and enhance user satisfaction towards the system.

### 3.2.2 System Requirement Analysis

A use case diagram helps identify functional requirements since it shows the interaction of the system with its users, or actors. In this way, it is easy to understand what the system is supposed to do for each actor. Each use case corresponds to a specific functional scenario that the system needs to support. Since the mapping between the actor and use cases are displayed clearly, a clear outline of the functionalities is available and hence no critical functions will be missing in the requirement analysis phase.

### 3.2.2.1 Functional Requirements

#### 3.2.2.1.1 Use Case Diagram

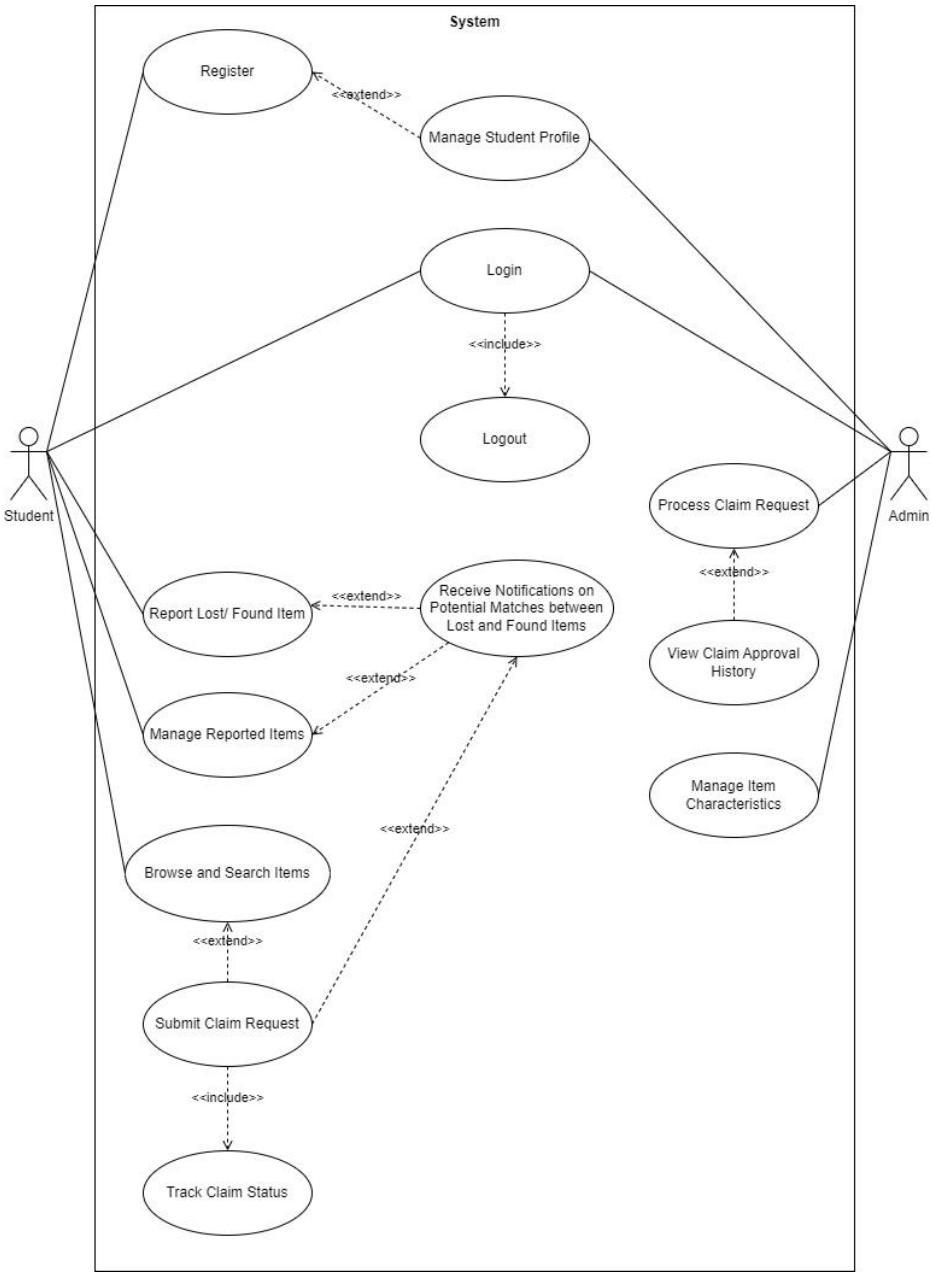



*Figure 3.18: Use Case Diagram.*

#### 3.2.2.1.2 Use Case Specification

*Table 3.1: Use Case Specification for Register.*

| Use Case Name: | Register |
|---|---|

| | |
|---|---|
| Use Case ID: | 1 |
| Actor(s): | Student |
| Description: | Create an account by providing required details. |
| Preconditions: | Student hasn't registered on the system yet. |
| Postconditions: | • Account is created.<br>• Student is logged in.<br>• Student's details are stored in the system. |
| Basic Path: | 1. Student navigates to the registration page.<br>2. Student enters name, email, and password.<br>3. Student clicks "Sign Up" button.<br>4. System validates the input.<br>5. System checks for duplicated email addresses.<br>6. System sends a verification email to the student.<br>7. Student click the URL in the email to verify their email.<br>8. New user account is created.<br>9. Student is logged in automatically after a successful registration.<br>10. Student receives notification on a successful registration. |
| Alternative Path: | A1.1: Invalid Email and Password Format<br>1. Error message displayed.<br>2. Student enters correct email and password.<br>3. Resubmit the registration form.<br><br>A1.2: Duplicate Email<br>1. Error message displayed.<br>2. Student use different email for registration. |

*Table 3.2: Use Case Specification for Login.*

| | |
|---|---|
| Use Case Name: | Login |
| Use Case ID: | 2 |
| Actor(s): | Student, Admin |
| Description: | Login to system using valid credentials. |

| | |
|---|---|
| Preconditions: | User complete registering an account. |
| Postconditions: | • Student is logged in. |
| Basic Path: | 1. Student/ Admin navigates to the login page.<br>2. Student/ Admin enters their email, and password.<br>3. Student/ Admin clicks "Sign In" button.<br>4. System verifies the credentials against the database.<br>5. Student/ Admin is redirected to the home page/ dashboard after login successful. |
| Alternative Path: | A2.1: Invalid Login Credentials<br>1. Error message displayed.<br>2. Student/ Admin re-enters email and password.<br>3. Student/ Admin submit login request again.<br><br>A2.2: Forgot Password<br>1. Click "Forgot Password" link.<br>2. Student enters registered email address.<br>3. Student receives password reset link in the email address.<br>4. Student inputs the new password and submits the form. |

*Table 3.3: Use Case Specification for Logout.*

| | |
|---|---|
| Use Case Name: | Logout |
| Use Case ID: | 3 |
| Actor(s): | Student, Admin |
| Description: | Logout to end the current session. |
| Preconditions: | User logged in |
| Postconditions: | • Student logged out successfully.<br>• Student/ Admin redirected to the login page. |
| Basic Path: | 1. Student/ Admin clicks on "Logout" option.<br>2. Student/ Admin clicks on the "Confirm" button in the confirmation dialog to logout. |
| Alternative Path: | A3.1: Inactivity<br>1. System logs the user out automatically due to inactivity. |

| | |
|---|---|
| | 2. Student/ Admin is redirected to the login page. |

*Table 3.4: Use Case Specification for Managing Student Profile (Student).*

| | |
|---|---|
| Use Case Name: | Manage Student Profile (Student) |
| Use Case ID: | 4 |
| Actor(s): | Student |
| Description: | Student can view and edit their own profile. |
| Preconditions: | Student logged in. |
| Postconditions: | • Student's profile is updated if any changes applied. |
| Basic Path: | 1. Student navigates to user profile section.<br>2. System displays student's profile information.<br>3. Student is allowed to modify the profile information like name and by choosing the option "edit profile".<br>4. Student click 'submit' button to submit the changes.<br>5. System displays the updated profile to the student. |
| Alternative Path: | A4.1: Invalid Input<br>1. Input with invalid format is found by the system.<br>2. Error message displayed.<br>3. Student re-enters the information and submit the changes. |

*Table 3.5: Use Case Specification for Managing Student Profile (Admin).*

| | |
|---|---|
| Use Case Name: | Manage Student Profile (Admin) |
| Use Case ID: | 5 |
| Actor(s): | Admin |
| Description: | Admin are allowed to view student's profile. |
| Preconditions: | Admin logged in |
| Postconditions: | • Student's profile is displayed successfully to the admin. |
| Basic Path: | 1. Admin navigates to student section.<br>2. System displays student's profile information. |

*Table 3.6: Use Case Specification for Report Lost/ Found Item.*

| | |
|---|---|
| Use Case Name: | Report Lost/ Found Item |
| Use Case ID: | 6 |
| Actor(s): | Student |
| Description: | Student report the item they found or they lost |
| Preconditions: | • Student logged in.<br>• Database is active and ready |
| Postconditions: | • The new reported lost/ found item is stored in the database. |
| Basic Path: | 1. Student clicks "+" button to initiate the reporting process.<br>2. Student fills in all the required field of the report form, eg. item type, item name, item description, category, colour, and location lost/ found.<br>3. Student uploads the image of the lost item if they have any.<br>4. Student submits the report.<br>5. System validates input in the report.<br>6. The reported lost/ found item is stored in the database. |
| Alternative Path: | A6.1: Missing Required Details<br>1. Error Message displayed.<br>2. Student fills in all the mandatory field and submits the form.<br>3. Reported lost/ found item is stored in the database. |

*Table 3.7: Use Case Specification for Receiving Notifications on Potential Matches between Lost and Found Items*

| | |
|---|---|
| Use Case Name: | Receive Notifications on Potential Matches between Lost and Found Items |
| Use Case ID: | 7 |
| Actor(s): | Student |
| Description: | Notifies students on potential matches between a lost and a found item to facilitate the recovery process. |
| Preconditions: | • Student has done reported a lost item in the system.<br>• Potential match is identified in the database.<br>• Student allows the system notifications. |
| Postconditions: | • Student is notified of the potential matches. |

| | |
|---|---|
| Basic Path: | 1. System looks for a potential match between the lost and found item based on the item characteristics like category, location, colour and the image similarity if images are provided by the students.<br>2. Student receives push notifications from the system if potential matches for the item they reports has a potential match with any found item reported in the system.<br>3. Students click on the push notifications.<br>4. System redirects students to the notification sections in the system for complete notification details. |
| Exception Path: | E7.1: No Match Found<br>1. There is no similar item in the system matching the item reported lost by the students.<br>2. No notifications sent to the students. |

*Table 3.8: Use Case Specification for Viewing Reported Items.*

| | |
|---|---|
| Use Case Name: | View Reported Items |
| Use Case ID: | 8 |
| Actor(s): | Student |
| Description: | Students can view the details of the items they have reported |
| Preconditions: | • Student logged in.<br>• Student has done reported a lost/found item in the system. |
| Postconditions: | • Item details displayed to the students. |
| Basic Path: | 1. Student navigates to “My Item” section<br>2. System displays a list of all the reported items separated by item type (lost item, found item).<br>3. Student select an item.<br>4. Student navigates to item details subsection to view the detailed item details. |
| Exception Path: | E8.1: No reported item found<br>1. System found no items reported by the student.<br>2. Student receives informative message from the system |

| | |
|---|---|
| | explaining the situation. |

*Table 3.9: Use Case Specification for Editing Reported Items.*

| | |
|---|---|
| Use Case Name: | Edit Reported Items |
| Use Case ID: | 9 |
| Actor(s): | Student |
| Description: | Allow students to update the item details |
| Preconditions: | • Student logged in.<br>• The reported item not involved in any claim. |
| Postconditions: | • The item details are updated. |
| Basic Path: | 1. Student navigates to "My Item" section.<br>2. Student selects an item from the item list to edit.<br>3. Student clicks the "Edit" button under the "Item details" subsection.<br>4. System displays the item details in an editable form.<br>5. Student updates the item details.<br>6. Student submits the form.<br>7. System validates all mandatory field is filled.<br>8. System display message on a successful edit. |
| Alternative Path: | A9.1: Missing Required Item Details<br>1. Student submit the request to edit item details.<br>2. System displays error message on some mandatory field in the form is not filled.<br>3. Student fills in the missing details and submits the form. |
| Exception Path: | E9.1: Item is associated with a claim.<br>1. Students click the "Edit" button.<br>2. System displays an error message indicating that any item involved in a claim and hence cannot be edited. |

*Table 3.10: Use Case Specification for Deleting Reported Items*

| | |
|---|---|
| Use Case Name: | Delete Reported Items |

| | |
|---|---|
| Use Case ID: | 10 |
| Actor(s): | Student |
| Description: | Allow students to removes the item they have reported from the system. |
| Preconditions: | • Student logged in<br>• The reported item not involved in any claim. |
| Postconditions: | • The reported item is removed from the system. |
| Basic Path: | 1. Student navigates to "My Item" section.<br>2. Student selects an item from the item list to edit.<br>3. Student clicks the "Delete" button under the "Item details" subsection.<br>4. System prompts students on their intention to confirm delete the reported item.<br>5. Student confirms delete the reported item. |
| Exception Path: | E10.1: Item is associated with a claim.<br>1. Students click the "Delete" button.<br>2. System displays an error message indicating that any item involved in a claim cannot be deleted. |

*Table 3.11: Use Case Specification for Browsing and Searching Items.*

| | |
|---|---|
| Use Case Name: | Browse and Search Items |
| Use Case ID: | 11 |
| Actor(s): | Student |
| Description: | Allow students to locate the lost or found items by applying search filters and keywords. |
| Preconditions: | • Student is logged in |
| Postconditions: | • Item list displayed to the students<br>• Students view the details of the specific item. |
| Basic Path: | 1. Student access the homepage (browse and search page) initially.<br>2. System displayed a list of items found in the order of date |

| | |
|---|---|
| | posted by default.<br>3. Student can choose to view list of items lost by choosing the "Lost Item" filter.<br>4. Student applies the filters "category", "colour" and "location".<br>5. Student search for items using keyword.<br>6. System displayed all the items with matching keywords in the search results.<br>7. Student clicks on the specific items to look for more details. |
| Alternative Path: | A11.1: Not Applying Filters (except the filter applied by default initially: lost item, found item) and Search Keyword<br>1. Scroll to find the desired items.<br>2. Choose specific items to view detailed item information.<br>A11.2: Not Applying Filters<br>1. Student inputs search keywords.<br>2. Choose desired item from the search result to view item details.<br>A3: Not Applying Search Keyword<br>1. Student applies filter.<br>2. Choose desired item from the filtered result to view item details. |
| Exception Path: | E11.1: No Matching Items Found<br>1. System displays message highlighting no items matched the filters or search keywords. |

*Table 3.12: Use Case Specification for Submitting Claim Request.*

| | |
|---|---|
| Use Case Name: | Submit Claim Request |
| Use Case ID: | 12 |
| Actor(s): | Student |
| Description: | Student can submit claim request on specific item to the admin for verification. Student can manually search for the item and submit request or accept the suggested matches from the system |

| | |
|---|---|
| | and submit request. |
| Preconditions: | • Student logged in |
| Postconditions: | • Admin receive the claim request. |
| Basic Path: | 1. Student access homepage (browse and search page).<br>2. Student search for or browse the list of items found.<br>3. Student chooses specific items.<br>4. Student clicks "Submit" button.<br>5. Student inputs the justification for claim in the text field.<br>6. Confirm to submit the claim request. |
| Alternative Path: | A12.1: Accept Suggested Matches and Submit Claim Request<br>1. Student reports the item they lost.<br>2. System displays students the potential matches of the found items to their lost items in the "My Item" section.<br>3. Student chooses specific item.<br>4. Student clicks "Claim" button to accept the suggestion and prepare to submit the claim request.<br>5. Student inputs the justification for claim in the text field.<br>6. Confirm to submit the claim request. |

*Table 3.13: Use Case Specification for Tracking Claim Status.*

| | |
|---|---|
| Use Case Name: | Tracking Claim Status |
| Use Case ID: | 13 |
| Actor(s): | Student |
| Description: | Allow student to monitor the status of all the item they have requested to claim. |
| Preconditions: | • Student logged in<br>• Student submits at least one claim. |
| Postconditions: | • Student views claim status. |
| Basic Path: | 1. Student access "My Claim" section.<br>2. All the submitted claims (via manual search or suggested matches) are displayed.<br>3. Student views the claim status directly from the list or click |

| | |
|---|---|
| | to see more details of the claim. |
| Alternative Path: | A13.1: Access “My Item” section to track claim status<br>1. Student access “My Item” section.<br>2. Student select the specific item.<br>3. Student navigates to “My Claims” subsection.<br>4. System displayed a list of claims submitted via suggested matches.<br>5. Student views the status of the claim directly from the list or click to see more details of the claim. |
| Exceptional Path: | E13.1: No Claims Submitted<br>1. The student has no submitted any claim.<br>2. System display message highlights that no claim is found. |

*Table 3.14: Use Case Specification for Processing Claim Request.*

| | |
|---|---|
| Use Case Name: | Process Claim Request |
| Use Case ID: | 14 |
| Actor(s): | Admin |
| Description: | Allow admin to approve or reject the claim requested by the students. |
| Preconditions: | • Admin logged in<br>• At least one pending claim request found in the system. |
| Postconditions: | • Claim request is approved or rejected.<br>• Claim status is updated. |
| Basic Path: | 1. Admin accesses “Claim Review” page.<br>2. All found items with pending claim request are displayed here.<br>3. Admin choose an found item.<br>4. System displayed all claim requests associated with the found item in the table and a section displaying the details of found item and the claimant justification.<br>5. Admin clicks “compare” button besides the claim requests to update the section including lost item details with the |

| | |
|---|---|
| | claimant's justification.<br>6. Admin repeat steps 5 until finish reviewing all the claim requests.<br>7. Admin approves or rejects the claim requests. |
| Exceptional Path: | E14.1: No Pending Claims<br>1. Admin accesses "Claim Review" page.<br>2. System displayed message highlighting that no claims are submitted for review. |

*Table 3.15: Use Case Specification for Viewing Claim Approval History.*

| | |
|---|---|
| Use Case Name: | View Claim Approval History |
| Use Case ID: | 15 |
| Actor(s): | Admin |
| Description: | Allow admin to view the history of all the request that had been processed. |
| Preconditions: | • Admin logged in<br>• At least one claim processed. |
| Postconditions: | • Admin views the history of claim approval. |
| Basic Path: | 1. Admin accesses "Claim Approval History" page.<br>2. System displayed all the processed claims.<br>3. Admin views the status directly from the list or click the "view" button for more details. |
| Exceptional Flow: | E15.1: No Processed Claims<br>1. Admin accesses "Claim Approval History" page.<br>2. System displays message highlighting that no claims has been processed so far. |

*Table 3.16: Use Case Specification for Managing Item Characteristics.*

| | |
|---|---|
| Use Case Name: | Manage Item Characteristics |
| Use Case ID: | 16 |
| Actor(s): | Admin |

| | |
|---|---|
| Description: | Allow admin to create, view, edit and delete the item characteristics (e.g. item category, colour and location) so that these characteristics will appear as the options in the dropdown menu for students to choose from during reporting an item. |
| Preconditions: | • Admin logged in |
| Postconditions: | • Admin views the existing item's characteristics.<br>• Any action made on the item characteristics are reflected (view, edit, delete) in the system. |
| Basic Path: | 1. Admin navigates to page of specific item characteristics (item category, colour, and location) to manage the correspond item characteristics.<br>2. System displays a list of instances of the specific item characteristics found in the system.<br>3. System displays all the options (create, edit, and delete) in the same page.<br>4. Admin click "create" button to create new instances of the item characteristics in the system.<br>5. Admin click "edit" to edit the current instances of the item characteristics in the system.<br>6. Admin click "delete" to remove the instances of the item characteristics from the system. |

#### 3.2.2.1.3 Functional Requirements

*Table 3.17: Functional Requirements.*

| **Requirement Description** | **Priority** |
|---|---|
| User: Admin | |
| Login to the system | High |
| Logout of the system | High |
| View Student Profile | High |
| View Pending Claim Requests | High |

| | |
|---|---|
| Approve/Reject Claim Requests | High |
| View Claim Approval History | High |
| View Details of both Lost and Found Items in a Claim | High |
| Create, View, Edit and Delete Item Characteristics | High |
| User: Student | |
| Register New Account | High |
| Login to the System | High |
| Logout of the System | High |
| Reset Password if Forgot the Password | Medium |
| View Profile Information | Medium |
| Edit Profile Information | Medium |
| Report Item as Lost/ Found | High |
| Upload Image for Lost/ Found Item | Medium |
| View all their Reported Items | High |
| Edit their Reported Items (for Items not involved in any claim request) | Medium |
| Delete their Reported Items (for items not involved in any claim request) | Medium |
| Search for Reported Items using Keywords | High |
| Filter Items | High |
| View Search Results | High |
| Submit Claim Request for Found Items | High |
| Provide Justification during claims (if any) | Medium |
| Track the Status of Submitted Claims | High |
| Notified on New Potential Matches for their Lost Items | Medium |
| View Potential Matches for their Lost Items | Medium |

## 3.3 System Design Phase

The focus of the system design phase is to convert requirements into blueprints which can guide the development process. Designs like system architectural design, activity diagram and class diagrams are created in this phase. System architectural design is developed to provides a high-level structure of the system, showing the interactions between different system components. Apart from that, activity diagram is built to illustrate the system workflow clearly and help to understand more complex system logic with more ease. Finally, class diagram is designed to outline the system structure clearly by modeling the classes, attributes, methods and relationships between classes.

### 3.3.1 Overall System Architectural Design

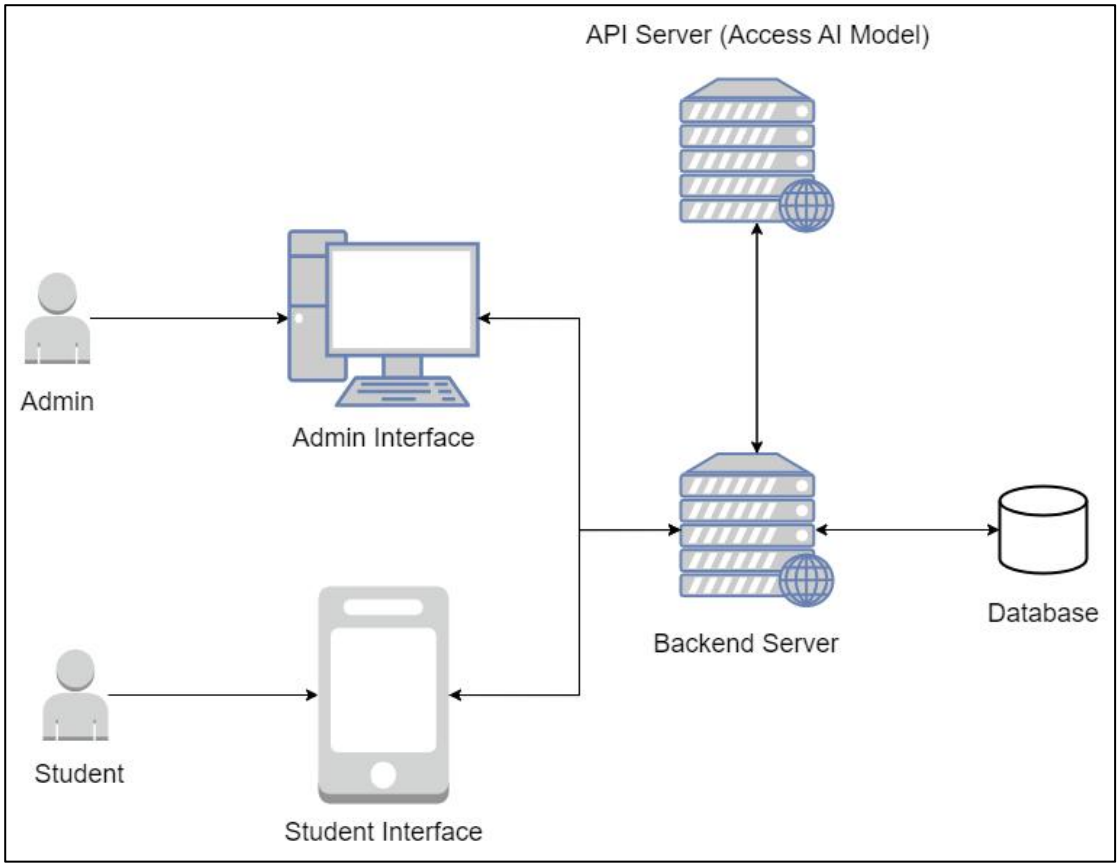



*Figure 3.19: Overall System Architecture of Proposed System (FindIt).*

Figure 3.19 shows the system architecture design for the proposed system, FindIt. There are 2 types of users in this system which are admin and student. Admin use website interface to access the system while student use mobile interface to access the system. A backend server is ready to accept the request from both admin and students and return corresponding data or results. Besides that, the web server also will store and protect the data. Furthermore, users interact indirectly with the shared database via backend server which plays the role to communicates with the shared database to store and retrieve data. Lastly, backend server plays the role to send request to api server to access the AI model when backend server receive request which needs the power of AI model, compare image and calculate similarity score in this case.

### 3.3.2 Activity Diagram

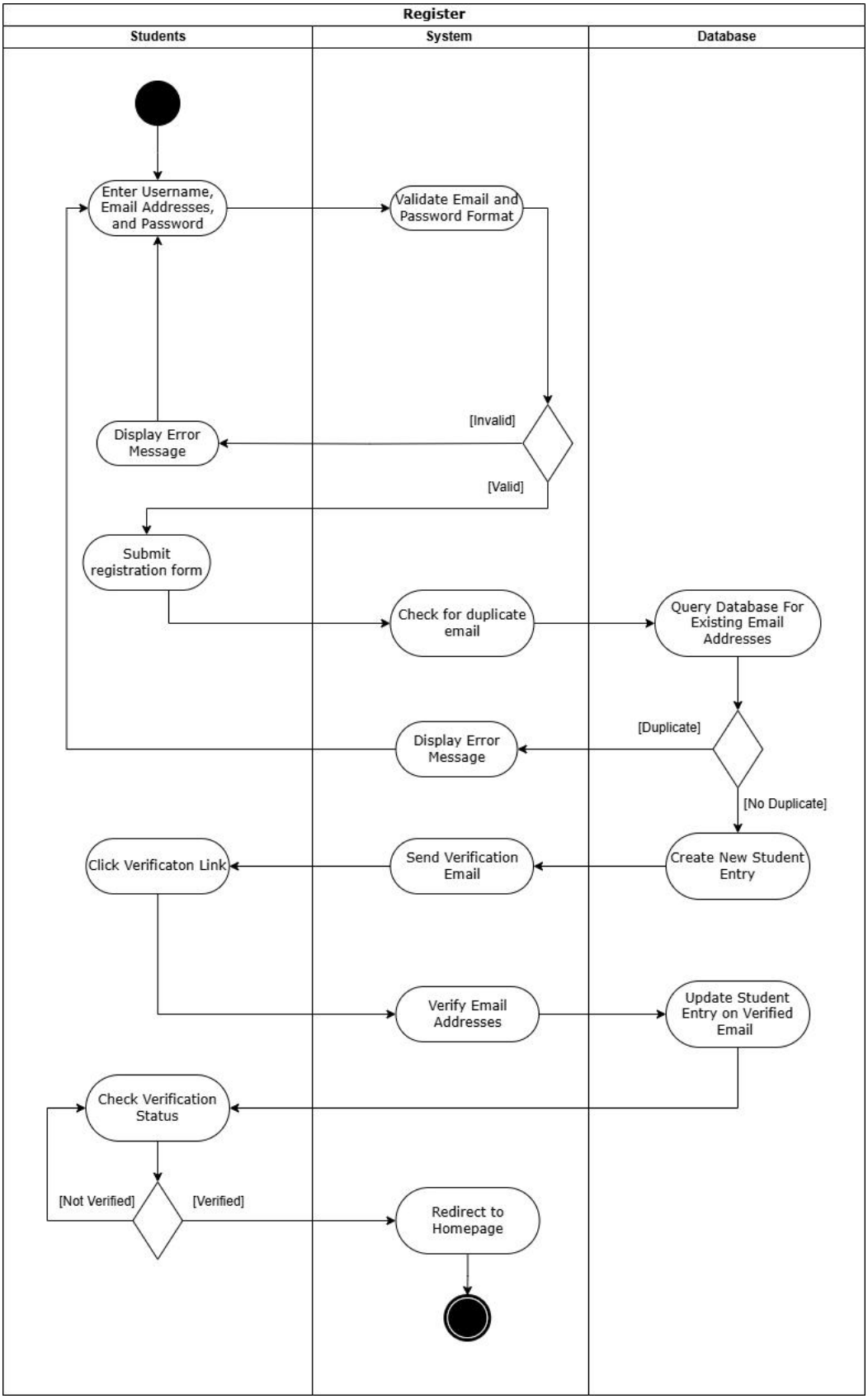



*Figure 3.20: Activity Diagram for Register.*

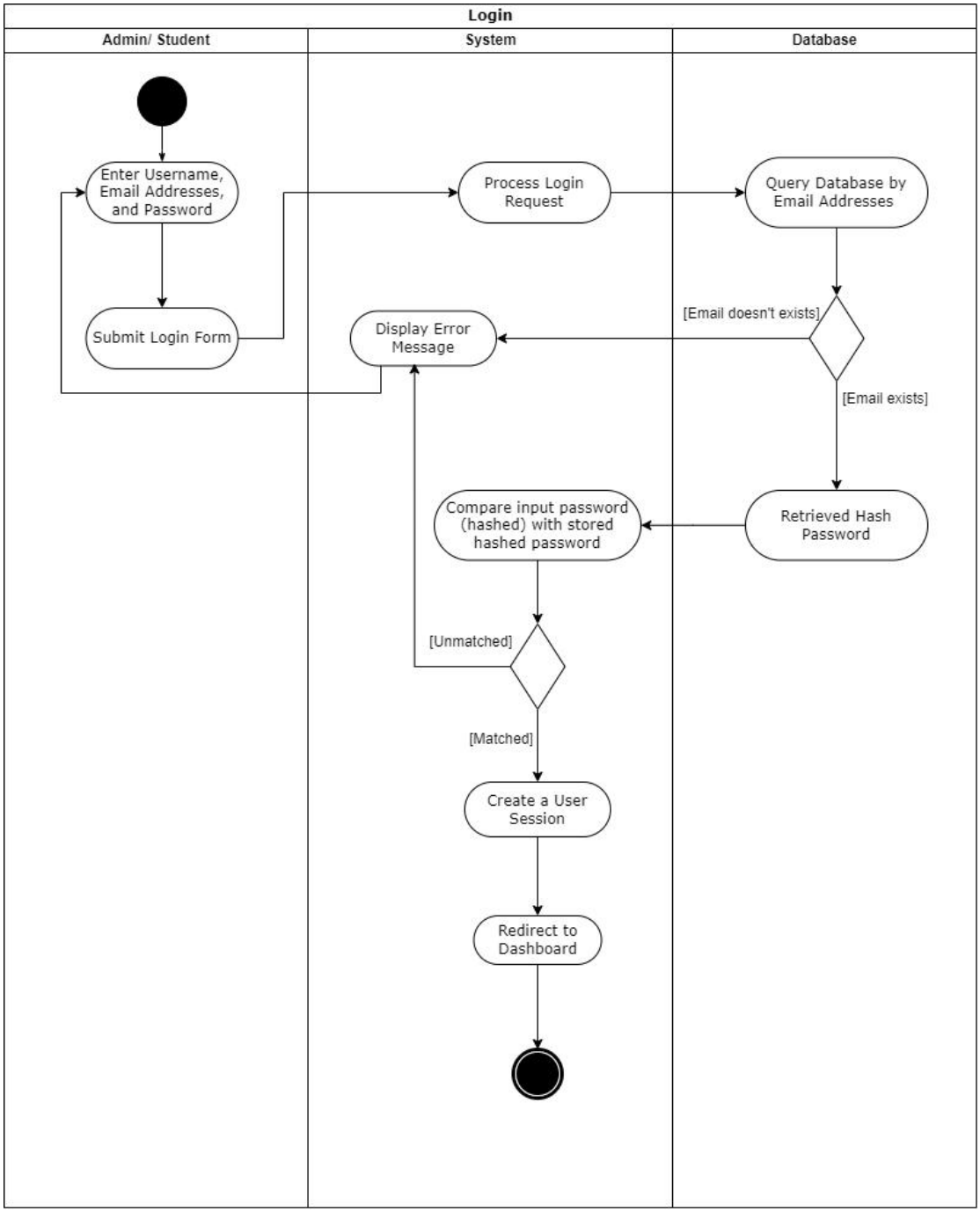



*Figure 3.21: Activity Diagram for Login.*

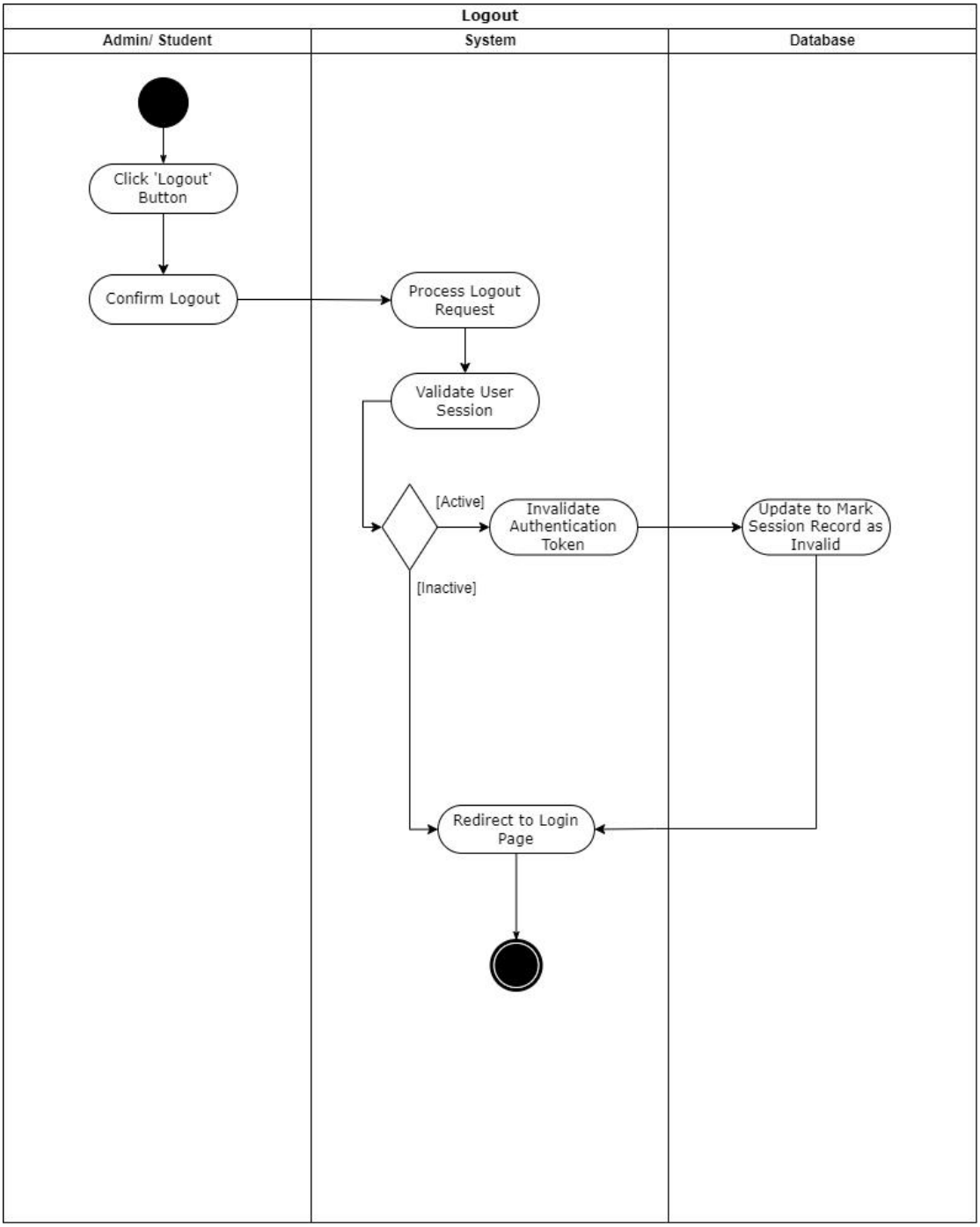



*Figure 3.22: Activity Diagram for Logout.*

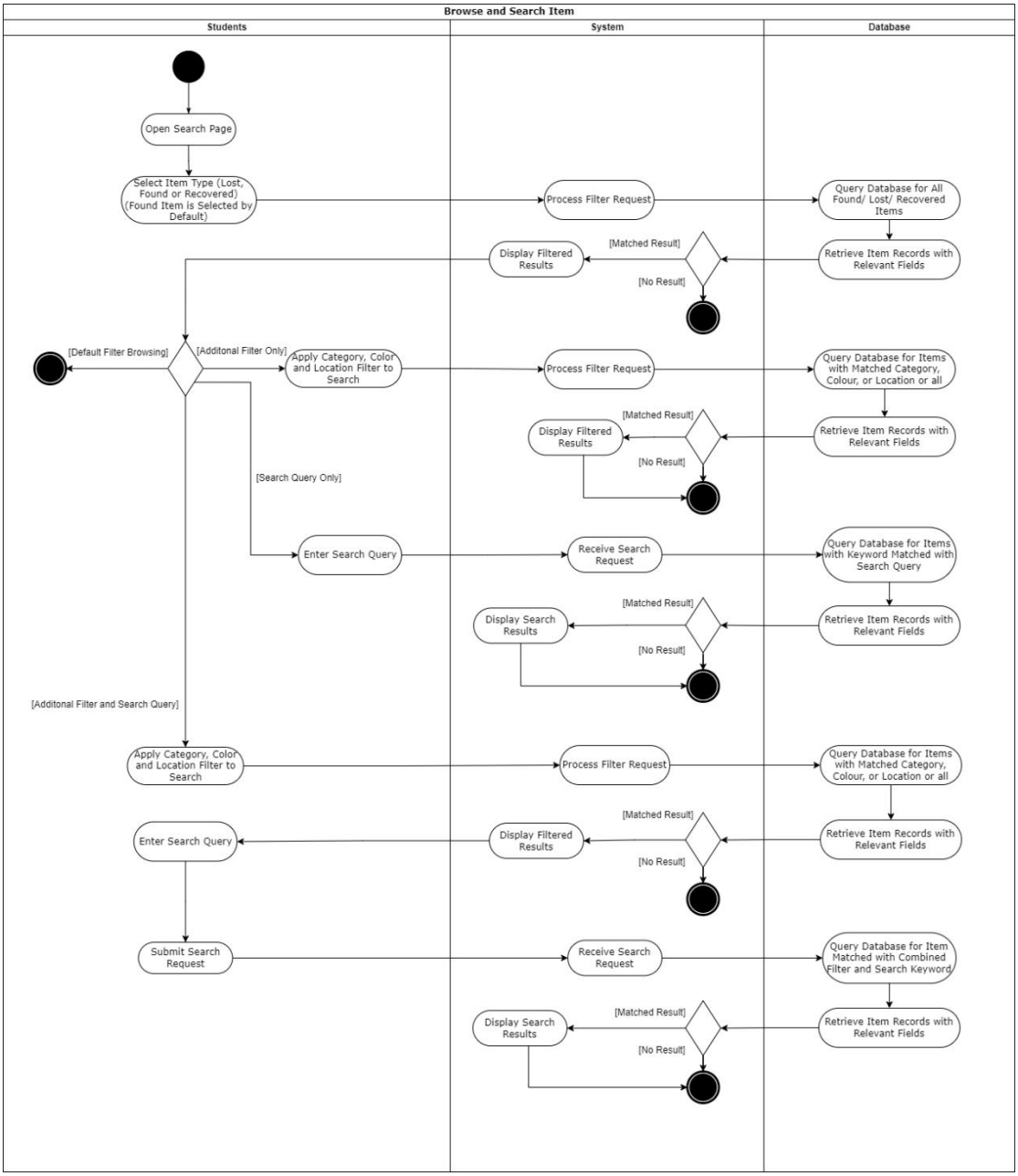



*Figure 3.23: Browse and Search Item.*

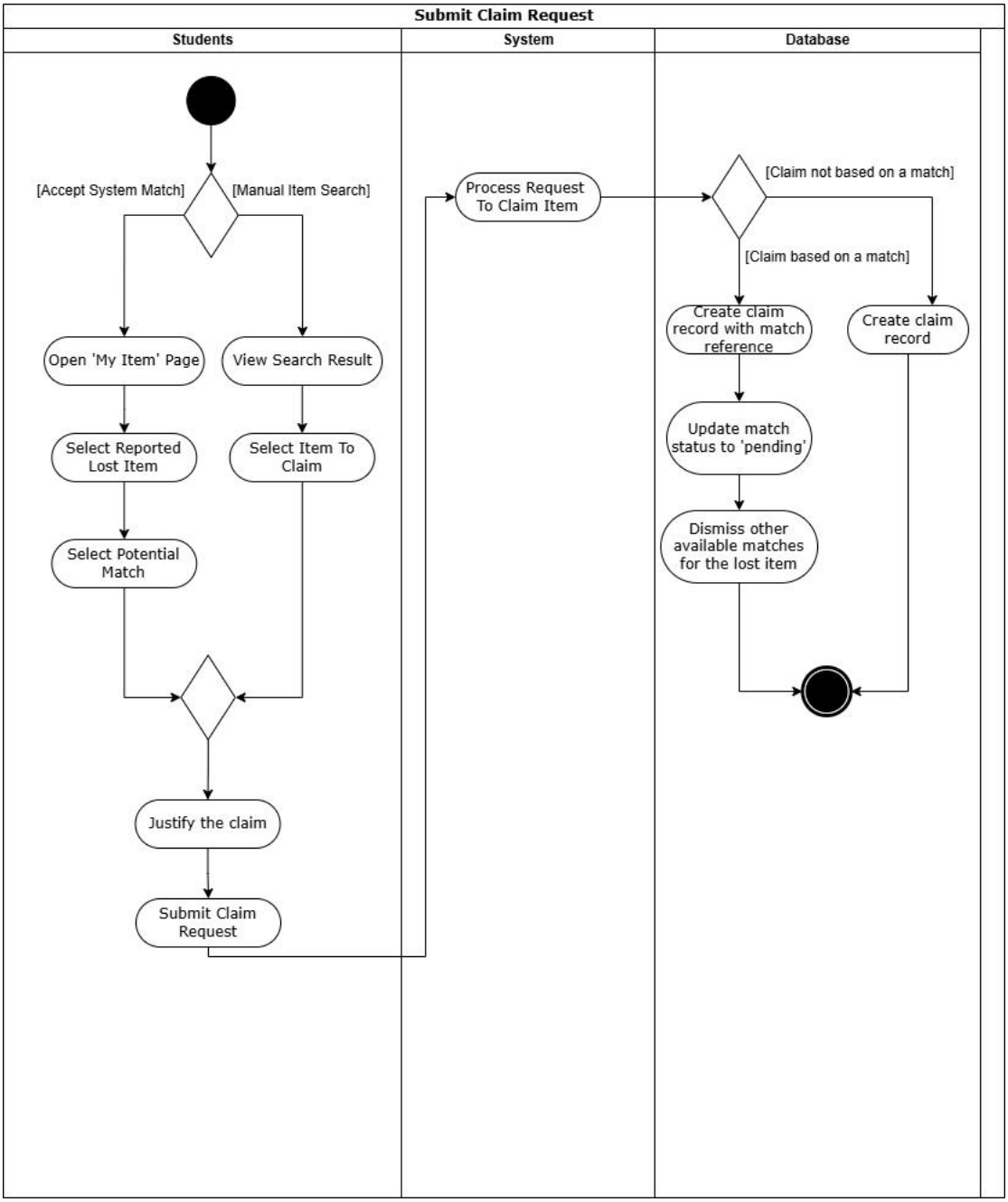



*Figure 3.24: Activity Diagram for Submit Claim Request.*

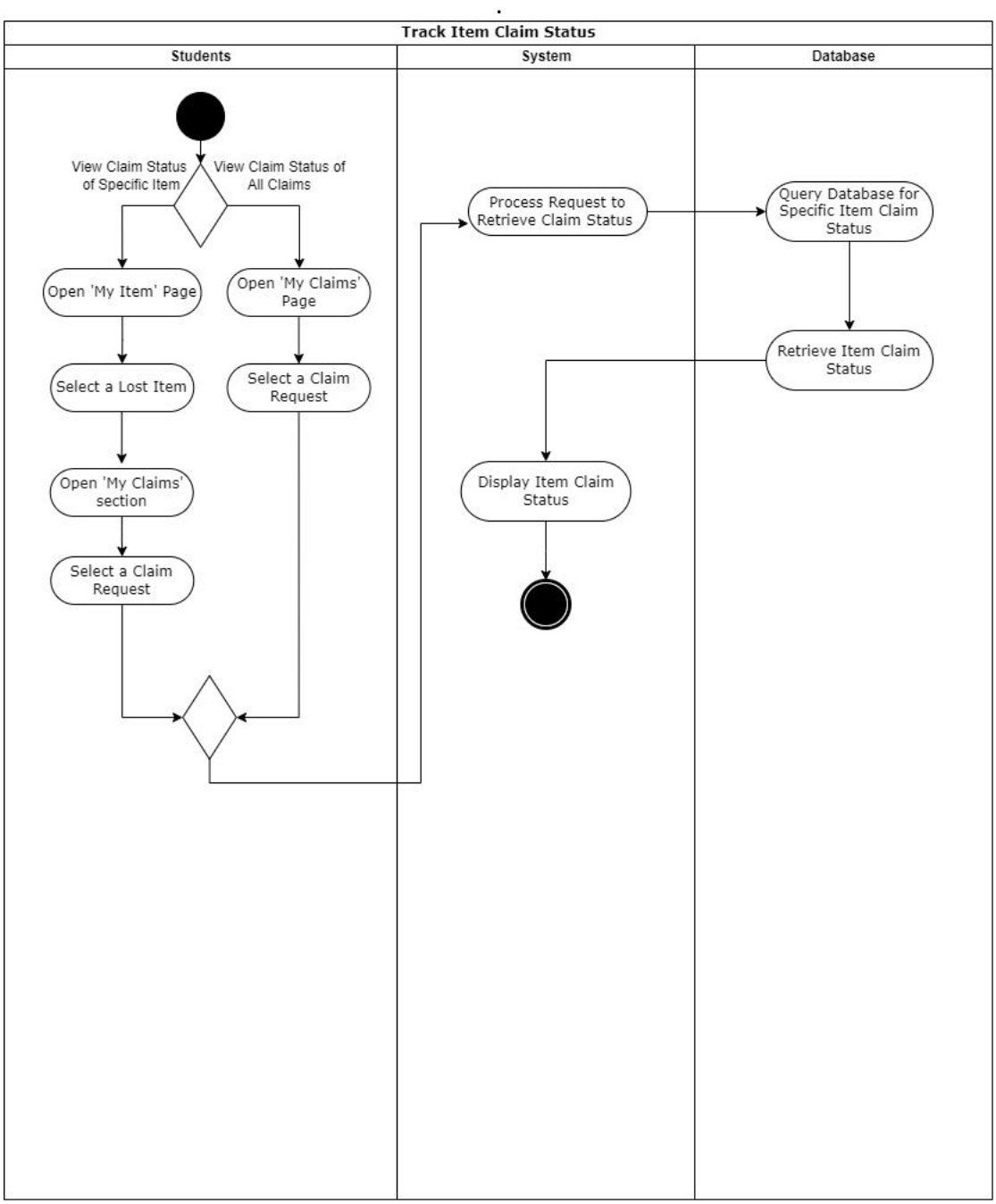



*Figure 3.25: Activity Diagram for Tracking Item Claim Status.*

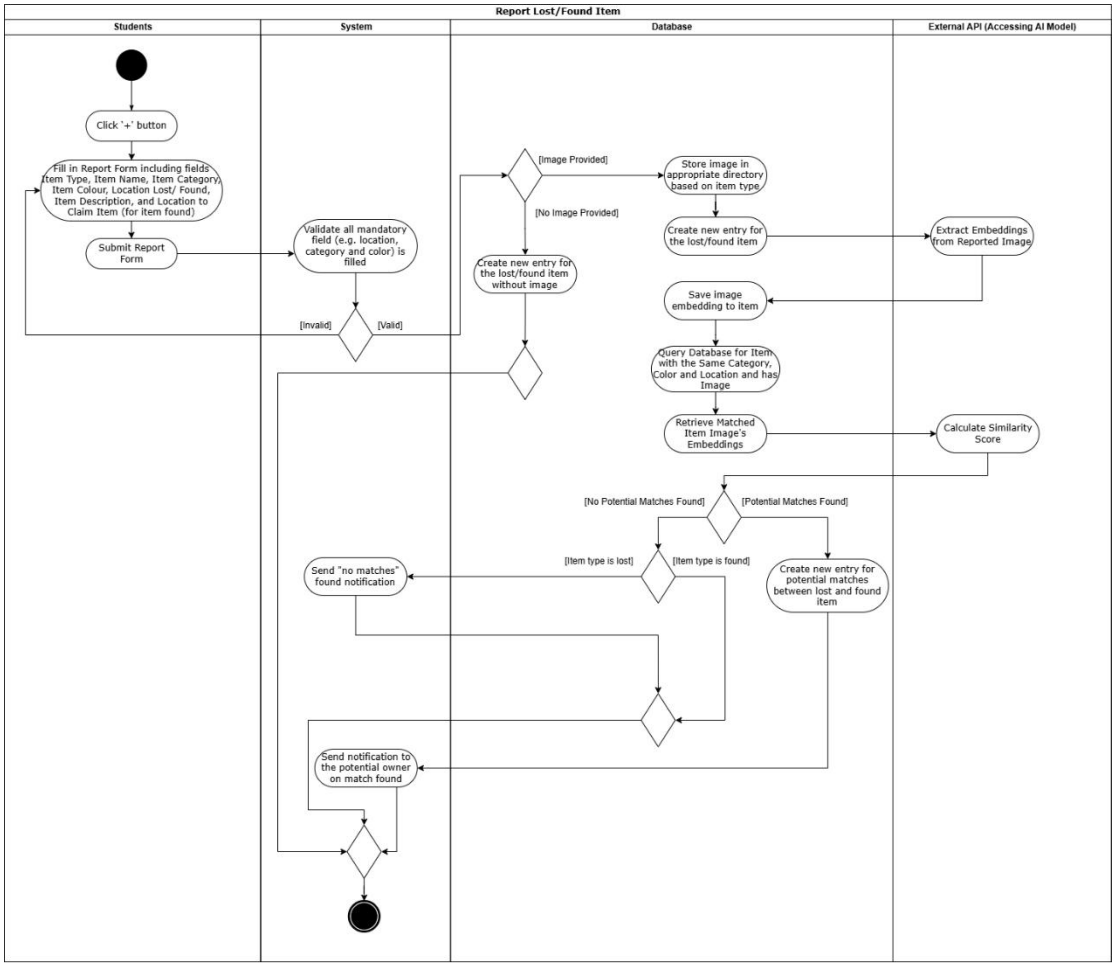



*Figure 3.26: Activity Diagram for Report Lost/ Found Item.*

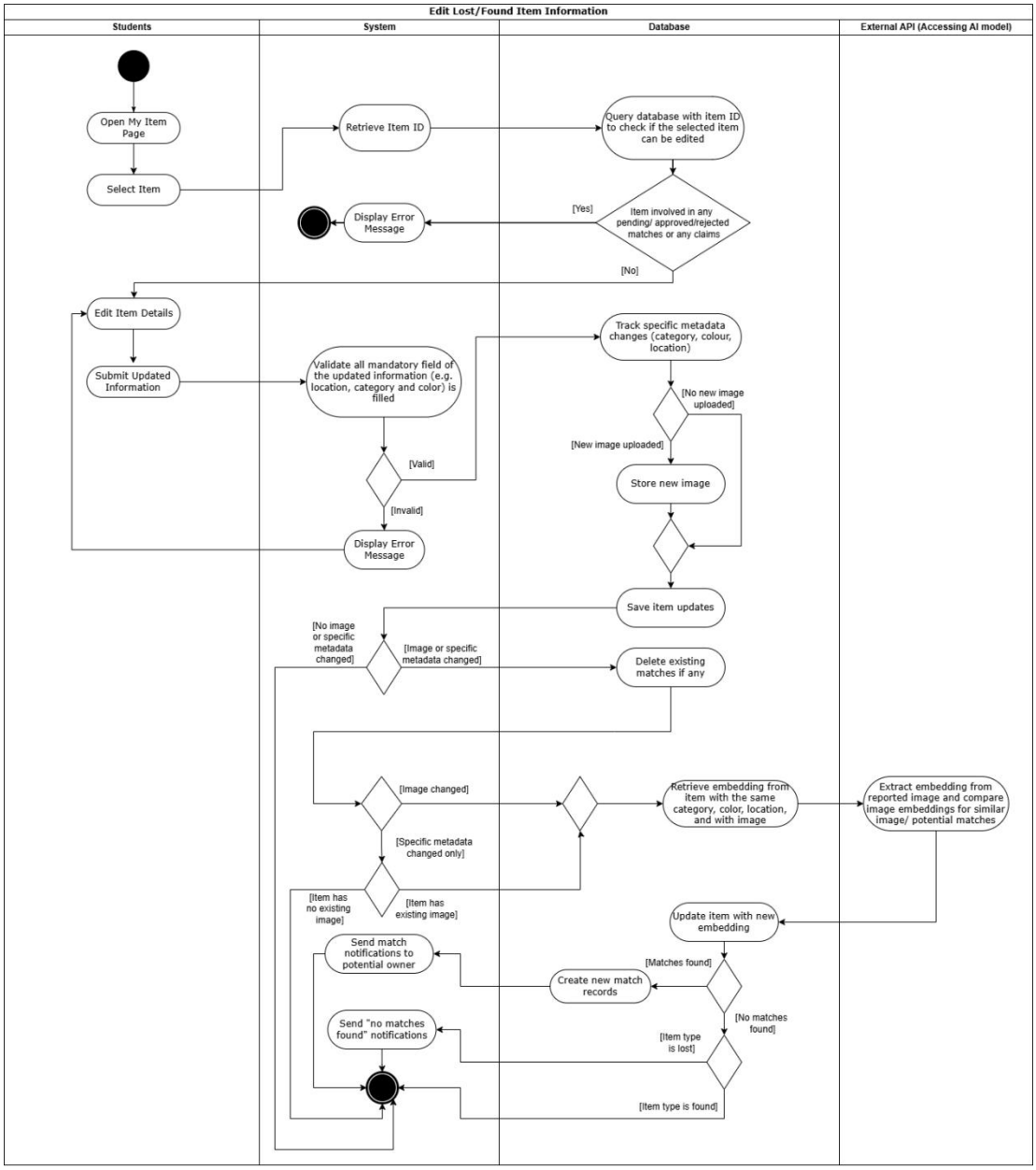



*Figure 3.27: Activity Diagram for Editing Lost/ Found Item Information.*

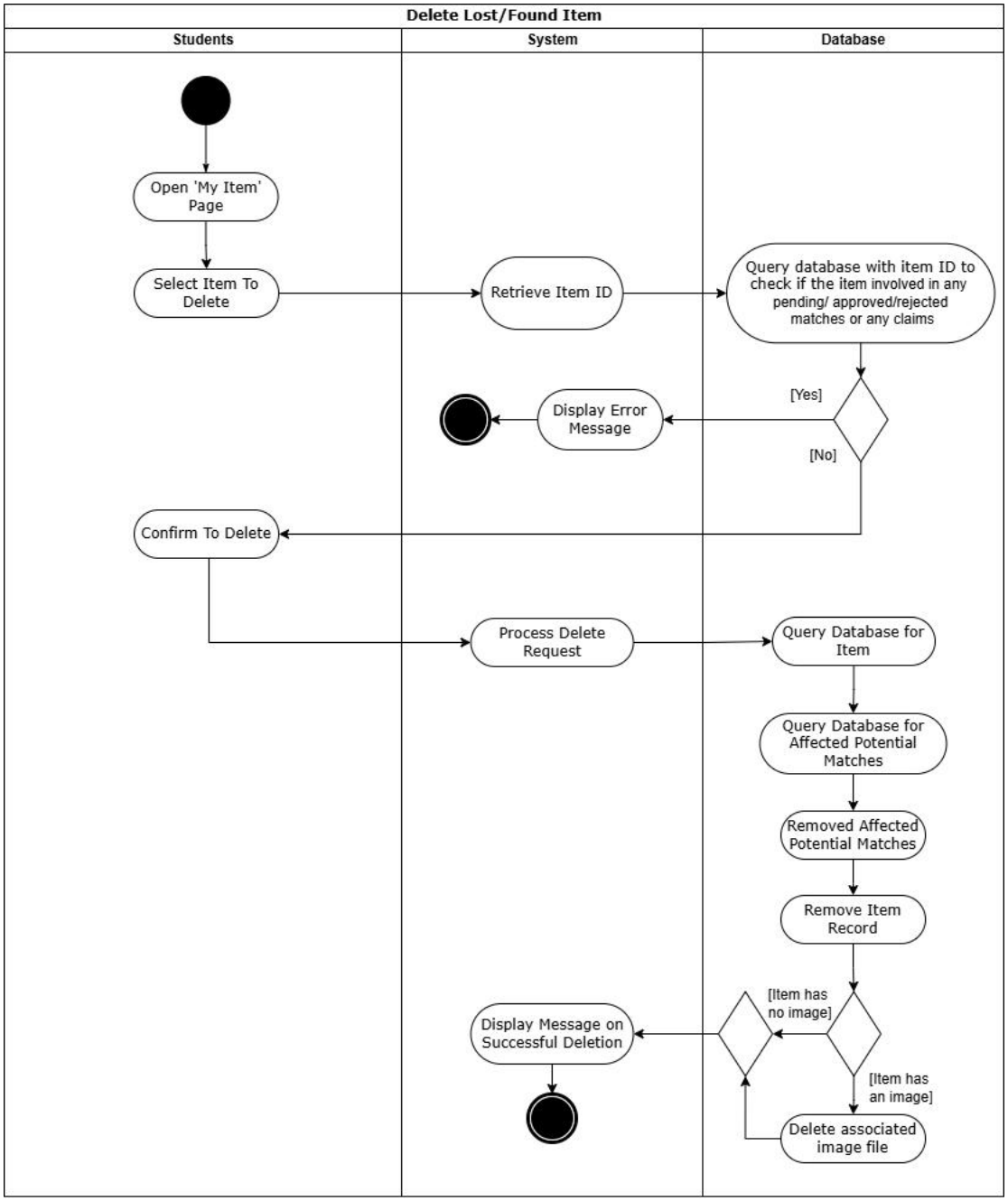



*Figure 3.28: Activity Diagram for Delete Lost/ Found Item.*

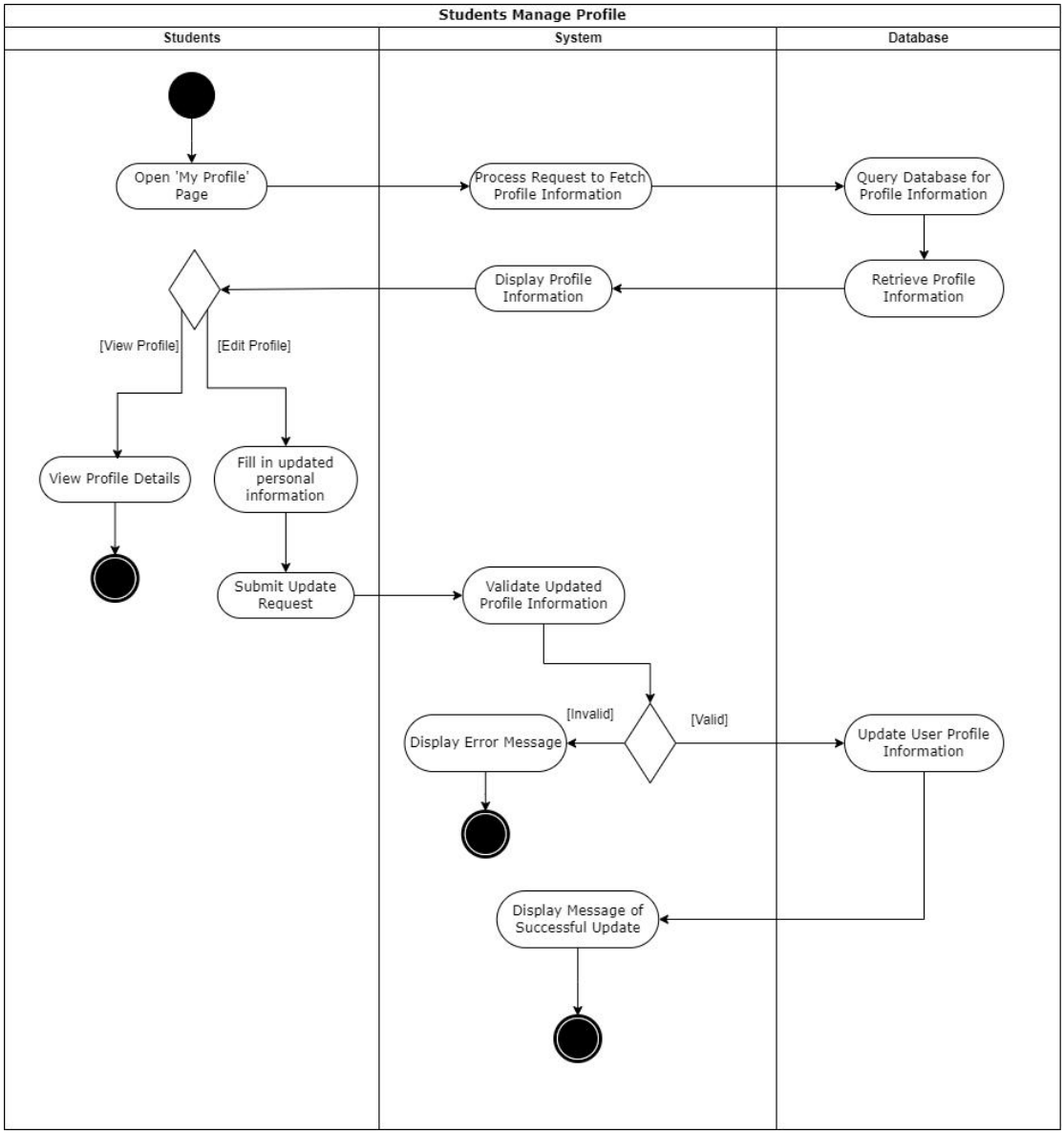



*Figure 3.29: Activity Diagram for Student Manage Profile.*

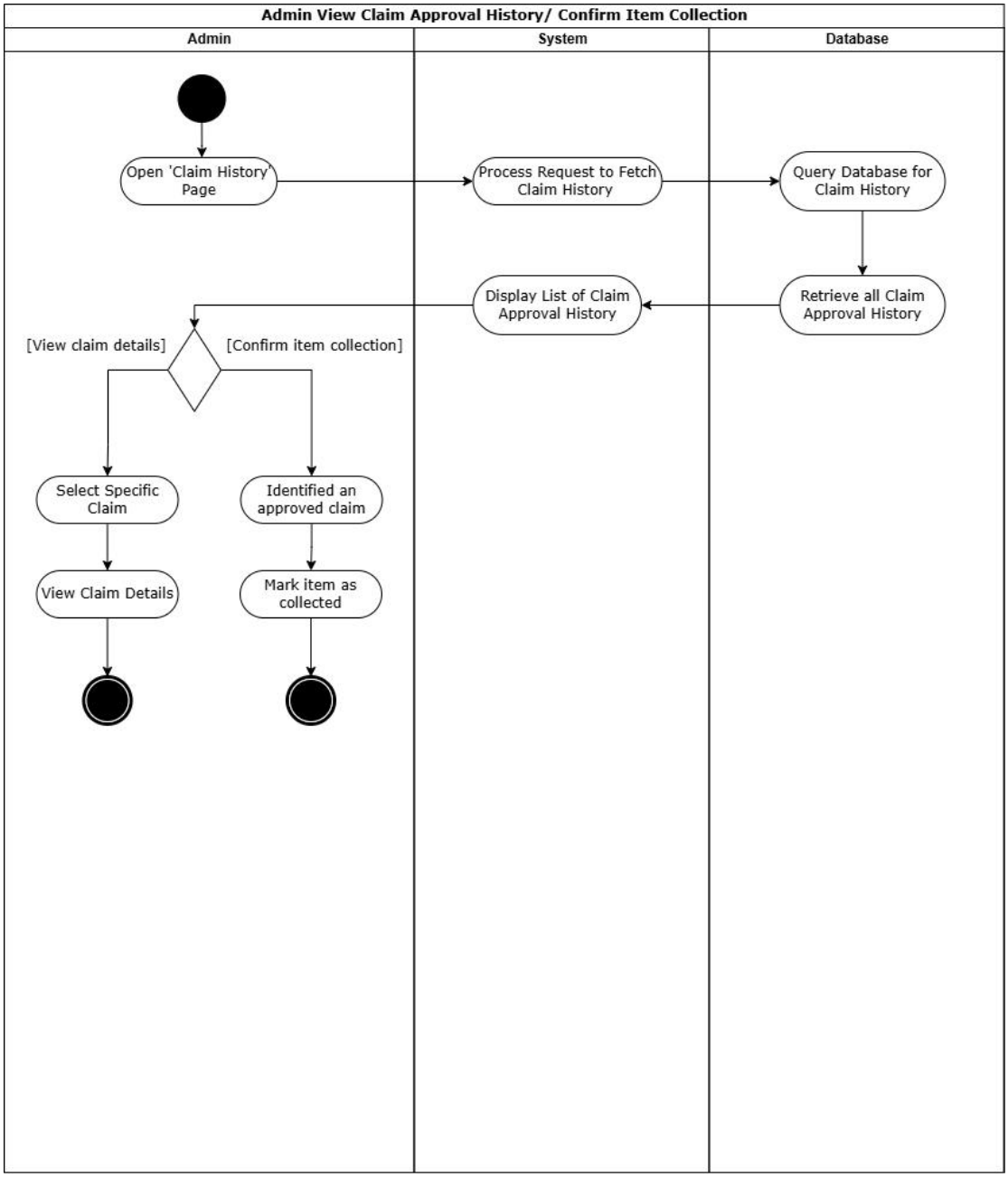



*Figure 3.30: Activity Diagram for Admin View Claim Approval History and Confirm Item Collection.*

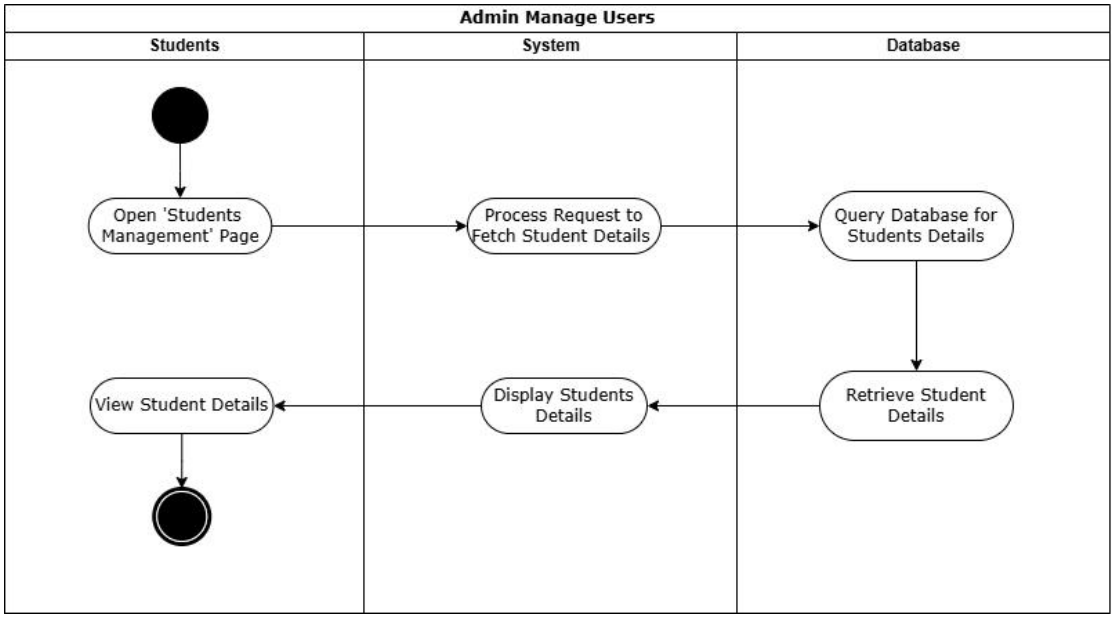



*Figure 3.31: Activity Diagram for Admin Manage Users*

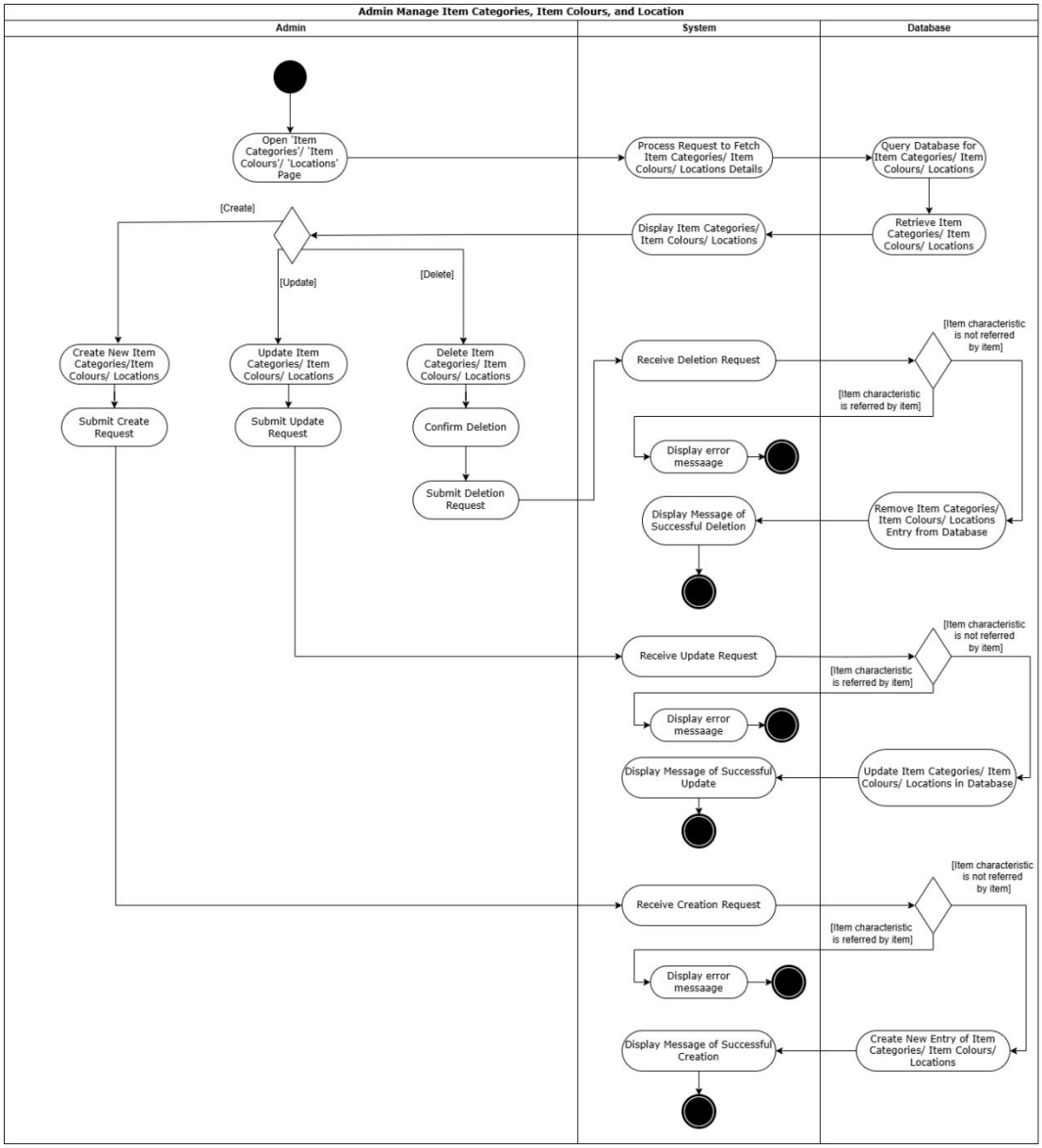



*Figure 3.32: Activity Diagram for Admin Manage Item Categories, Colours and Location.*

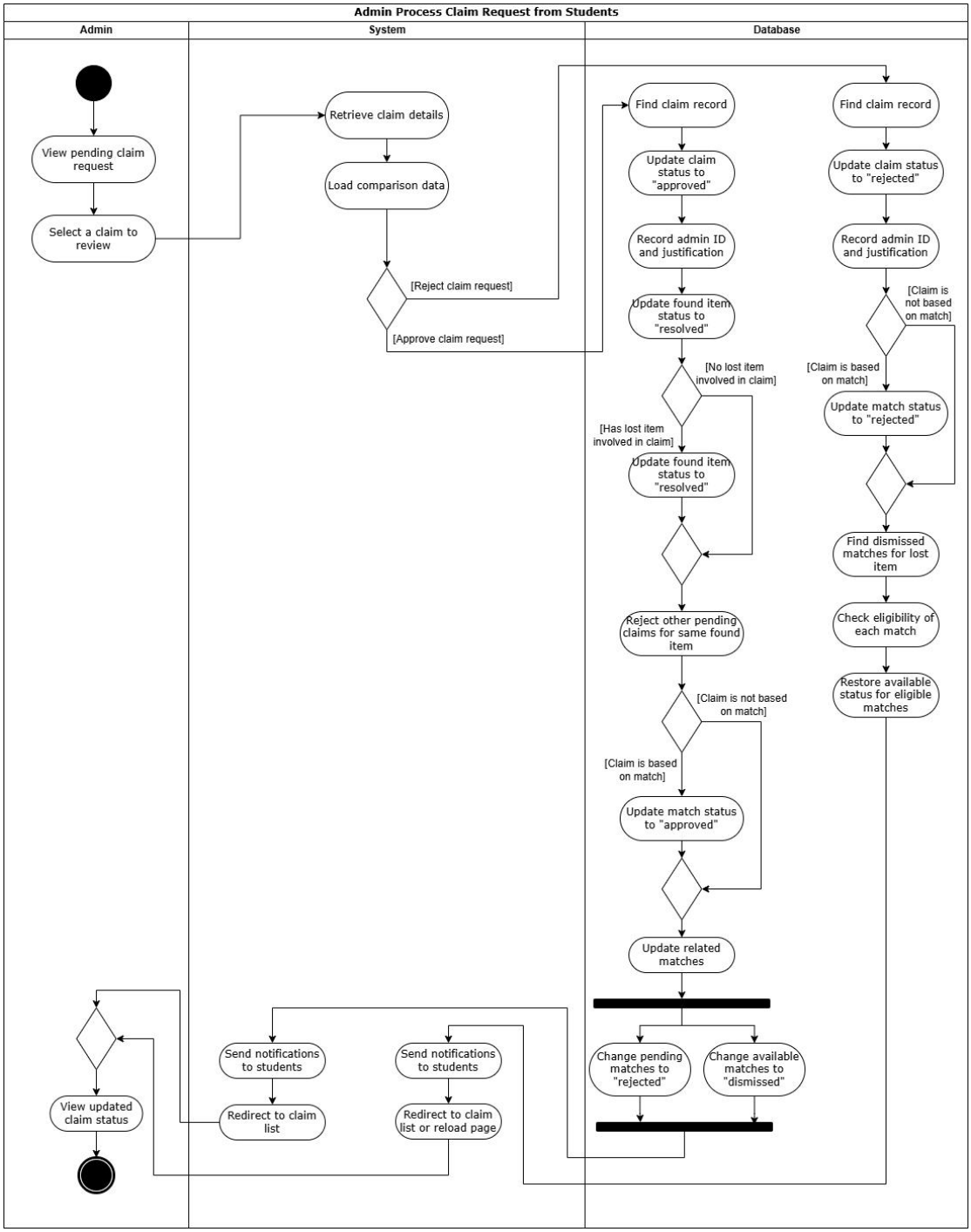



*Figure 3.33: Activity Diagram for Admin Process Claim Request from Students*

### 3.3.3 Class Diagram

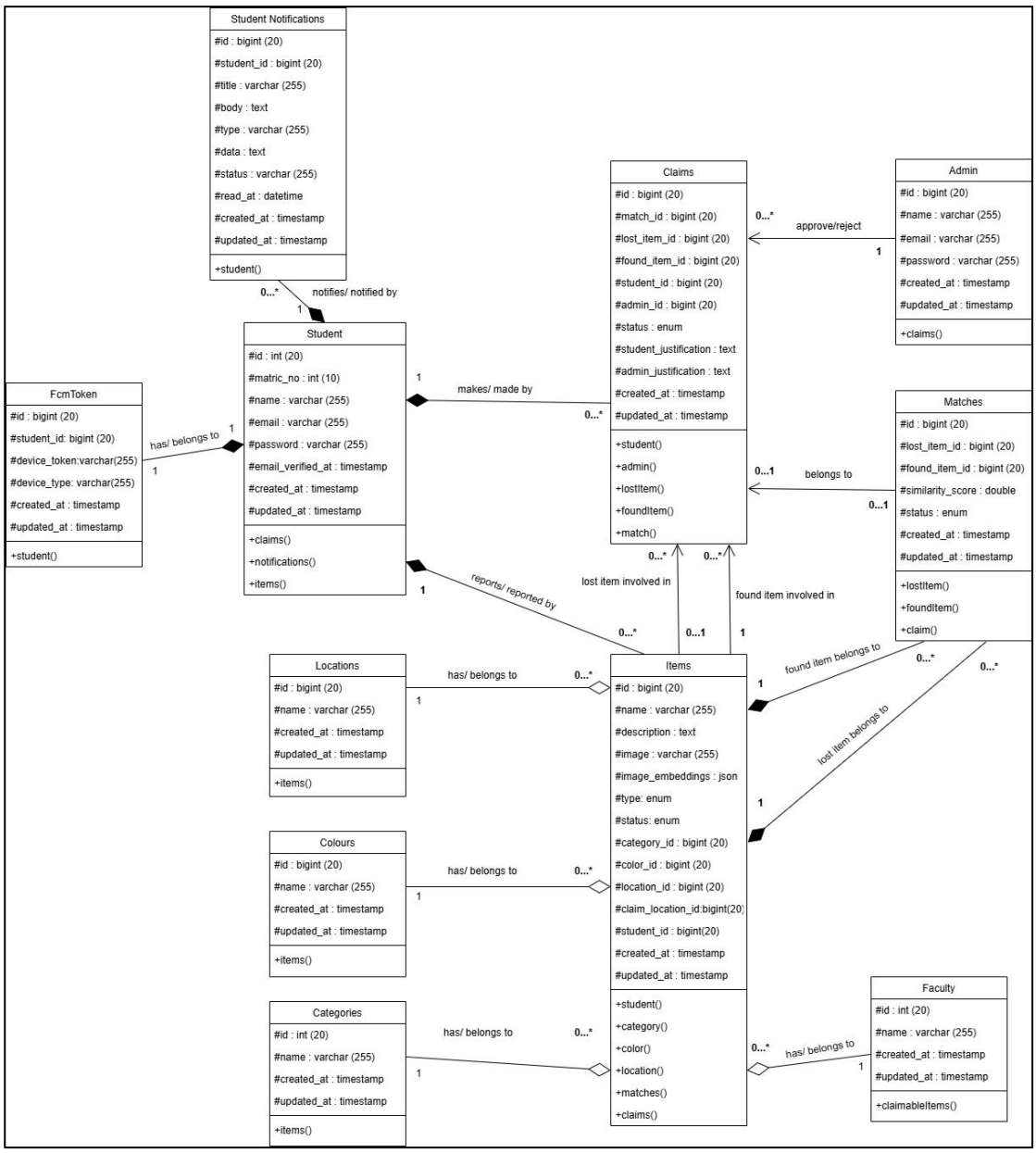



*Figure 3.34: Class Diagram of Proposed System.*

A class diagram is used to represent the structure of the system by giving details about the entities, attributes, methods and relationship between entities. As shown in Figure 3.34, there are nine classes involved in the class diagram, including “Students”, “Admins”, “Items”, “Locations”, “Categories”, “Colors”, “Claims”, “Matches” and “Notifications”, “FcmToken”, and “Faculty”.

The “Student” class contains attributes such as “id”, “matric_no”, “name”, “email”, “password”, “email_verified_at”, “created_at”, “updated_at”, with methods “claims()”, “notifications()”, and “items()” which retrieves the claims made, notifications received and items reported by the student. Class “Students” establishes a one-to-many composition relationships with the class “Claims”, indicated by the “1” on Students side and “0...*” on the Claims side because each student can claim none or more found item, while one claim can only be initiated by one student. The composition relationship between the classes shows that a claim cannot exist without the existence of a student. Meanwhile, the “Students” class has a one-to-many compositions relationships to the “Items” class, with “1” on Student side, and “0...*” on Items side because each student can report none or many items in the system, while an item can only be reported by one student only. Apart from that, “Student” class has a one-to-one compositions relationship to the “FcmToken” class, depicting that a student can only have one fcm token at a time to receive notification. The student() method in “FcmToken” class is used to retrieve the student associated with the fcm token. The composition relationship depicts that items cannot without the context of user.

The “Notifications” class contains attributes such as “id”, “student_id”, “title”, “body”, “type”, “data”, “status”, “read_at”, “created_at”, and “updated_at”. The methods “student()” is used to retrieve the students who had received the notifications. Class “Notifications” maintains a many-to-one composition relationships with the “Student” class, indicated by the “0…*” on Notifications side while “1” on Student side. It is because each student can receive none or more notifications, while one notification only sent to one student. The composition relationship shows a notification cannot exist without the linked student.

The “Admin” class has attributes like “id”, “name”, “email”, “password”, “created_at”, and “updated_at”. The class includes methods “claims()” retrieving all the claims approved or rejected by the admin. Class “Admin” has a one-to-many relationships

with “Claims” table, where the cardinality is “0...*” on Claims side and “1” on Admin side because each admin can approve/ reject none or many claims, while a claim can only be approved/ rejected by one admin only.

The “Claims” class has attributes like “id”, “match_id”, “lost_item_id”, “found_item_id”, “student_id”, “admin_id”, “status”, “user_justification”, “admin_justification”, “created_at” and “updated_at”. Methods included in the class are “students()”, “admin()”, “lostItem()”, “foundItem()” and “match()”. They are responsible to retrieve the students who made the claim, admin who approve or reject the claim, the lost item involved in the claim, the found item involved in the claim, and also the match which triggers the claim respectively. “Matches” class has a one-to-one relationship with “Claims” class, indicated by “0…1*” on both side of Matches and Claims because each match can be ignored or accepted by the user one time only, if the claim based on the matches is rejected by the admin, the student cannot submit claim based on the rejected matches again. Also each claim can only involves none or one match only because a claim not necessary submitted based on a potential match suggested by the system.

The class “Matches” contains attributes like “id”, “lost_item_id”, “found_item_id”, “similarity_score”, “status”, “created_at” and “updated_at”, with methods “lostItem()”, “foundItem()” and “claim()” “Items”, with the roles to retrieve the lost and found item involved in a match, and also the claim triggered by the match respectively. Class “Items” has a one-to-many composition relationships with “Matches” table for a single lost item, indicated by “1” on Items side and “0…*” on Matches side. This is because each lost item can be involved in none or many potential matches generated by the system, while a match must involve one lost item. Next, “Items” class has a one-to-many composition relationships with “Matches” class for a single found item, indicated by “1” on Items side and “0…*” on Matches side because each found item can be involved in none or many potential matches

generated by the system, while a match must involved one found item. The composition relationships shows that the matches will be removed from the system once the item involved in the matches is removed.

The class “Items” contains attributes such as “item_id”, “name”, “description”, “image”, “image_embeddings”, “type”, “status”, “category_id”, “color_id”, “location_id”, “claim_location_id”, “student_id”, “created_at”, “updated_at”, and methods such as “student()”, “category()”, “color()”, “location()”, “matches()”, and “claims()”. These methods allow to retrieve the student who reported the item, category and color of the item, location of the item lost or found, matches which involves the item, and also claims involves the item. “Items” class has a one-to-many relationships with “Claims” class for single found item, indicated by “0…*” at Claims side and “1” at Items side because each found item can be claimed by none or many students at the same time, while a claim must mention which found item would be claimed by the student. Furthermore, “Items” class has a one-to-many relationships with “Claims” table for single lost item, indicated by “0…*” at Claims side and “0…1” at Items side . This is because every claim may not include a reported lost item because the student can directly submit claim request for a specific found item without reporting a lost item.

Class “Locations”, “Colours” and “Categories”, “Faculty” have common attributes “id”, “name”, “created_at”, “updated_at” with a method “items()” to retrieve items associated with these classes except that “Faculty” class has method “claimableItems()” instead of method “items()”. The relationships are shown as aggregation, where locations, colors, and categories, and faculty can exist independently but are used to classify multiple items. The relationship between these classes and class “Items” is one-to-many, indicated by “1” on Locations, Colours and Categories, Faculty/Claim location side, and “0..*” on “Items” side. This is because many items may be located at the same place, can be under the same category,

claimed at the same location, and also share the same color. On the other hand, each item can only be reported lost or found at only one specific location, classified under one category, claimed at one location and associated with only one primary color.

### 3.3.4 Wireframe

#### 3.3.4.1 Student User Interface

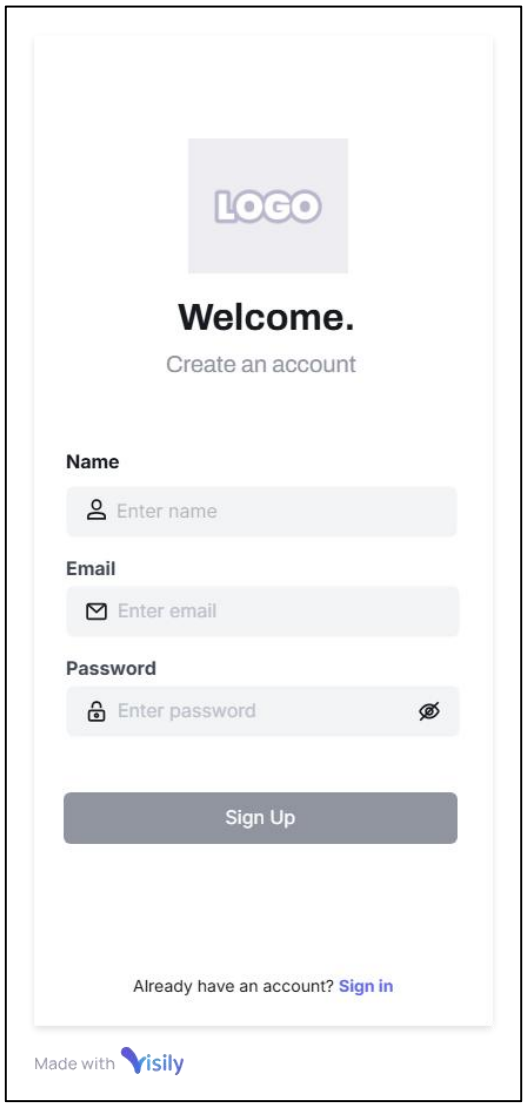



*Figure 3.35: Student Register.*

Figure 3.35 shows a simple user registration screen for a mobile app. The screen includes fields likes name, email, and password, a button to submit registration request, and a link for users who already have an account to "Sign in.".

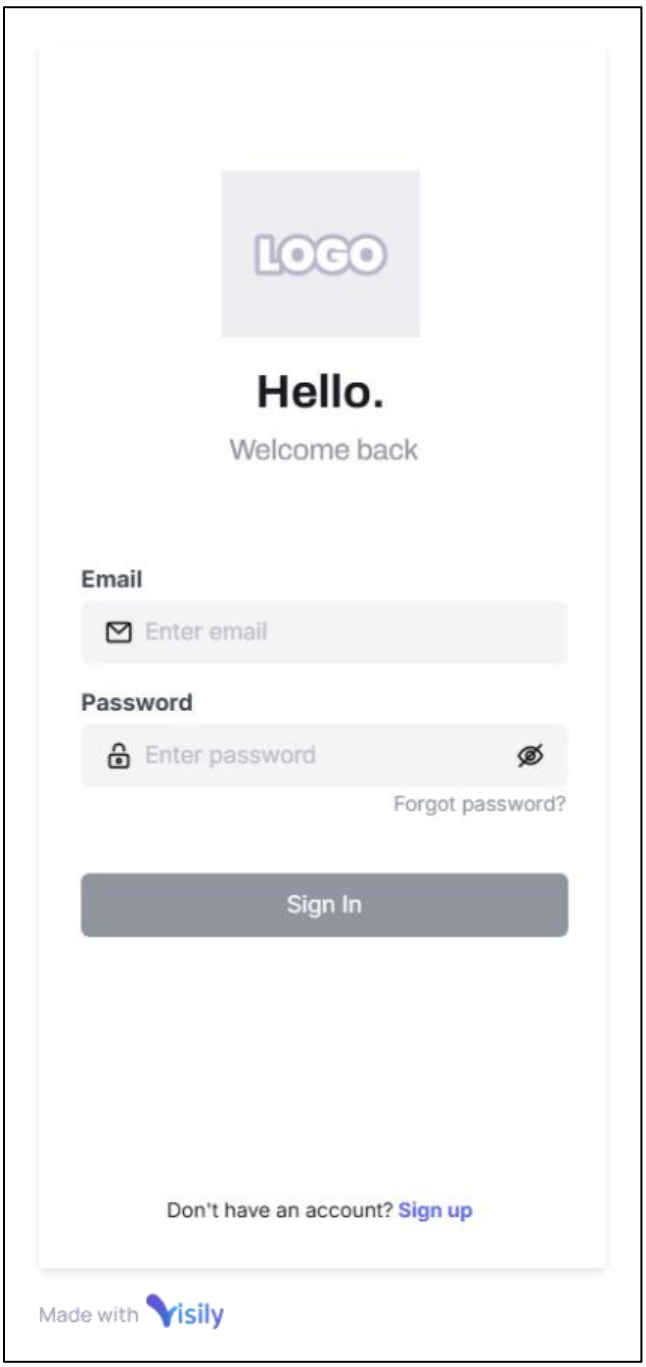



*Figure 3.36: Student Login.*

As shown in Figure 3.36, this is a login screen allows student to enter email and password to login. The user can navigate to 'Sign Up' page by clicking the link 'Sign Up' if the user hasn't created an account yet.

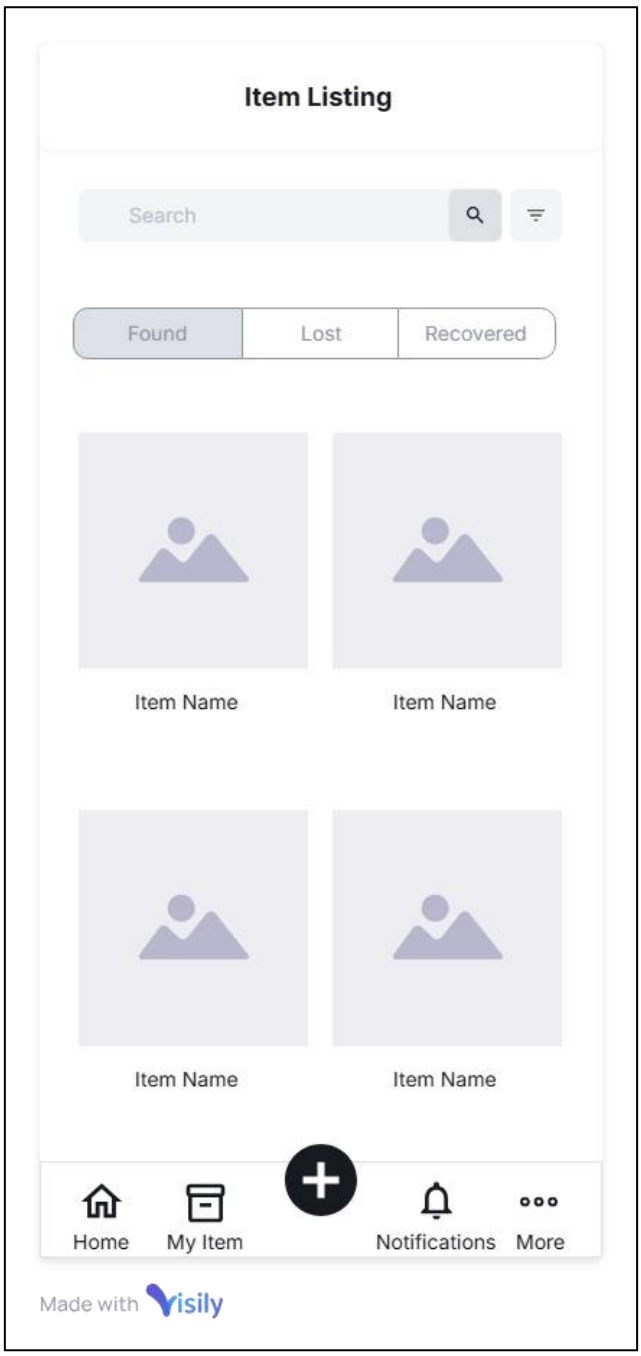



*Figure 3.37: Homepage/ Item Listing Page.*

Figure 3.37 depicts a screen displaying a list of items and allow user to search for and filter the items. The screen includes a search bar, filters for "Found," "Lost," and "Recovered" items. Item name and item images will be shown in this screen to give student a quick overview of the items. A navigation bar at the bottom provides access to "Home," "My Item," "Notifications," and "More" sections.

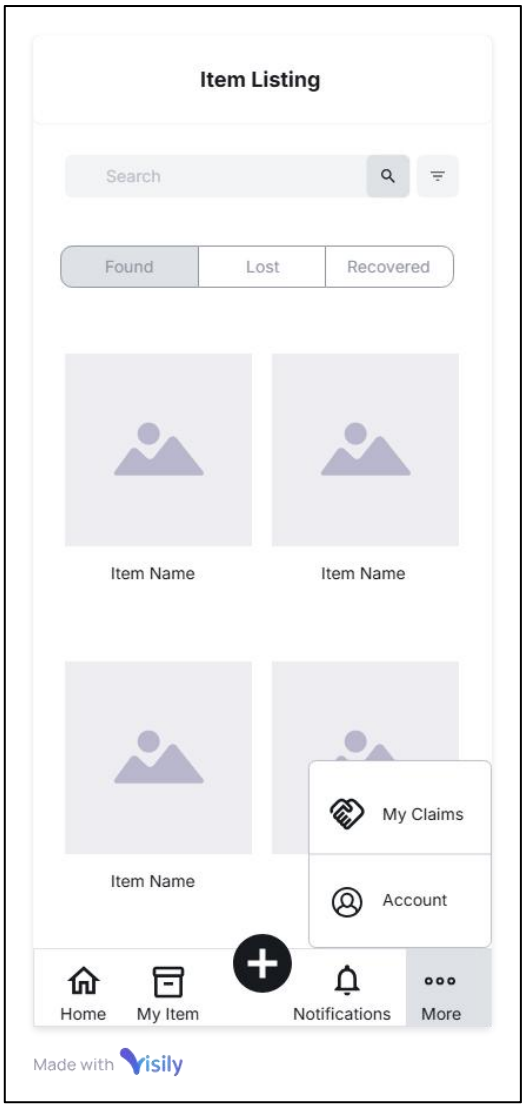



*Figure 3.38: More Option.*

The screen in Figure 3.38 shows that there are options 'My Claims' and 'Account' popped up once the 'More' button is clicked.

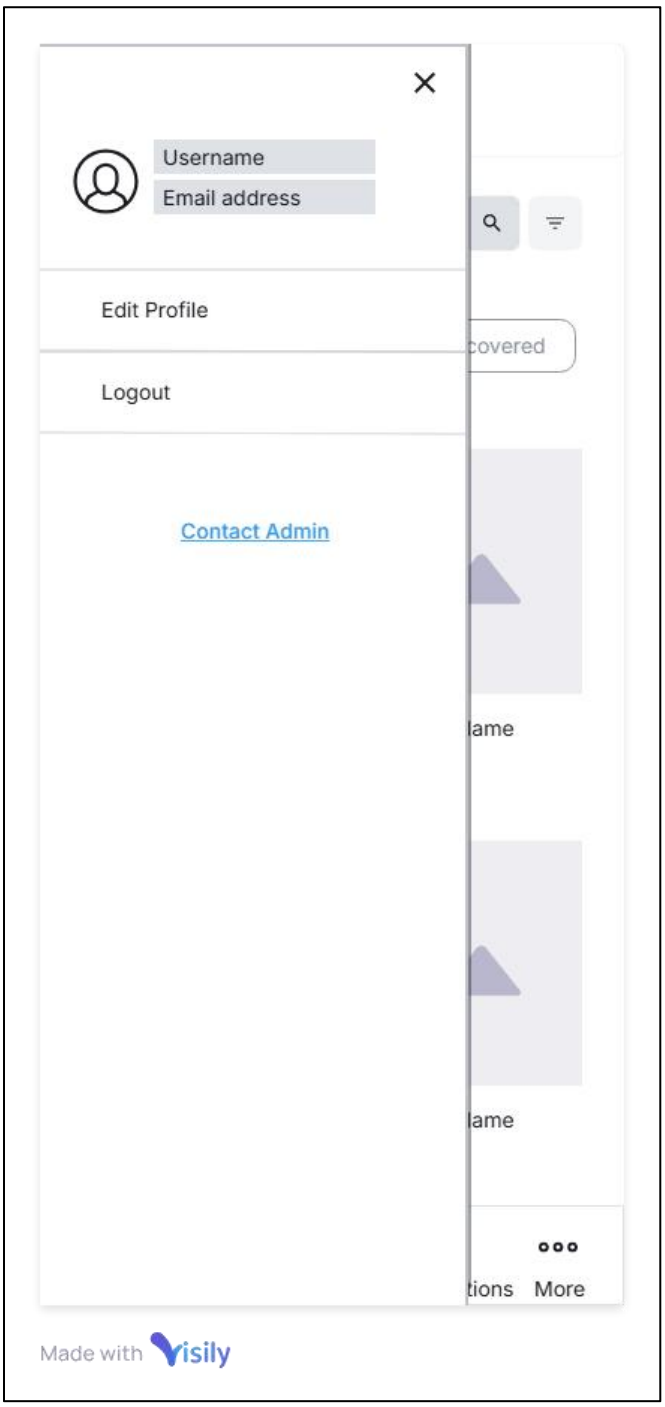



*Figure 3.39: Profile Menu*

The screen in Figure 3.39 depicts a profile menu, where user can have a quick overview of their username and email addresses, and offer option to edit the profile, logout of the system, and a link redirect user to an email application to contact admin.

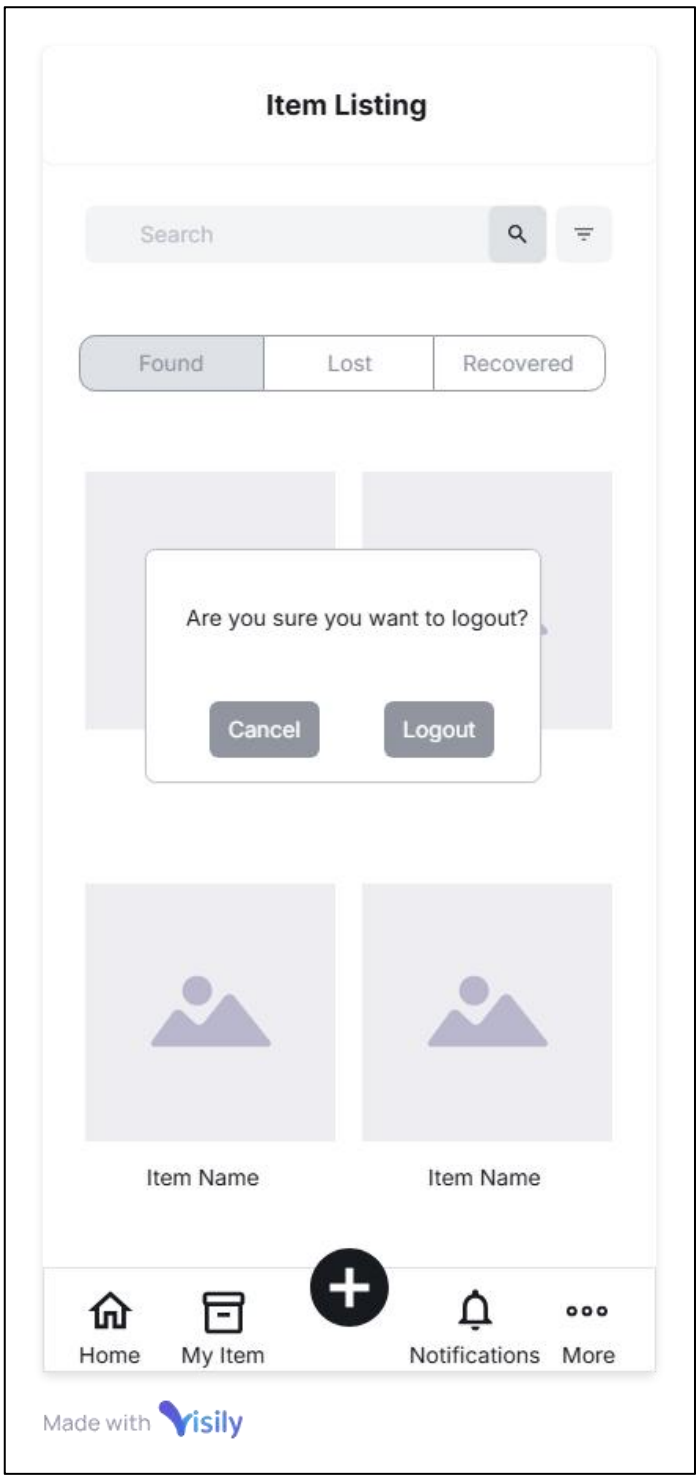



*Figure 3.40: Confirm to logout.*

A confirmation message will be shown to the user as shown in Figure 3.40 to confirm their intents to logout after clicking 'Logout' button to avoid any careless mistakes.

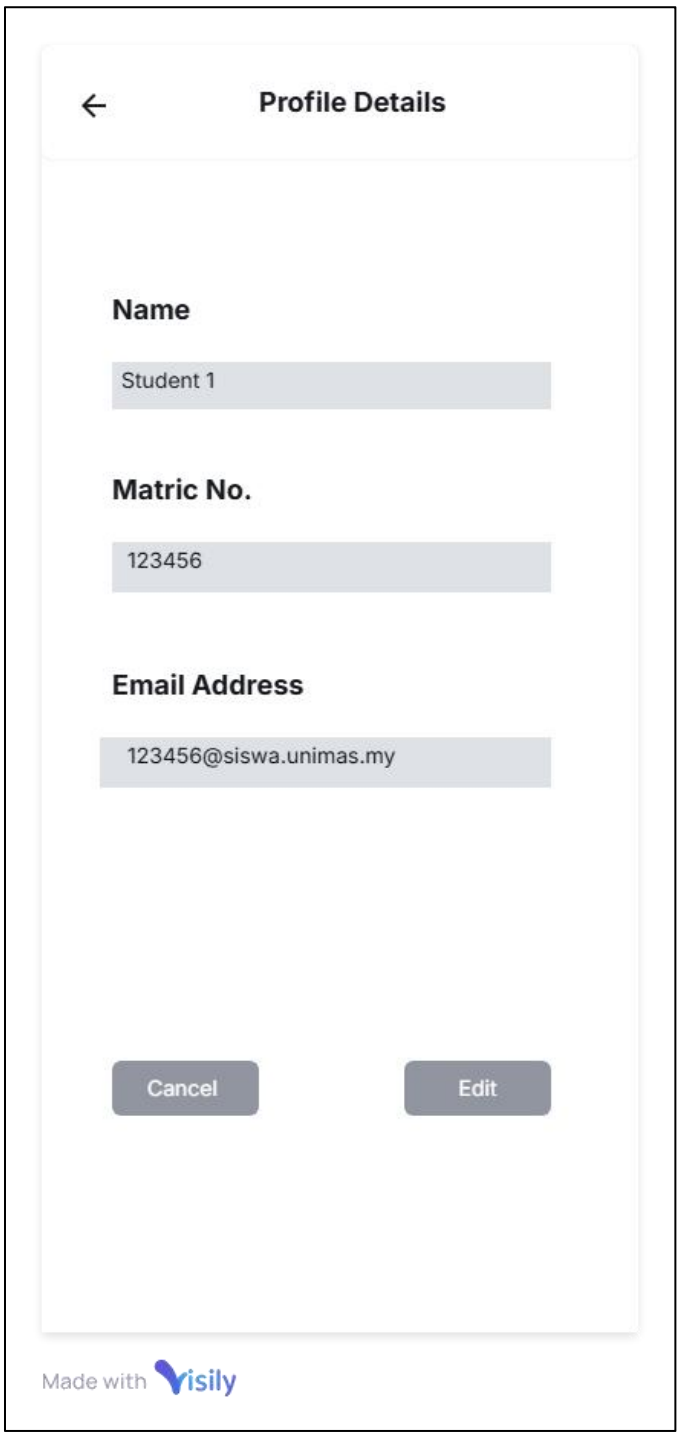



*Figure 3.41: Profile Details.*

Figure 3.41 shows a screen that allow user to edit their profile details. 'Cancel' button is provided to cancel the action of editing and user profile details will be updated once they click 'Edit' button.

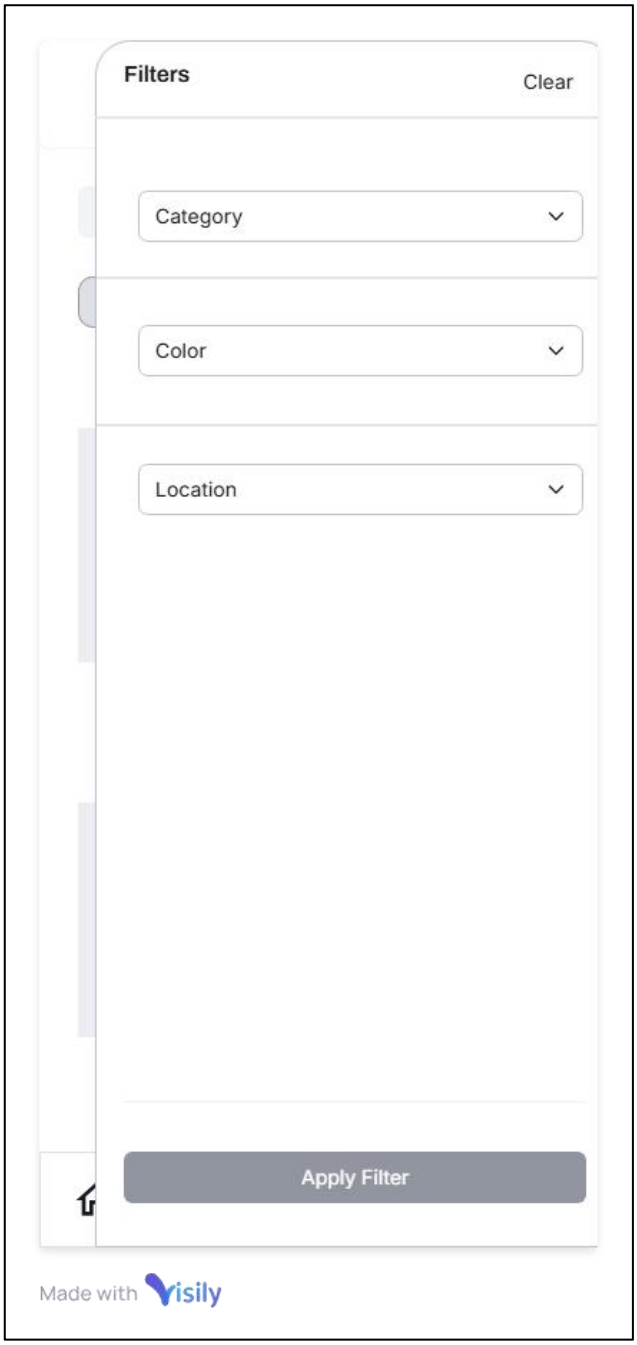



*Figure 3.42: Additional Filter*

There are 3 filters, 'Category', 'Color', and 'Location' for the user to apply to the search result as shown in Figure 3.42. The filter will be applied to the search result right after the 'Apply Filter' button is clicked. All filter will be cleared once the 'Clear' word on the top right corner is clicked.

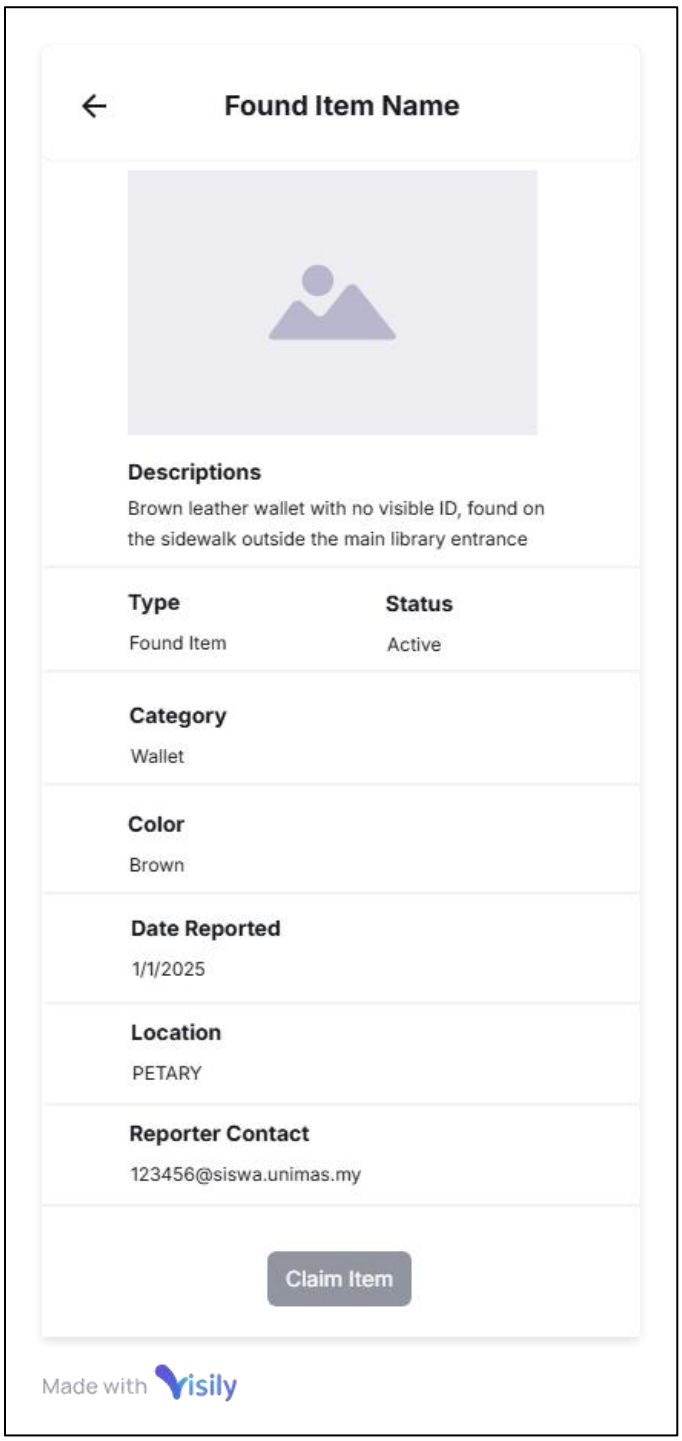



*Figure 3.43: Item Details of a Found Item.*

Figure 3.43 illustrates a screen where the user can view more details for a found item and continue to claim it by clicking the button 'Claim item'.

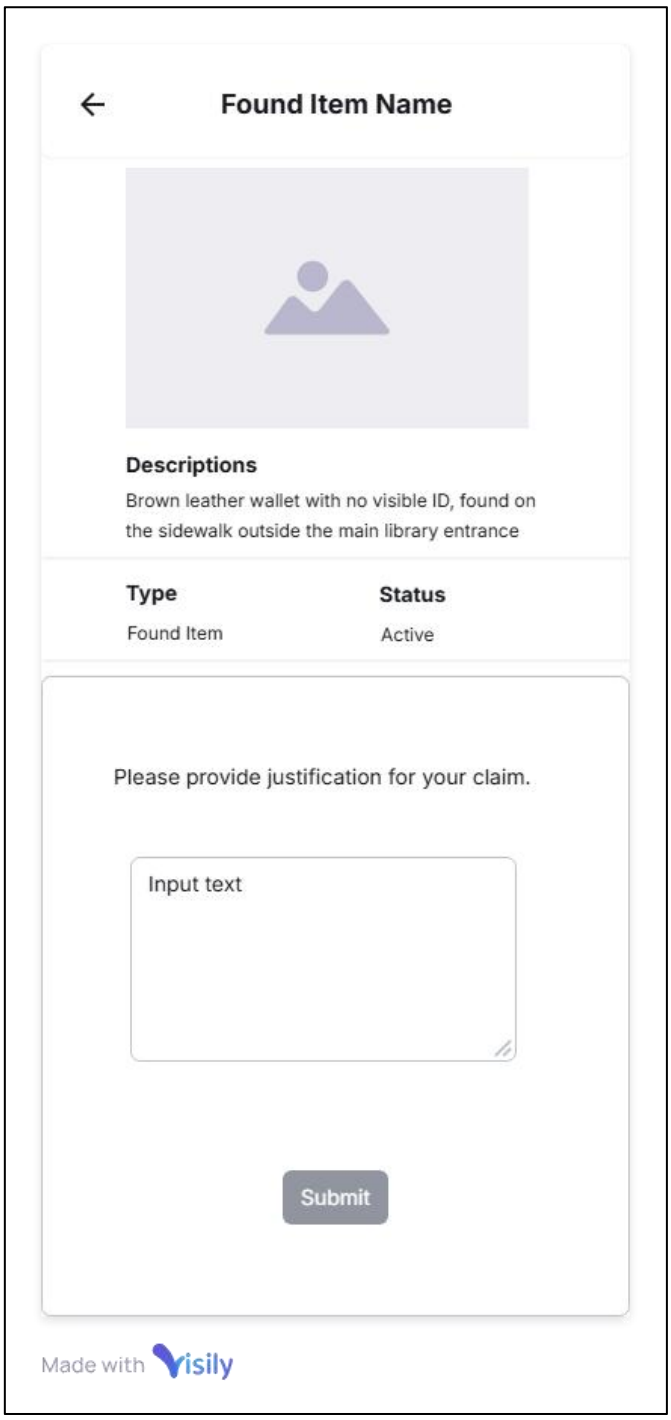



*Figure 3.44: Claim Justification.*

A user has to provide a justification for their claim to convince the admin to approve this claim as shown in Figure 3.44. Clicking 'Submit' button to send a claim request alongside with justification to the admin for approval.

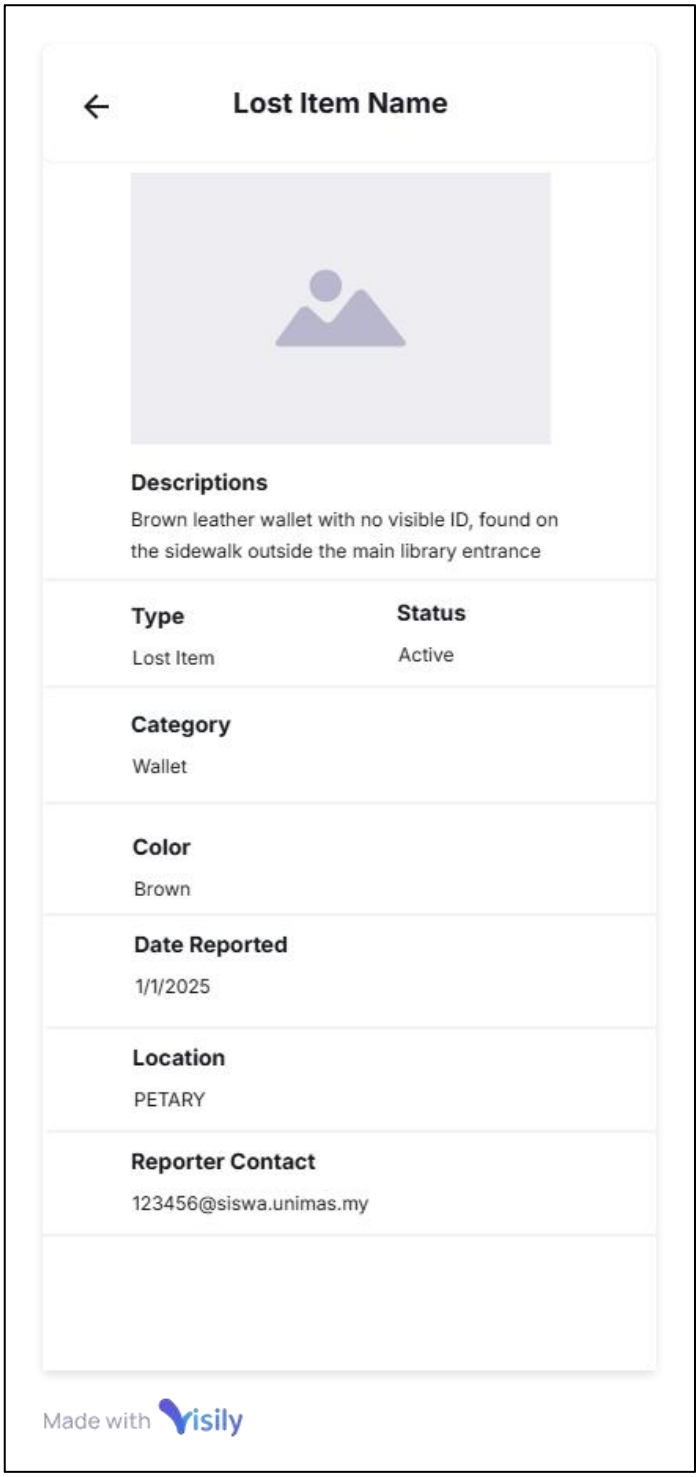



*Figure 3.45: Item Details of a Lost Item.*

Figure 3.45 shows a screen where user can see the details of a lost item.

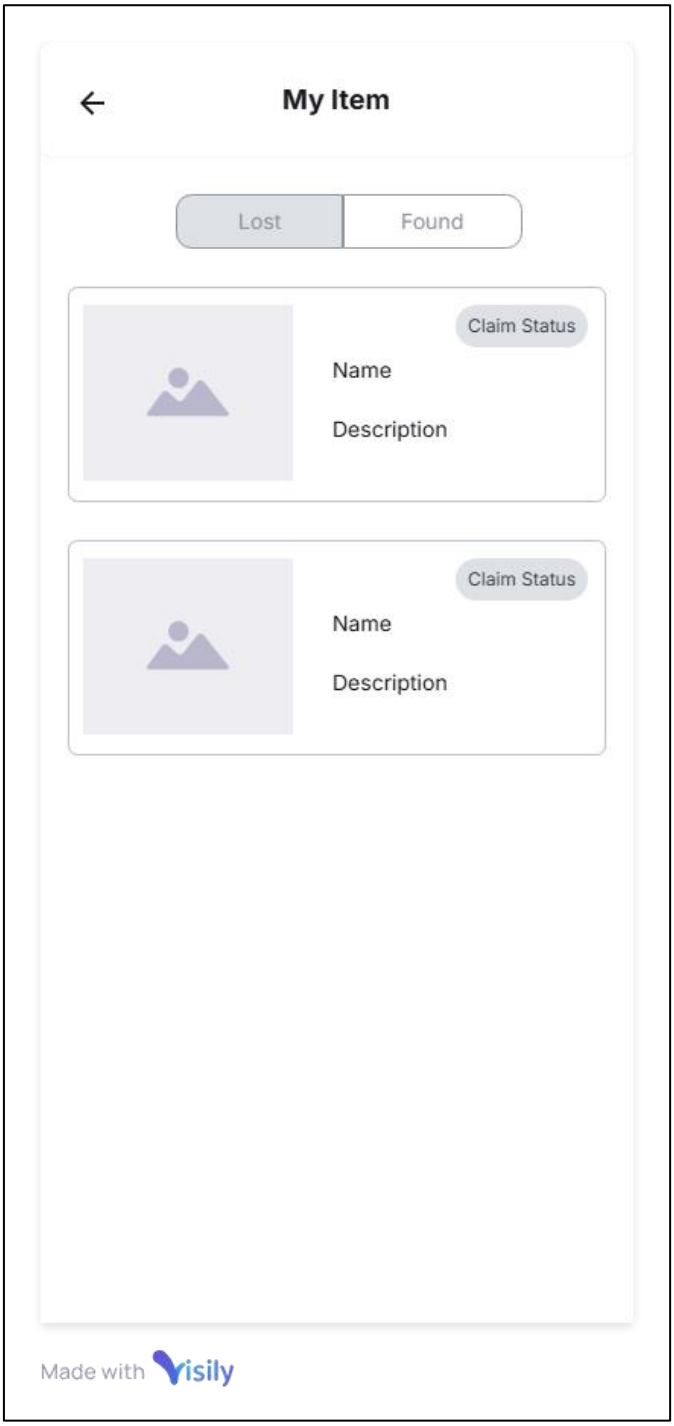



*Figure 3.46: My Item.*

Figure 3.46 shows a screen users can tap on 'Lost' or 'Found' tabs to view all the items they have reported as lost or found respectively.

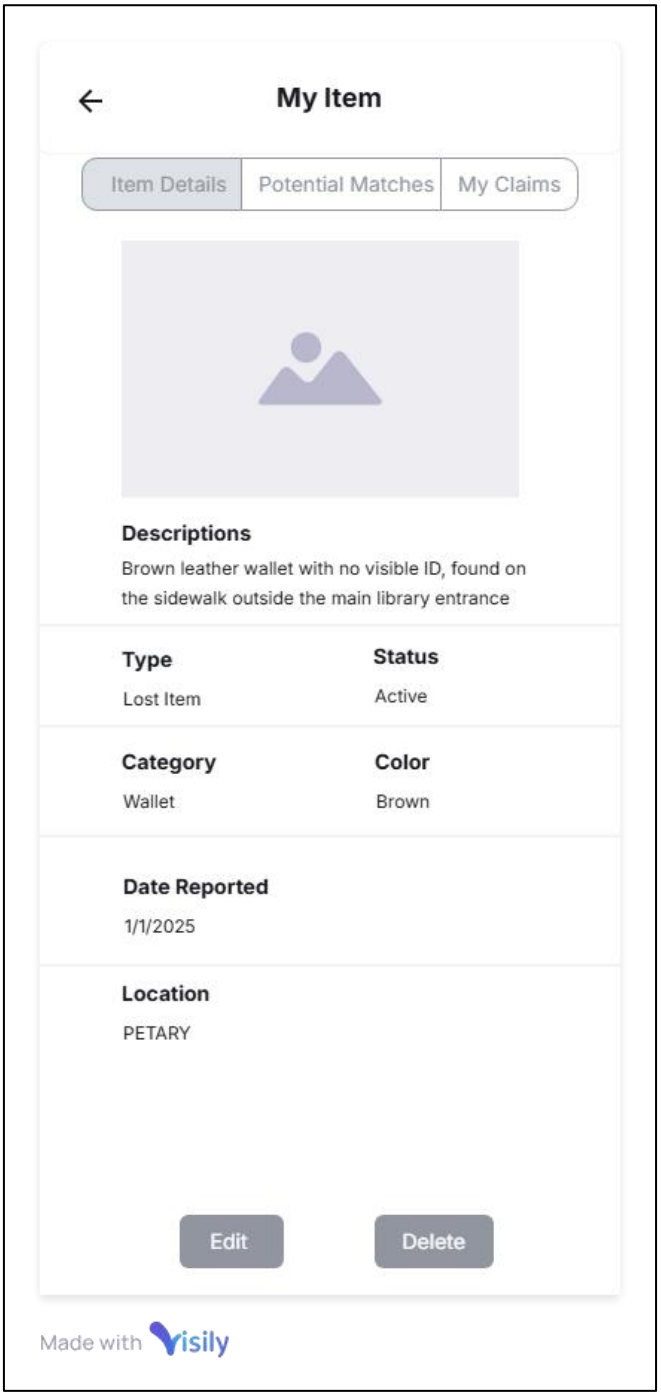



*Figure 3.47: Item Details of 'My Lost Item'.*

Figure 3.47 shows a screen displayed once the tab 'Item Details' is clicked. All item details are shown here, and the user can choose to edit or delete the item.

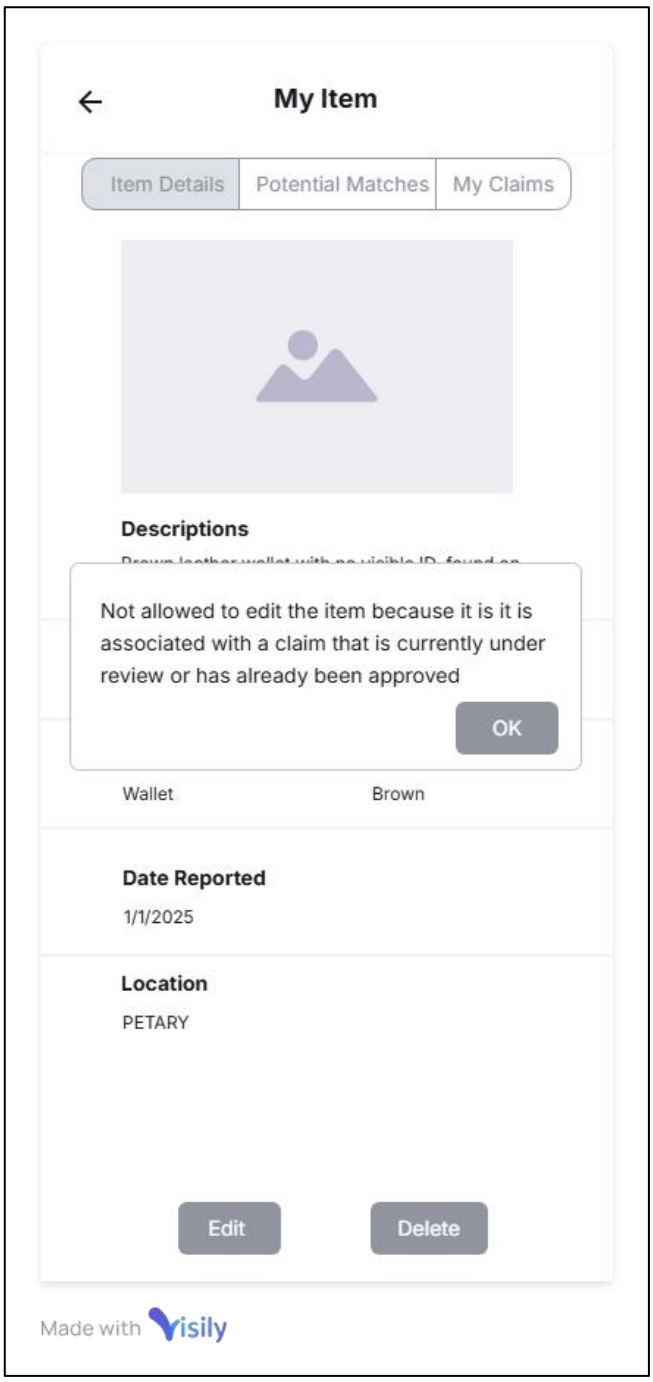



*Figure 3.48: Error Message for Edit Item Details.*

An error message will be displayed to the user explaining the reason they can't edit the item as shown in Figure 3.48.

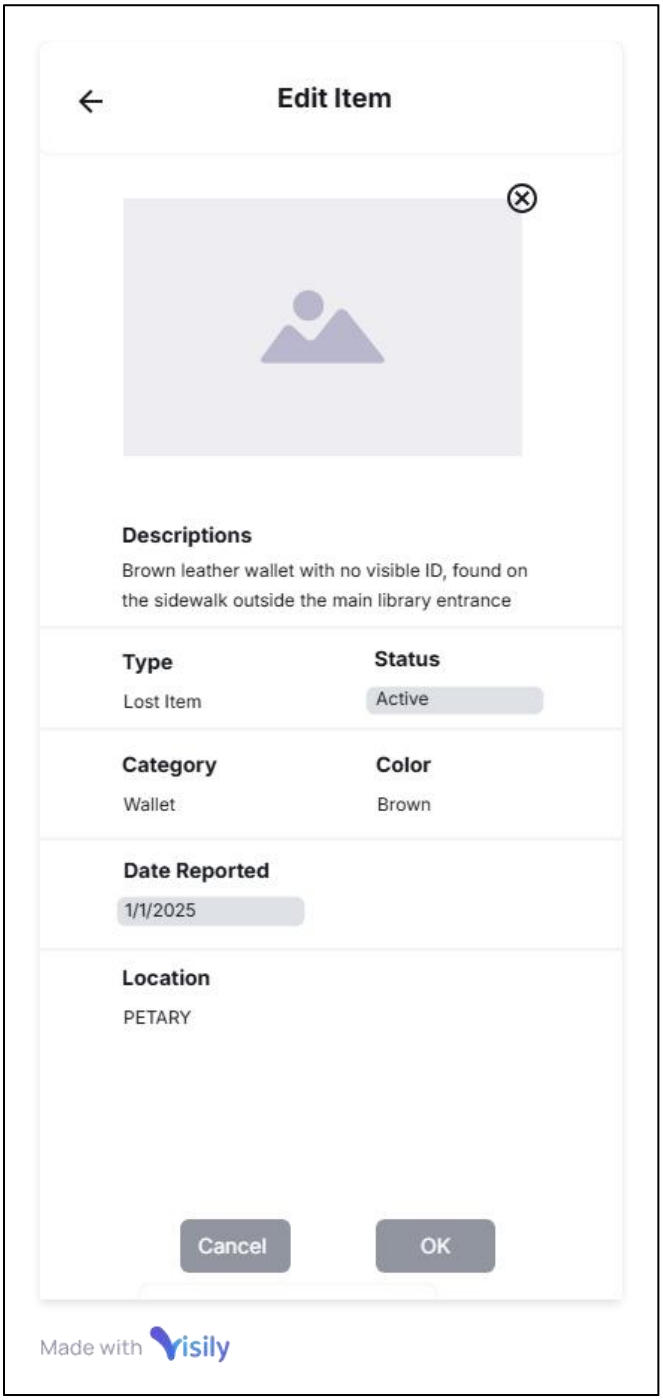



*Figure 3.49: Edit Item page.*

As shown in Figure 3.49, user is allowed to edit the item details not in the grey area. Clicking 'OK' button to edit the item details and 'Cancel' button to cancel the action and redirect back to previous screen.

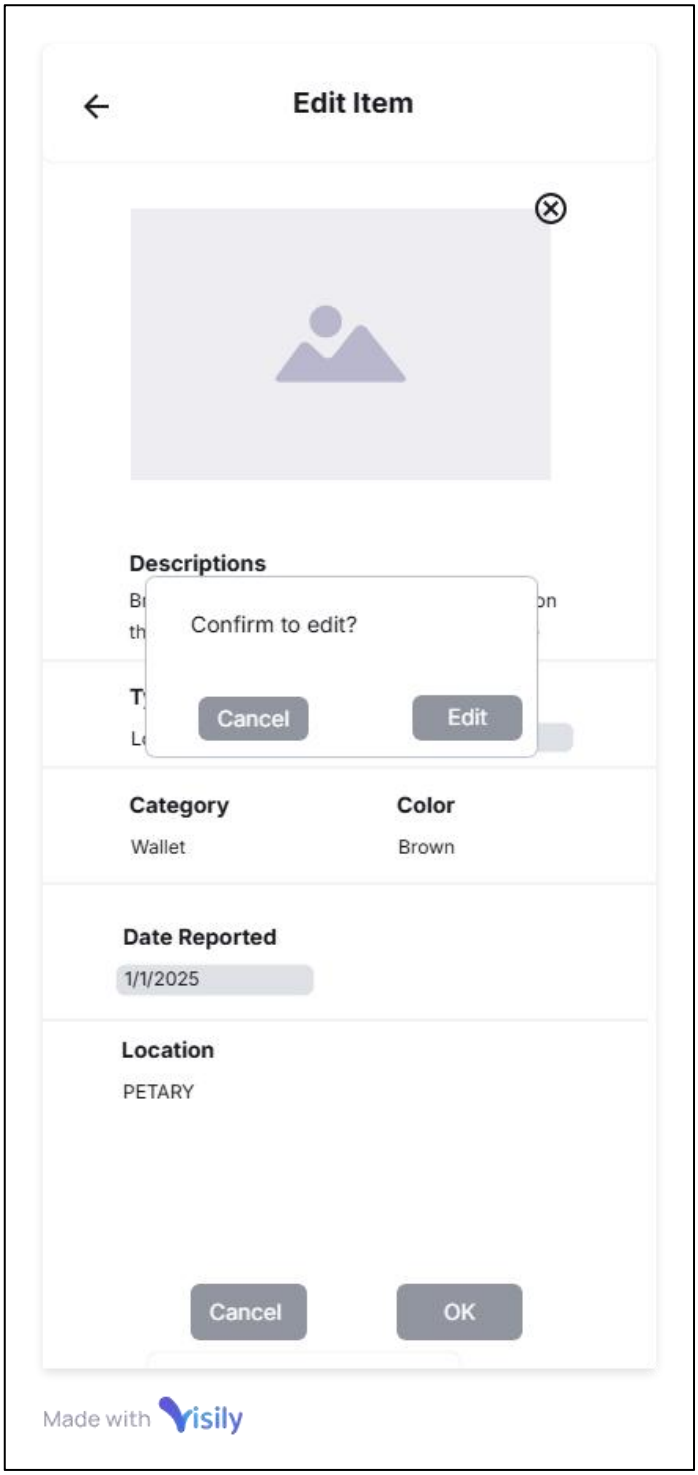



*Figure 3.50: Confirm to Edit Item Details.*

Figure 3.50 depicts that the user will be prompted to confirm edit the item details or cancel the action to edit.

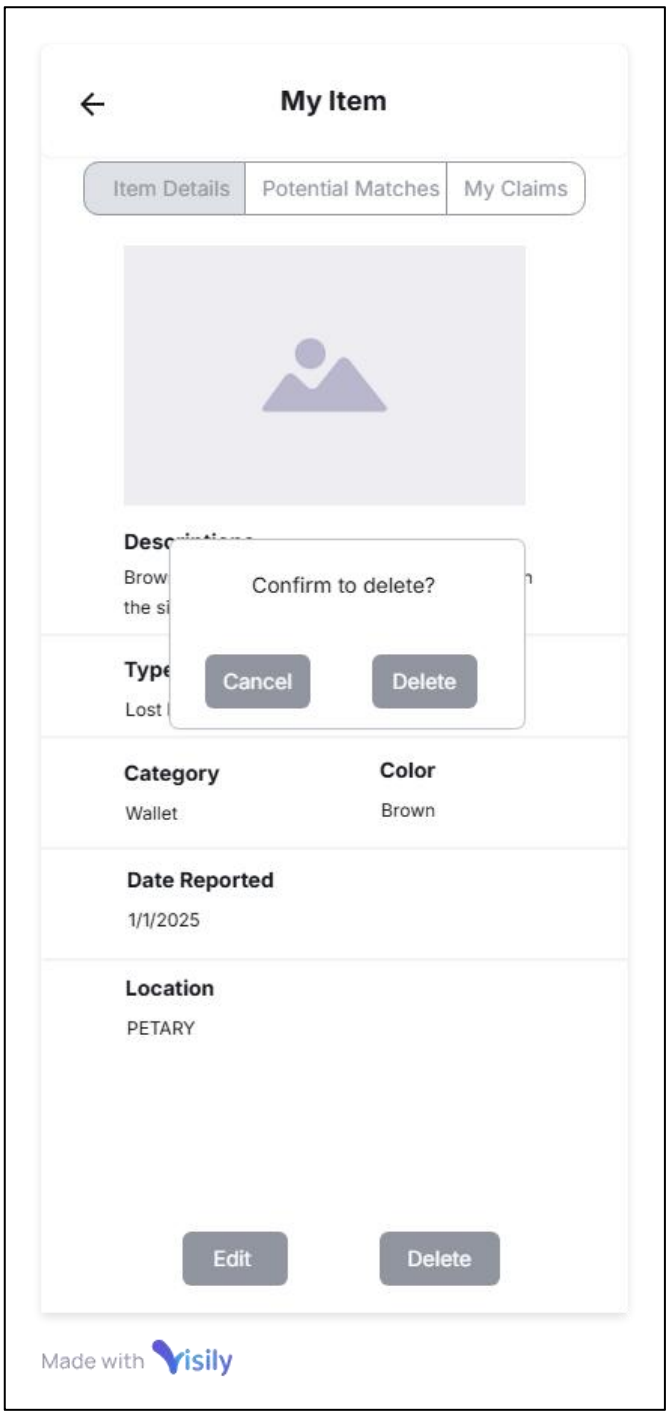



*Figure 3.51: Confirm to Delete Item.*

Screen in Figure 3.51 shows the user is prompted to confirm delete or cancel delete with the buttons ‘Delete’ and ‘Cancel’ provided.

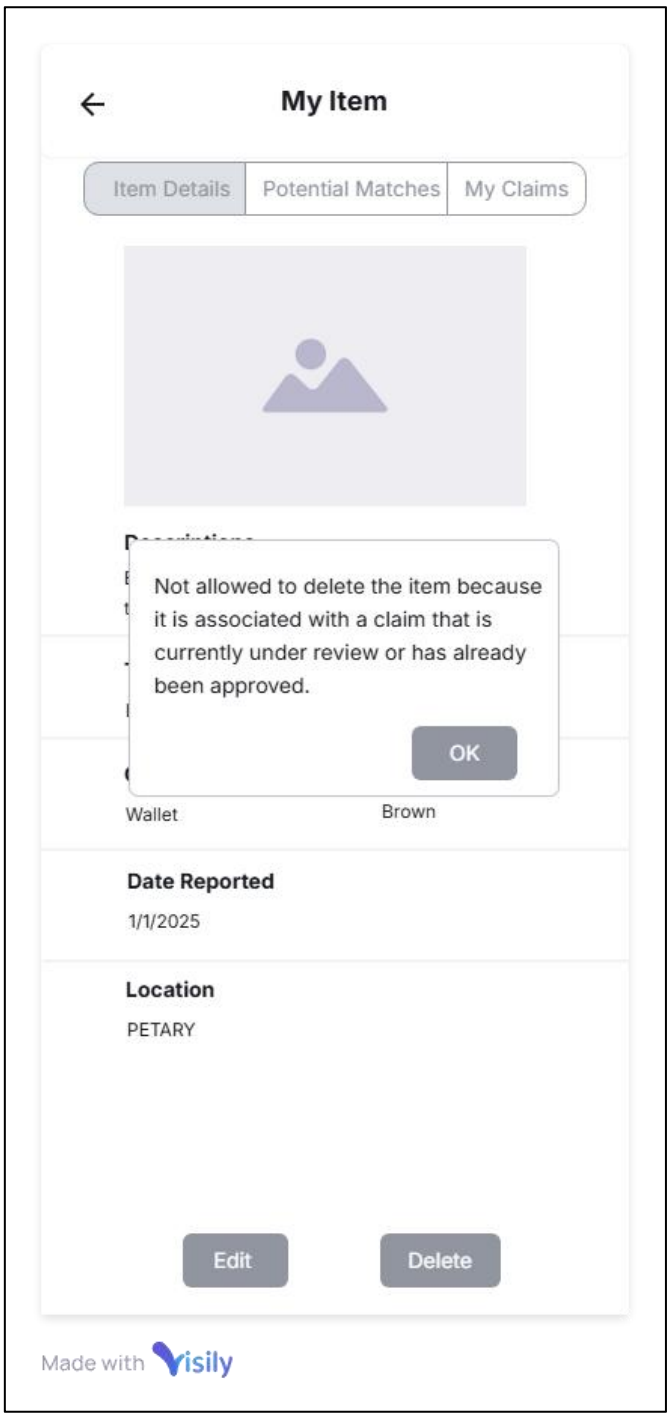



*Figure 3.52: Error to Delete Item.*

Figure 3.52 shows an error message during editing an item.

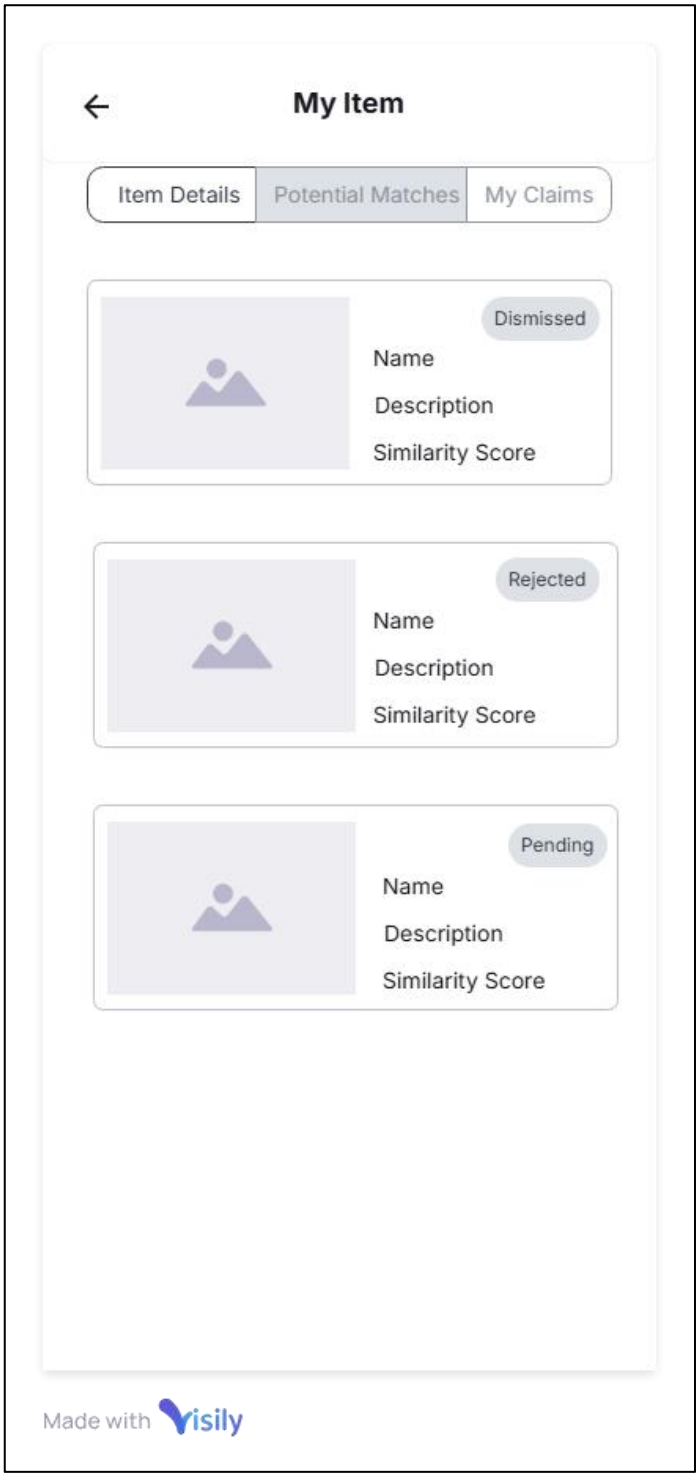



*Figure 3.53: Potential Matches (My Item).*

By clicking the tab 'Potential Matches' as shown in Figure 3.53, the screen displaying the lists of found items which may be a potential match for lost item that user has reported, with an overview of the status of the claim associated with the match, found item name, description and similarity score.

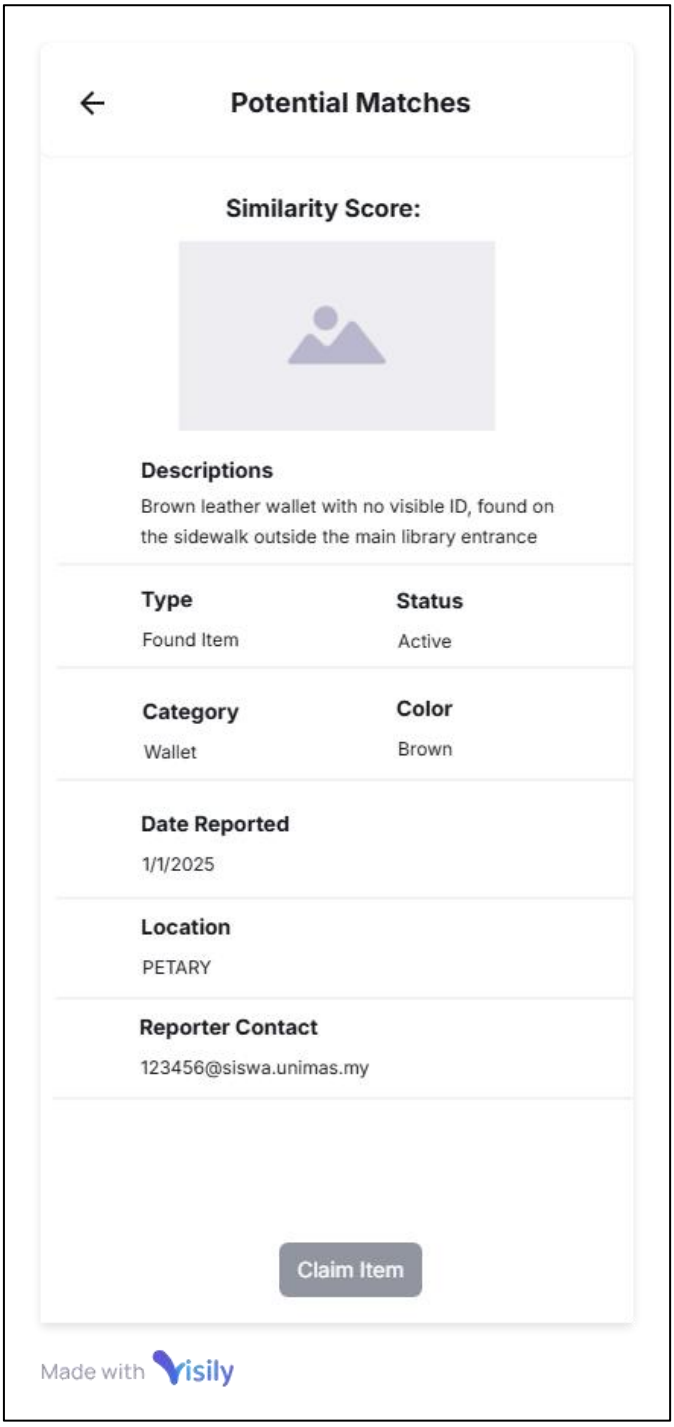



*Figure 3.54: Accept Potential Matches and Claim Item.*

Click 'Claim Item' button as shown in Figure 3.54 to submit a claim request for the item.

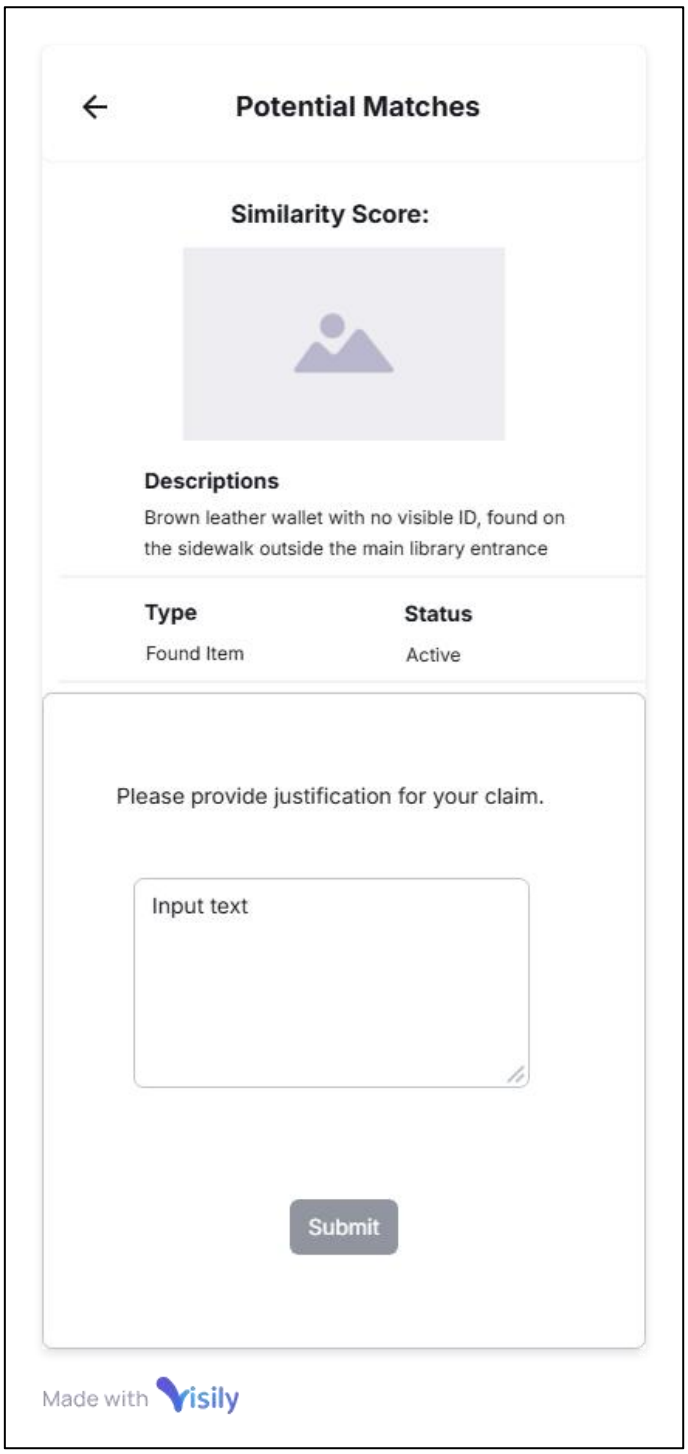



*Figure 3.55: Claim Justification.*

User has to provide to justification for the claim of an item as shown in Figure 3.55.

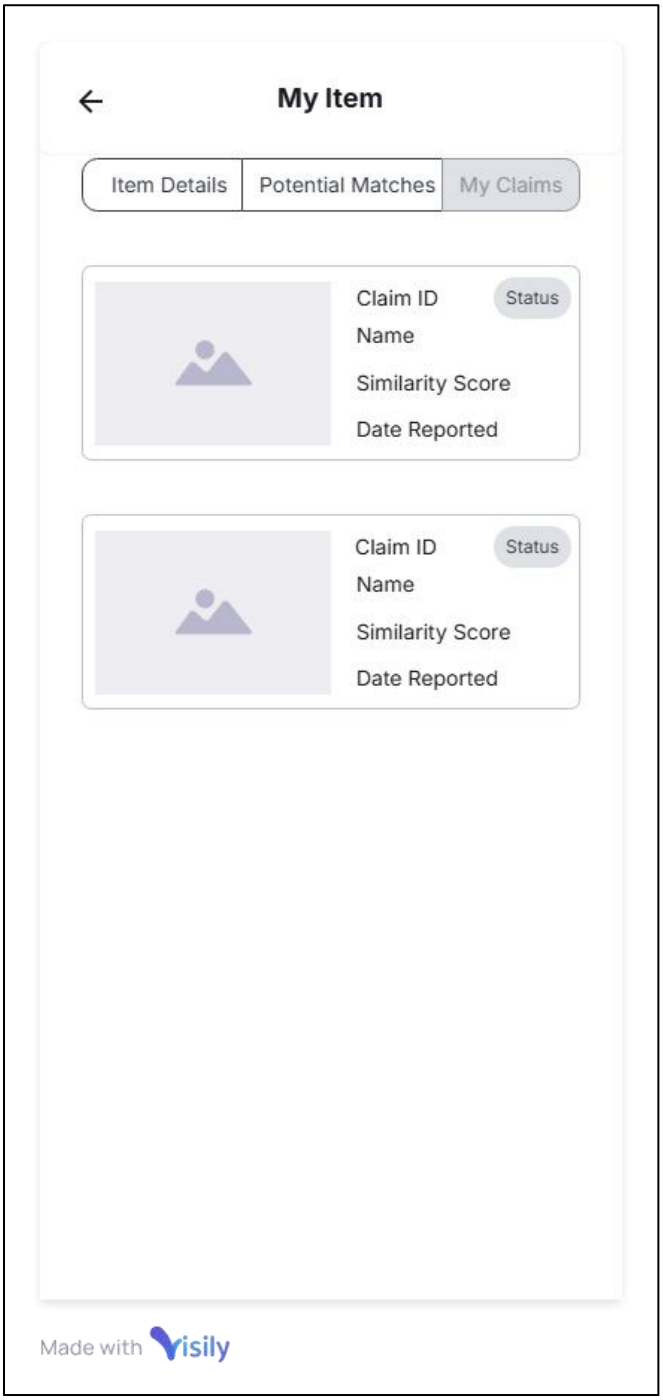



*Figure 3.56: My Claims (My Item).*

Clicking on tab 'My Claims' will display all the overview of past and ongoing claim for the potential matches of item suggested to the user as shown in Figure 3.56.

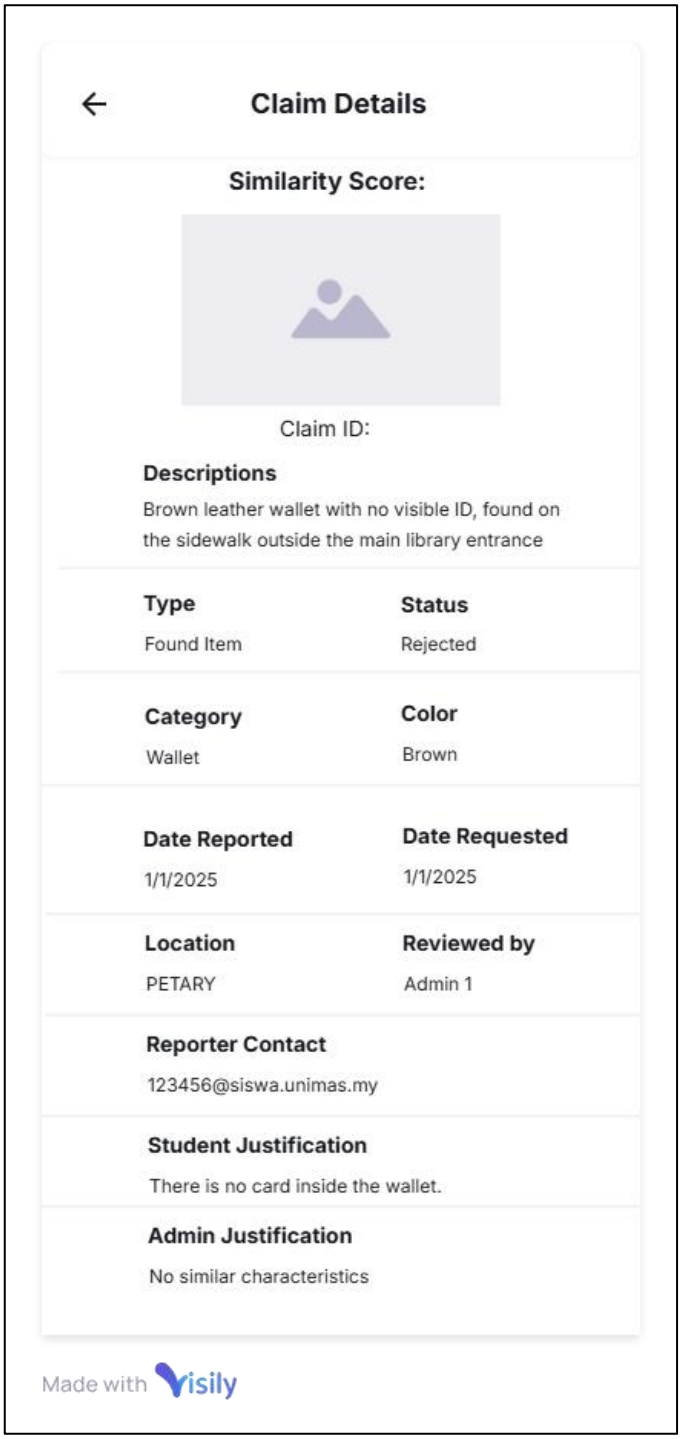



*Figure 3.57: Claim Details (My Claims).*

Figure 3.57 shows all the details related to a claim after the user click on specific claim.

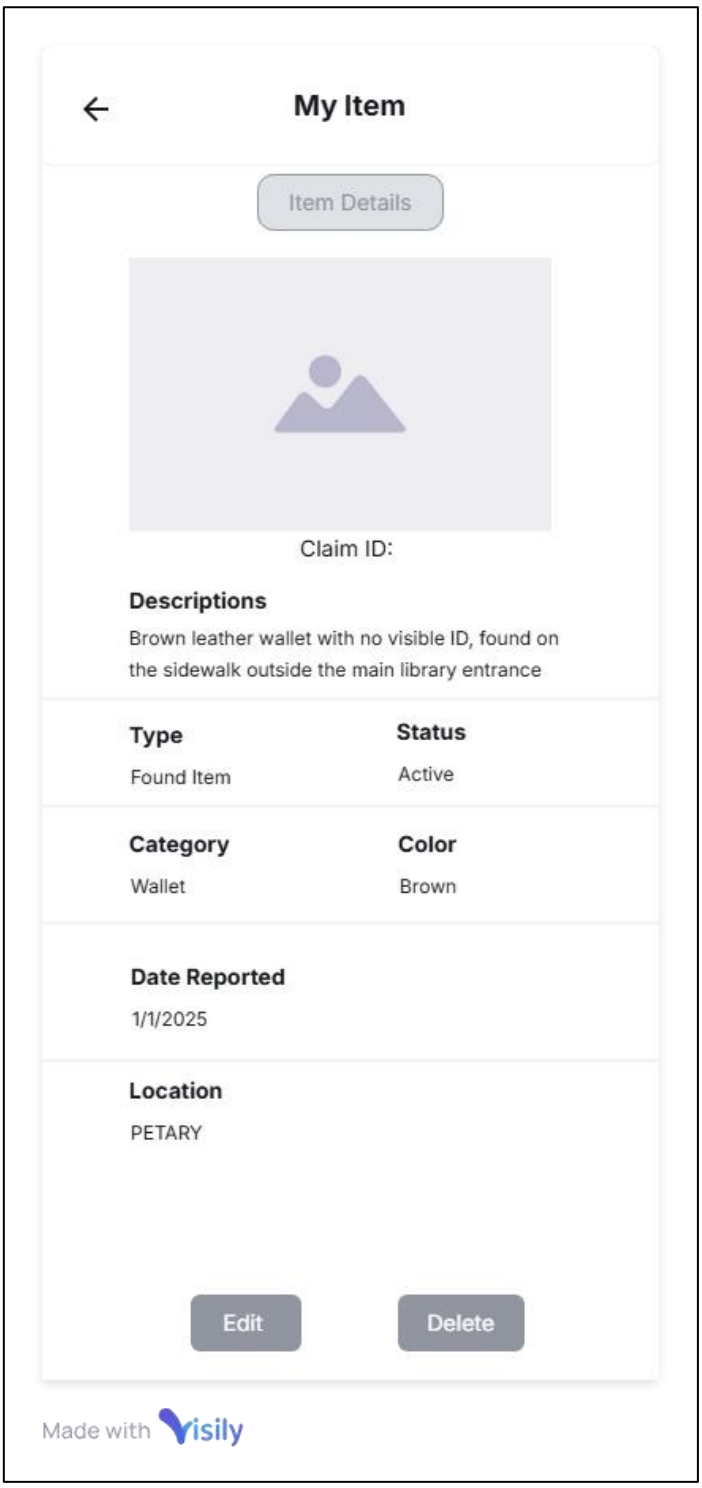



*Figure 3.58: My Found Item (Item Details).*

For the reported found item, there is only one tab which is 'Item Details' available where shows the item details and status of the item, active or resolved as shown in Figure 3.58.

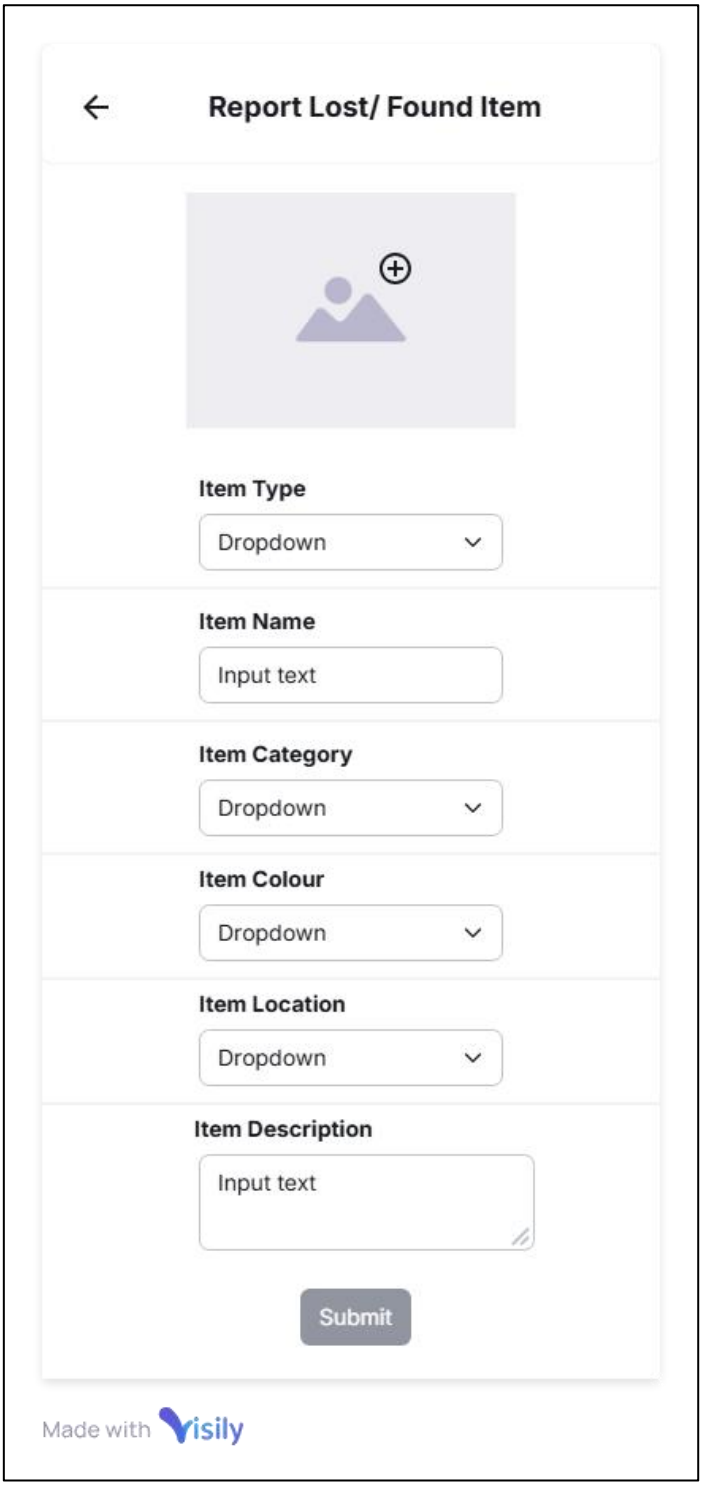



*Figure 3.59: Report Lost/ Found Item.*

Figure 3.59 shows a form for users to report their lost items or found items.

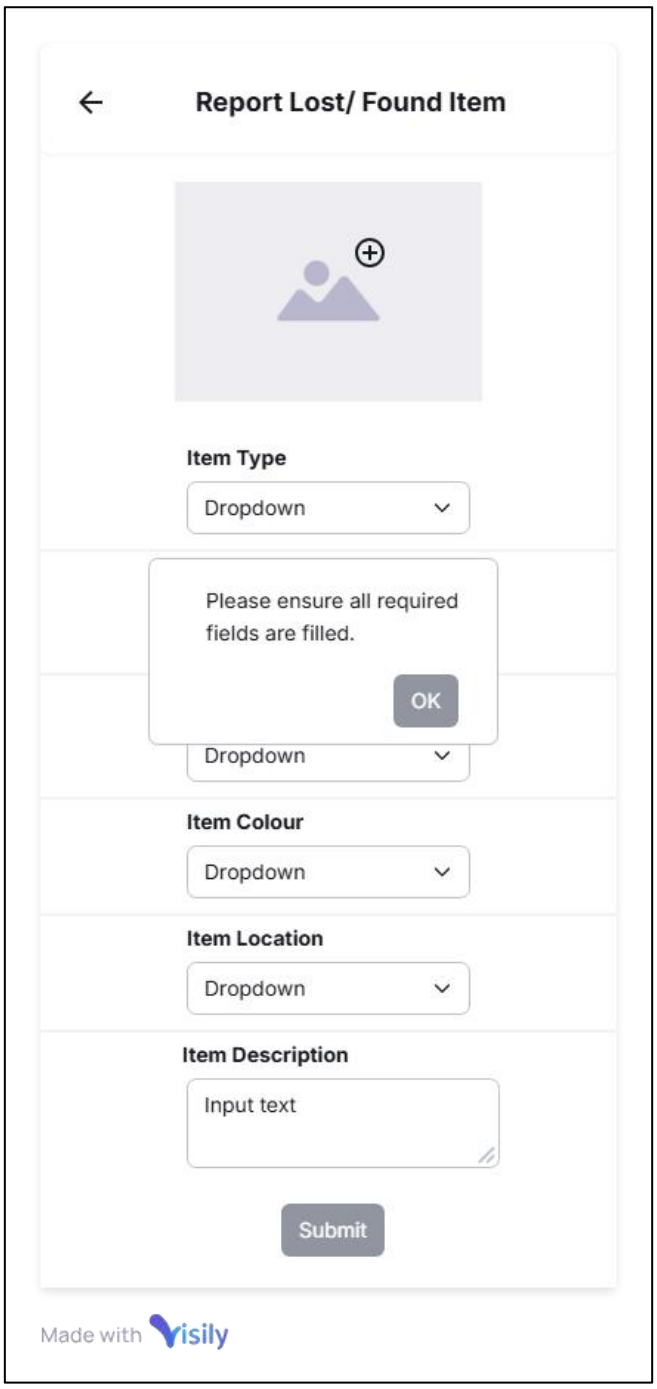



*Figure 3.60: Error Message of Reporting Item.*

Figure 3.60 shows an error message displayed when there is empty field.

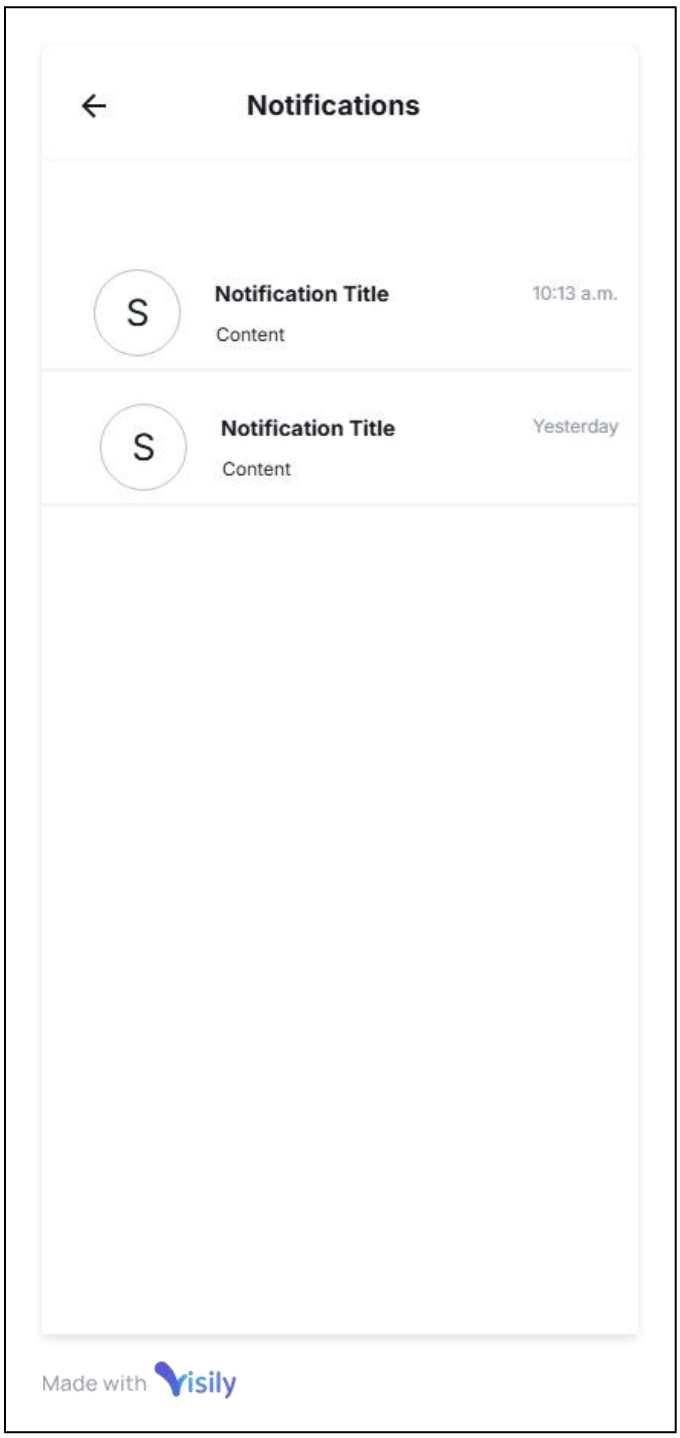



*Figure 3.61: Notifications*

Notifications are all listed at the notification page as shown in Figure 3.61.

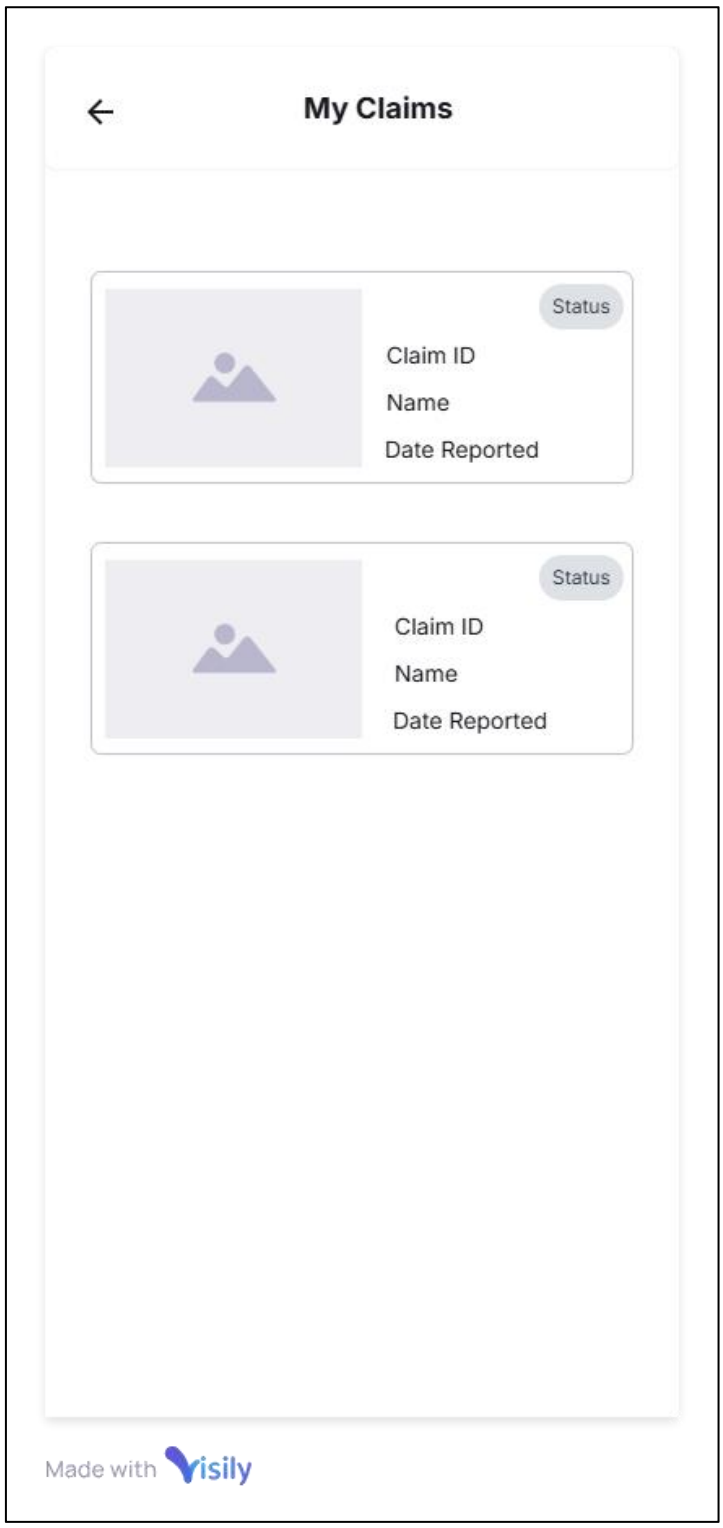



*Figure 3.62: My Claims.*

Figure 3.62 shows the screen where all claims are listed, not limited to specific item.

### 3.3.4.2 Admin's User Interface

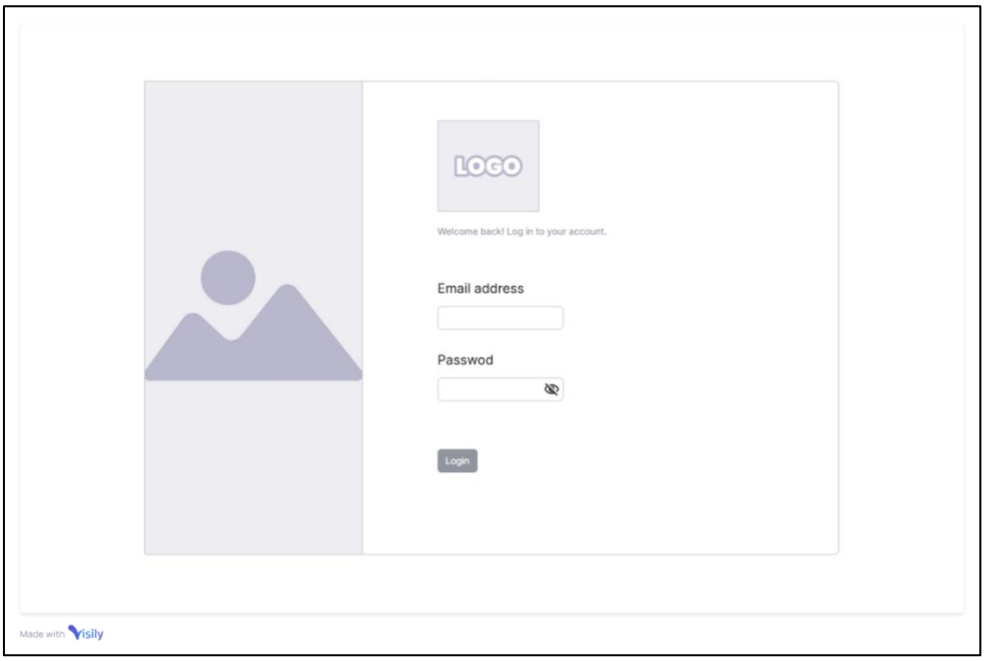



*Figure 3.63: Admin Login.*

Figure 3.63 illustrates a login webpage for admin. Admin need to provide email address and password to login.

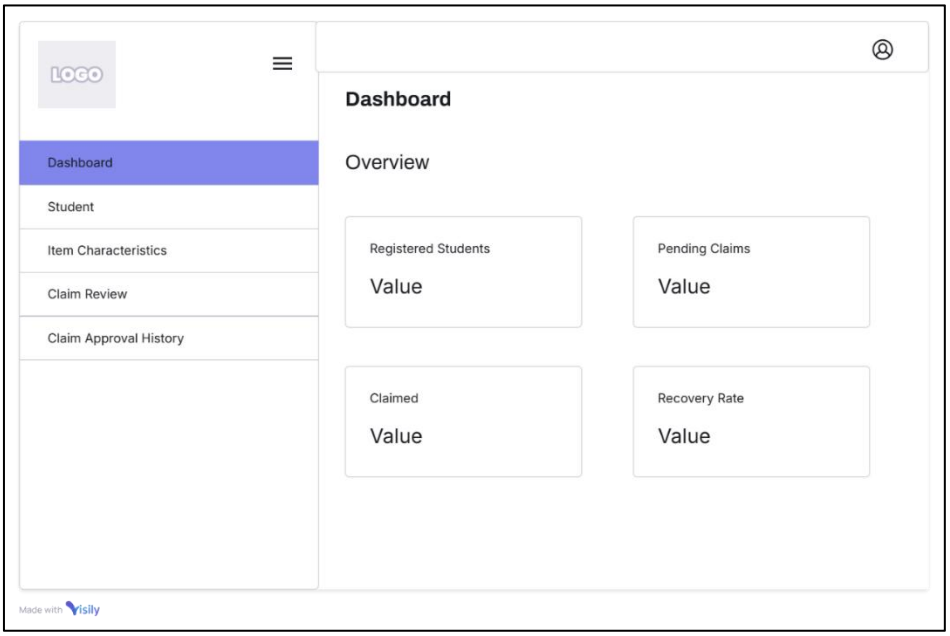



*Figure 3.64: Admin Dashboard.*

Figure 3.64 depicts a dashboard for admin to have a quick overview on no. of registered students, no. of pending claims, no. of claimed item, and recovery rate.

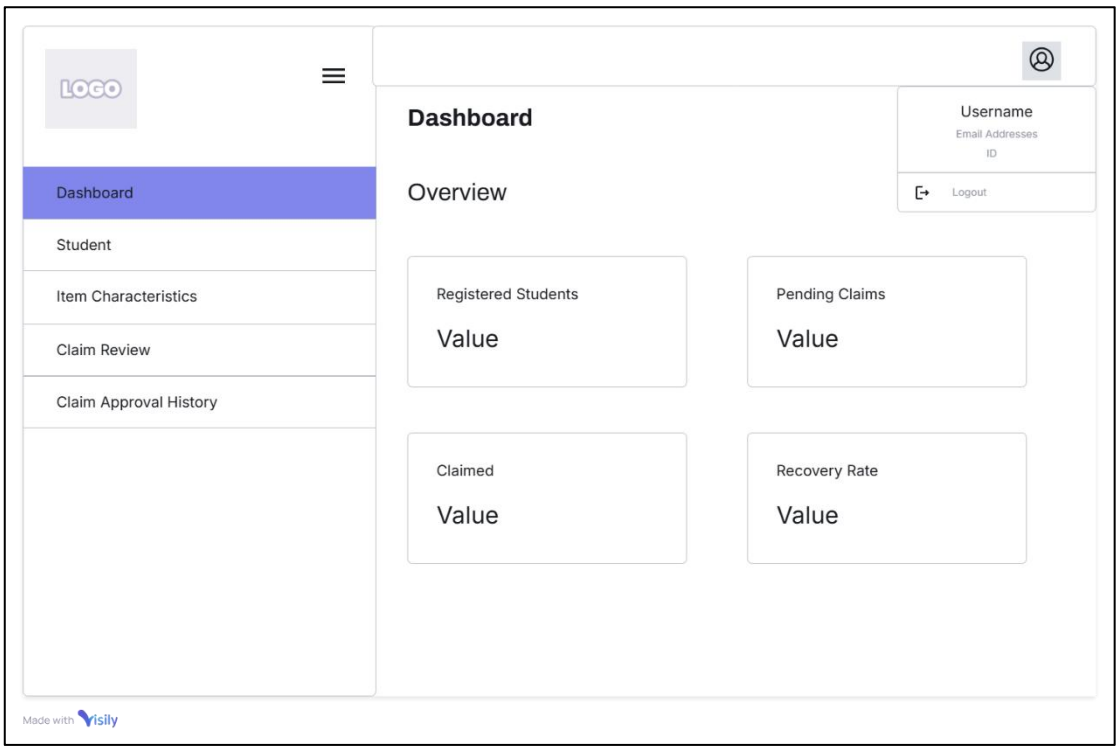



*Figure 3.65: Admin Profile Menu.*

Figure 3.65 shows the screen with profile menu and logout option appears after the 'person' icon is clicked.

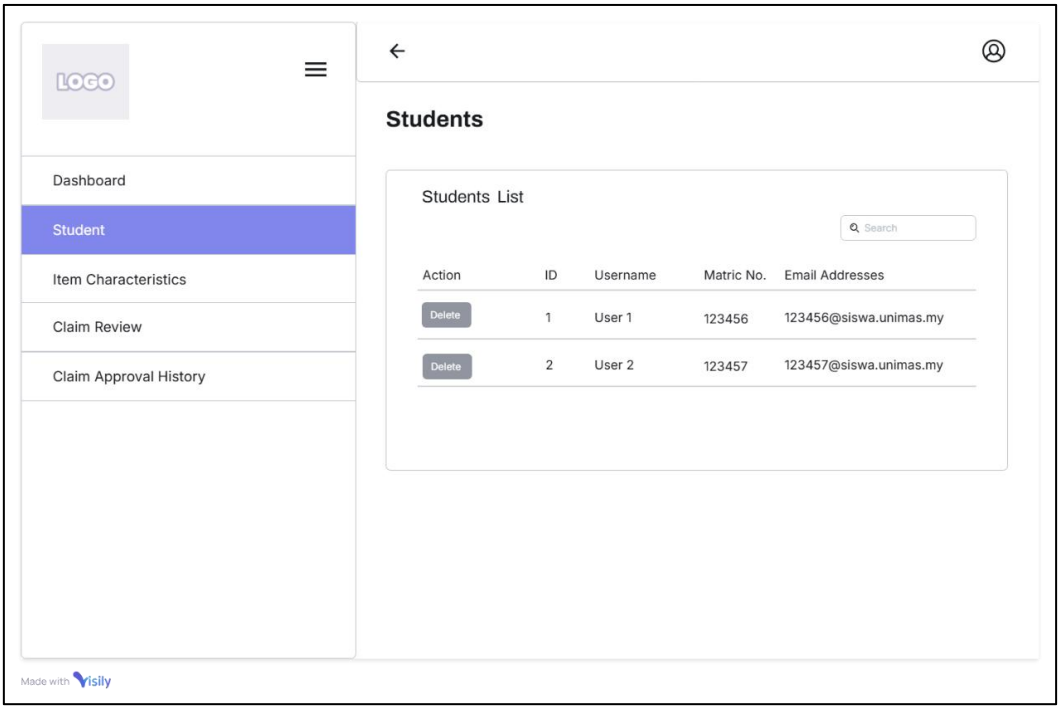



*Figure 3.66: Manage Students.*

Figure 3.66 shows the screen that the user can view or delete user profile.

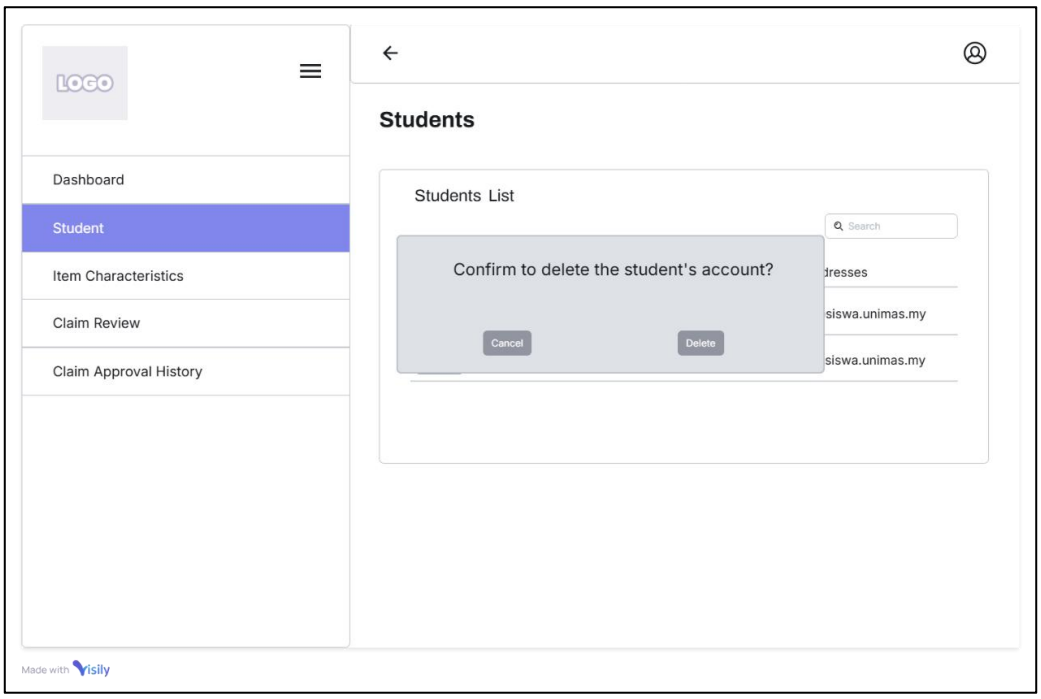



*Figure 3.67: Confirm Delete Student's Account.*

Figure 3.67 shows a screen that the admin confirms to delete the student's account.

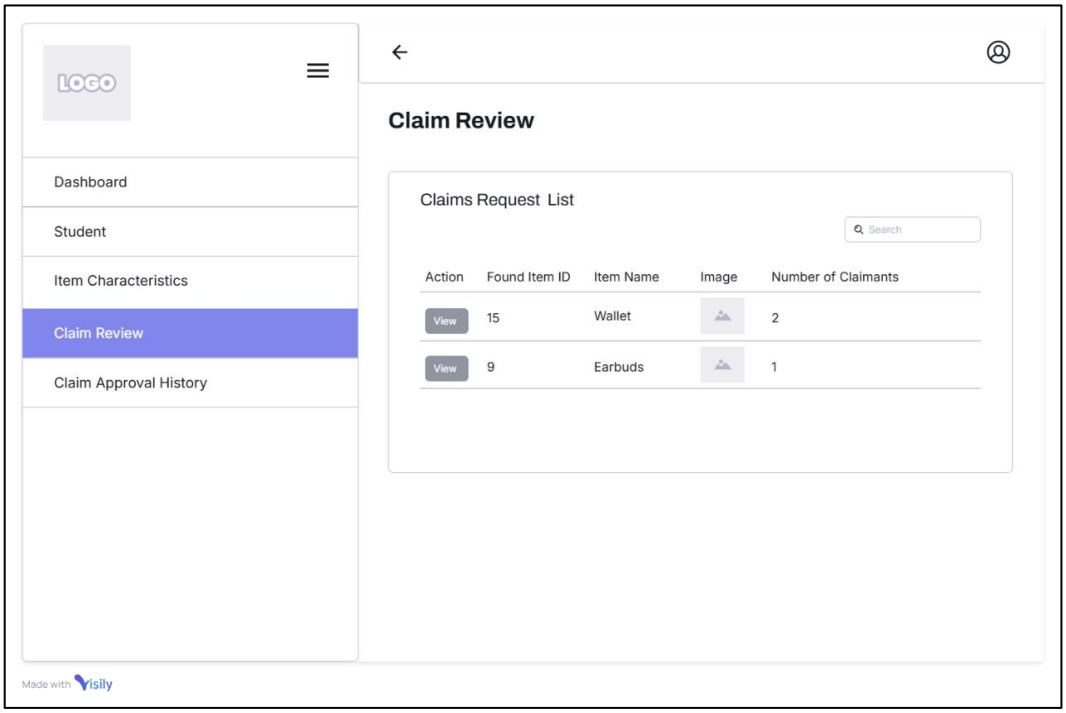



*Figure 3.68: Review Claim Request.*

Figure 3.68 shows all the found items request to be claimed by the students in the table form.

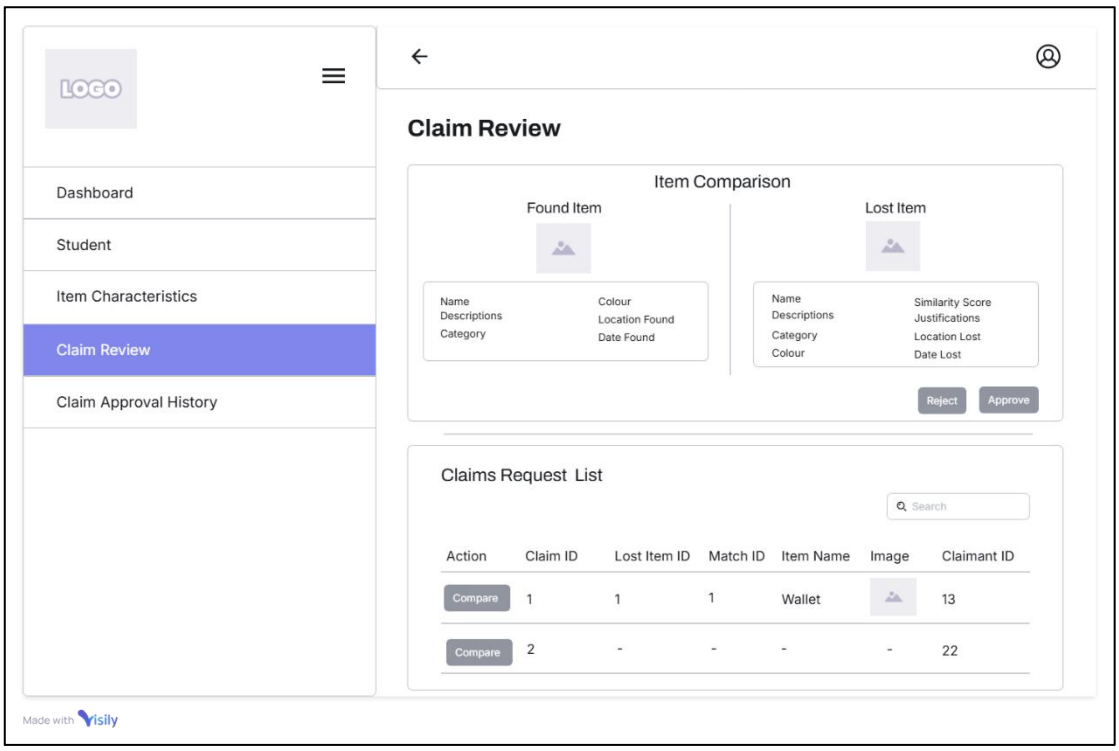



*Figure 3.69: Claim Requests.*

Figure 3.69 shows a section displaying the details of found item and lost item for the user to compare the similarity between the items easily. Admin can switch to compare with item in another claim request by clicking the 'Compare' button. After clicking 'Compare button', the section displaying previous lost item details will be updated with the associated lost item details.

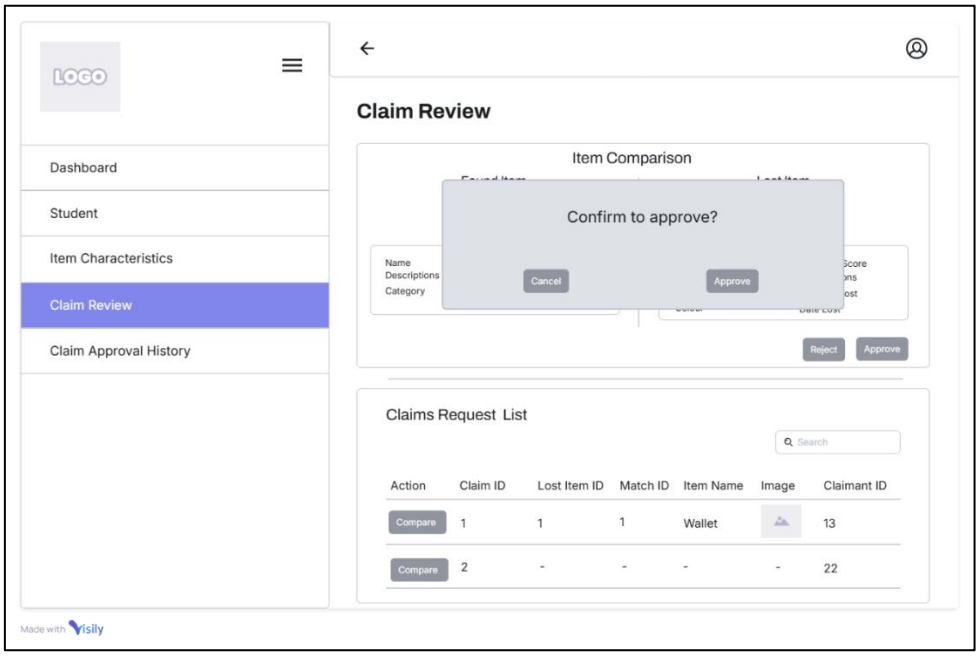



*Figure 3.70: Confirm Approve Claim Request.*

Figure 3.70 shows the screen admin need to confirm their intents to approve the claim request.

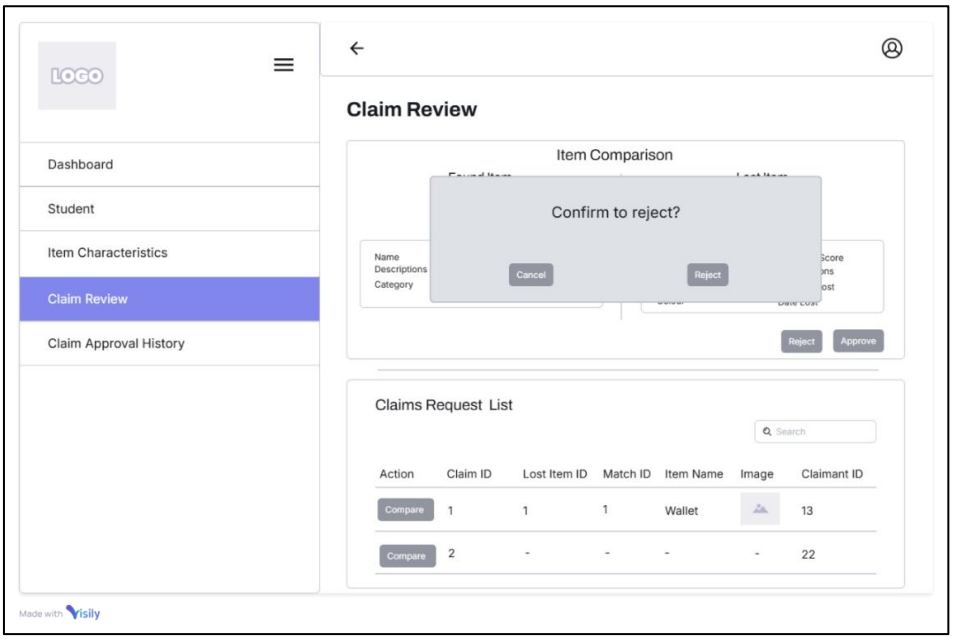



*Figure 3.71: Confirm Reject Claim Request.*

Figure 3.71 shows the screen admin need to confirm their intents to reject the claim request.

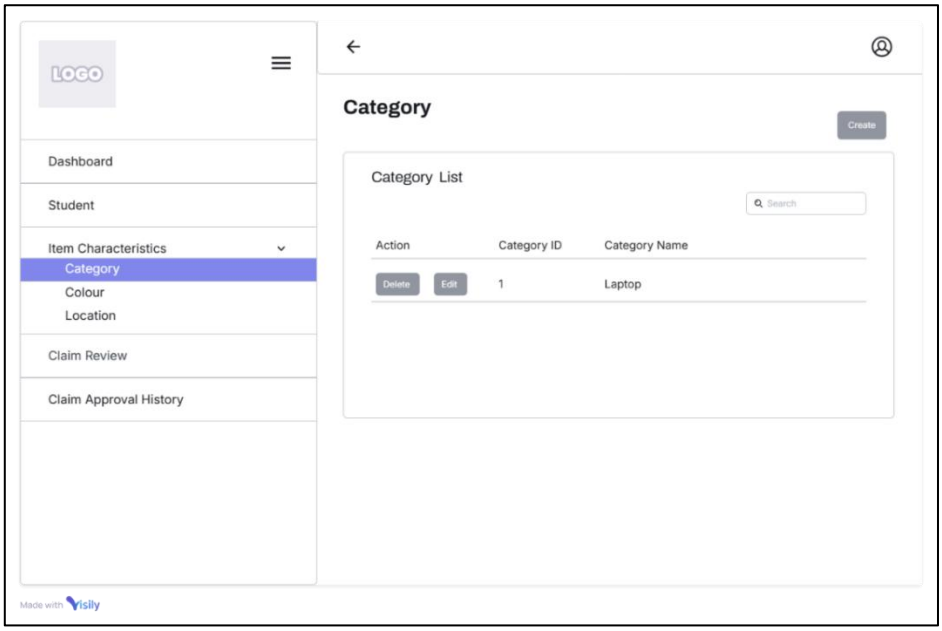



*Figure 3.72: Manage Item Category.*

Figure 3.72 shows a screen that admin can create, view, edit and delete the item category options.

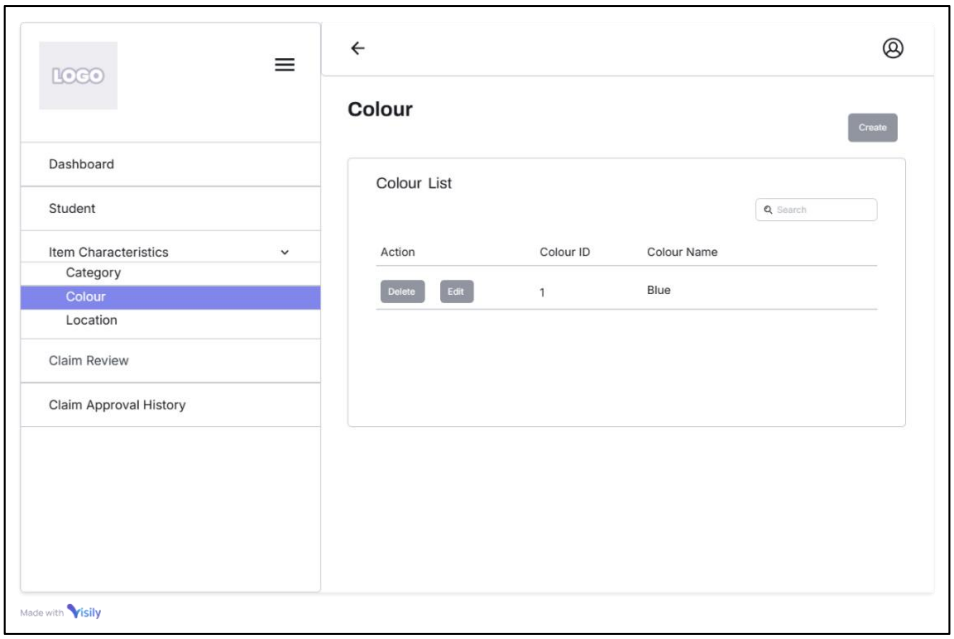



*Figure 3.73: Manage Item Color.*

Figure 3.73 shows a screen that admin can create, view, edit and delete the item color options.

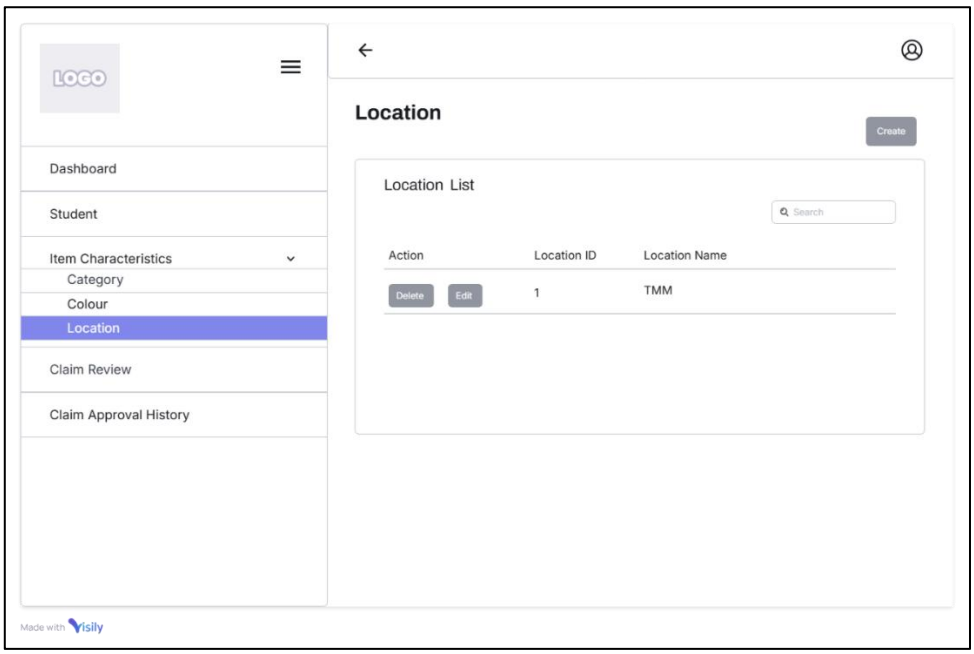



*Figure 3.74: Manage Item Location.*

Figure 3.74 shows a screen that admin can create, view, edit and delete the location options.

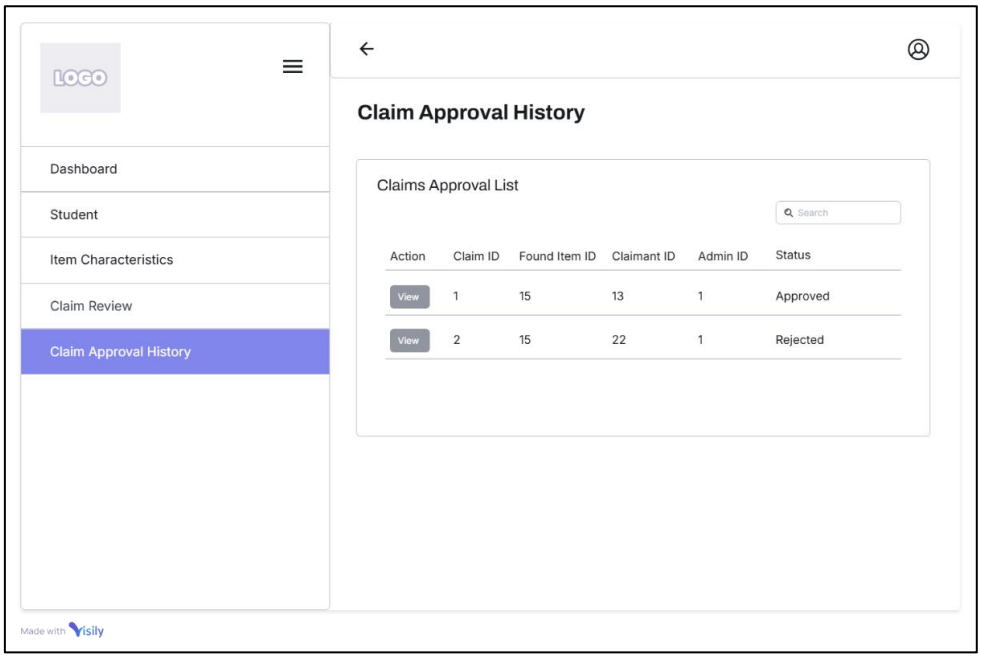



*Figure 3.75: Claim Approval History Overview.*

Figure 3.75 shows an overview of all the claims that were approved or rejected by the admin.

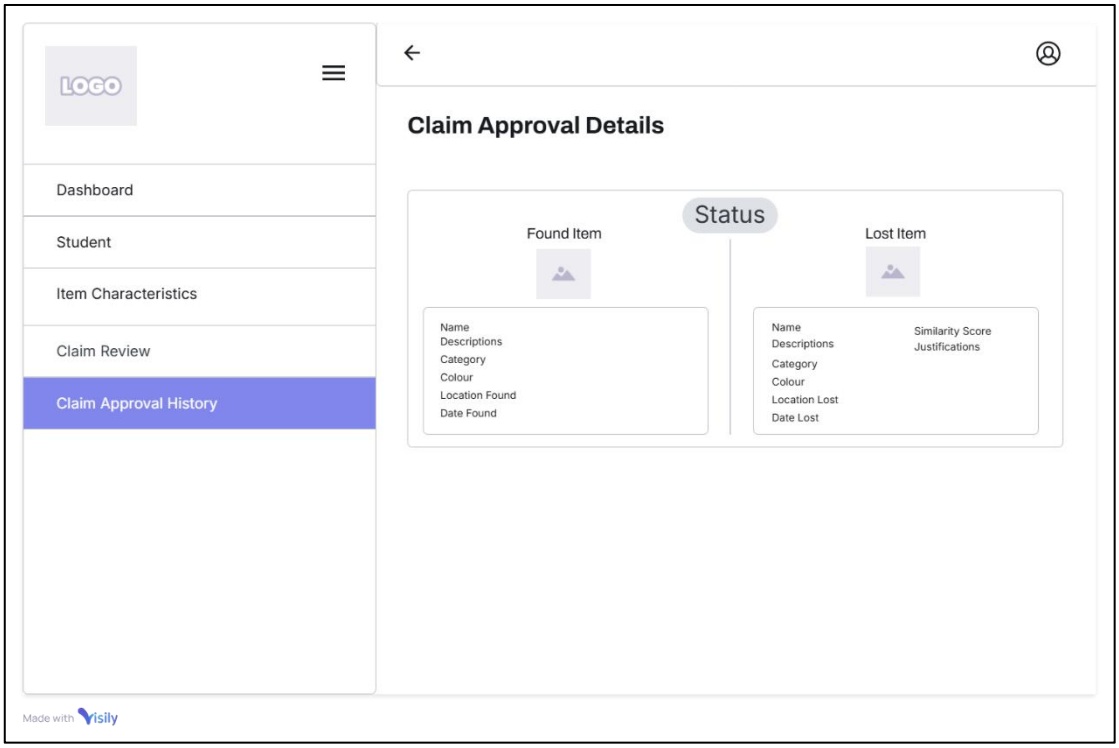



*Figure 3.76: Claim Approval Details.*

Figure 3.76 depicts a screen where more details of a claim are displayed to the admin, which include the lost item details and found item details alongside the status of the claim (approved or rejected).

## 3.4 Implementation Phase

The development of the system for lost and found begins right after the phase of system design. This is the implementation phase where coding is necessary for the translation of the system design into an effective application. The proposed system will be both a mobile and a web app. In developing the web application, the front-end pages will be built using HTML, CSS, Bootstrap, and JavaScript.

The mobile application will be done using Flutter, which gives an intuitive, responsive UI for mobile use. An API endpoint will be attached to the back-end server that connects through the API, facilitating the communication of the mobile application with the

backend for actions: registering an item; an image-matching request coming from the user's side; and approvals by admins.

The proposed backend server will be powered by the Laravel framework, which will handle all the server-side activities: processing the requests, interaction with the database, and execution of the image matching algorithm. A database will be set up for storing information regarding users, item details, image information, and match results. It will also run on the API server: hosting the pre-trained AI model for doing image comparisons that could suggest possible matches of lost-found pairs. The AI model connects with the back-end server for automated matching.

### 3.5 Verification Phase

This stage is focused on validating whether the system meets its requirements or not. Accordingly, test cases are designed based on the system. Also, the functional testing ensures that main features are functioning according to specifications and the usability testing will make the user interface of the system in an accessible manner. This ensures that the system is on target and also delivers initial design and requirement by users.

### 3.6 Maintenance Phase

This is the phase where all the bugs encountered will be addressed to ensure user satisfaction and prolongs the system’s lifespan.

### 3.7 Conclusion

In short, requirement analysis and system design phases of the Waterfall model is discussed in detailed in Chapter 3 which facilitate the development of the system in Chapter 4. In the requirement analysis phase, undergraduate students in UNIMAS are taken as the sample for the questionnaire to gather system requirements. Apart from that, use case diagram and use

case specifications are built to ensure critical functionalities of the systems are considered and not missed. In the system design phase, system architectural design, activity diagrams, and class diagrams are designed. Furthermore, wireframes are designed to outline the user interface of the proposed system. This chapter ends with brief introduction to the other phases taken part in Waterfall.

# Chapter 4: Implementation

## 4.1 Introduction

This chapter describes how the FindIt system is implemented in practice, transforming the documented requirements and system design into a functional, AI-driven solution for campus item recovery. The implementation phase involves developing a mobile application for students and a web-based administrative interface, with the integration of pretrained DINOv2 model to help students locate their belongings faster from the recommendations of the system. As a result, the claim management process became efficient.

## 4.2 Development Tools

The tools like Laravel, Bootstrap, Flutter, Laragon, Visual Studio Code, PyTorch, FastAPI, Android Studio, Hugging Face are being used in developing and implementing FindIt system.

### 4.2.1 Laravel

Laravel played an important role in the system development by providing a structured MVC architecture and Eloquent ORM for database interactions, and enable database management with migrations and seeders. Apart from that, it uses Blade for rendering of HTML views and responsible for jobs like creating RESTful API endpoints to communicate with Flutter app, and accessing AI model via FastAPI. It also helps complete tasks like manage user accounts and process items' claims. Additionally, Laravel is responsible to handle authentication with pre-built or custom middleware to ensure system security and integrity.

### 4.2.2 Bootstrap

Bootstrap enables the creation of an adaptive admin dashboard by providing a responsive and pre-built UI framework. For instance, 12-column grid system, pre-built UI components for navigation, tables, buttons, forms, and modals while Bootstrap's utility classes and CSS variables allow customization to tailor to project's need. It is integrated with the Laravel

backend with ease through Blade template. Furthermore, Bootstrap's responsive design utilities ensured that the admin dashboard was usable and responsive across different devices, enhancing the overall user experience and administrative efficiency.

### 4.2.3 Flutter

Flutter helps in developing mobile app with intuitive and responsive screens by leveraging its widget-based architecture like Material Design components and custom UI elements aligned with project requirements. Using Flutter will enable integrating native device features to the mobile application such as select image from gallery and taking photo using device's camera. Flutter's capability extends to handle real-time notifications by implementing Firebase Cloud Messaging (FCM) to keep users informed about potential matches and updates. Additionally, Flutter's hot reload feature accelerated the development process by eliminating the need to restart the whole mobile application to reflect the UI updates in the emulator or mobile phone.

### 4.2.4 Laragon

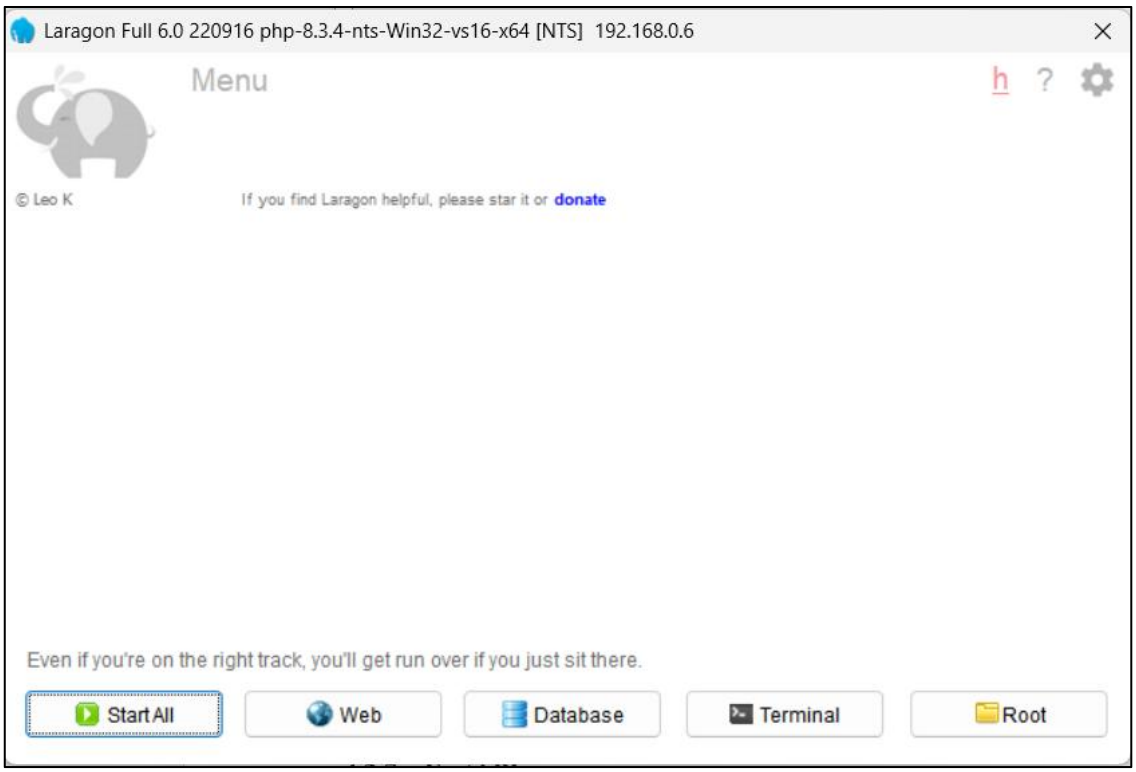



*Figure 4.1 Laragon*

During development, Laragon provides an all-in-one local development environment that facilitated the setup and management of the Laravel backend. It provided a pre-configured PHP runtime environment, and hence no need to manually install PHP and its extensions required by Laravel. It also automatically configured Apache, Nginx, and MySQL services, eliminating the need of manual setup and allowing the developer to focus on the backend logic. Database management with MySQL/MariaDB and local testing of Laravel APIs are also provided by Laragon. Furthermore, its GUI provided one-click control for starting and stopping Apache, MySQL and PHP services enabled an efficient development workflow.

### 4.2.5 Visual Studio Code

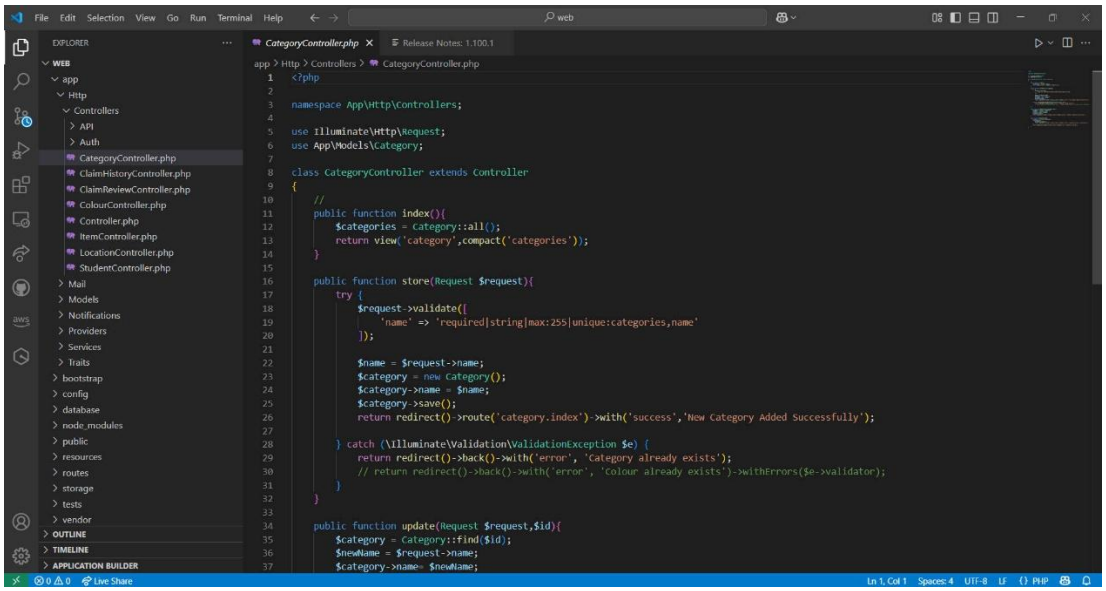



*Figure 4.2 Visual Studio Code*

Visual Studio Code is an integrated development environment used to build Laravel backend (i.e. database management, APIs communicate with Flutter mobile application and FastAPI endpoint) and frontend web interface. Visual studio Code provide code completion, debugging syntax highlighting and autocompletion features after the extension “PHP Intelephese” is configured properly, providing a smooth PHP development experience.

### 4.2.6 PyTorch

In the FastAPI service, PyTorch is important in AI model's operations—handling the model's architecture (ViT layers), and provide tensor operations like embedding calculations, L2 normalization and similarity comparison. It also ensures efficient inference by disabling gradient tracking to reduce memory overhead.

### 4.2.7 FastAPI

FastAPI plays a vital role in enabling a seamless integration of ai model's capabilities into the system. FastAPI serves as an API layer to expose the AI model as a RESTful endpoint for Laravel backend to communicate with. For instance, Laravel could forward image from Flutter mobile application to the endpoint provided by FastAPI. This setup enables the access to AI model for image similarity task. To facilitate this, FastAPI are installed in Hugging Face to handle HTTP requests.

### 4.2.8 Android Studio

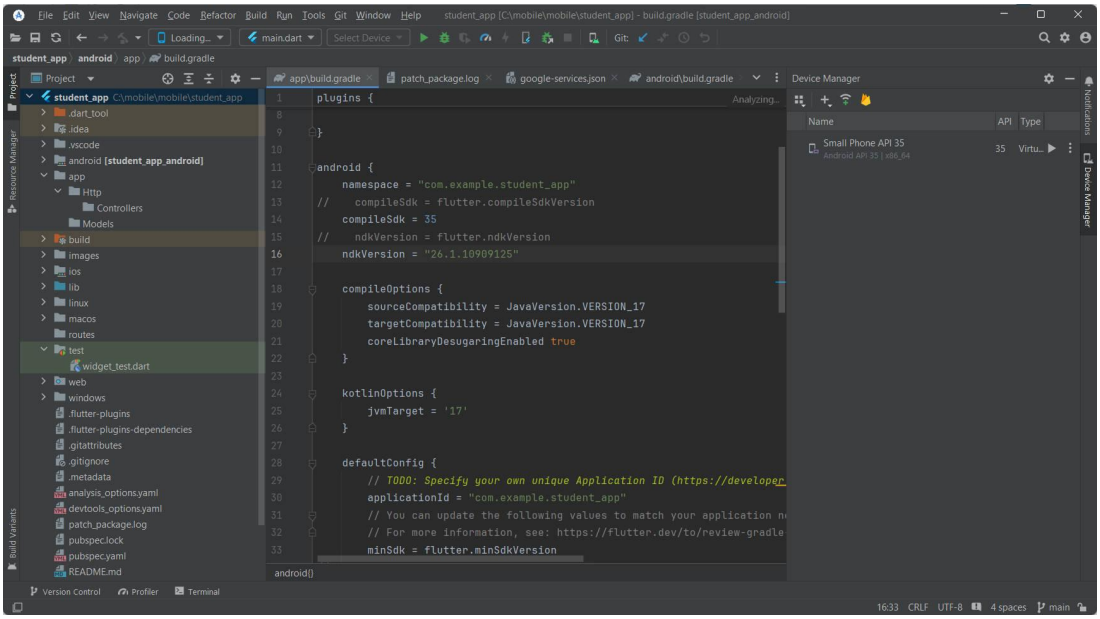



*Figure 4.3 Android Studio*

Android studio provides integrated development environment to build the Flutter mobile application, with tools and environment to develop, test and debug the application. Flutter and Dart plugin is configured in Android Studio to enable the syntax highlighting and widget

inspector features, while Android SDK and Flutter SDK are configured in Android Studio to allow seamless compilation to physical devices or Android emulators. Furthermore, it is used in generating apk file for release and distribution purpose.

### 4.2.9 Hugging Face

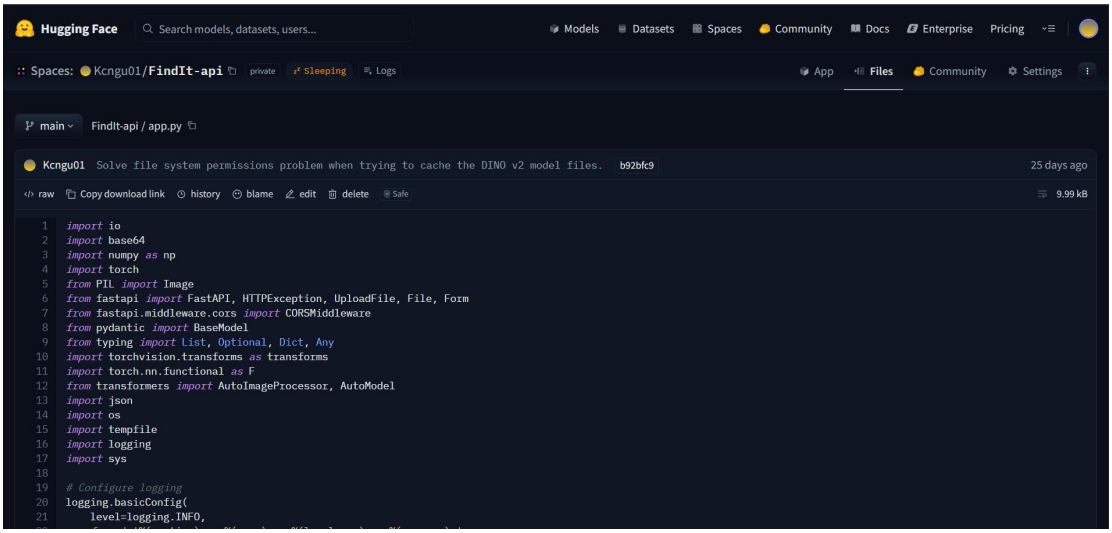



*Figure 4.4 Hugging Face*

Hugging Face is utilized to load the AI model and enable the Laravel backend to access the AI model for image similarity task through FastAPI. The FastAPI server is hosted in Hugging Face to expose the AI model to the internet for external access, i.e. Laravel backend. To establish a seamless connection between Flutter mobile app and the AI model hosted remotely on Hugging Face, Laravel act as the middleware routing the request between them.

## 4.3 System Implementation

This section will discuss the system features and functionalities in detail. The section will be divided into 2 subsections, i.e. students and admin for better organization of the system implementation. Then, the functionalities provided to both students and admins will be elaborated alongside with the screenshot of user interfaces, ensuring a comprehensive overview of system implementation is delivered.

### 4.3.1 Student

#### 4.3.1.1 Student Registration

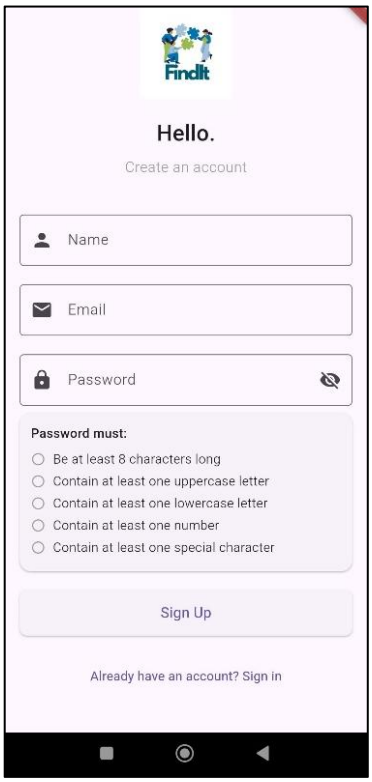



*Figure 4.5 Student Registration*

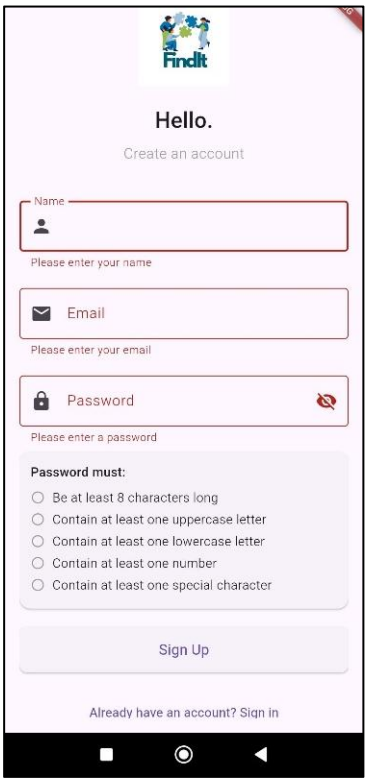



*Figure 4.6 Error Handling of Empty Fields*

The student registration page as shown in Figure 4.5 serves as the entry point for new users to create accounts within the FindIt system before they can access all the provided platform features. Student need to provide basic user information like username, email and password. The student has to provide password with at least 8 characters, one uppercase letter, one lowercase letter, one number and one special character for a better password security. The circles next to the password requirements will display a green tick once the corresponding requirement is fulfilled.

Errors will be prompted if the students missed filling any fields as shown in Figure 4.6.

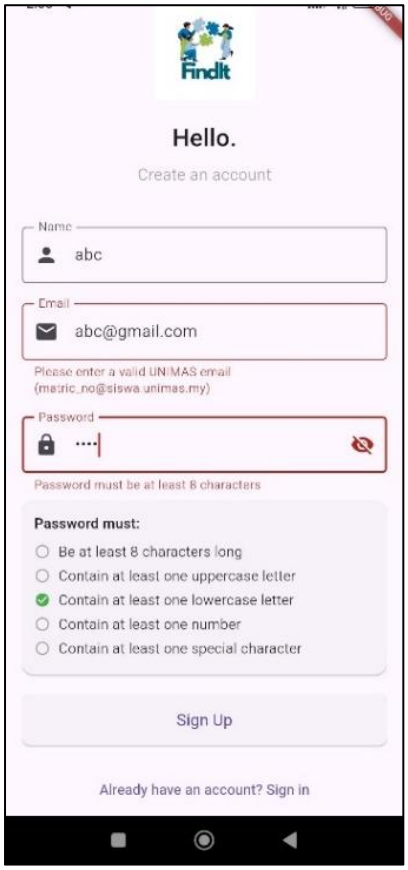



*Figure 4.7 Error Handling of Invalid Email and Password*

As shown in Figure 4.7, the student can provide any username they want upon registration, but it is mandatory for them to register with the email they use for UNIMAS system because the system is only allowed for UNIMAS student to use only. Additionally, the password provided by the students must satisfied all the requirements.

#### 4.3.1.2 Email Verification

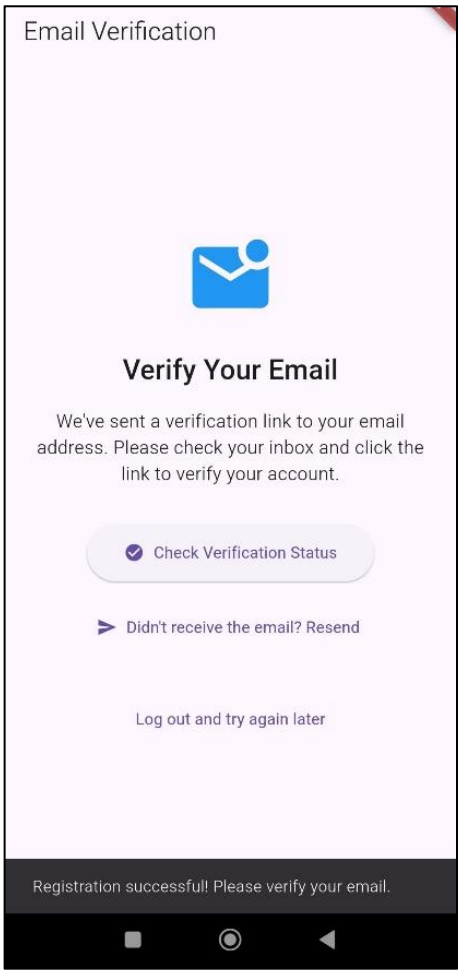



*Figure 4.8 Email Verification*

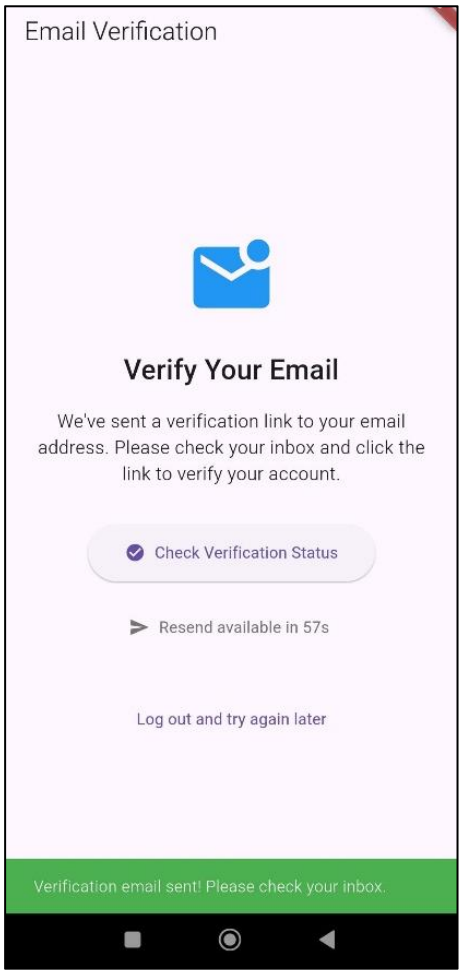



*Figure 4.9 Email Resend Countdown*

The student will be navigated to the page as shown in Figure 4.8 after they register an account to verify the ownership of the email they provide. In this page, the student can click the 'Check Verification Status' button to sign in to the system and navigated to the homepage after they verify their email. Furthermore, the student is given with option to request system to resend the verification email and log out of the system.

From Figure 4.9, it is clear that the student can only request for a new verification email after 60s cooldown of the first attempt. The resend button will be temporarily disabled in the 60s cooldown duration.

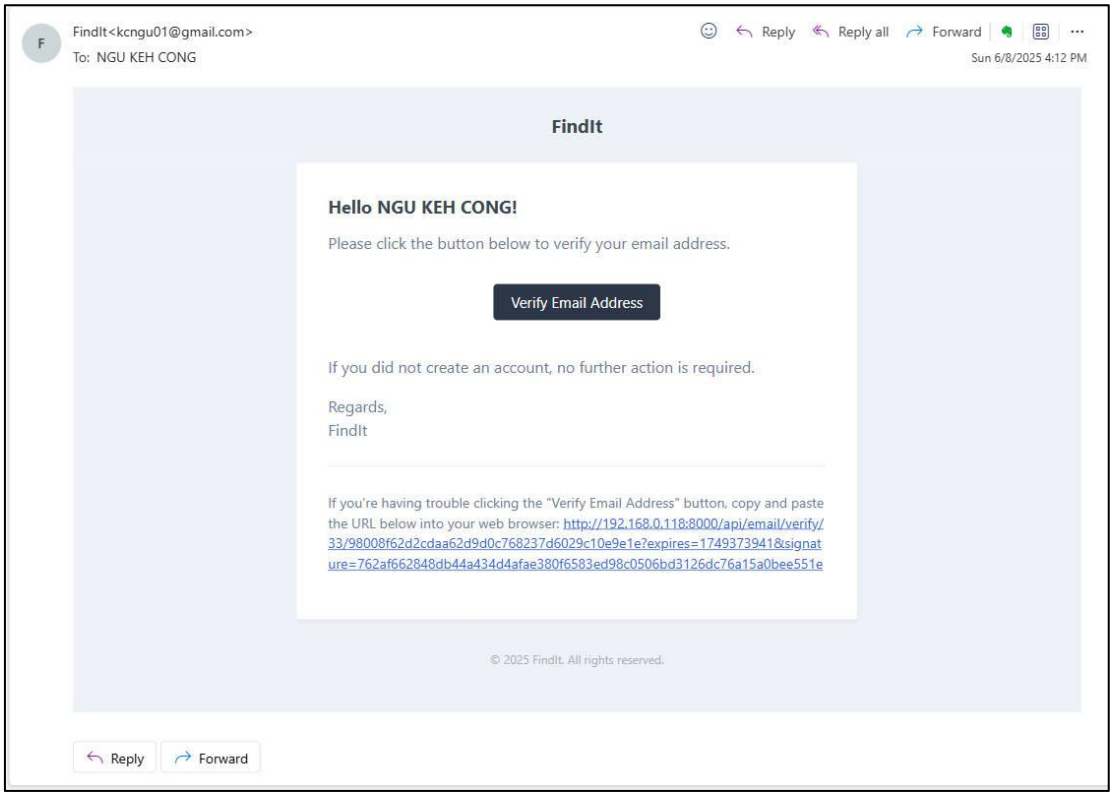



*Figure 4.10 Verification Email Received*

Figure 4.10 reveals the verification email received by the student after a successful registration. The student will need to click the 'Verify Email Address' button to verify their email so can later access the full system functionalities. If the student believe they are not the one has done the register action, they can ignore the message.

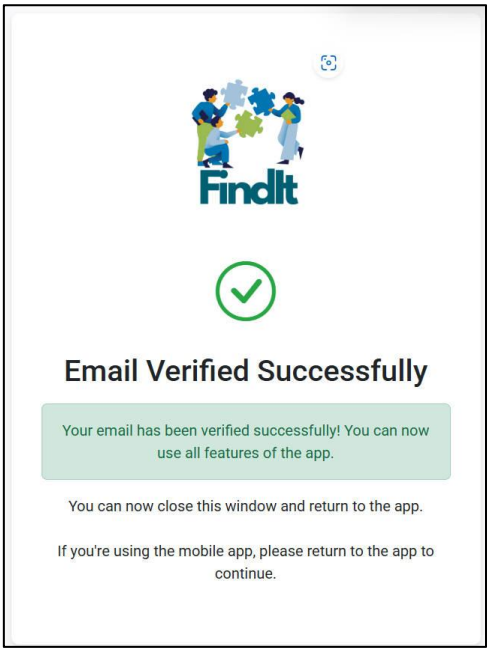



*Figure 4.11 Verification Passed*

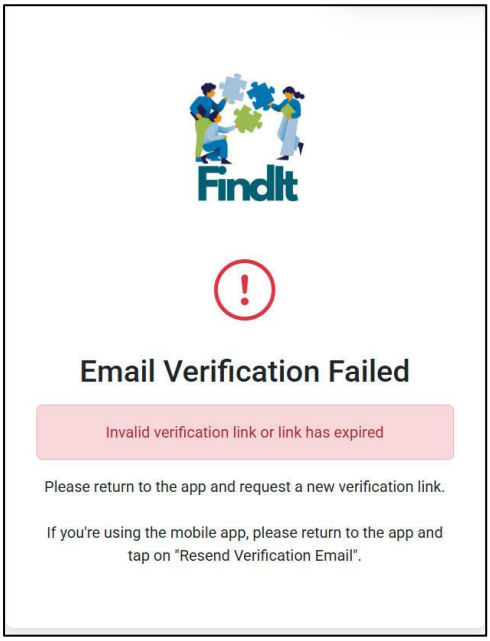



*Figure 4.12 Verification Failed*

As seen in Figure 4.11, a webpage will be opened indicating the student has verified their email successfully. This webpage is shown after the student click the 'Verify Email Address' button in the verification email within an hour as illustrated in Figure 4.10.

From Figure 4.12, it is obviously that the student will fail to verify their email when they only attempt to do so after an hour the verification is received when the verification link has expired. They will have to request a new verification link from the system to verify their email.

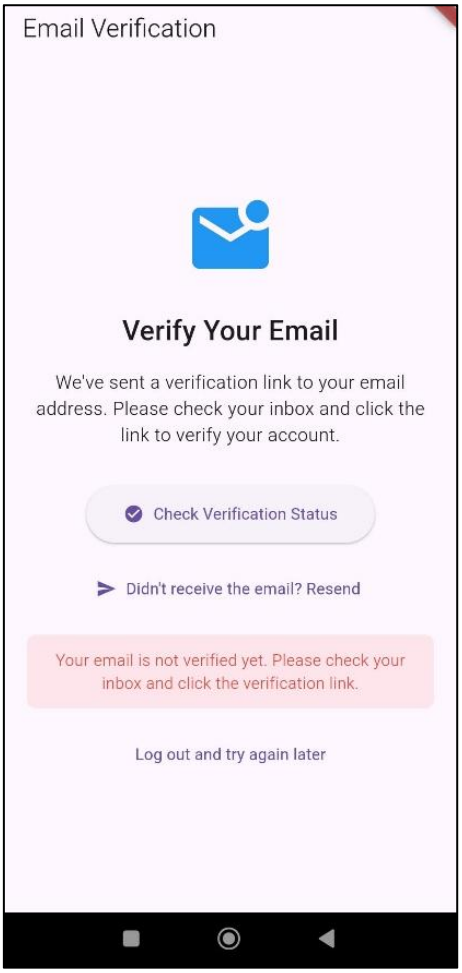



*Figure 4.13 Check Verification Status before Emal Verified*

As illustrated in Figure 4.13, an error message will be displayed showing that the email is not verified when they click the 'Check Verification Status' button before their email is verified successfully.

#### 4.3.1.3 Student Login

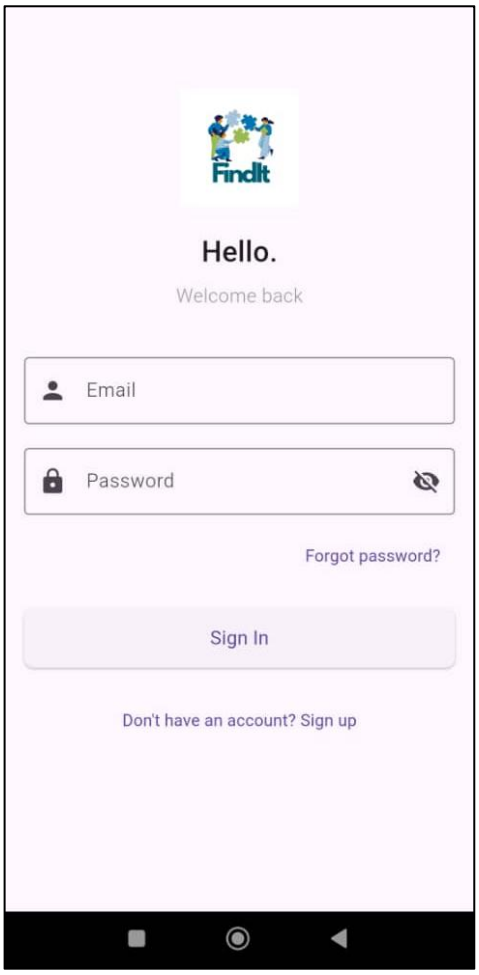



*Figure 4.14 Login Page*

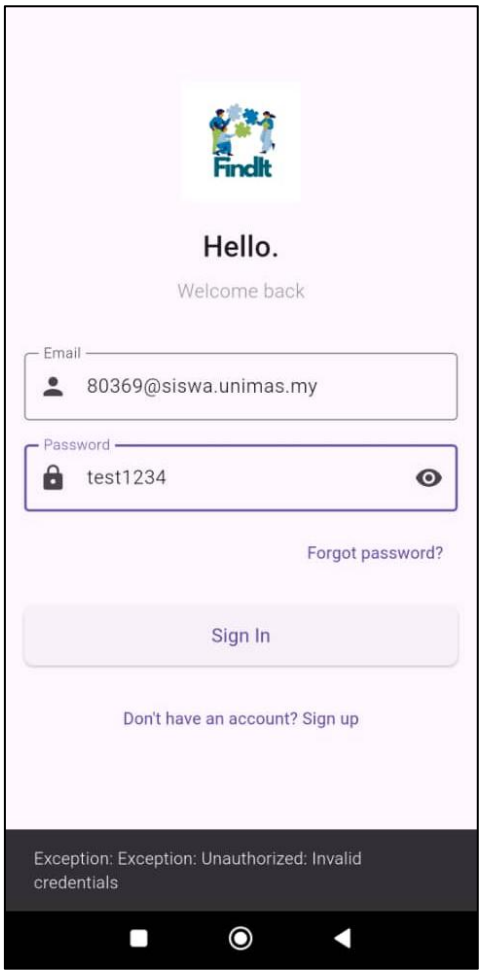



*Figure 4.15 Login Failed*

Figure 4.14 depicts that a student will have to input his email and password as the credentials to login to the system as an important flow in authenticating their identity.

As shown in Figure 4.15, an error message will be displayed if they user login with wrong credentials.

#### 4.3.1.4 Student Logout

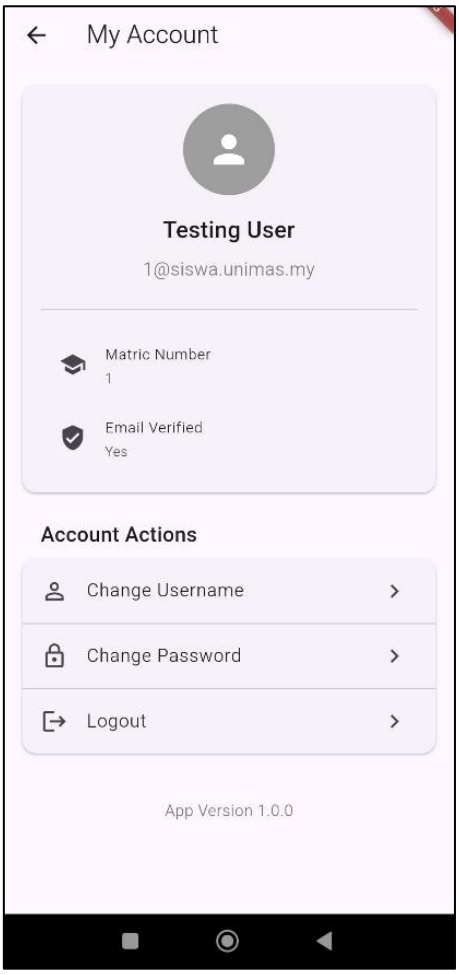



*Figure 4.16 Logout button*

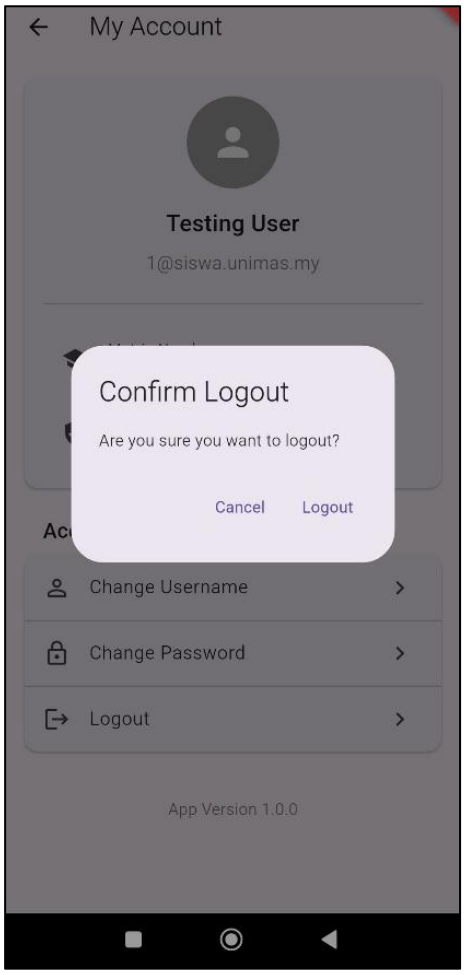



*Figure 4.17 Confirm logout*

Figure 4.15 illustrates the 'logout' button is located in the 'My Account' page, and clicking it will trigger a dialog asking for confirmation to logout of the application as shown in Figure 4.17. After the student logout successfully, they will be redirected to the login page.

#### 4.3.1.5 Forget Password

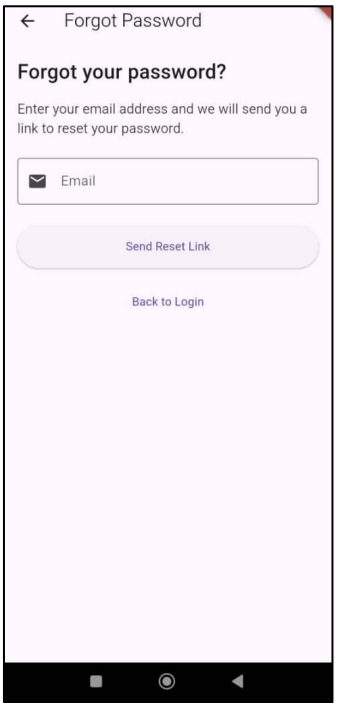



*Figure 4.18 Enter email address to receive password reset link*

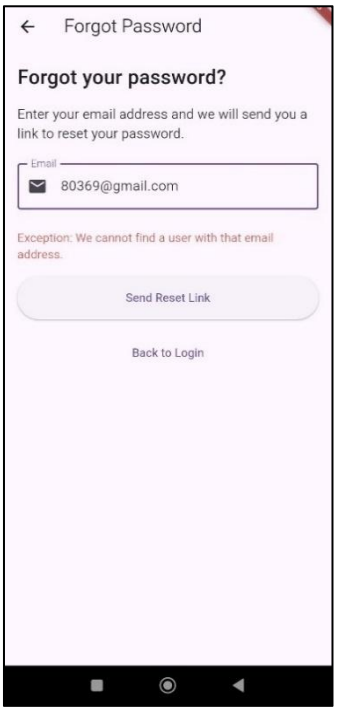



*Figure 4.19 Email not found*

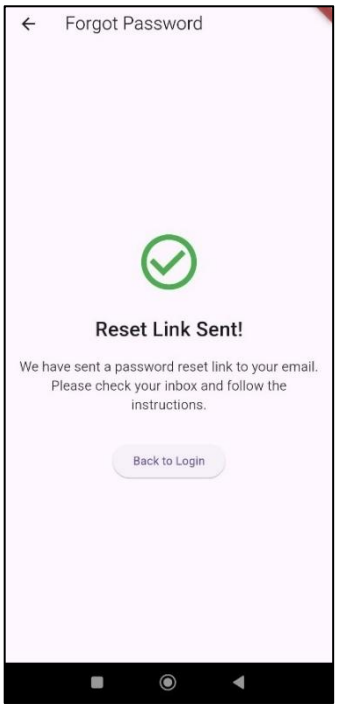



*Figure 4.20 Reset link sent*

Figure 4.18 depicts that the student can enter their email address to receive a password reset link if they found forgotten their password.

Figure 4.19 illustrates the scenario that an error message is appeared after the student submits an email address and the system cannot find a matching account. It means the email address is not registered and hence cannot send a password reset link to them.

The screen as shown in Figure 4.20 will appear and a password reset link will be sent to the student's email if the entered email address is found in the system. The student is also instructed to check their inbox and follow the instructions to reset their password.

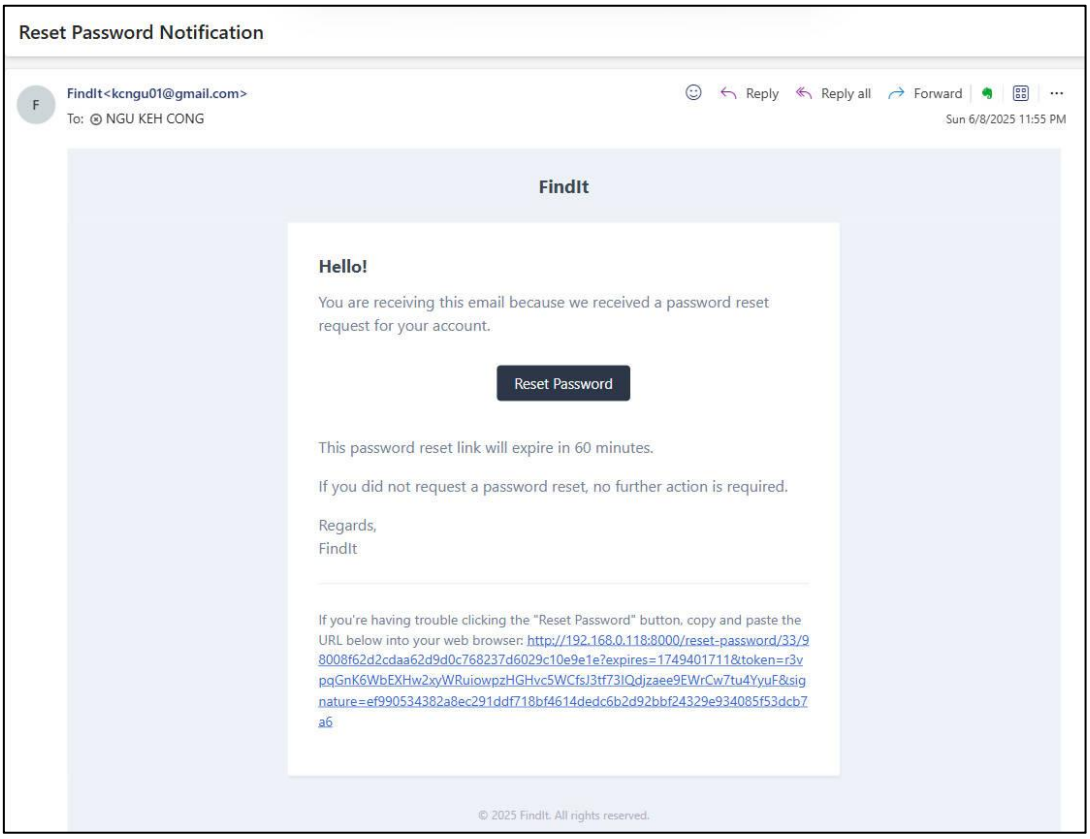



*Figure 4.21 Password Reset Email*

Figure 4.21 shows a password reset email sent from the system to a student. The student is required to click the button 'Reset password' to initiate the process of resetting password. It also tells the user to reset password within 60 minutes. Additionally, a direct URL is provided for resetting the password in case the button does not work.

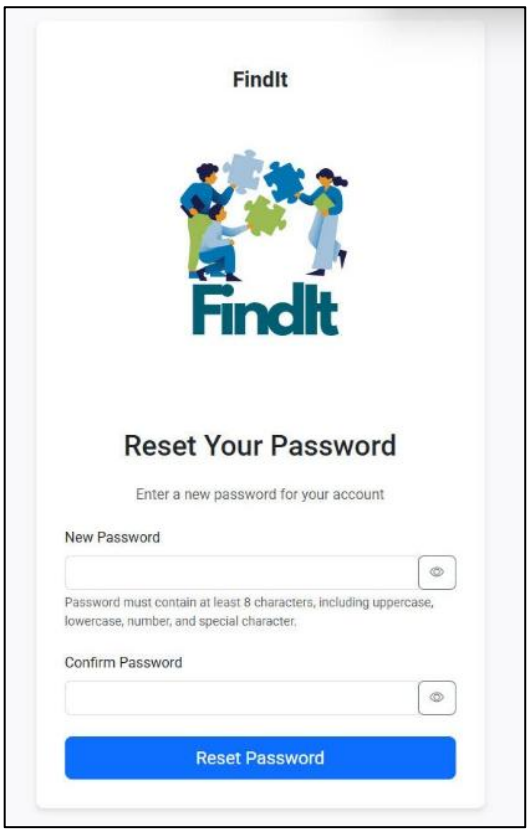



*Figure 4.22 Password reset form*

*Figure 4.23 Password does not match*

Figure 4.22 prompts the user to enter and confirm a new password, and the new password must meet the specified requirements to ensure password security.

Figure 4.23 illustrates a case that an error message is displayed if the confirmed password does not match the new password.

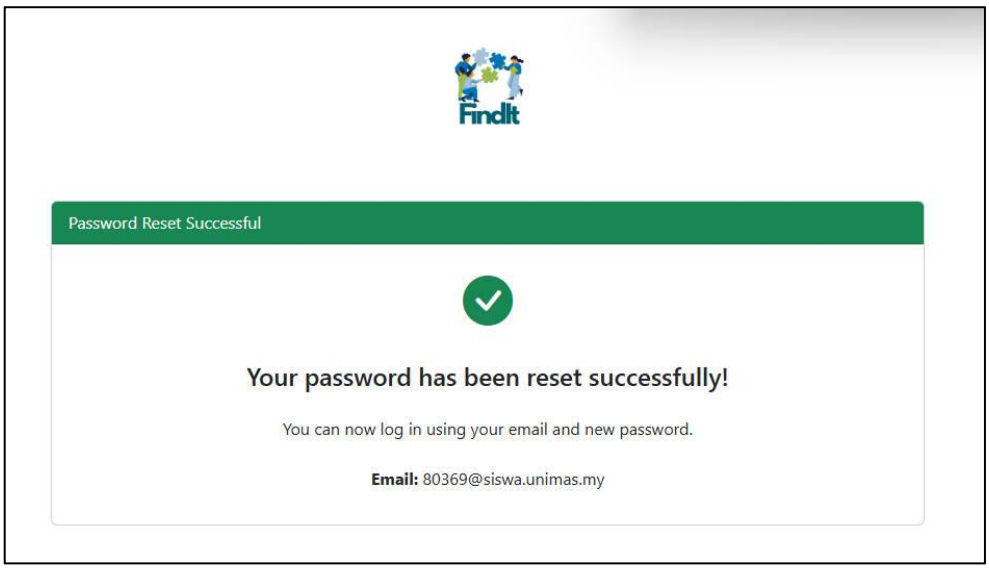



*Figure 4.24 Password reset successfully*

Student is navigated to a page as shown in Figure 4.24 to notify them they have reset the password successfully.

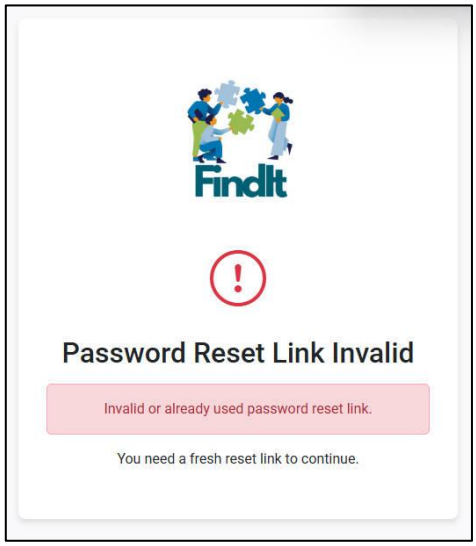



*Figure 4.25 Password reset link invalid*

If the password reset link is expired or was used to reset the password successfully, then Figure 4.25 will display an error message instead of the password reset screen.

#### 4.3.1.6 Change Password

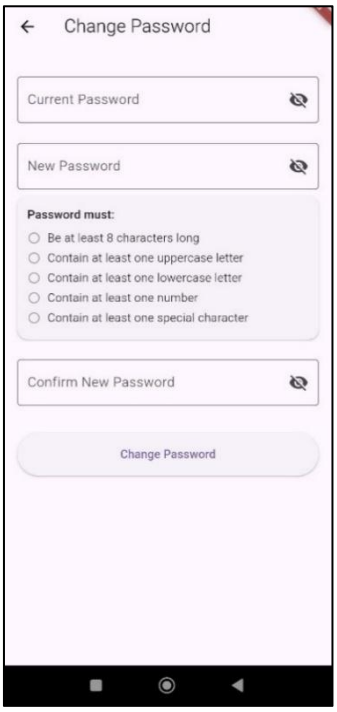



*Figure 4.26 Change password screen*

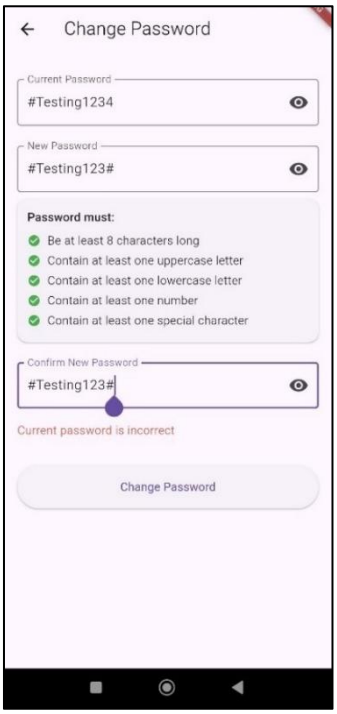



*Figure 4.27 Current password incorrect*

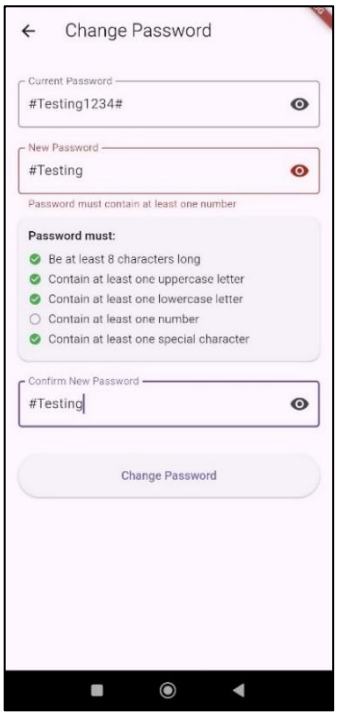



*Figure 4.28 New password invalid*

Figure 4.26 illustrates a change password screen where user need to enters their current password correctly, set a new password meets all the listed requirements (the circle box will be filled with green tick once the corresponding requirement is met) and confirms the new password.

A wrong current password entered by the student will trigger an error message in Figure 4.27, prompting the user to re-enter the correct current password.

An invalid new password not meeting all listed requirements will make the change password failed until the student resolve the error as shown in Figure 4.28.

#### 4.3.1.7 Report Lost/ Found Item

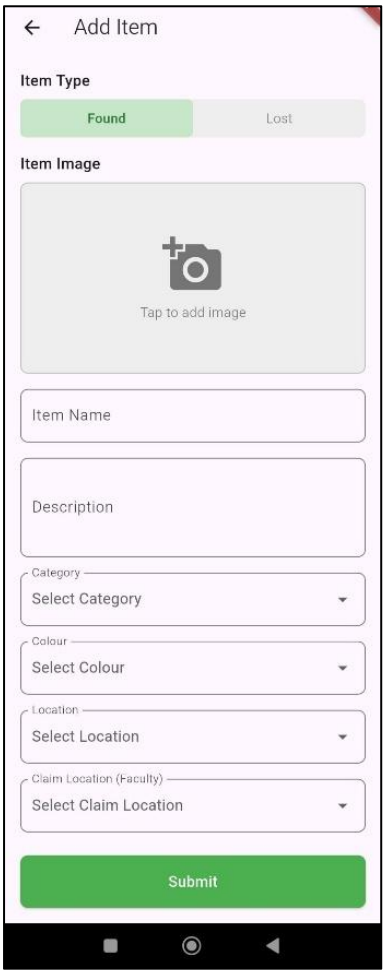



*Figure 4.29 Add a new lost/found item*

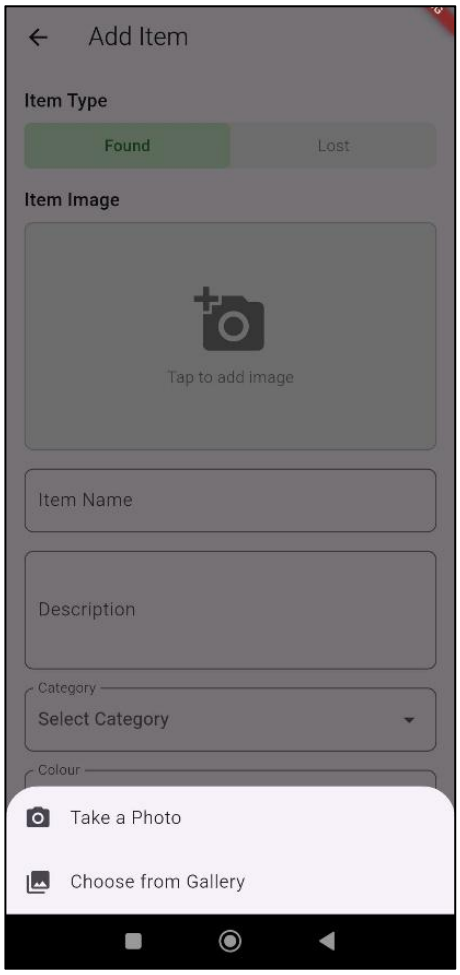



*Figure 4.30 Insert photo*

After student clicks the "+" button in the bottom navigation bar, it will redirect to the page as shown in Figure 4.29. Student can toggle between 'found' or 'lost' tab to submit different types of items. All input fields (item type, image, item name, category, location, claim location) are mandatory except the image and description field. To add an image, tap on the image section will prompt 2 options in Figure 4.30 for students to choose from, either to take a photo using the camera or choose an existing image from gallery.

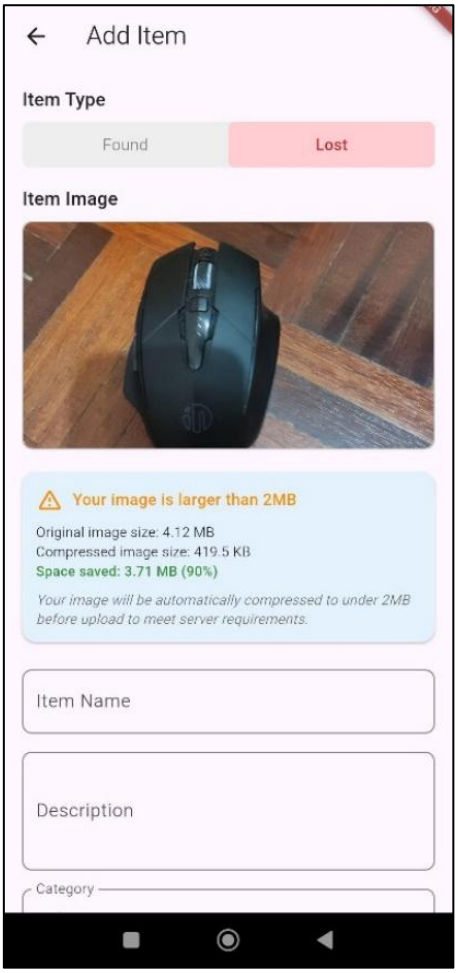



*Figure 4.31 Compress large image*

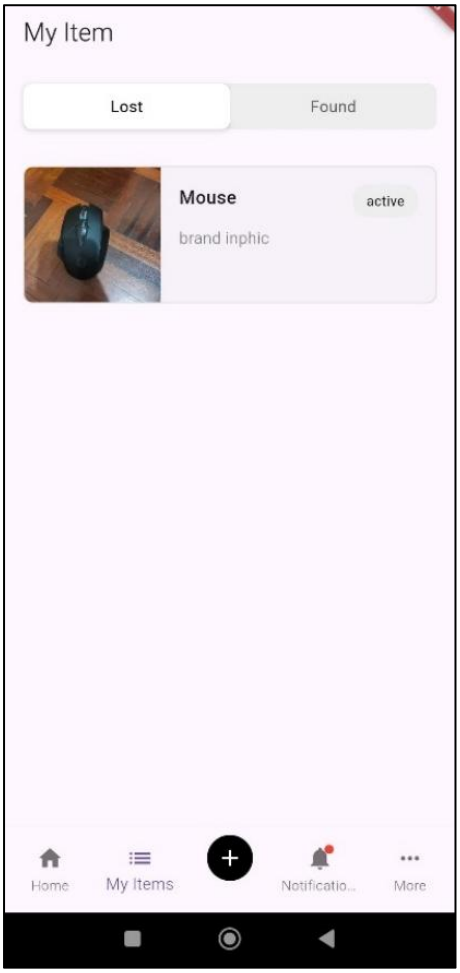



*Figure 4.32 My reported item*

Figure 4.31 illustrates an info message is appeared when the image uploaded by the student is larger than the acceptable size, 2MB. The message informs the student that the image is too large and hence it is compressed.

After the item created successfully from Figure 4.31, it will appear in Figure 4.32, where all the lost items and found items reported by the account owner (student) are listed here by toggling between 'lost' and 'found' tabs. Student can access My Item page by tapping 'My Items' in the bottom navigation bar.

#### 4.3.1.8 Manage Reported Item

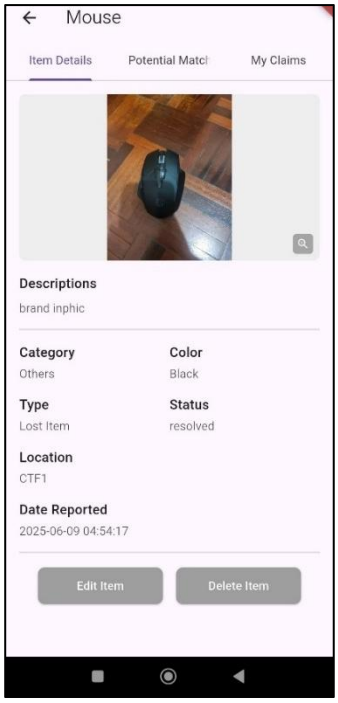



*Figure 4.33 Manage Reported Item*

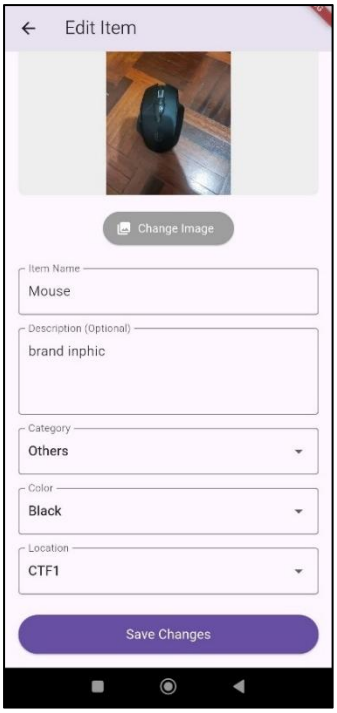



*Figure 4.34 Edit item information*

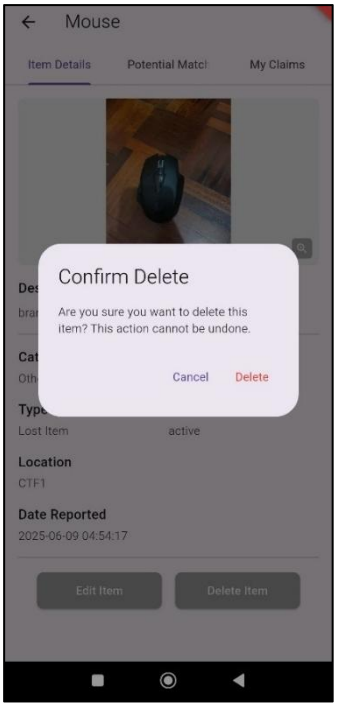



*Figure 4.35 Delete reported item*

Tapping the item in Figure 4.32 will redirect student to Figure 4.33 which shows the item details. When student feels like want to edit item details, they can click on the “Edit Item” button in Figure 4.33 and then will be navigated to Figure 4.34. Item image can be changed through a “Change Image” button, while all other item details like item name, description, category, colour, location, claim location (for found item only) are editable. It is important to tap on save changes button in Figure 4.34 to complete the update action. Students also have the ability to delete their item reports through the “Delete Item” button in Figure 4.33, which then a confirmation dialog in Figure 4.35 will be displayed to prevent them from deleting the item unintentionally.

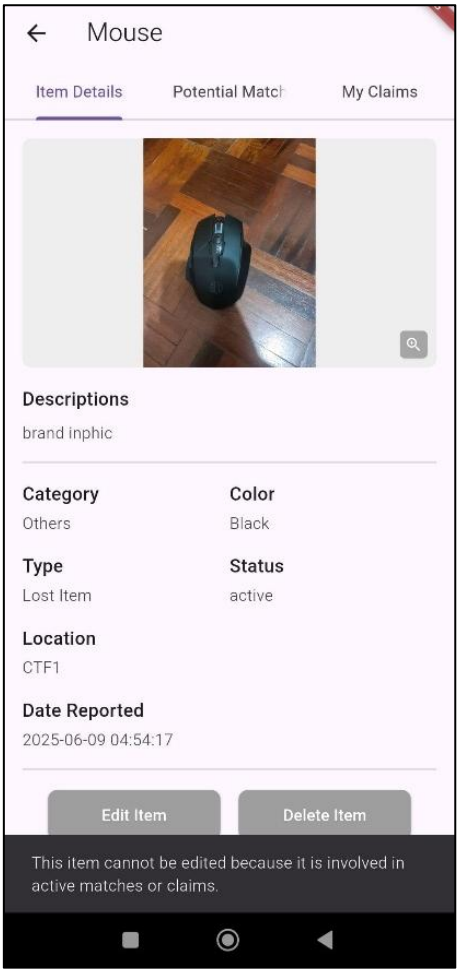



*Figure 4.36 Edit item information failed*

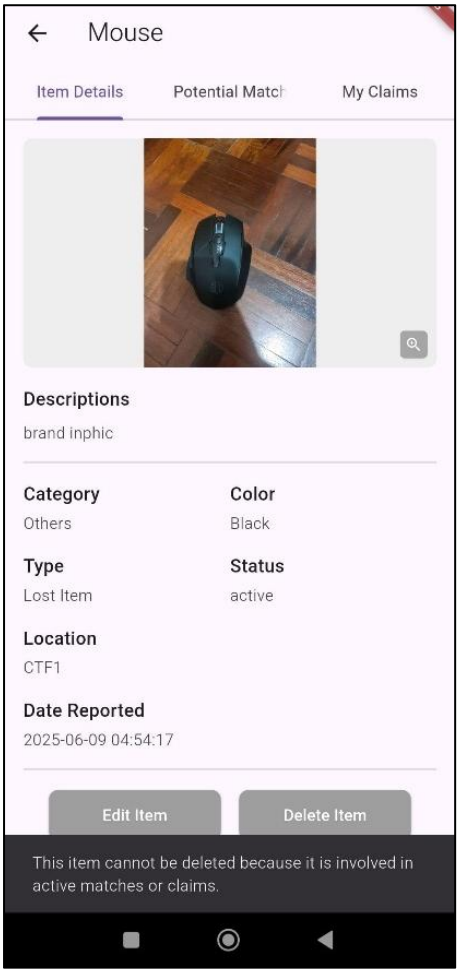



*Figure 4.37 Delete failed*

As shown in Figure 4.36 and Figure 4.37, the item is in “active” status, which indicates it is involved in ongoing claims with other students. This scenario will restrict the students from edit or delete item to maintain data integrity. Any attempts to edit or delete items will be failed and an error message will be displayed explaining the reasons.

#### 4.3.1.9 Search and Filter Item

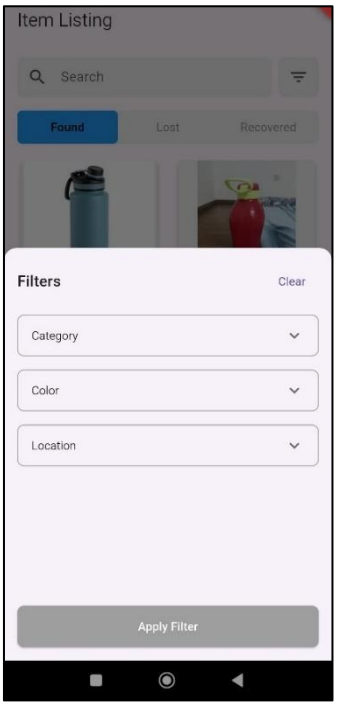



*Figure 4.38 Filter options*

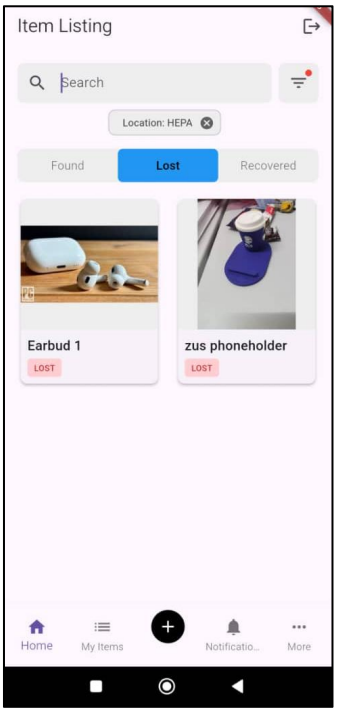



*Figure 4.39 Filter applied (Homepage)*

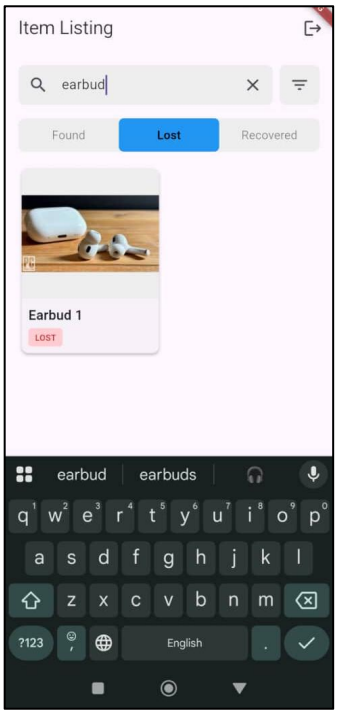



*Figure 4.40 Search item*

Figure 4.39 demonstrates the homepage student will be navigated to when they login successfully. Tapping on home icon in the bottom navigation bar or tapping system back button (numbers of times of tapping needed is based on user's current location in the app) from any other screens can help redirecting to this screen. The student will exit the app completely once they tapping system back button in homepage. "Found" items are displayed to students by default, while they can choose to look for other types of items by toggling between "Found", "Lost" and "Recovered" tabs. Pull-down-to-refresh feature is provided so students can easily retrieve all the latest items reported from the system. "Recovered" here refers to the items which the ownership is confirmed (i.e. claims on the items approved by admin).

Clicking the filter icon besides the search bar in Figure 4.39 will trigger a detailed filter options in Figure 4. Figure 4.39 demonstrates the students can narrow down their search by

selecting specific categories, colors and location where the item is lost or found. After selecting the desired filters, tapping “Apply Filter” button will return all the items which meets the filter applied. Students can remove all the filters applied by tapping on the word “Clear” in Figure 4.38, or if they prefer to remove specific filter only they can tap on the “x” icon near the filter applied under search bar as shown in Figure 4.39. Furthermore, a red dot near the filter icon will only be visible when there is any filter applied to help the student know the filter is applied to the search result.

The search functionality shown in Figure 4.40 enable students to type keywords into the search bar to refined their search result. These integrated features work together to help students efficiently locating the desired items reported in the system.

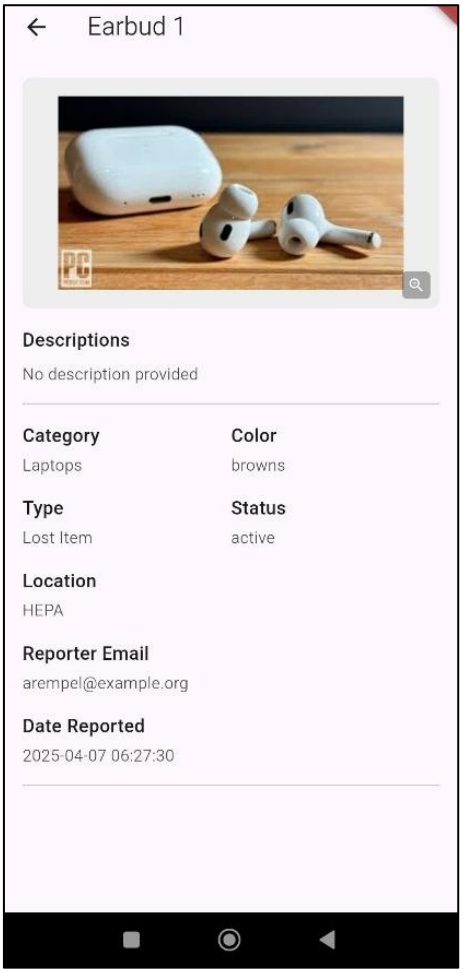



*Figure 4.41 Lost item information*

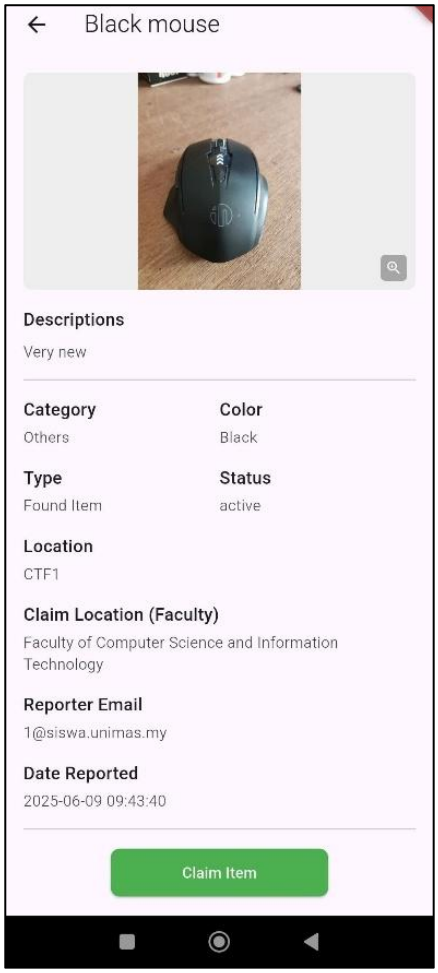



*Figure 4.42 Found item information*

Tapping on any lost item from the search results in Figure 4.39 and Figure 4.40 will display the detailed item information to students, which include item image, descriptions, category, color, type, status, location, reporter email, and date reported. No “Claim Item” button is available as illustrated in Figure 4.41 since it is a lost item.

Figure 4.42 depicts the found item information which is similar to lost item information, but with an extra information, location to claim the found item. Besides, a “Claim Item” button is available for the students to claim this item without reporting any lost item.

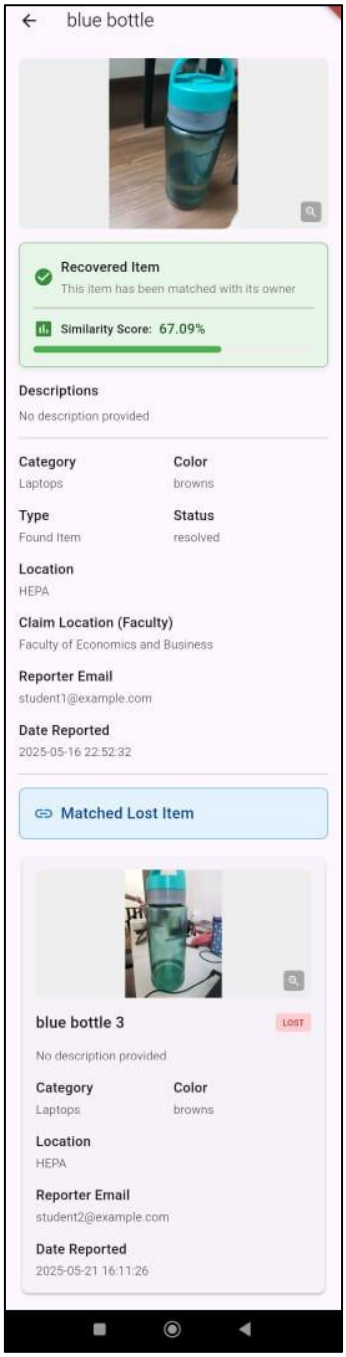



*Figure 4.43 Recovered item information (matched lost item)*

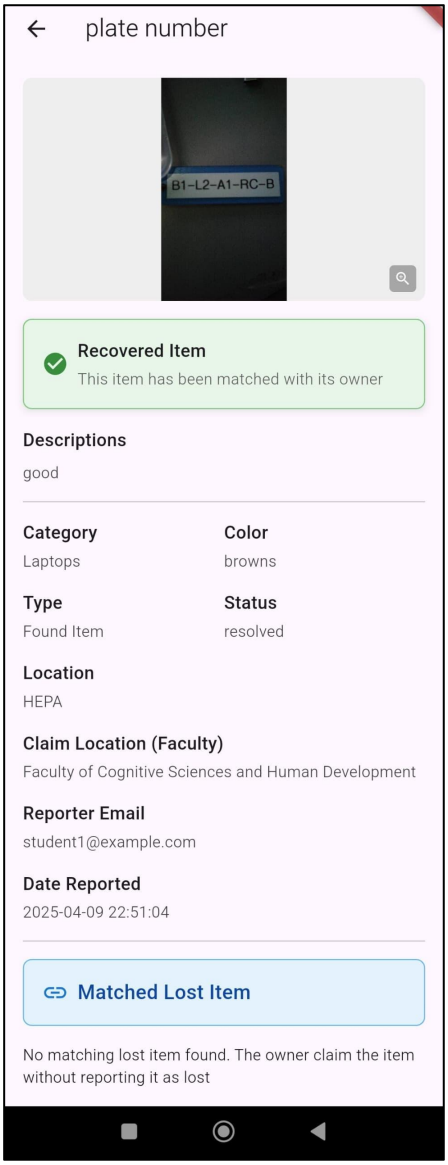



*Figure 4.44 Recovered item information (no matched lost item)*

Figure 4.43 illustrates the detailed information of a found item and its matched lost item reported by students. Student can view recovered item information by selecting a recovered item from the homepage. Upper section as shown in Figure 4.43 displaying found item information with the similarity score between found item and the matched lost item, while the section below shows the matched lost item information.

Figure 4.44 demonstrates the case where a found item is recovered and returned to the owner who submit a claim request without having previously reported the item as lost. As a result, no similarity score between found item and the matched lost item is found, and no information of lost item will be shown.

#### 4.3.1.10 Claim Items without Report Item Lost

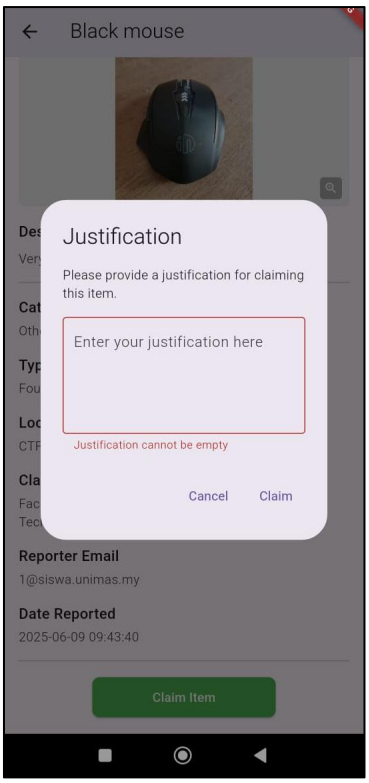



*Figure 4.45 Submit claim request with justification*

When a student found their lost item is reported by others as a found item in the homepage as shown in Figure 4.39, they can tap the found item and then tap the button "Claim Item" in Figure 4.42, followed with enters justification when submitting claim request as shown in Figure 4.45. The justification is for the reference of admin to decide approving or rejecting the claim.

#### 4.3.1.11 Claims Possible Matches for the Lost Item Reported

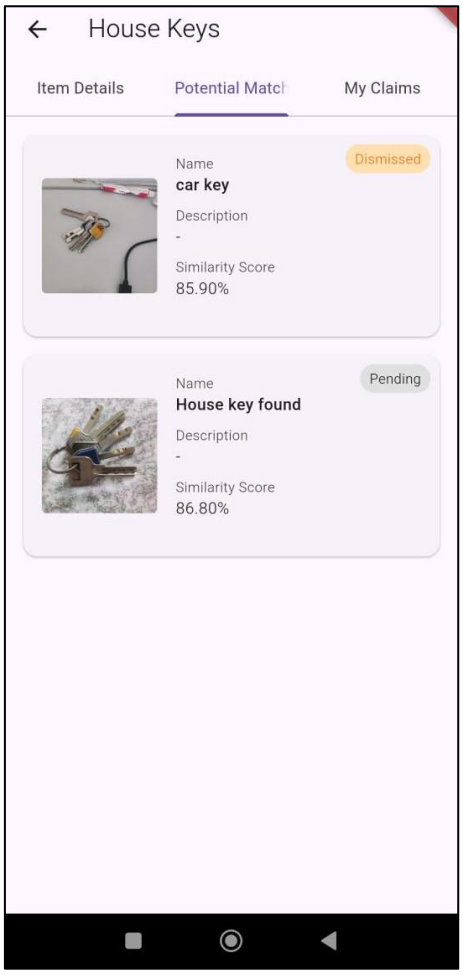



*Figure 4.46 List of matched found items*

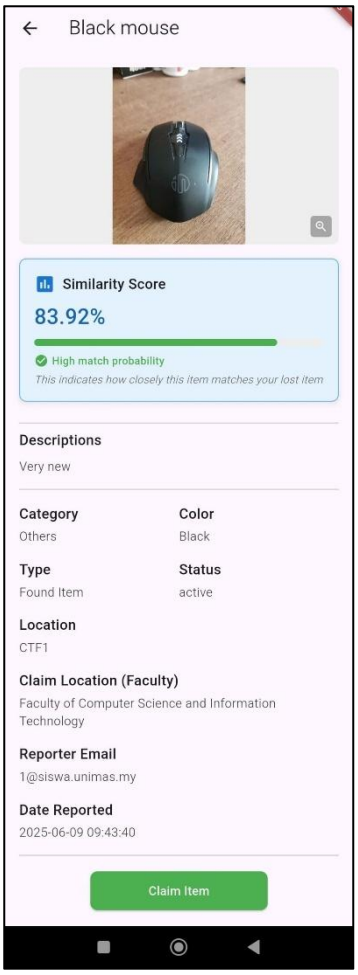



*Figure 4.47 Claim matched found item*

Figure 4.46 illustrates all matched found items related to the reported lost item discovered by the system. Student can navigate to this page by tapping 'My Item' in bottom navigation bar, select a lost item with matched found item and then finally toggle to 'Potential Match' tab. The matched found item overview includes information like item name, description, similarity score between lost and found item, and the match status. There are 5 match statuses in total: available, pending, approved, rejected and dismissed, covering all possible scenarios. Available is used when the item's ownership is not yet confirmed and open to receive claim request. Pending is used when the student (account holder) has submitted a claim request for the item. Approved is used when the admin approved the claim request for the found item. Rejected is used when the admin rejected the claim request for the item. Dismissed is used

when the student (account holder) has ongoing claim request for the same lost item or the item already claimed by other students.

Figure 4.47 illustrates the found item details with a claim button available when student tap on a match found item with status ‘available’. The students also have to provide brief justification upon submitting claim request as shown in Figure 4.45. After submitting claim request, the corresponding matched found item will have status changed from ‘available’ to ‘pending’, while statuses of all other matched found item for the same lost item will be changed from ‘available’ to ‘dismissed’ to restrict students from submitting more than 1 claim request for the found item related to same lost item at a time. When the claim request submitted previously is rejected by the admin, all matched found item will have status changed from ‘dismissed’ to ‘available’ now if the item is not yet claimed anyone. If the claim request submitted is approved by the admin, then the other found items will remain in status ‘dismissed’ instead of changing to status ‘available’ to prevent the action of claiming more than 1 item for the same lost item.

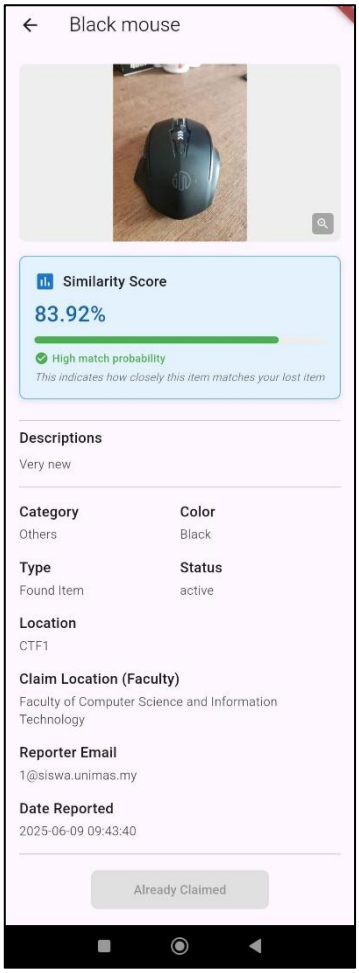



*Figure 4.48 Already claimed*

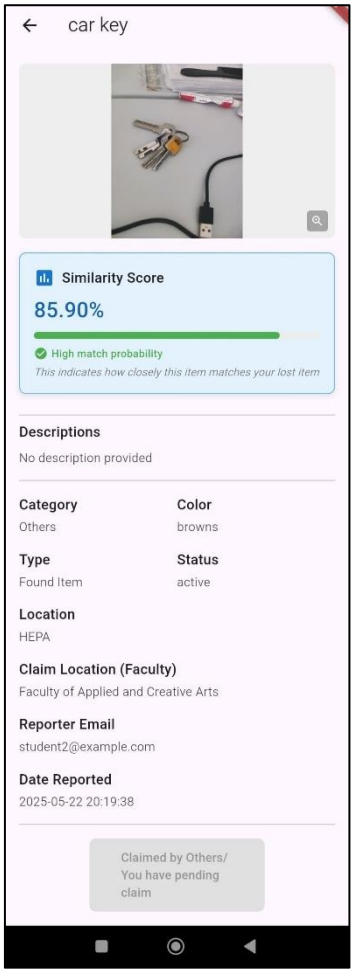



*Figure 4.49 Claimed by others / other claim ongoing*

As shown in Figure 4.48, after submitting the claim request, the claim button will be disabled and the text 'Already Claimed' will be shown to inform the student that they have already submitted a claim for this found item before. A student can only submit one claim request for one found item, which meant they cannot submit claim request for the same found item anymore if they are rejected.

Figure 4.49 illustrates that the text "Claimed by Others / You have pending claim" will be shown when the student attempts to claim a matched found item with status 'dismissed'. It means the student will have to wait the ongoing claim request being processed by the admin before they can submit a claim request for the found item related to the same lost item or they can no longer submit claim request for this found item if it is already claimed by others.

#### 4.3.1.12 Track Claim Status

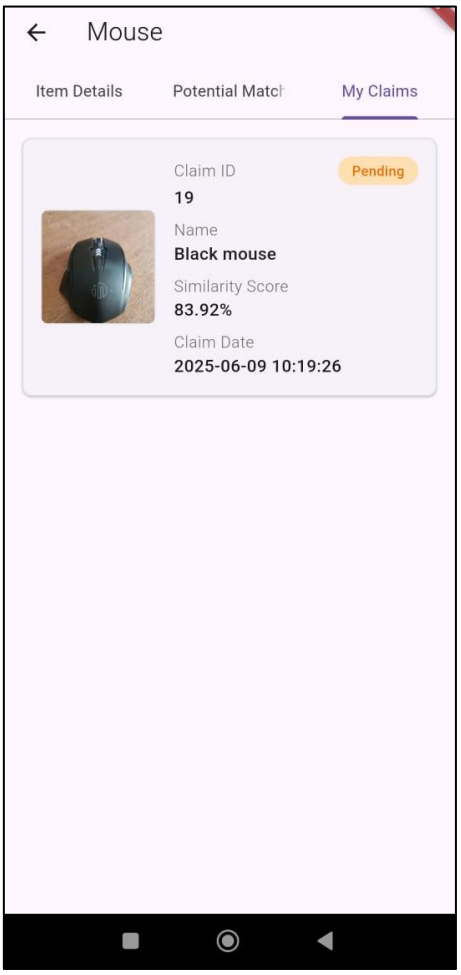



*Figure 4.50 All claims for the same reported lost item*

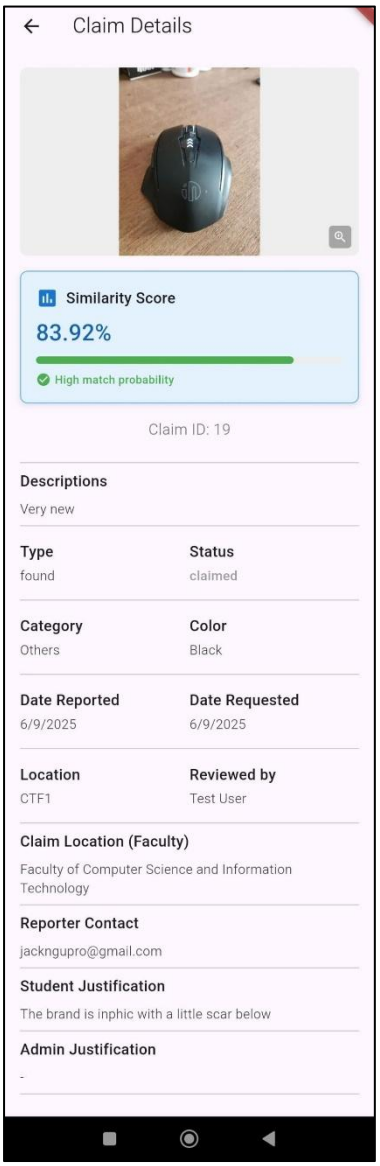



*Figure 4.51 Claim details (involves matched found item)*

In Figure 4.50, all the claims for found items corresponding to a lost item is listed, with an overview of claim status is displayed for each claim to let the student track the claim status efficiently, so they only will have to tap on the claims when they want to see detailed claim details illustrated in Figure 4.51. The important information included in Figure 4.51 includes claim ID, location to claim the item if approved, and the claim status.

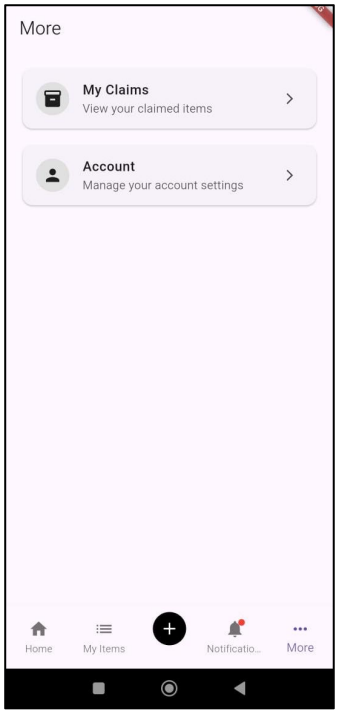



*Figure 4.52 More Options*

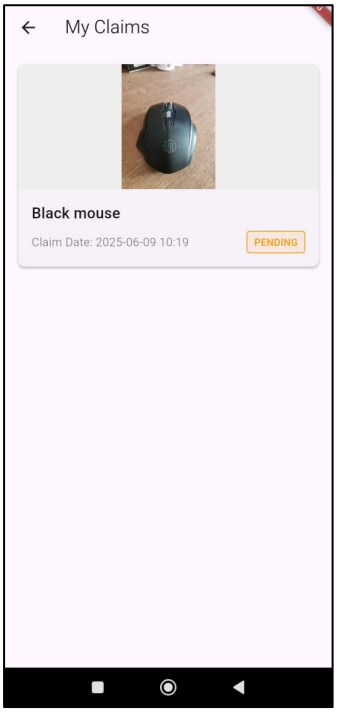



*Figure 4.53 All submitted claims*

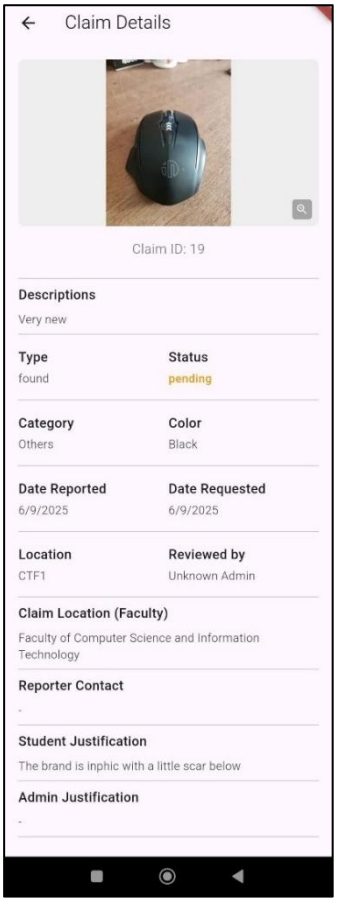



*Figure 4.54 Claim details*

Figure 4.52 depicts that the students will have to first tap on 'More' in bottom navigation bar and then tap on the option 'My Claims' before they are navigated to Figure 4.53 which displays all the claims submitted before.

Figure 4.54 demonstrates a page showing detailed claim details. The claim details screen won't display similarity score between the lost item and found item if the claim does not involve any matched found item or the claim details screen is access through navigation route ['More' in bottom navigation bar > 'My Claims' option > Select claim]. Accessing claim details (if the claim involves found item from potential matches generated by the system) through navigation route ['My Item' in bottom navigation bar > Toggle 'lost' tab > Select lost item > Toggle 'My Claims' tab > Select claim] will shows the similarity score as shown in Figure 4.51.

#### 4.3.1.13 Notifications

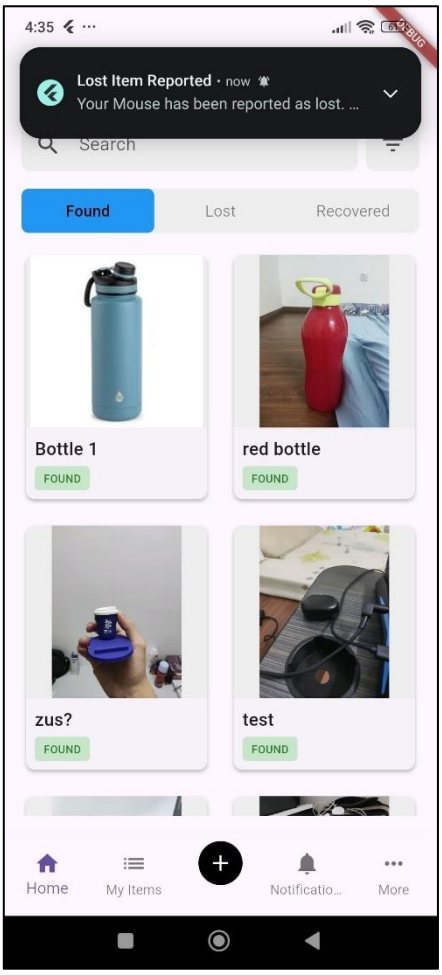



*Figure 4.55 Match notification*

Figure 4.55 illustrates that the pop-up notification will be sent to the students regarding claim updates and matches when they are logged-in. Students will be notified when the admin approved or rejected their claim request. Apart from that, the students will be notified immediately to inform them whether an existing match was found or no current match exists in the system when they first reported the item as lost. Students will also receive notifications in the future whenever any potential matches for their lost items are discovered, ensuring they stay informed of any possible new found items that could belong to them.

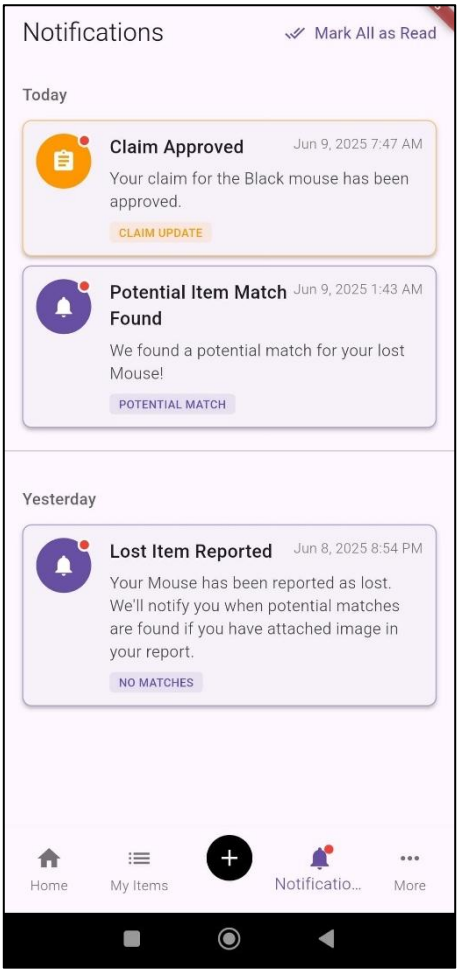



*Figure 4.56 Notifications*

Figure 4.56 demonstrates all the notifications received by the students. The page is accessed via tapping 'Notification' in the bottom navigation bar. If there are any notifications unread, there will be a red dot beside the notification icon, informing the student they may have missed some important notifications. Each notification is come along with notification type labeled as 'CLAIM UPDATE', 'POTENTIAL MATCH', or 'NO MATCHES' to quickly identify the nature of the alert. Students can mark the notification as read by tapping on it or tap on 'Mark All as Read'.

### 4.3.2 Admin

#### 4.3.2.1 Admin Login

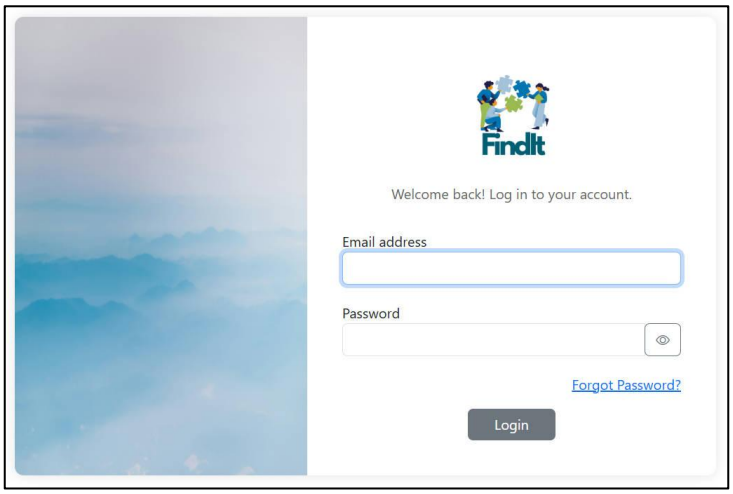



*Figure 4.57 Admin login screen*

Figure 4.57 illustrates the login screen for admin where they need to input registered email address and correct password in order to login successfully. The admin account is registered by the developer at the backend, so the admin is not able to register a new account themselves. Admin can request the credentials from the developer to login the system.

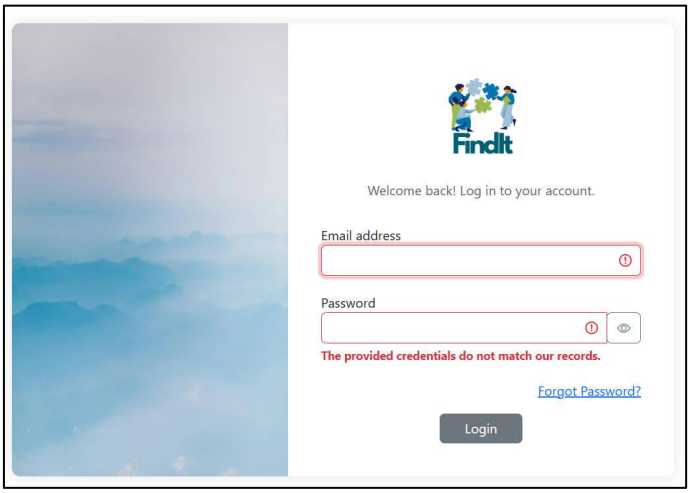



*Figure 4.58 Admin login failed*

If any of the provided credentials is incorrect, there will be an error message informing this as shown in Figure 4.58.

#### 4.3.2.2 Reset Password

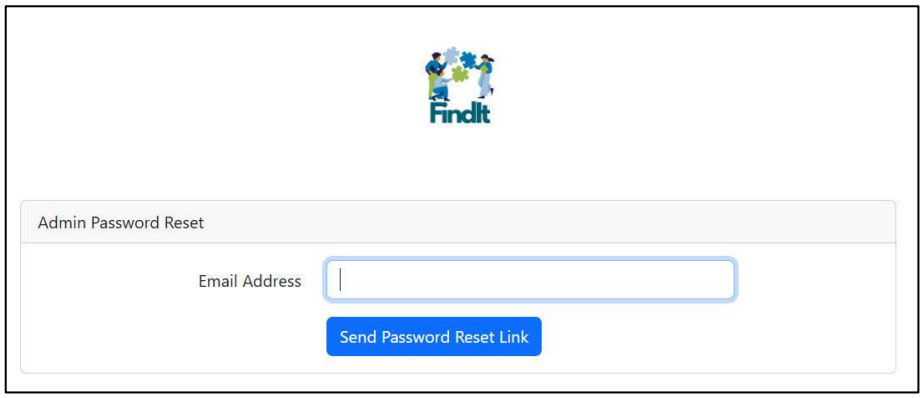



*Figure 4.59 Send password reset link screen*

Click "Forgot Password" hyperlink in Figure 4.57 will navigate admin to Figure 4.59 where they have to enters registered email address to receive the password reset link successfully.

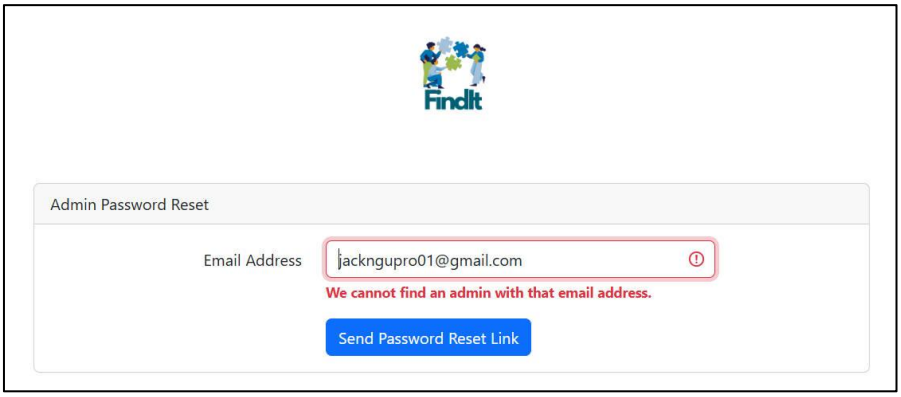



*Figure 4.60 Not existing email address*

As indicated by the error in Figure 4.60, any attempts to receive password reset link with an unregistered email address will fail.

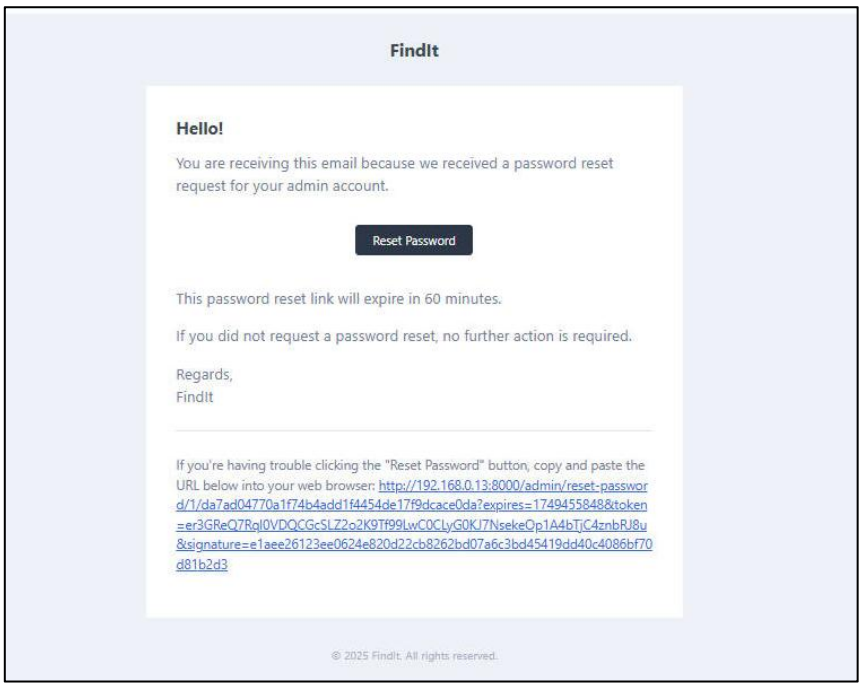



*Figure 4.61 Password reset email for admin*

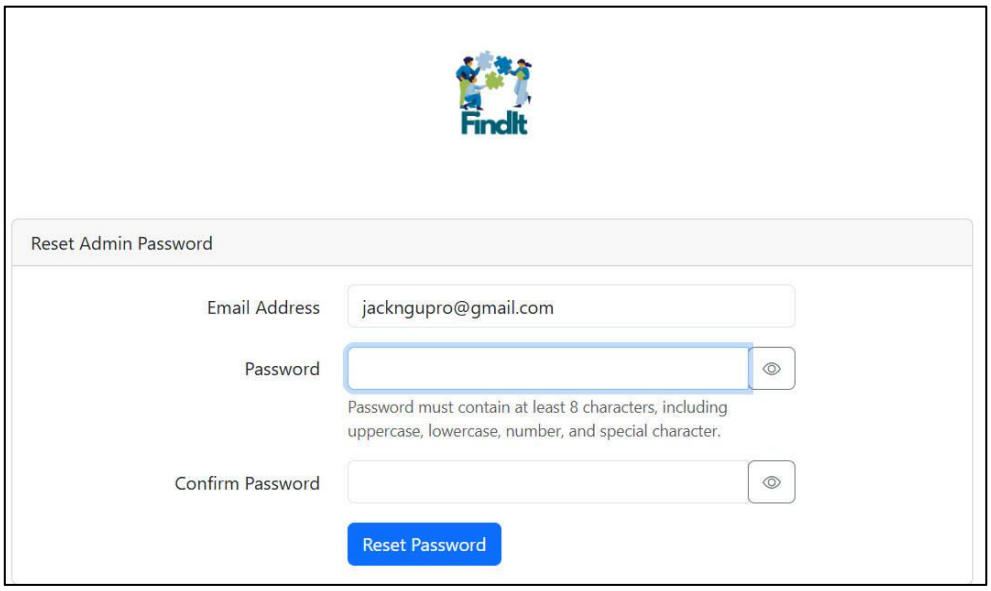



*Figure 4.62 Reset email password for admin*

If a correct email address is entered in Figure 4.59, an email in Figure 4.61 will be sent to the admin. In the email, a brief overview of the purpose of the password reset email, a “Reset Password” button, and a reset password link are given. The admin can click the button to open the reset password web page, but if any error occurs when opening the page, they may copy and paste the link in the browser to open the page.

Figure 4.62 illustrates the page to reset password. To reset password successfully, admin need to enter a new password which fulfil all the requirements (i.e. at least 8 characters, 1 uppercase, 1 lowercase, 1 number and 1 special character), and the password need to confirmed successfully.

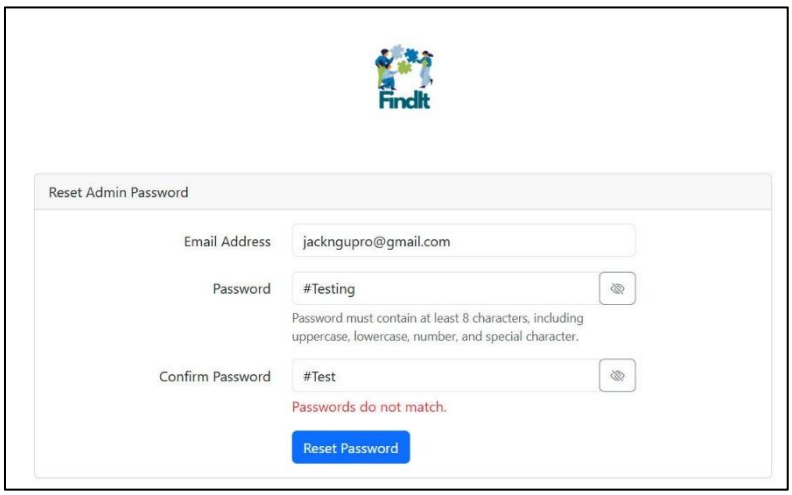



*Figure 4.63 Password not matched*

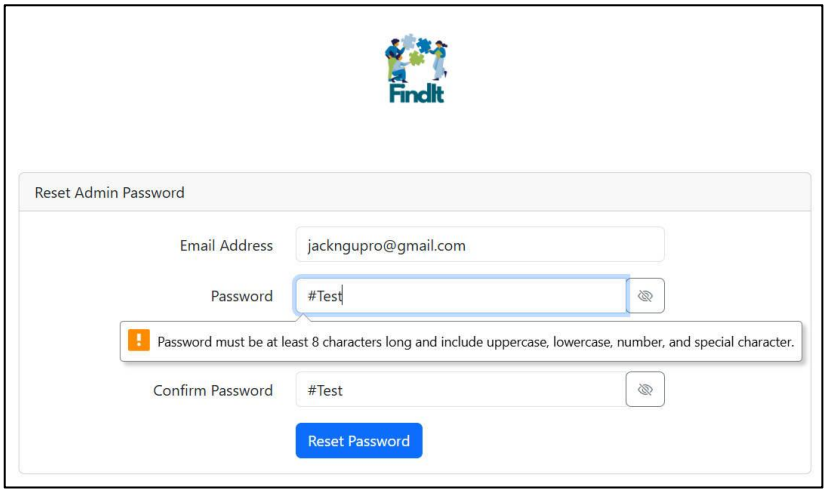



*Figure 4.64 New password requirements*

The error in Figure 4.63 arose when the new password is not the same in both 'Password' and 'Confirm Password' fields.

Figure 4.64 illustrates the error message prompted when admin attempts to set a new password does not match all the password requirements.

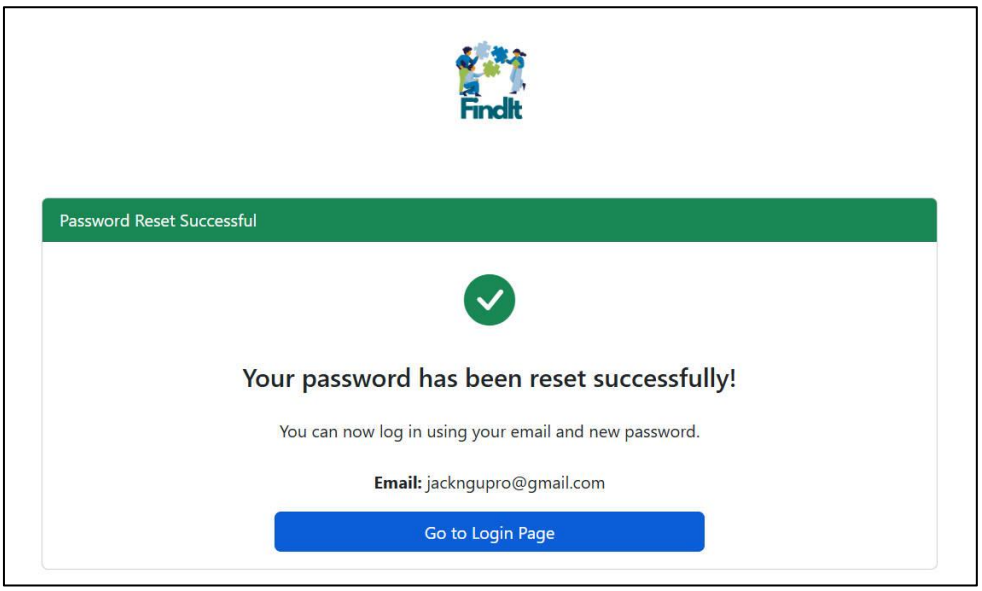



*Figure 4.65 Password reset successfully*

Admin will be redirected to Figure 4.65, which informs them the password is reset successfully. A "Go to Login Page" button is provided so the admin can efficiently continue with the login process using the new password set.

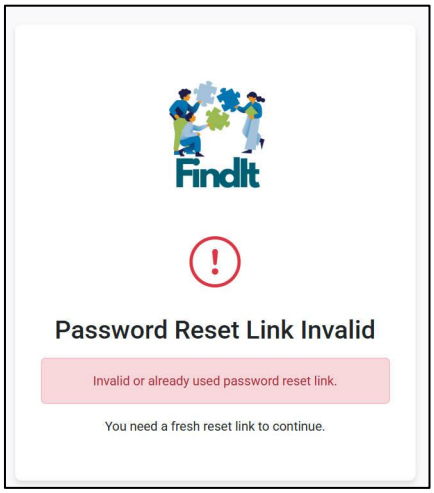



*Figure 4.66 Invalid password reset link*

If the password reset link had been used to reset a new password successfully or the link is used after exceeds 1 hour from the email is received, admin will be navigated to the page in Figure 4.66 explaining the possible error.

#### 4.3.2.3 Admin Logout

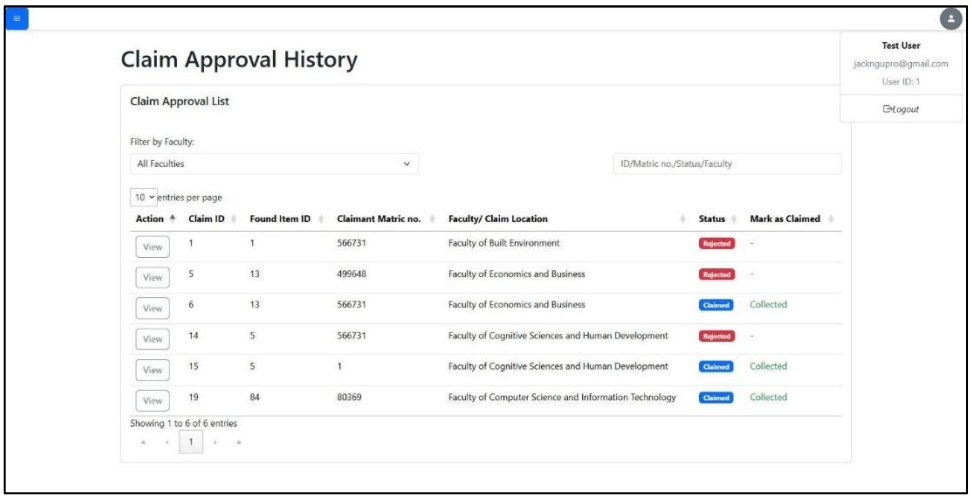



*Figure 4.67 Admin logout*

In Figure 4.67, it is illustrated that the admin will have to click on the user icon on the top right corner of top navigation bar, and then click the logout button in the profile dropdown menu section to logout from the system. In addition, the username, email address, and user id are displayed in the profile dropdown menu section.

#### 4.3.2.4 Side Menu

*Figure 4.68 Side menu*

As seen in Figure 4.68, a side menu is sliding out from the left side when the hamburger icon in the top navigation bar is clicked. The menu contains navigation items including 'Dashboard', 'Student', 'Item Characteristics', 'Category', 'Colour', 'Location', 'Claim

Review', and 'Claim Approval History', each navigating admin to the respective pages when selected.

#### 4.3.2.5 Dashboard

*Figure 4.69 Dashboard*

Figure 4.69 displays a dashboard for the admin after selecting navigation item 'Dashboard' from the side menu in Figure 4.68. The dashboard is important to provide admin with an overview of key metrics related to student registration and claims management. From the dashboard, the admin will be clear about the total number of registered students across all faculties. For the claim management section, there are 3 charts, each showing metrics of 'number of pending claims', 'number of approved claims', and 'recovery rate' respectively. The admin is allowed to apply filter to access the data for each faculty instead of all faculty by default. By hovering over the 'info' icon near the chart 'recovery rate', the calculation formula will be clearly stated as '(Approved Claims/ Total claims) x 100%' to let the admin have a clear and better understanding of this performance metrics. This dashboard layout consolidates administrative data into easily digestible visual components (i.e. charts), enabling quick assessment of student enrollment status, claim processing workflow and also overall system performance.

#### 4.3.2.6 View Student Information

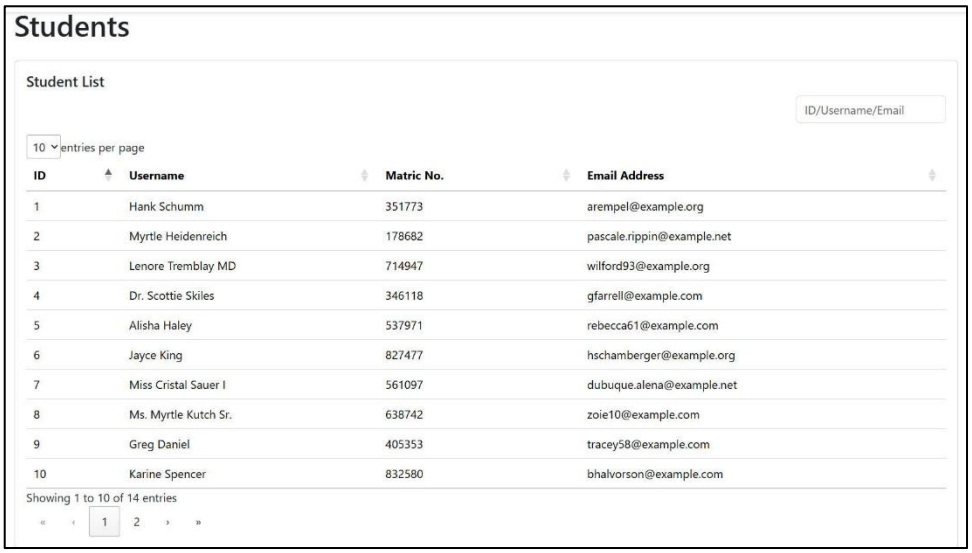



*Figure 4.70 Students information*

Figure 4.70 displays a list of student information after selecting navigation item 'Student' from the side menu in Figure 4.68. Admin is able to efficiently locate the student through the search functionality which allows the user to search by ID, username, matric no, or email address. Besides, the admin can sort the students' information based on student ID, username, matric no., and email address by clicking the arrow next to the columns. Lastly, admin can control the number of records visible simultaneously via configurable display option showing "10 entries per page".

#### 4.3.2.7 Manage Item Characteristics

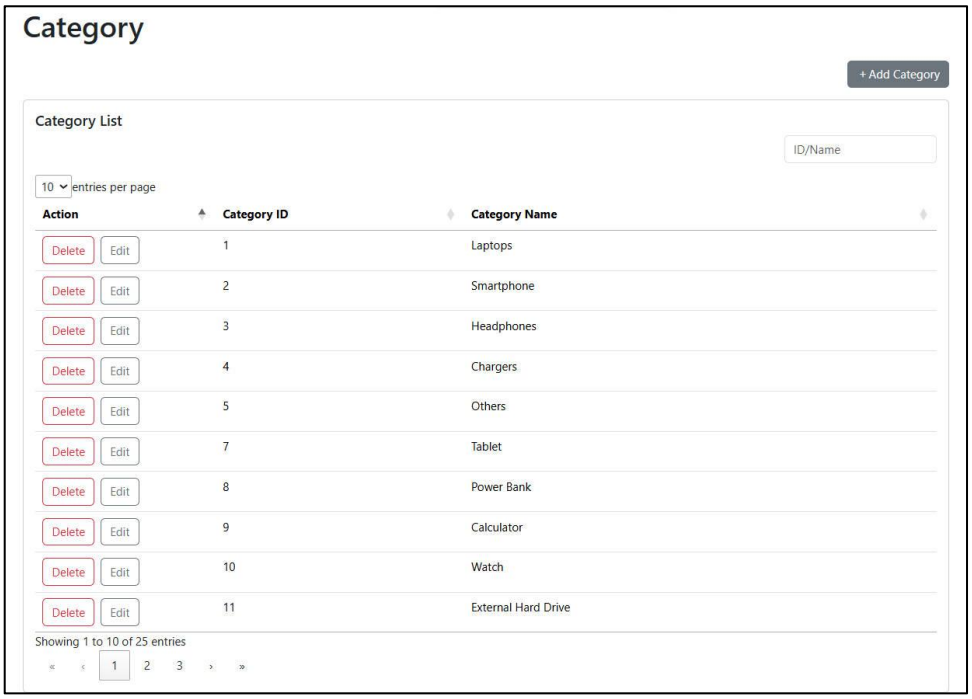



*Figure 4.71 Category Management*

Figure 4.71 illustrates an interface where admins are able to carry out full CRUD operations for managing item categories. Admin can add new categories via the "+Add Category" button positioned at the top right corner of the screen, while can modify and remove each existing category via the "Edit" and "Delete" buttons. Search functionality is provided so admin can locate desired existing category faster using category ID, or category name. Besides, the categories can be sorted based on alphabet or number using the arrow icon besides the column field name. Lastly, admin can control the number of records visible simultaneously via configurable display option showing "10 entries per page". This standardized interface design and functionality set serves as a template for other item characteristic management screens like Colour and Location modules.

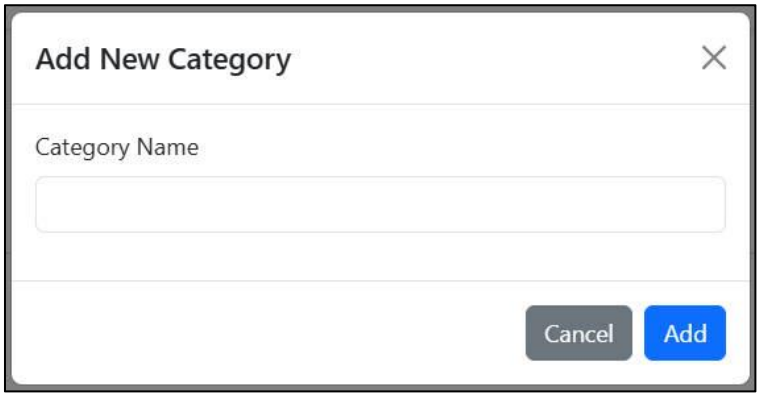



*Figure 4.72 Add new category*

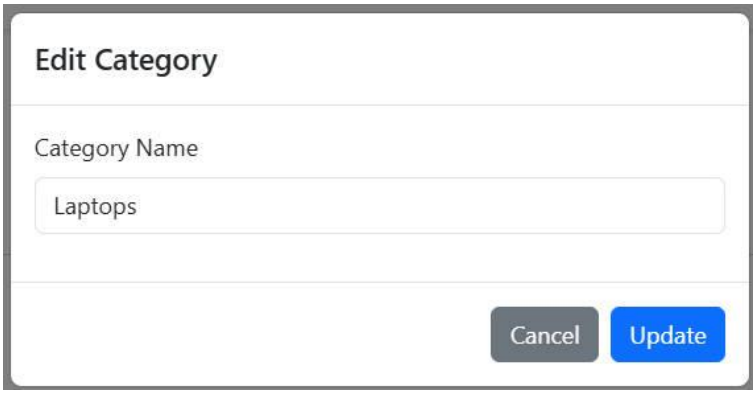



*Figure 4.73 Edit existing category*

*Figure 4.74 Category exists*

*Figure 4.75 Update fail (category name is referenced by items)*

*Figure 4.76 Delete fail (category name is referenced by items)*

Figure 4.72 shows a modal to add new category which will be triggered by clicking "+Add New Category" button in Figure 4.71, while clicking "Edit" button in any row of table will show Figure 4.72, which the existing category name will pre-populate the input field with current value automatically.

Figure 4.74 demonstrates an error message will be displayed at top of the page as shown in Figure 4.71 when the admin attempts to create a duplicate category name. Figure 4.75 and

Figure 4.76 reveals that the system prevents admin from modify or delete categories that are actively referenced by items reported by students.

This standardized approach to manage data, use of modal dialogs, and validation systems are consistently applied across all of the item characteristic modules within the system including the colour management and location management screens.

#### 4.3.2.8 Process Claim Request

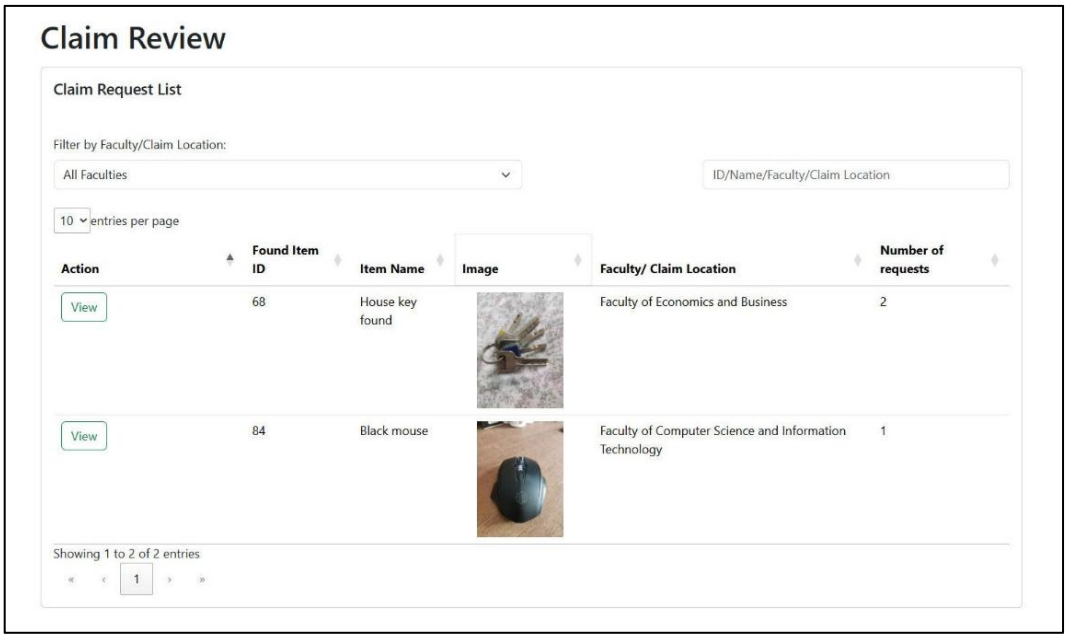



*Figure 4.77 Claim review interface with found item list*

Figure 4.77 shows a list of found items which are involved in the claim request waiting for the admin to approve or reject. Searching, filtering and sorting functionalities are provided for admins to narrow down the search results efficiently. After the student found an item, they will have to return the item back to any faculty (nearer to the location of item found is better). As a result, apply filtering by faculty or claim location is convenient for admin to quickly locate and handle the claim requests for specific found item in their faculties. To approve or reject the claim requests for a found item, click the “View” button that allows them to see more details about all claim requests related to one found item. The design (i.e.

column which shows number of requests for each item) also allows admin to quickly see which items have the most requests and then prioritize their review work accordingly.

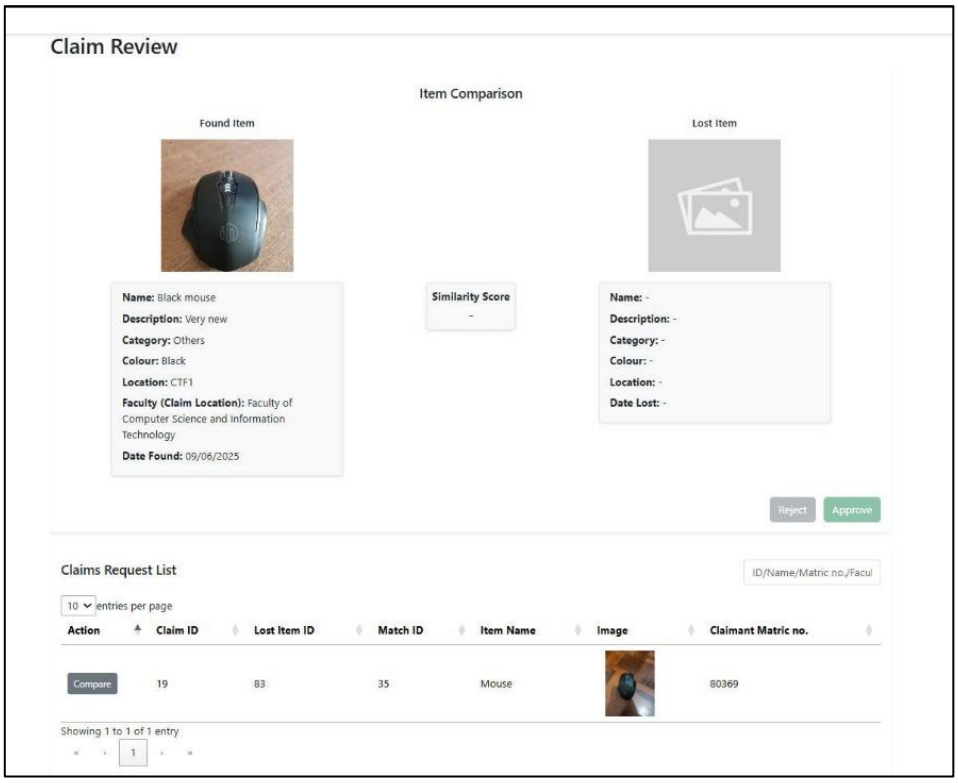



*Figure 4.78 Item comparison interface for matching lost and found items*

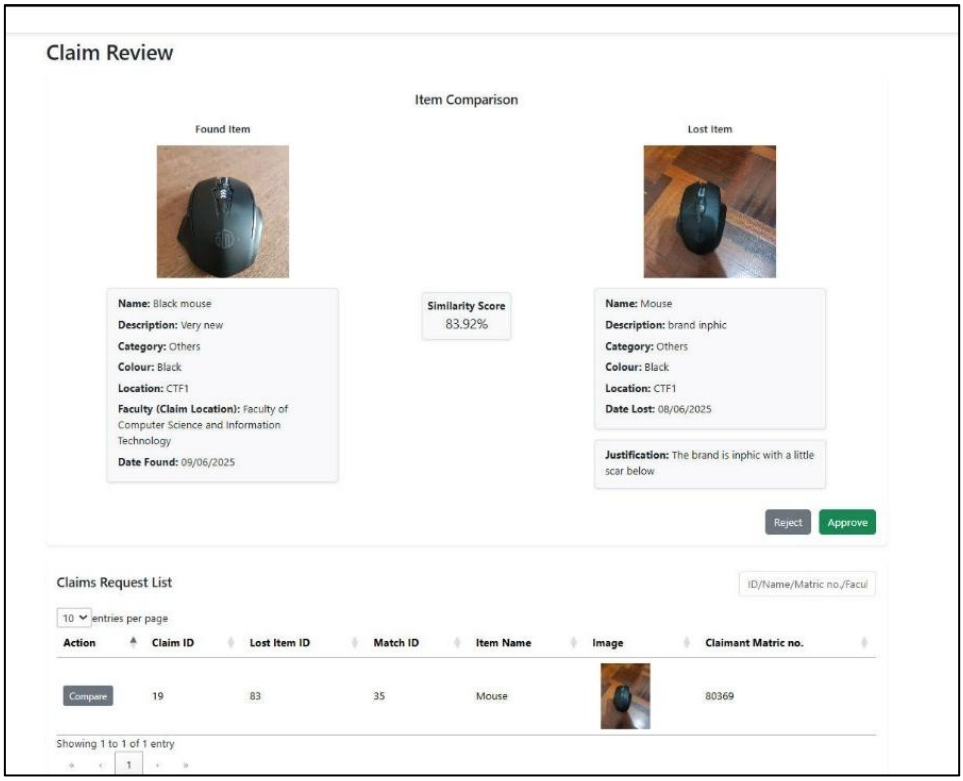



*Figure 4.79 Item comparison screen showing matched item with similarity score*

Clicking the “View” button in Figure 4.77 will navigate admin to Figure 4.78. Figure 4.78 demonstrates the initial stage of the item comparison interface when the matched lost items are not yet compared with the found item by the admin. The found item section on the left shows the detailed information of a found item, while the right side is a lost item section remains unpopulated with placeholder fields for corresponding item details until the specific lost item claim is selected for comparison. The similarity score section at the center is also empty currently, and will only be populated once the lost item is selected for comparison. The lost item section and similarity score section will remain empty if the claim request does not involve any lost item (student submit claim request without reporting item as lost) except justification section since it is compulsory for student to provide when submitting a claim request. Below this item comparison section, the claim request list displays all pending claims for a found item with the essential information provided. The “reject” and “approve” button is temporarily disabled until a lost item claim in the claim request list is selected for comparison.

Figure 4.79 illustrates the system has populated both sides of the item comparison section with available information when the “Compare” button for specific lost item claim in the claim request list below is clicked. The interface now includes actionable elements which are “reject” and “approve” buttons, enabling the admin to make informed decisions about claim validity based on the similarity score, lost item information and student’s justification.

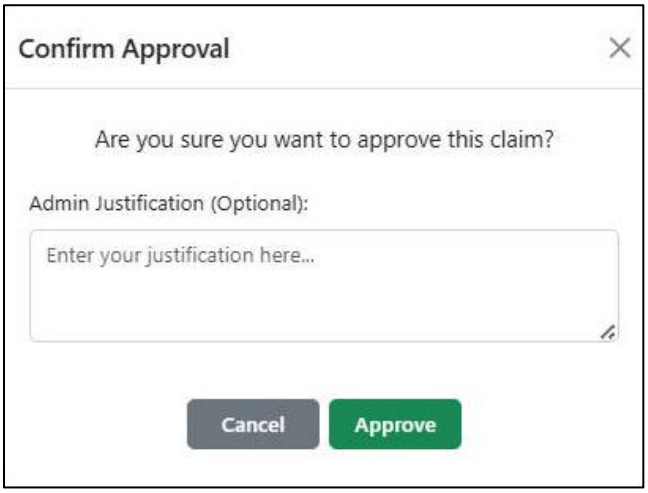



*Figure 4.80 Approve claim request*

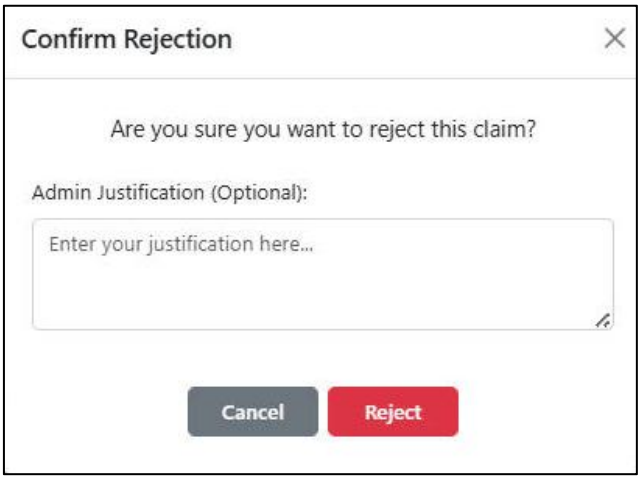



*Figure 4.81 Reject claim request*

Figure 4.80 and Figure 4.81 reveal a confirmation modal will arose when the button "Approve" or "Reject" in Figure 4.79 is clicked. The admin is free to leave down any justification when rejecting or approving the claim request for the claimant's reference on why their claim request is rejected/approved. If one claim request for a found item is approved, all other claim requests for the same found item will be rejected and a default justification will be created for the claimants, informing them their claim requests are fail because another claim request for the same found item is approved.

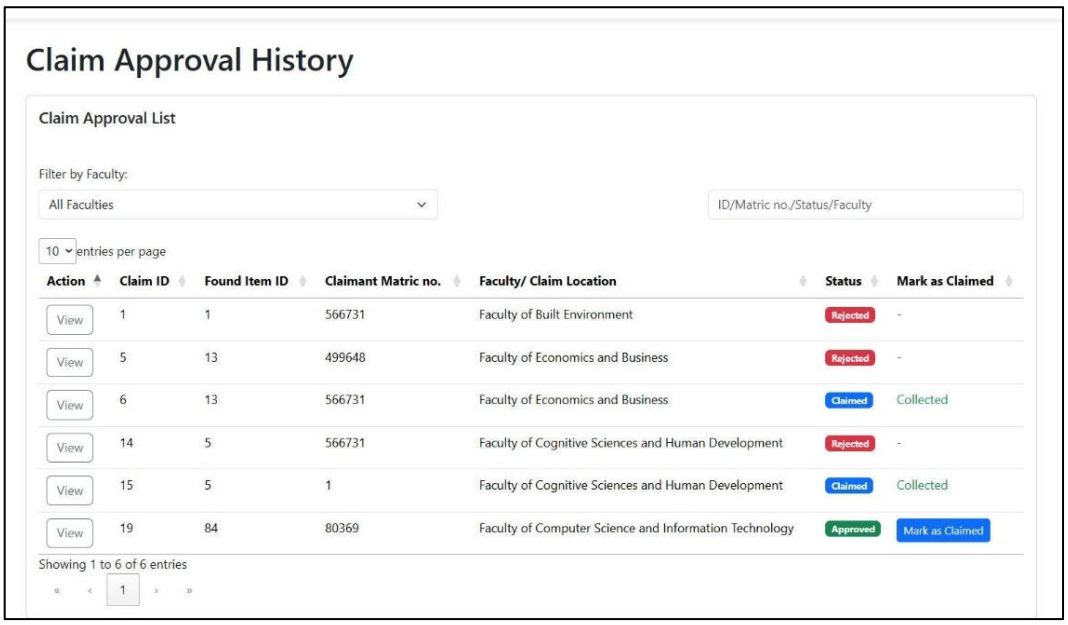



*Figure 4.82 Claim Approval History Interface with Status Tracking*

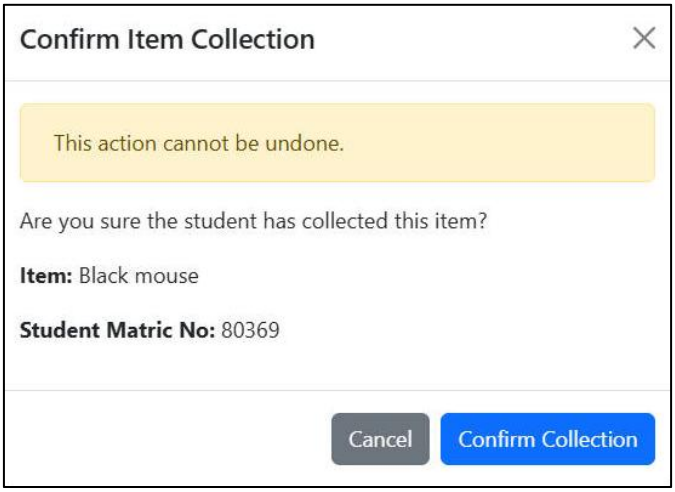



*Figure 4.83 Confirm item collection*

After the claim requests are processed in Figure 4.79, they will appear in the claim approval history here in Figure 4.82 for admin's reference. Searching, filtering and sorting functionalities are provided here to help admin locate the desired claim details faster. Admin have to click the button "Mark as Claimed" in Figure 4.82 after the owners claimed their items back.

Clicking the button will prompt a modal in Figure 4.83 asking for confirmation that the item was collected by the student, and also informing that this action cannot be undone.

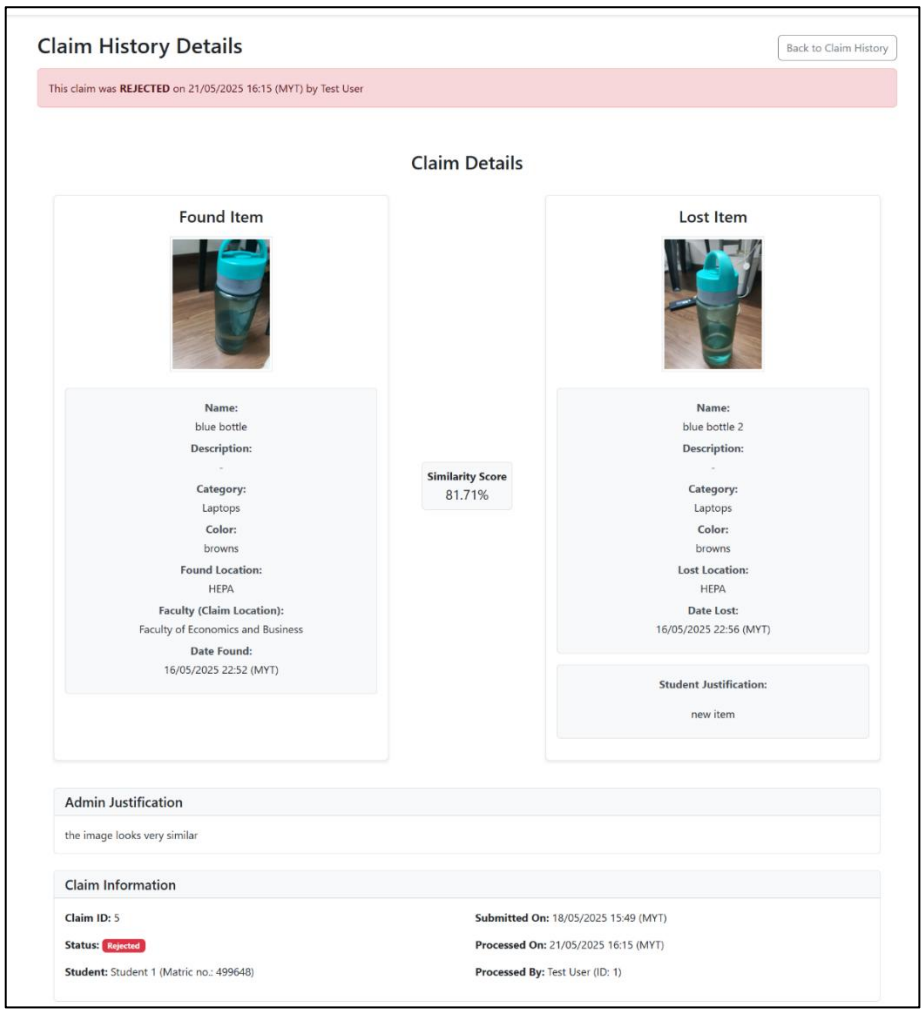



*Figure 4.84 Claim details*

*Figure 4.85 Claim approved message*

*Figure 4.86 Item collected message*

Figure 4.84 illustrates the comprehensive claim details for admin's reference or proof in future, including found item information, similarity score (if involves matched lost item), lost item information (if any), student's justification, claim status, admin justification, and finally claim information.

Figure 4.85 and Figure 4.86 depicts the message shown when the claim is in status approved or claimed instead of rejected as shown in Figure 4.84.

# Chapter 5: Testing

## 5.1 Introduction

Testing is a software development phase that checks if the system satisfies its requirements and behaves properly in different usage scenarios. In this chapter we report the results of the evaluation of the FindIt system obtained through functional testing, aimed at assessing the achievement of system functionality with respect to requirements and usability testing, which focused on assessing the user experience of the system.

Testing ensures that all the implemented functionality works as expected, identifies potential defects or weaknesses in the system, and hence ensure system's reliability, usability and overall effectiveness in supporting both student and administrative workflows.

## 5.2 Functional Testing

Functional testing is a type of software testing that validates the software system against the functional requirements and specifications. The purpose of functional testing is to test each function of the software application using appropriate input and checking the output against the functional requirements.

Functional testing was conducted over all major modules to ensure comprehensive coverage of system capabilities. The carefully designed test scenarios is used to test each module of the FindIt system. These test scenarios are designed in such a way that these cover all the possible scenarios in which the system is used by the users. These test scenarios cover the normal use of the system as well as the edge cases where the system is used. Functional testing covers the following modules of the system:- Authentication, Item management, Item search and discovery, Claim management, Student profile management,

Admin dashboard. The following sections present detailed test cases and results for each functional module.

### 5.2.1 Authentication Module

*Table 5.1 Student register*

| Test Case ID: AUTH-01<br>Module: Authentication<br>User Role: Student<br>Test Case Title: Verify a student can register with valid inputs.<br>Preconditions:<br>1. Student is on registration page. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Register with all valid inputs. | 1. Enter any username.<br>2. Enter a valid UNIMAS email with format 'matricNo@siswa.unimas.my'.<br>3. Enter a valid password that meets all the requirements: min 8 chars, 1 uppercase, 1 lowercase, 1 number, 1 special char.<br>4. Tap the "Sign Up" button. | • Registration is successful.<br>• A success message is displayed.<br>• A verification email is sent to user email address.<br>• User is redirected to the email verification screen.<br>• Checklist for the password requirements under password input field illustrates that all of the requirements are met. | As expected | Pass |
| 2. | Register with all input fields empty. | 1. Tap the "Sign Up" button without filling in any input fields. | • Registration failed.<br>• Validation error message will be displayed under each required input field.<br>• Checklist for | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | | the password requirements under password input field illustrates that none of the requirements are met. | | |
| 3. | Register with non-UNIMAS email. | 1. Enter any username.<br>2. Enter non-UNIMAS email (e.g. yourname@gmail.com).<br>3. Enter a valid password that meets all the requirements: min 8 chars, 1 uppercase, 1 lowercase, 1 number, 1 special char.<br>4. Tap the "Sign Up" button. | • Registration failed.<br>• Validation error message will be displayed under email input field.<br>• Checklist for the password requirements under password input field illustrates that all of the requirements are met. | As expected | Pass |
| 4. | Register with invalid UNIMAS email. | 1. Enter any username.<br>2. Enter invalid UNIMAS email (e.g. testing@siswa.unimas.my )<br>3. Enter a valid password that meets all the requirements: min 8 chars, 1 uppercase, 1 lowercase, 1 number, 1 special char.<br>4. Tap the "Sign Up" button. | • Registration failed.<br>• Validation error message will be displayed under email input field.<br>• Checklist for the password requirements under password input field illustrates that all of the requirements are met. | As expected | Pass |
| 5. | Register with invalid password. | 1. Enter any username.<br>2. Enter valid UNIMAS email (e.g. 1@siswa.unimas.my).<br>3. Enter an invalid password (i.e. does not meet all password requirements: min 8 chars, 1 uppercase, 1 lowercase, 1 number, 1 | • Registration failed.<br>• Validation error message will be displayed under password input field.<br>• Checklist for the password requirement under password input | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | special char). | field illustrates that some or all of the requirements are not met. | | |
| Post Condition: Upon successful registration, user account is created in the database with "unverified" status, verification email is sent to the registered email address, and user is redirected to email verification page. | | | | | |

*Table 5.2 Student login with verified email*

| Test Case ID: AUTH-02<br>Module: Authentication<br>User Role: Student<br>Test Case Title: Verify a student can login with valid credentials and verified email.<br>Preconditions:<br>1. Student had a registered account.<br>2. Student is on login page. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Login with verified email and correct password | 1. Enter verified email.<br>2. Enter correct password.<br>3. Tap "Sign In" button. | • Login successfully.<br>• User is redirected to home page. | As expected | Pass |
| 2. | Login with verified email and wrong password | 1. Enter verified email.<br>2. Enter wrong password.<br>3. Tap "Sign In" button. | • Login failed.<br>• Error message is displayed to indicate invalid credentials. | As expected | Pass |
| 3. | Login with unverified | 1. Enter unverifie | • Login failed.<br>• A prompt displayed | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | email and correct password | d email.<br>2. Repeat steps 2-3 in scenario 1. | asking user intention in resending verification email before navigated to email verification screen or directly navigated to the screen.<br>• Student is navigated to email verification screen. | | |
| 4. | Login with unverified email and wrong password | 1. Enter unverified email.<br>2. Repeat steps 2-3 in scenario 2. | • Login failed.<br>• Error message is displayed to indicate invalid credentials. | As expected | Pass |
| Post Condition: After login successfully, user is redirected to the homepage/dashboard. | | | | | |

*Table 5.3 Student verify email address*

| Test Case ID: AUTH-03<br>Module: Authentication<br>User Role: Student<br>Test Case Title: Verify a student can verify their email addresses using a valid link received in the verification email.<br>Preconditions:<br>1. Student has a registered account. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Student verify the email address using a valid link and unexpired link. | 1. Tap and open the link received in email. | • Email verified successfully.<br>• A success message is displayed. | As expected | Pass |
| 2. | Student verify the email address using an expired link. | 1. Tap and open the link received in email after 1 hour. | • Email verification failed.<br>• Error message is displayed to indicate the | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | | reasons of failed email verification. | | |
| 3. | Student verify the email address using an invalid link. | 1. Copy the link received from email.<br>2. Paste the link in browser.<br>3. Edit the link.<br>4. Access the link. | • Email verification failed.<br>• Error message is displayed to indicate the reasons of failed email verification. | As expected | Pass |
| Post Condition: After verifying the email successfully, student account status changed to "verified". | | | | | |

*Table 5.4 Student check verification status to login*

| Test Case ID: AUTH-04<br>Module: Authentication<br>User Role: Student<br>Test Case Title: Verify a student can login successfully after their email is checked as verified.<br>Preconditions:<br>1. Student has a registered account.<br>2. Student is on the email verification page. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Student check the verification status after verifying the email. | 1. Tap and open the link received in email.<br>2. Go back to the mobile application.<br>3. Tap "Check Verification Status" button. | • Login successfully.<br>• Student is redirected to the home page. | As expected | Pass |
| 2. | Student check the verification status before verifying the | 1. Tap "Check Verification Status" | • Login failed.<br>• Error message is displayed to | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | email. | button. | indicate the email is not verified yet. | | |
| Post Conditions: After verifying the email successfully and login, student can now access full system functionality. | | | | | |

*Table 5.5 Student request new verification email*

| Test Case ID: AUTH-05<br>Module: Authentication<br>User Role: Student<br>Test Case Title: Verify a student can request for a verification email again if they found no verification email is received.<br>Preconditions:<br>1. Student has a registered account.<br>2. Student has unverified email (show either an email verification page or a prompt for resending email). | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Student attempts to resend verification email on email verification page. | 1. Navigated to email verification page after register successfully .<br>2. Tap "Didn't receive the email? Resend" button. | • Verification email sent.<br>• A success message is displayed to indicate the email is successfully sent.<br>• The resend button is disabled for 60 seconds and a timer is shown. | As expected | Pass |
| 2. | Student attempts to resend verification email (via prompt) when trying to login using unverified email. | 1. Login with unverified email and correct password.<br>2. Tap "Resend | • Verification email sent.<br>• A snackbar is displayed showing that the email is sent | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | Verification Email" button in the prompt. | successfully.<br>• Redirected to email verification page. | | |
| Post Conditions: New verification email is sent to student's registered email successfully, cooldown timer is activated to prevent spam requests, and verification link will be expired after 1 hour from the time it is generated. | | | | | |

*Table 5.6 Student logout*

| Test Case ID: AUTH-06<br>Module: Authentication<br>User Role: Student<br>Test Case Title: Verify a logged-in student can logout.<br>Preconditions:<br>1. Student has a registered account. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Student logout before the email is verified. | 1. Tap the "logout" button on email verification page. | • Student logout successfully.<br>• Student redirected to login page. | As expected | Pass |
| 2. | Student logout after the email is verified. | 1. Tap "More" on bottom navigation bar.<br>2. Tap "Account"<br>3. Tap "Logout" button.<br>4. Confirm to logout from the modal displayed. | • Student logout successfully.<br>• Student redirected to login page. | As expected | Pass |
| Post Conditions: After login successfully, student's session is terminated, authentication token is removed, and redirected to login page. | | | | | |

*Table 5.7 Student receive password reset email*

| Test Case ID: AUTH-07<br>Module: Authentication<br>User Role: Student<br>Test Case Title: Verify a student can receive password reset email by providing a registered or verified email address.<br>Preconditions:<br>1. Student has a registered account.<br>2. Student is on login page. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Student provides registered email address to receive the reset link. | 1. Tap "Forgot Password" button.<br>2. Enter registered email.<br>3. Tap "Sent Reset Link" button. | • Reset password link is sent to student's email.<br>• Redirected to a screen showing a success message that the reset link is sent. | As expected | Pass |
| 2. | Student provides unregistered email address to receive the reset link. | 1. Tap "Forgot Password" button.<br>2. Enter unregistered email.<br>3. Tap "Sent Reset Link" button. | • Reset password link is not sent successfully.<br>• Error message displayed indicating that there is no such student with the email address. | As expected | Pass |
| Post Conditions: Student received a password reset email by providing a registered email address, password reset token is generated and stored in database, password reset token is generated with 1 hour expiration time. | | | | | |

*Table 5.8 Student and Admin reset password before login successfully*

Test Case ID: AUTH-08

Module: Authentication

User Role: Student & Admin

Test Case Title: Verify a student/admin can reset password via the valid link received in password reset email via 'Forgot password' flow.

Preconditions:

1. Student/admin has a registered account.
2. Student/admin receive password reset email.

| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
|---|---|---|---|---|---|
| 1. | Student/admin provides valid new password and matches the confirmed password. | 1. Click the "Reset Password" button in email.<br>2. Enter valid new password.<br>3. Enter confirmed password which matches the new password. | • Redirected to a reset password webpage.<br>• Reset password successfully. | As expected | Pass |
| 2. | Student/admin provides invalid new password. | 1. Click the "Reset Password" button in email.<br>2. Enter invalid new password.<br>3. Enter confirmed password which matches the new password. | • Redirected to a reset password webpage.<br>• Error message displayed indicating the new password does not fulfil the password requirement.<br>• Reset password failed. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| 3. | Student/admin provides valid new password but does not match the confirmed password. | 1. Click the "Reset Password" button in email.<br>2. Enter valid new password.<br>3. Enter confirmed password which does not matches the new password. | • Redirected to a reset password webpage.<br>• Error message displayed indicating the new password does not match the confirmed password.<br>• Reset password failed. | As expected | Pass |
| 4. | Student/admin use the same password reset link to reset password after it had already been used successfully. | 1. Click the "Reset Password" button in the same email which had been used to reset password successfully. | • Error message displayed indicating the link is invalid. | | |
| 5. | Student/admin use expired password reset link to reset password. | 1. Click the "Reset Password" button in the email after 60 minutes the email is sent. | • Error message displayed indicating the link is invalid. | As expected | Pass |
| Post Conditions: By using a valid link, student and admin can reset and login with the new password, the password reset token will be deleted after the link is used to reset the password successfully. | | | | | |

*Table 5.9 Logged-in student reset password*

| |
|---|
| Test Case ID: AUTH-09<br>Module: Authentication<br>User Role: Student<br>Test Case Title: Verify a logged-in student can reset password. |

| Preconditions:<br>1. Student has a registered account.<br>2. Student is logged in.<br>3. Student has navigated to Change Password screen (via: More (in bottom nav bar) → Account → Change Password). | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Student provides correct current password. | 1. Enter correct current password in password reset screen.<br>2. Enter new password that follows all the password requirement displayed below the password input field.<br>3. Enter a confirmed password matched the new password.<br>4. Tap "Change Password" button. | • Reset password successfully.<br>• A success message is shown to indicate password reset successfully.<br>• Navigated to My Account screen. | As expected | Pass |
| 2. | Student provides wrong current password | 1. Enter wrong current password in password reset screen.<br>2. Enter new password that follows all the password requirement displayed below the password | • Failed to reset password.<br>• Error message is displayed to show that the current password is incorrect. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | input field.<br>3. Enter a confirmed password matched the new password.<br>4. Tap “Change Password” button. | | | |
| 3. | Student provides invalid new password. | 1. Enter correct current password in password reset screen.<br>2. Enter invalid password that follows all the password requirement displayed below the password input field.<br>3. Enter a confirmed password matched the new password.<br>4. Tap “Change Password” button. | • Failed to reset password.<br>• Error message is displayed to show that new password is invalid. | As expected | Pass |
| 4. | Student’s confirmed password does not match the new password. | 1. Enter correct current password in password reset screen.<br>2. Enter valid password.<br>3. Enter a confirmed password not | • Failed to reset password.<br>• Error message is displayed to show that the confirmed password not matched with new password. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | matched the new password.<br>4. Tap “Change Password” button. | | | |
| 5. | Student leave required input field empty. | 1. Leave any one or all input fields empty.<br>2. Tap “Change Password” button. | • Failed to reset password.<br>• Error message is displayed to show that the input field is empty. | As expected | Pass |
| Post Conditions: Old password is replaced with new password after the password changed successfully. | | | | | |

*Table 5.10 Admin login*

| Test Case ID: AUTH-10<br>Module: Authentication<br>User Role: Admin<br>Test Case Title: Verify an admin can login with valid credentials.<br>Preconditions:<br>1. Admin has a registered account.<br>2. Admin is on login page. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Admin login with registered email and correct password. | 1. Enters registered email.<br>2. Enters correct password.<br>3. Click “Login” button. | • Login successfully.<br>• Redirected to dashboard. | As expected | Pass |
| 2. | Admin login with registered email and wrong password. | 1. Enters registered email.<br>2. Enters wrong password.<br>3. Click | • Login failed.<br>• Error message displayed showing that the credentials is invalid. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | "Login" button. | | | |
| 3. | Admin login with unregistered email and correct password. | 1. Enters unregistered email.<br>2. Enters correct password.<br>3. Click "Login" button. | • Login failed.<br>• Error message displayed showing that the credentials is invalid. | As expected | Pass |
| 4. | Admin login with unregistered email and wrong password. | 1. Enters unregistered email.<br>2. Enters wrong password.<br>3. Click "Login" button. | • Login failed.<br>• Error message displayed showing that the credentials is invalid. | As expected | Pass |
| 5. | Admin login with required fields empty. | 1. Click "Login" button. | • Login failed.<br>• Error message displayed showing the required field is empty. | As expected | Pass |
| Post Conditions: After admin login successfully, the session is established, and they are redirected to dashboard. | | | | | |

*Table 5.11 Admin receive password reset email*

| | | | | | |
|---|---|---|---|---|---|
| Test Case ID: AUTH-11<br>Module: Authentication<br>User Role: Admin<br>Test Case Title: Verify an admin can receive password reset email by providing a registered email address/ account.<br>Preconditions:<br>1. Admin has a registered account.<br>2. Admin is on login page. | | | | | |
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Admin attempts | 1. Click | • Password reset link is | As | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | to reset password for a registered account. | "Forgot Password" .<br>2. Click "Send Password Resent Link" button. | sent to admin's email successfully.<br>• A success message is displayed indicating that the password reset link is sent to admin's email. | expected | |
| 2. | Admin attempts to reset password for an unregistered account. | 1. Click "Forgot Password" .<br>2. Click "Send Password Resent Link" button. | • Password reset link could not be delivered to admin's email.<br>• Error message displayed indicating that the email is no such admin with the email address found in the system. | As expected | Pass |
| Post Condition: Admin received a password reset email by providing a registered email address, password reset token is generated and stored in database, password reset token is generated with 1 hour expiration time. | | | | | |

### 5.2.2 Item Management Module

*Table 5.12 Student report new lost / found item*

| Test Case ID: ITEM-01<br>Module: Item Management<br>User Role: Student<br>Test Case Title: Verify a student can report new item as lost or found when all required fields/ information is filled.<br>Preconditions:<br>1. Student has a verified account.<br>2. Student is logged-in. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Student report item with all required fields filled. | 1. Tap "+" button at the center of the bottom navigation bar. | • Item reported successfully.<br>• A success message is displayed to show | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | 2. Choose an item type.<br>3. Upload an item image by taking a photo using camera or choosing an image from gallery.<br>4. Enter item name.<br>5. Enter item description.<br>6. Choose an item category.<br>7. Choose an item colour.<br>8. Choose a location where the item is lost/found.<br>9. Choose a location where the owner can claim their belongings (for item type "found" only).<br>10. Click "Submit" button. | the item is created and reported successfully.<br>• Student is redirected to the previous screen where the "+" button is tapped.<br>• System checks for potential matches and notifies the owner (who reported their item lost) if any are found. Future matches will notify the owner.<br>• Student will receive notifications in the app (via Notification in bottom navigation bar). | | |
| 2. | Student report item with at least one required field left empty. | 1. Tap "+" button at the center of the bottom navigation bar.<br>2. Leave anyone or all of the required fields (i.e. item type, item name, category, | • Item could not be reported/created.<br>• An error message displayed indicating that the required field is not filled. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | colour, location and claim location (for item type "found" only)) empty.<br>3. Click "Submit" button. | | | |
| 3. | Student uploads an image larger than allowed size when reporting an item. | 1. Tap "+" button at the center of the bottom navigation bar.<br>2. Choose an item type.<br>3. Upload an item image larger than the allowed size (i.e. larger than 2MB).<br>4. Enter item name.<br>5. Enter item description.<br>6. Choose an item category.<br>7. Choose an item colour.<br>8. Choose a location where the item is lost/found.<br>9. Choose a location where the owner can claim their belongings (for item type "found" only).<br>10. Click "Submit" | • Item reported successfully.<br>• A success message is displayed to show the item is created and reported successfully.<br>• Student is redirected to the previous screen where the "+" button is tapped.<br>• System checks for potential matches and notifies the owner (who reported their item lost) if any are found. Future matches will notify the owner.<br>• Student will receive notifications in the app (via Notification in bottom navigation bar). | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | button. | | | |
| Post Conditions: Upon successful item creation, item is in “active” status, item matching process is triggered if image is provided, item appears in “My Item” list, notification is sent informing the match result, item becomes searchable by other students. | | | | | |

*Table 5.13 Student view own reported lost / found items details*

| Test Case ID: ITEM-02<br>Module: Item Management<br>User Role: Student<br>Test Case Title: Verify a student can view the details of their own reported lost/found items.<br>Preconditions:<br>1. Student has a verified account.<br>2. Student is logged-in. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Student view the item details he/she has reported before. | 1. Tap “My Items” at the bottom navigation bar.<br>2. Tap the ‘Lost’/’Found’ tab at the top of the screen to see the list of lost/found items.<br>3. Tap to view specific item details. | • All items reported by the current student are displayed.<br>• Item details is shown after choosing to view specific item. | As expected | Pass |
| 2. | Student attempts to view item details before reporting any item. | 1. Tap “My Items” at the bottom navigation bar.<br>2. Tap the ‘Lost’/’Found’ tab at the top of the screen to see the list of lost/found items. | • An info message appears showing there is no item reported yet. | As expected | Pass |
| Post Conditions: Item information is displayed to students if any. | | | | | |

*Table 5.14 Student edit own reported lost / found item details*

| Test Case ID: ITEM-03<br>Module: Item Management<br>User Role: Student<br>Test Case Title: Verify a student can edit the details of their own reported lost/found item/s.<br>Preconditions:<br>1. Student has a verified account.<br>2. Student is logged-in.<br>3. Student had reported an item. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Student edit item details when the item is not involved in any claim request yet. | 1. Tap "My Items" at the bottom navigation bar.<br>2. Tap the 'Lost'/'Found' tab at the top of the screen to see the list of lost/found items.<br>3. Tap on the specific items.<br>4. Tap "Edit Item" button.<br>5. Tap the input fields to edit the item details.<br>6. Tap "Save Changes" button. | • Student is redirected to the edit item page.<br>• The original item image is displayed if any image is uploaded for this item before.<br>• The original item information is displayed/chosen by default in the input fields.<br>• Item information is updated successfully. | As expected | Pass |
| 2 | Student edit item details when the item is not involved in any claim request yet, | 1. Tap "My Items" at the bottom navigation bar. | • An error message is displayed indicating that the required field is empty. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | but with required field empty. | 2. Tap the 'Lost'/'Found' tab at the top of the screen to see the list of lost/found items.<br>3. Tap on the specific items.<br>4. Tap "Edit Item" button.<br>5. Tap the input fields to edit the item details.<br>6. Left the required field, i.e. item name empty.<br>7. Tap "Save Changes" button. | • Item information could not be updated successfully. | | |
| 3. | Student attempts to edit item details when the item is already involved in a claim request. | 1. Tap "My Items" at the bottom navigation bar.<br>2. Tap the 'Lost'/'Found' tab at the top of the screen to see the list of lost/found items.<br>3. Tap on the specific items.<br>4. Tap "Edit Item" button. | • An error message is displayed indicating that the item details cannot be edited because it had involved in claim request. | As expected | Pass |
| Post Conditions: After student edit successfully, item information is updated in screen and database, item matching process is triggered if either item category, colour, location or image is modified, or image is added. | | | | | |

*Table 5.15 Student delete own reported lost / found item details*

| Test Case ID: ITEM-04<br>Module: Item Management<br>User Role: Student<br>Test Case Title: Verify student can delete the details of their own reported lost/found item/s when the item is not involved in any claim request.<br>Preconditions:<br>1. Student has a verified account.<br>2. Student is logged-in.<br>3. Student had reported an item. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Student delete item details when the item is not involved in any claim request yet. | 1. Tap "My Items" at the bottom navigation bar.<br>2. Tap the 'Lost'/'Found' tab at the top of the screen to see the list of lost/found items.<br>3. Tap on the specific items.<br>4. Tap "Delete Item" button.<br>5. Tap "Delete" button in the prompt. | • A prompt is displayed to ask for user's confirmation in deleting the item.<br>• The item is deleted from the database successfully. | As expected | Pass |
| 2. | Student attempts to delete item details when the item is already involved in a claim request. | 1. Tap "My Items" at the bottom navigation bar.<br>2. Tap the 'Lost'/'Found' tab at the top of the screen to see the list of | • An error message is displayed indicating that the item details cannot be deleted because it had involved in claim request. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | lost/found items.<br>3. Tap on the specific items.<br>4. Tap "Delete Item" button. | | | |
| Post Conditions: After the item is deleted successfully, item record is removed from database, the associated matches are removed, item images are removed from the server if any. | | | | | |

*Table 5.16 Admin create new item characteristics instance*

| Test Case ID: ITEM-05<br>Module: Item Management<br>User Role: Admin<br>Test Case Title: Verify admin can create new item characteristics (category, colour, location) instance with a unique name.<br>Preconditions:<br>1. Admin has a registered account.<br>2. Admin is logged-in.<br>3. There is an item category in the system with name 'Electronics'. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Admin create new item characteristics with a unique name. | 1. Click the hamburger icon in the top navigation bar to open sidebar menu.<br>2. Select "Item Characteristics" from the sidebar menu to expand its dropdown options.<br>3. Select "Category" from the dropdown options. | • A new item characteristic (Category/Colour/Location) is successfully created.<br>• A success message is displayed to notify the successful creation of a new item characteristic. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | 4. Click “Add Category” button at the top right of the list page.<br>5. Enter a unique name for item category not currently found in the system.<br>6. Click “Add” button to submit the form.<br>(Note: The same test steps apply to item characteristics 'Colour' and 'Location' as they follow the same validation rules as 'Category'.) | | | |
| 2. | Admin create new item characteristics with a duplicate name. | 1. Click the hamburger icon in the top navigation bar to open sidebar menu.<br>2. Select “Item Characteristics” from the sidebar menu to expand its dropdown options.<br>3. Select “Category” from the dropdown options. | • An error message is displayed indicating that the item characteristics already exists in the system. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | 4. Click “Add Category” button at the top right of the list page.<br>5. Enter a duplicated name (e.g. ‘Electronics’) for item category.<br>6. Click “Add” button to submit the form.<br>(Note: The same test steps apply to item characteristics 'Colour' and 'Location' as they follow the same validation rules as 'Category'.) | | | |
| 3. | Admin create new item characteristics with a duplicate name, and variations (i.e. case or spacing) | 1. Click the hamburger icon in the top navigation bar to open sidebar menu.<br>2. Select “Item Characteristics” from the sidebar menu to expand its dropdown options.<br>3. Select “Category” from the dropdown options. | • An error message is displayed indicating that the item characteristics already exists in the system. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | 4. Click “Add Category” button at the top right of the list page.<br>5. Enter a duplicated name with variations in case or spacing (e.g. ‘electronics’, ‘ELECTRONICS’, ‘ electronics’, ‘electronics ’, ‘ electronics ’) for item category.<br>6. Click “Add” button to submit the form.<br>(Note: The same test steps apply to item characteristics 'Colour' and 'Location' as they follow the same validation rules as 'Category'.) | | | |
| Post Conditions: Upon successful creation of item characteristics, it can be found in database, it is available for item reporting. | | | | | |

*Table 5.17 Admin view item characteristics*

| |
|---|
| Test Case ID: ITEM-06<br>Module: Item Management<br>User Role: Admin<br>Test Case Title: Verify an admin can view item characteristics list (category, colour, location).<br>Preconditions: |

| | | | | | |
|---|---|---|---|---|---|
| 1. Admin has a registered account.<br>2. Admin is logged-in. | | | | | |
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Admin access the item characteristics list when entries already exist. | 1. Click the hamburger icon in the top navigation bar to open sidebar menu.<br>2. Select “Item Characteristics” from the sidebar menu to expand its dropdown options.<br>3. Select “Category” from the dropdown options.<br>4. Enter category ID or category name to search for and display the specific category (optional).<br>(Note: The same test steps apply to item characteristics 'Colour' and 'Location'.) | • All the item categories are displayed.<br>• Specific category is displayed (if use search functionality). | As expected | Pass |
| 2. | Admin access the item characteristics list when no entries have been created. | 1. Click the hamburger icon in the top navigation bar to open sidebar menu.<br>2. Select “Item | • An info message is displayed indicating that there is no item category created and | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | Characteristics" from the sidebar menu to expand its dropdown options.<br>3. Select "Category" from the dropdown options.<br>(Note: The same test steps apply to item characteristics 'Colour' and 'Location'.) | found in the system. | | |
| Post Conditions: All item characteristics are shown. | | | | | |

*Table 5.18 Admin edit item characteristics*

| Test Case ID: ITEM-07<br>Module: Item Management<br>User Role: Admin<br>Test Case Title: Verify admin can edit item characteristics (category/colour/location) when they are not referenced by any item.<br>Preconditions:<br>1. Admin has a registered account.<br>2. Admin is logged-in.<br>3. Admin had created an item category named 'Electronics'.<br>4. [Navigation] Admin is on the [Category/Colour/Location] List page (via hamburger menu > Item Characteristics dropdown> [Category/Colour/Location]). | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Admin edit an item characteristic which is not referenced by any item with a unique name. | 1. Select "Category" from the dropdown options.<br>2. Click "Edit button" for the specific category.<br>3. Enters a unique name for item category (e.g. 'Book'). | • A success message is displayed indicating that the item characteristic's name is edited successfully. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | 4. Click “Update” button.<br>(Note: The same test steps apply to item characteristics 'Colour' and 'Location'.) | | | |
| 2. | Admin edit an item characteristic which is already referenced by the item. | 1. Select “Category” from the dropdown options.<br>2. Click “Edit button” for the specific category.<br>3. Enters any category name.<br>4. Click “Update” button.<br>(Note: The same test steps apply to item characteristics 'Colour' and 'Location'.) | • An error message is displayed indicating that the item characteristic’s name could not be edited because is referenced by specific amounts of items. | As expected | Pass |
| 3. | Admin edit an item characteristic which is not referenced by any item with a duplicate name. | 1. Select “Category” from the dropdown options.<br>2. Click “Edit button” for the category which is not referenced by any item.<br>3. Enters a duplicate name for item category (e.g. ‘Electronics’).<br>4. Click “Update” button.<br>(Note: The same test steps apply to item characteristics 'Colour' and 'Location'.) | • An error message is displayed indicating that the item characteristic’s name could not be edited because the item characteristic already exist. | As expected | Pass |
| 4. | Admin edit an item characteristic which is not referenced by | 1. Select “Category” from the dropdown options.<br>2. Click “Edit button” for the specific | • An error message is displayed indicating that the item | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | any item with a duplicate name and variations (i.e. case or spacing) | category.<br>3. Enters a duplicate name for item category with variations in case or spacing (e.g. ‘electronics’, ‘ELECTRONICS’, ‘ Electronics’, ‘Electronics ’, ‘ Electronics ’).<br>4. Click “Update” button.<br>(Note: The same test steps apply to item characteristics 'Colour' and 'Location'.) | characteristic’s name could not be edited because the item characteristic already exist. | | |
| Post Conditions: After admin edit item characteristics successfully, it is reflected in the screen and database. | | | | | |

*Table 5.19 Admin delete item characterstics*

| Test Case ID: ITEM-08<br>Module: Item Management<br>User Role: Admin<br>Test Case Title: Verify an admin can delete item characteristics when they are not referenced by any item.<br>Preconditions:<br>1. Admin has a verified account.<br>2. Admin is logged-in.<br>3. Admin had created an item category named ‘Electronics’.<br>4. [Navigation] Admin is on the [Category/Colour/Location] List page (via hamburger menu > Item Characteristics dropdown> [Category/Colour/Location]). | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Admin delete item characteristic which is not referenced by any item. | 1. Select “Category” from the dropdown options.<br>2. Click “Delete button” for | • A prompt is displayed to ask for admin’s confirmation in deleting the item characteristic.<br>• The item characteristic is | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | the specific category.<br>(Note: The same test steps apply to item characteristics 'Colour' and 'Location'.) | deleted from the database successfully. | | |
| 2. | Admin delete item characteristic which is referenced by the item. | 1. Select "Category" from the dropdown options.<br>2. Click "Delete button" for the specific category.<br>(Note: The same test steps apply to item characteristics 'Colour' and 'Location'.) | • A prompt is displayed to ask for admin's confirmation in deleting the item characteristic.<br>• An error message is displayed indicating that the item characteristic cannot be deleted because it is referenced by specific amounts of item. | As expected | Pass |
| Post Conditions: After item characteristic is deleted successfully, it is removed from the screen and database. | | | | | |

### 5.2.3 Item Search and Discovery Module

*Table 5.20 Student search items*

| |
|---|
| Test Case ID: ITEMSD-01<br>Module: Item Search and Discovery<br>User Role: Student<br>Test Case Title: Verify student can search for item using keywords, filters and item types.<br>Preconditions:<br>1. Student has a verified account.<br>2. Student is logged-in.<br>3. Student is on homepage.<br>4. Default selected item type is "Found".<br>5. Items exist in all 3 types:<br>• Found: "Black Wallet", "Earbuds" |

- Lost: “Gold Watch”, “Gaming Laptop”
- Recovered: “Blue Bottle”

6. Filters available: Category (Wallet, Electronics, Bottle), Location (HEPA, CTF1), Colour (Black, Blue, Others).

| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
|---|---|---|---|---|---|
| 1. | Search item with exact matching keywords. | 1. Stay on “Found” tab.<br>2. Enter “Earbuds” in search bar. | • Shows only item with named “Black Wallet” | As expected | Pass |
| 2. | Search item with partial keyword. | 1. Switch to “Lost” tab.<br>2. Enter “Gaming” in search bar. | • Shows only item with named “Gaming Laptop” | As expected | Pass |
| 3. | Search item with all capitalized keywords. | 1. Switch to “Recovered” tab.<br>2. Enter “BLUE BOTTLE” in search bar. | • Shows only item with named “Blue Bottle” | As expected | Pass |
| 4. | Search item with filter + item type only | 1. Stay on “Found” tab.<br>2. Tap on filter icon.<br>3. Select category = “Wallet”<br>4. Select colour = “Black”<br>5. Select location = “CTF1” | • Shows only item with named “Black Wallet” | As expected | Pass |
| 5. | Search item with keywords + filter + item type | 1. Stay on “Found” tab.<br>2. Tap on filter icon.<br>3. Select | • Shows only item with named “Black Wallet”. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | category = "Wallet"<br>4. Select colour = "Black"<br>5. Select location = "CTF1"<br>6. Enter "Black Wallet" in search bar. | | | |
| 6. | Search item with non-matching keyword. | 1. Stay on "Found" tab.<br>2. Enter "Leather Wallet" in search bar. | • Shows an info message indicating that no result is found. | As expected | Pass |
| 7. | Search item with extra spaces (trailing or starting spaces or both). | 1. Stay on "Found" tab.<br>2. Enter " Earbuds " in search bar. | • Shows an item with name "Earbuds". | As expected | Pass |
| Post Conditions: Item matched with the search query, filter, and item type is displayed to the student. | | | | | |

*Table 5.21 Student clear filter / search keywords*

| | | | | | |
|---|---|---|---|---|---|
| Test Case ID: ITEMSD-02<br>Module: Item Search and Discovery<br>User Role: Student<br>Test Case Title: Verify a student can clear filters / search keywords to reset search results.<br>Preconditions:<br>1. Student has a verified account.<br>2. Student is logged-in.<br>3. Student is on homepage.<br>4. Student has applied filters or entered keywords in search bar. | | | | | |
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Clear the specific filters. | 1. Locate the applied filter chip below the search bar | • The filter is removed from the active filters list. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | 2. Click the "x" icon on the specific filter chip. | • Search results update to reflect the removed filter. | | |
| 2. | Clear all filters. | 1. Click the filter icon to open the filters panel.<br>2. Click the "Clear" button at the top of the filters panel. | • All applied filters (Category, Color, Location) are reset.<br>• Search results revert to unfiltered state. | As expected | Pass |
| 3. | Clear the search keywords. | 1. Click the “x” icon in the search bar. | • The search keyword is removed from the search bar.<br>• Search results revert to default state. | As expected | Pass |
| Post Conditions: All applied filters are removed, search results returned to default view. | | | | | |

*Table 5.22 Student view item details*

| | | | | | |
|---|---|---|---|---|---|
| Test Case ID: ITEMSD-03<br>Module: Item Search and Discovery<br>User Role: Student<br>Test Case Title: Verify student can view item details for the item listed in search results.<br>Preconditions:<br>1. Student has a verified account.<br>2. Student is logged-in.<br>3. Student is on homepage.<br>4. Search results are displayed after performing a search or applying filters. | | | | | |
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | View details of a found item in search result. | 1. Locate an item of interest in the search result. | • The item in search result is visible with item image (if any), item name, and item type.<br>• Navigated to item details page. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | 2. Click the item image or title.<br>3. View details of the found item. | • All relevant item details are displayed (e.g. item image, descriptions (if any), category, colour, type, status, found location, claim location, reporter email, date reported).<br>• Item image can be enlarged after tapped.<br>• A claim button is displayed. | | |
| 2. | View details of a lost item in search result. | 1. Locate an item of interest in the search result.<br>2. Click the item image or title.<br>3. View details of the found item. | • The item in search result is visible with item image (if any), item name, and item type.<br>• Navigated to item details page.<br>• All relevant item details are displayed (e.g. item image, descriptions (if any), category, colour, type, status, lost location, reporter email, date reported).<br>• Item image can be enlarged after tapped.<br>• No claim button is displayed. | As expected | Pass |
| 3. | View details of a recovered item in search result which has matching lost item. | 1. Locate an item of interest in the search result.<br>2. Click the item image or title. | • The item in search result is visible with found item image (if any), item name, and item type.<br>• Navigated to item details page.<br>• All relevant found item details are displayed (e.g. item image, name, similarity score between | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | 3. View details of the recovered item. | the found item and matched lost item, descriptions (if any), category, colour, type, status, found location, claim location, reporter email, and date reported).<br>• All relevant lost item details are displayed (e.g. item image, name, descriptions(if any), category, colour, lost location, reporter email, and date reported.)<br>• Item image can be enlarged after tapped.<br>• No claim button is displayed. | | |
| 4. | View details of a recovered item in search result which has none matching lost item | 1. Locate an item of interest in the search result.<br>2. Click the item image or title.<br>3. View details of the recovered item. | • The item in search result is visible with found item image (if any), item name, and item type.<br>• Navigated to item details page.<br>• All relevant found item details are displayed (e.g. item image, name, descriptions (if any), category, colour, type, status, found location, claim location, reporter email, and date reported).<br>• No matched lost item information is displayed.<br>• Item image can be enlarged after tapped.<br>• No claim button is displayed. | As expected | Pass |
| Post Conditions: Comprehensive item information is displayed. | | | | | |

### 5.2.4 Claim Management Module

*Table 5.23 Student submits claim request for a found item without reporting item as lost*

| Test Case ID: CM-01<br>Module: Claim Management<br>User Role: Student<br>Test Case Title: Verify a student can submit a claim request for a found item (as the potential owner) without reporting a lost item.<br>Preconditions:<br>1. Student has a verified account.<br>2. Student is logged-in.<br>3. Student is on homepage. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Submit a first-time claim request for the found item. | 1. Locate the desired found item.<br>2. Tap the item image/name.<br>3. Tap "Claim Item" button.<br>4. Enter justification in the dialog.<br>5. Tap "Claim" button. | • "Claim Item" button is available.<br>• A dialog is displayed asking for justification.<br>• A success message is displayed indicating that the claim request is created successfully.<br>• A claim request is submitted successfully. | As expected | Pass |
| 2. | Attempts to submit another claim request for the same found item. | 1. Locate the desired found item.<br>2. Tap the item image/name. | • "Claim Item" button is unavailable.<br>• The claim button is greyed out and show the text "Already Claim" instead of "Claim | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | | Item”.<br>• Failed to submit claim request. | | |
| Post Conditions: After claim request is submitted successfully, claim record is created with “pending” status, | | | | | |

*Table 5.24 Student submits claim request for a matched found item related to own reported lost item*

Test Case ID: CM-02

Module: Claim Management

User Role: Student

Test Case Title: Verify that a student can submit a claim for a found item (as the potential owner) after reporting a lost item and receiving potential matches of the found items from the system.

Preconditions:

1. Student has a verified account.
2. Student is logged-in.
3. Student has reported a lost item.
4. System has generated potential matches between the lost item and found item.
5. [Navigation] Student is on the potential matches page (via My Items (in bottom navigation bar) > ‘Lost’ tab > Choose item > ‘Potential Matches’ tab).

| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
|---|---|---|---|---|---|
| 1. | Submits a claim request for a matched found item marked as 'available'. | 1. Tap the desired matched found item that is marked as ‘available’.<br>2. Tap the “Claim Item” button.<br>3. Enter justification in the dialog.<br>4. Tap “Claim” button. | • “Claim Item” button is available.<br>• A dialog is displayed asking for justification.<br>• A success message is displayed indicating that the claim request is created successfully.<br>• Claim request is submitted successfully.<br>• The matched found item is now marked as ‘pending’. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | | • Other matched found items with status 'available' is now marked as 'dismissed'. | | |
| 2. | Attempt to submit a claim request for a matched found item marked as 'approved'. | 1. Tap the desired matched found item that is marked as 'approved'. | • "Claim Item" button is unavailable.<br>• The claim button is greyed out and show the text "Already Claim" instead of "Claim Item".<br>• Failed to submit claim request. | As expected | Pass |
| 3. | Attempt to submit a claim request for a matched found item marked as 'rejected'. | 1. Tap the desired matched found item that is marked as 'rejected'. | • "Claim Item" button is unavailable.<br>• The claim button is greyed out and show the text "Already Claim" instead of "Claim Item".<br>• Failed to submit claim request. | As expected | Pass |
| 4. | Attempt to submit a claim request for a matched found item marked as 'pending'. | 1. Tap the desired matched found item that is marked as 'pending'. | • "Claim Item" button is unavailable.<br>• The claim button is greyed out and show the text "Already Claim" instead of "Claim Item".<br>• Failed to submit claim request. | As expected | Pass |
| 5. | Attempt to submit | 1. Tap the | • "Claim Item" | As | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | a claim request for a matched found item marked as 'dismissed'. | desired matched found item that is marked as 'dismissed'. | button is unavailable.<br>• The claim button is greyed out and show the text "Claimed by Others/ You have pending claim" instead of "Claim Item".<br>• Failed to submit claim request. | expected | |
| Post Conditions: After the claim request is submitted successfully, a claim record linking both lost and found item is created, other potential matches for the same lost item are marked as "dismissed" temporarily until the ongoing claim for the item is rejected, claim appears in claim tracking list. | | | | | |

*Table 5.25 Student track claim request status of a matched found item*

| | | | | | |
|---|---|---|---|---|---|
| Test Case ID: CM-03<br>Module: Claim Management<br>User Role: Student<br>Test Case Title: Verify student can track the claim request status of found item/s from potential match for a lost item.<br>Preconditions:<br>1. Student has a verified account.<br>2. Student is logged-in.<br>3. Student has reported a lost item.<br>4. System has generated potential matches between the lost item and found item.<br>5. Student has submitted a claim request for a found item from the potential match.<br>6. [Navigation] Student is on the claims overview page (which lists all claims of found item/s for a lost item) (via My Items (in bottom navigation bar) > 'Lost' tab > Choose item). | | | | | |
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Track claim request status after submitting a claim request for a found item from the potential match. | 1. Tap "My Claims" tab.<br>2. Locate the interest claim | • All claim request for the found item related to the specific lost item are displayed.<br>• The status of | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | request<br>3. Tap the claim request (optional).<br>. | claim request is visible at the claim requests overview page and claim details page. | | |
| 2. | Track claim request status before submitting a claim request for a found item from the potential match. | 1. Tap "My Claims" tab. | • An info message is displayed indicating that no claim request is found for the lost item. | As expected | Pass |
| Post Conditions: Current claim status of the item is displayed, admin justification is displayed if available. | | | | | |

*Table 5.26 Student track status of all claim requests*

| | | | | | |
|---|---|---|---|---|---|
| Test Case ID: CM-04<br>Module: Claim Management<br>User Role: Student<br>Test Case Title: Verify a student can track the status of every claim request they have submitted.<br>Preconditions:<br>1. Student has a verified account.<br>2. Student is logged-in.<br>3. Student has reported a lost item.<br>4. [Navigation] Student is on the claims overview page (which lists all claims of found item/s) (via More (in bottom navigation bar) > 'My Claims' option). | | | | | |
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Track the claim request status for a found item after submitting a claim request from the potential match. | 1. Locate the interest claim request<br>2. Tap the claim request (optional).<br>. | • All claim requests (whether from potential matches or direct claims) are displayed for found items.<br>• The status of claim request is visible at the claim requests | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | | overview page and claim details page. | | |
| 2. | Track the claim request status for a found item after submitting a claim request without making any lost item report. | 1. Locate the interest claim request<br>2. Tap the claim request (optional).<br>. | • All claim requests (whether from potential matches or direct claims) are displayed for found items.<br>• The status of claim request is visible at the claim requests overview page and claim details page. | As expected | Pass |
| 3. | Attempt to track the claim request status for a found item without submitting any claim request. | - | • An info message is displayed indicating that there is no claim request made by the student yet. | As expected | Pass |
| Post Conditions: All user claims are displayed, claim status is displayed, admin justification for the claim is displayed if available. | | | | | |

*Table 5.27 Admin compare items' information in a claim request*

| Test Case ID: CM-05<br>Module: Claim Management<br>User Role: Admin<br>Test Case Title: Admin compare information of found item and lost item (if any) in a claim request.<br>Preconditions:<br>1. Admin is logged-in. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Process the claim request (based on potential matches) | 1. Click the hamburger icon in the top | • A list of pending claim request for | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | in consideration of information of found item, lost item, student's justification and similarity score. | navigation bar to open sidebar menu.<br>2. Select "Claim Review" from the sidebar menu.<br>3. Apply filter (i.e. claim location/ faculty) or/and search keywords to locate the interest found item faster.<br>4. Select an item from the claim request list by clicking the "View" button.<br>5. Click the "Compare" button of the interested claim request (with information like claim ID, lost item ID, match ID, item name, image and claimant's matric no. provided) in the claim request list section.<br>6. Compare item information.<br>7. Click "Approve" or "Reject" button.<br>8. Enter justification in confirmation | different found items are displayed.<br>• The lost item section in the claim review page is originally empty until user compare item information.<br>• Similarity score and student's justification are shown.<br>• "Approve" and "Reject" buttons are enabled now.<br>• A confirmation dialog is displayed.<br>• A success message is displayed indicating that the claim request is approved or rejected successfully. | | |

| | | | | | |
|---|---|---|---|---|---|
| | | dialog (optional).<br>9. Click “Approve” or “Reject” button in confirmation dialog. | | | |
| 2. | Process the claim request (student not reporting any lost item) in consideration of found item information and student’s justification. | 1. Repeat steps 1 to 4 in scenario 1.<br>2. Click the “Compare” button of the interested claim request (with claimant’s matric no. provided only) in the claim request list section.<br>3. Repeat steps 6 to 9 in scenario 1. | • A list of pending claim request for different found items are displayed.<br>• Only student’s justification is shown in lost item section.<br>• “Approve” and “Reject” buttons are enabled now.<br>• A confirmation dialog is displayed.<br>• A success message is displayed indicating that the claim request is approved or rejected successfully. | As expected | Pass |
| Post Conditions: Item comparison interface is populated with found item information, lost item information (if any), justification score (if any), and student justification. | | | | | |

*Table 5.28 Admin process claim request*

| Test Case ID: CM-06<br>Module: Claim Management<br>User Role: Admin<br>Test Case Title: Admin processes the claim request.<br>Preconditions:<br>1. Admin is logged-in.<br>2. Admin is on the claim review page.<br>3. There is information displayed in lost item section of the claim review page. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | Admin approves the claim request (from potential matches). | 1. Click "Approve" button.<br>2. Enter justification in confirmation dialog (optional).<br>3. Click "Approve" button. | • A confirmation dialog is displayed.<br>• A success message is displayed indicating that the claim request is approved successfully.<br>• All other claim requests involve the same found item are rejected automatically with justification "Automatically rejected as another claim for this item was approved." given.<br>• The matched found item in the successful claim request is now marked as 'approved', while the one involves in the rejected claim request is marked as 'rejected'.<br>• Other matched found items which are related to the lost item from the rejected claim request is now marked as 'available' from 'dismissed' if it is not | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | | yet claimed by others.<br>• Student involves in approved claim request is not allowed to submit claim request for the other found items related to the same lost item.<br>• Student involves in rejected claim request can now make claims of other matched found items for the lost item involved in the rejected claim request.<br>• Notifications on claim updates are sent to all the students submitting claim request for the found item. | | |
| 2. | Admin approves the claim request (which student submits without reporting a lost item). | 1. Repeat steps 1 to 3 in scenario 1. | • A confirmation dialog is displayed.<br>• A success message is displayed indicating that the claim request is approved successfully.<br>• All other claim requests involve the same found item are rejected automatically with justification "Automatically rejected as another claim for this item was approved." given.<br>• Notifications on claim updates are sent to all the students submitting claim request for the found item. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| 3. | Admin rejects the claim request (from potential matches). | 1. Click "Reject" button.<br>2. Enter justification in confirmation dialog (optional).<br>3. Click "Reject" button. | • A confirmation dialog is displayed.<br>• A success message is displayed indicating that the claim request is rejected successfully.<br>• The matched found item involves in the rejected claim request is marked as 'rejected'.<br>• Other matched found items which are related to the lost item from the rejected claim request is now marked as 'available' from 'dismissed' if it is not yet claimed by others.<br>• Student involves in rejected claim request can now make claims of other matched found items for the lost item involved in the rejected claim request.<br>• Notifications on claim updates are sent to the students. | As expected | Pass |
| 4. | Admin rejects the claim request (which student submits without reporting a lost item). | 1. Repeat steps 1 to 3 in scenario 3. | • A confirmation dialog is displayed.<br>• A success message is displayed indicating that the claim request is rejected successfully.<br>• Notifications on claim updates are sent to the students. | As expected | Pass |
| Post Conditions: If one claim is approved, other claims for the item is rejected automatically, and no more claim request is accepted for this item. If the claim is rejected, those who get rejected can submit claim request for | | | | | |

| |
|---|
| other items matched with their lost item. All claim notifications are sent to all affected students (claims rejected or approved). |

*Table 5.29 Admin manage previously processed claims*

| Test Case ID: CM-08<br>Module: Claim Management<br>User Role: Admin<br>Test Case Title: Admin manage previously processed claims.<br>Preconditions:<br>1. Admin is logged-in.<br>2. Student has reported a lost item.<br>3. System has generated potential matches between the lost item and found item. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | View previously processed claims. | 1. Click the hamburger icon in the top navigation bar to open sidebar menu.<br>2. Select "Claim Approval History" from the sidebar menu.<br>3. Apply filter (i.e. claim location/ faculty) or/and search keywords to locate the interest processed claim request faster.<br>4. Select a processed claim from the claim request list by clicking the "View" button. | • A list of processed claims are displayed.<br>• In the claim details page, found item information, claim information, lost item information (if any), similarity score (if any), admin justification (if any) are displayed. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| 2. | Confirm the item collection by the owner. | 1. Repeat steps 1 to 3 in scenario 1.<br>2. Click "Mark as Claimed" button.<br>3. Click "Confirm Collection" button in the confirmation dialog. | • A confirmation dialog on the action of marking item as collected is displayed.<br>• A success message about the item collection by student is displayed.<br>• The word "-" changed to "Collected" in the "Mark as Claimed" column of the processed claim list. | As expected | Pass |
| Post Conditions: All approved or rejected claims details are displayed. Item collection status cannot be undone after being confirmed. | | | | | |

### 5.2.5 Student Profile Management Module

*Table 5.30 Student manage profile*

| Test Case ID: SPM-01<br>Module: Student Profile Management<br>User Role: Student<br>Test Case Title: To verify student can view their profile information and edit their username.<br>Preconditions:<br>1. Student is logged-in. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | View user profile information | 1. Click "More" in bottom navigation bar.<br>2. Click "My Account" option in More page. | • Student's name, email account, matric number, and email verification status are displayed. | As expected | Pass |
| 2. | Edit username using a unique | 1. Repeat steps 1 to 2 | • Navigate user to a page to edit the | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | name. | in scenario 1.<br>2. Click "Change Username" option.<br>3. Enter a new username.<br>4. Click "Change Username" button. | username.<br>• A success message is displayed indicating that the username is changed successfully.<br>• Navigate user backs to "My Account" Page. | | |
| 3. | Edit username using a name taken by others. | 1. Repeat steps 1 to 4 in scenario 2. | • An error message appeared indicating that the username has already been taken by others. | As expected | Pass |
| Post Conditions: After change username successfully, the changes will be reflected in screen and database. | | | | | |

*Table 5.31 Admin view student profile information*

| Test Case ID: SPM-02<br>Module: Student Profile Management<br>User Role: Admin<br>Test Case Title: To verify admin can view student profile information.<br>Preconditions:<br>1. Admin is logged-in. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | View student profile information | 1. Click the hamburger icon in the top navigation bar to open sidebar menu.<br>2. Select "Student" from the sidebar menu.<br>3. Enter search keywords to | • Student's id, username, matric no., and email address are displayed. | As expected | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | locate the student faster (optional). | | | |
| Post Conditions: All student information are displayed. | | | | | |

### 5.2.6 Admin Dashboard Module

*Table 5.32 Admin view charts in dashboard*

| Test Case ID: AD-02<br>Module: Admin Dashboard<br>User Role: Admin<br>Test Case Title: To verify admin can view all charts loaded in dashboard properly.<br>Preconditions:<br>1. Admin is logged-in. | | | | | |
|---|---|---|---|---|---|
| Scenario | Description | Test Steps | Expected Result | Actual Result | Status (Pass/Fail) |
| 1. | View charts in dashboard | 1. Click the hamburger icon in the top navigation bar to open sidebar menu.<br>2. Select "Dashboard" from the sidebar menu.<br>3. Apply filter (i.e. faculty) to display data for specific faculty. | • A chart illustrates the number of registered students from all faculty using the system.<br>• 3 additional charts displaying the number of pending claims, approved claims, and recovery rate, each of which updates dynamically based on the selected faculty filter. If no faculty is chosen ("All Faculty" selected), the 3 charts will display aggregated data from all faculties. | As expected | Pass |
| Post Conditions: Current system metrics are displayed. | | | | | |

## 5.3 Usability Testing

Usability testing is an evaluation method that measures how effectively users can interact with the interface design as well as the system functions. This evaluation uses a set of questionnaires to gather feedbacks about user satisfaction with the system from the user perspective.

The questionnaire examines 2 main parts of user experience. The first section of the evaluation is interface satisfaction, while the second section of the evaluation is functionality satisfaction. These evaluations are important to ensure intuitive interface design and powerful, reliable functionality. Usability testing was performed with two user groups, administrators and students, each with different workflows. In the following sections we show the analysis of the questionnaire results from both user groups. There is a total of 30 responses were gathered through the Google survey attached in APPENDIX C.

### 5.3.1 Student Questionnaire Results

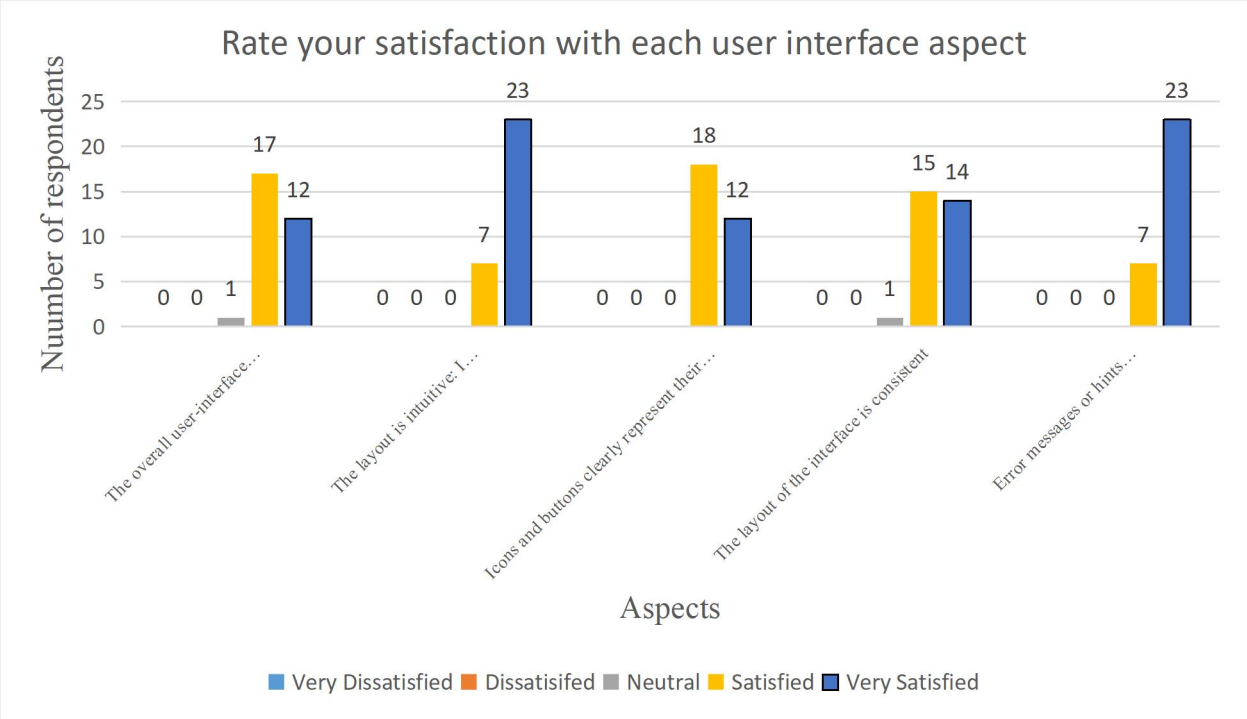



*Figure 5.1 Satisfaction Level on Student Interface*

Figure 5.1 reveals that the respondents are generally having positive experience with the system interface as no respondents were dissatisfied across 5 aspects of the interface. The strongest areas are navigation and handling of errors - 23 out of 30 respondents gave highest satisfaction rating for the question "The layout is intuitive I can quickly find what I need" and "Error messages or hints helps recovering from mistakes", while the other 7 users felt satisfied. 17 respondents were satisfied and 12 very satisfied with the overall visual appearance of the app. Icons and buttons got good ratings as well: 18 respondents were satisfied and 12 very satisfied with how they represent their functions. Besides, there are 15 satisfied respondents and 14 very satisfied respondents for layout consistency. As a result, it suggests that the system is easy to use, visually appealing and reliable for completing their tasks.

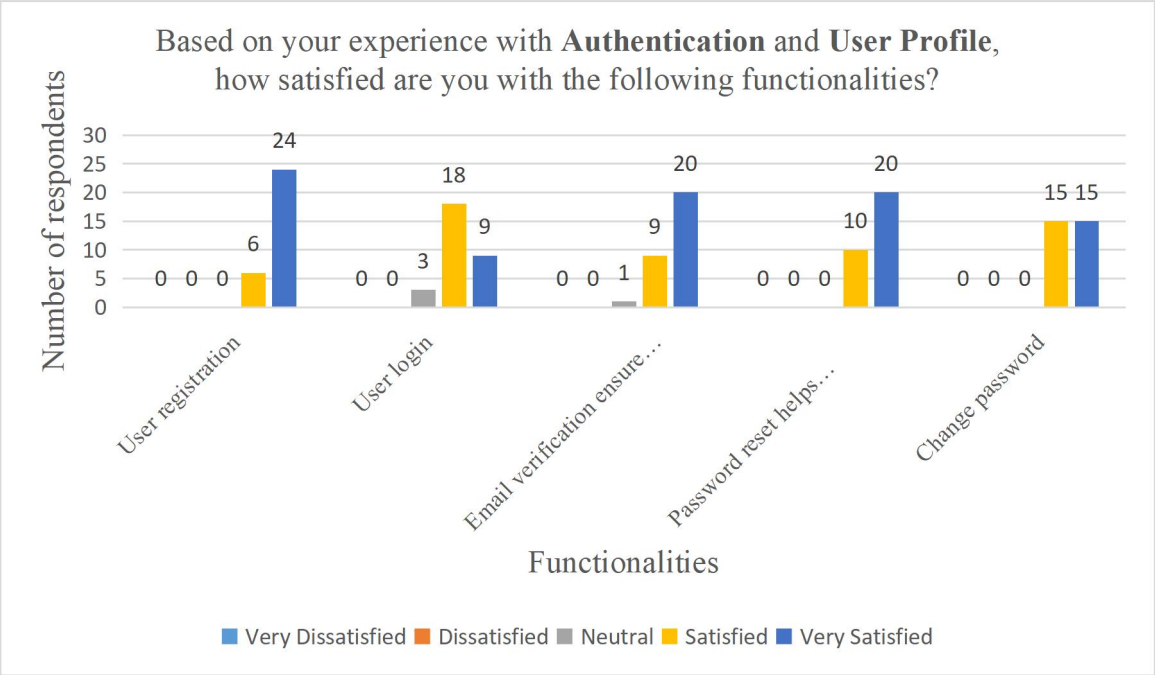



*Figure 5.2 Satisfaction Level on Authentication and Student Profile*

Figure 5.2 demonstrates overall positive user satisfaction with the functionalities in authentication and user profile module. Satisfaction level of registration was very high with 24 respondents very satisfied and 6 satisfied. User login also shows strong satisfaction levels with 18 respondents rating it as satisfied and 9 as very satisfied, though there are 3

respondents rating it as neutral (but is acceptable since they represent a very small minority). Email verification for account security was also very well received with 20 very satisfied and 9 satisfied and 1 neutral. Apart from that, password reset functionality performs well with 20 very satisfied and 10 satisfied. Finally, 15 respondents each rating feature 'change password' as satisfied and very satisfied. The high satisfaction rating across authentication and user profile functionality shows that the system provides good user experience across security related functionality.

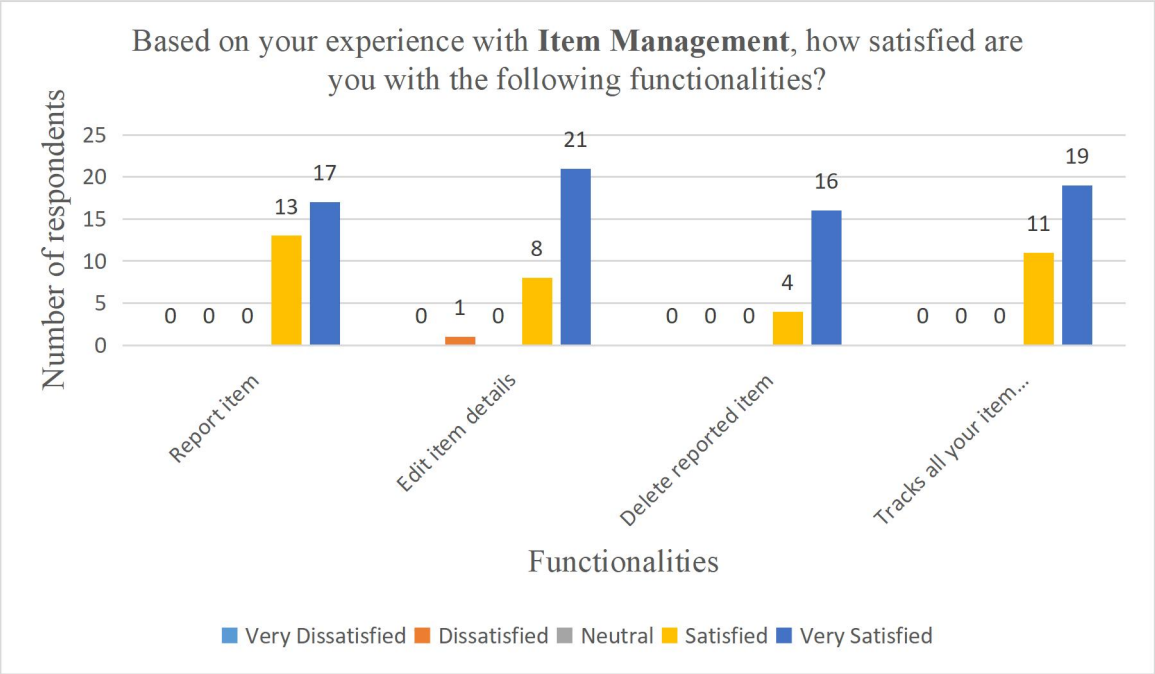



*Figure 5.3 Satisfaction Level on Item Management by Student*

Figure 5.3 demonstrates the satisfaction level on item management by student. The "report item" feature shows excellent user acceptance with 13 respondents satisfied and 17 respondents very satisfied. For the feature "edit item details", there are 8 respondents satisfied with it, 21 respondents very satisfied with it but there is 1 user feeling dissatisfied with it, possibly due to unclear error message causing them fail to edit item details. Regarding the feature of deleting reported item, 4 respondents were satisfied and 16 were satisfied. Additionally, 11 respondents were satisfied with the feature to track all the item

reports, and 19 were very satisfied. Overall, the data supports the statement that all the functionalities in item management are highly usable.

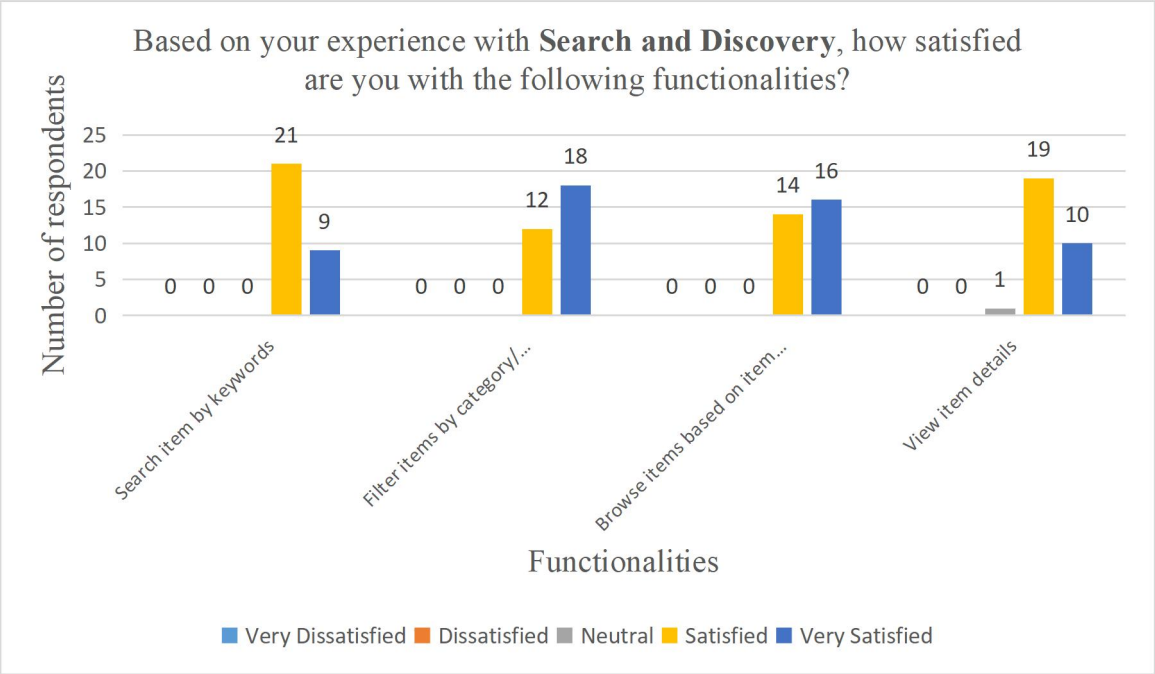



*Figure 5.4 Satisfaction Level on Search and Discovery by Student*

Figure 5.4 demonstrates the satisfaction level on search and discovery module. 21 respondents were satisfied, and 9 were very satisfied with search item by keywords functionality. The "Filter items by category/ colour/ location" feature receive the best feedback here from 12 satisfied and 18 very satisfied respondents. Besides, 14 respondents were satisfied, and 16 were very satisfied with the feature to browse item based on item type. Finally, the "view item details" feature works well with 19 satisfied respondents and 10 very satisfied respondents. There is none negative feedback as shown in the image, but there is a neutral response recorded for feature "view item details", which does not significantly impact the high satisfaction level among the respondents. These results indicate that the search and discovery module depicts high usability effectiveness, as respondents can efficiently locate items through multiple intuitive pathways—keyword search, categorical filtering, and type-based browsing, while the view functionality provides comprehensive information.

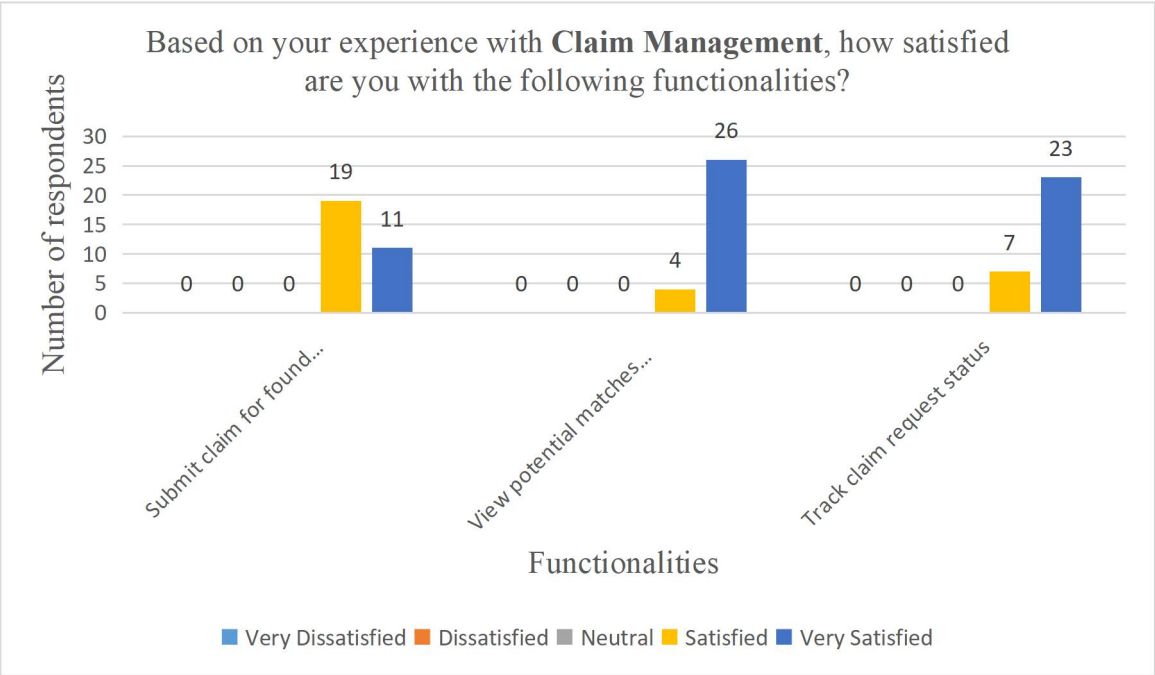



*Figure 5.5 Satisfaction Level on Claim Management by Student*

Figure 5.5 demonstrates the satisfaction level on claim management module. Satisfaction level of the functionality "submit claim for found items" was very high with 19 respondents satisfied and 11 very satisfied. The feature to view potential matches for lost items also shows strong satisfaction levels with 4 respondents rating it as satisfied and 26 as very satisfied. Finally, 7 respondents were satisfied, and 23 respondents were very satisfied with the functionality to track claim request status. As a result, respondents were satisfied with the usability design of the system in helping them navigate the claim management process efficiently.

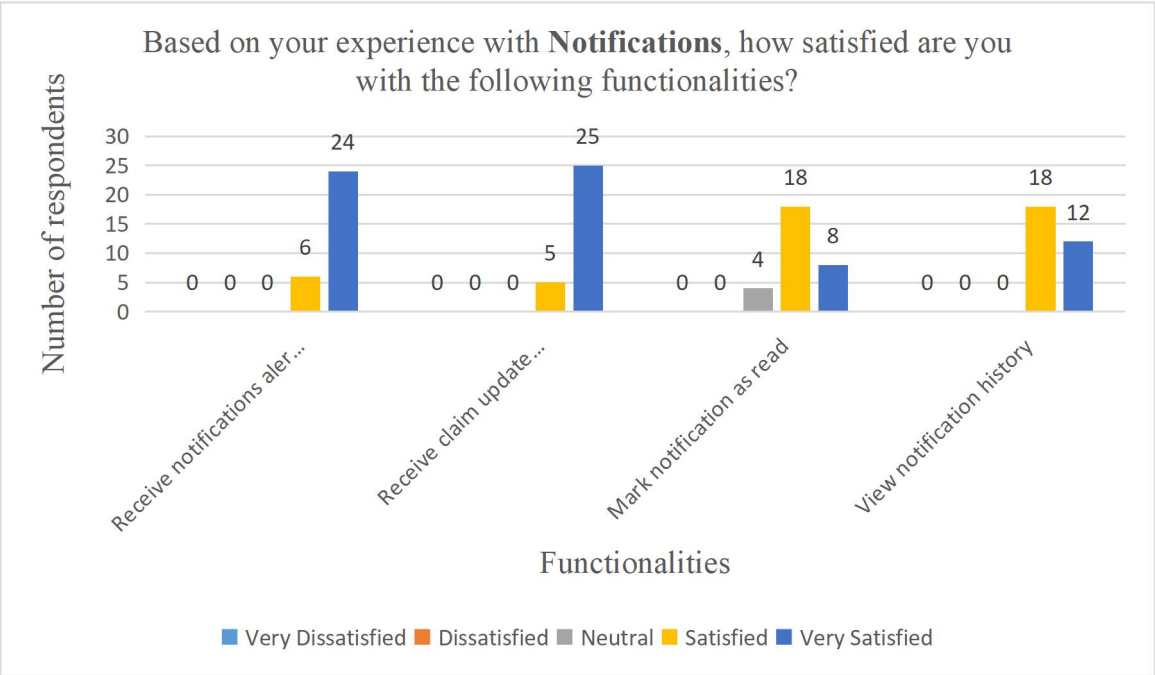



*Figure 5.6 Satisfaction Level on Student Notifications*

Figure 5.6 demonstrates the satisfaction level on notification module. The "Receive notifications alert about potential item matches" performs well with 6 satisfied respondents and 24 very satisfied respondents. Additionally, 5 respondents were satisfied, and 25 respondents were very satisfied with the functionality "receive claim update notifications". 18 satisfied respondents and 8 very satisfied respondents were recorded for the "Mark notification as read" feature. Apart from that, respondents seem to have positive experience when viewing notification history as 18 respondents feeling satisfied and 12 feeling very satisfied for this feature. Regarding neutral responses which was recorded from the feature "mark notification as read" has no significant impact on the system's robust design approach. These results illustrates that the system delivers high usability effectiveness, enabling respondents to efficiently manage their notifications and maintain awareness of system activities.

### 5.3.2 Admin Questionnaire Results

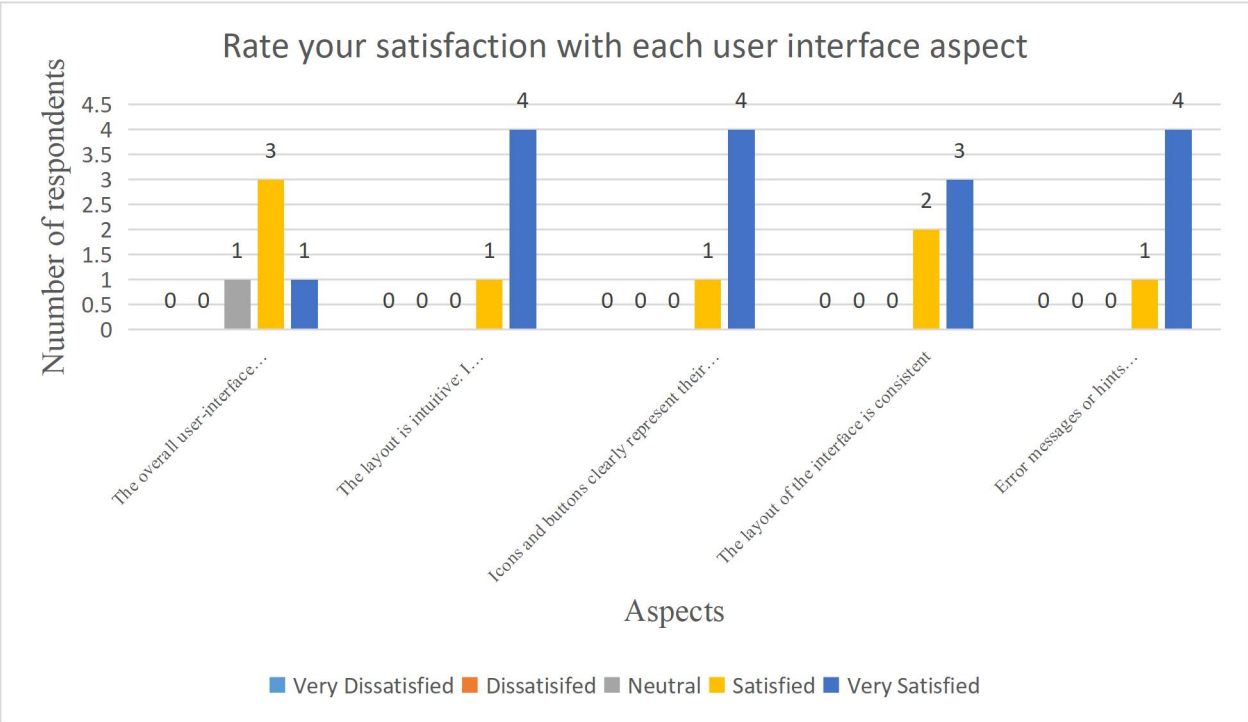



*Figure 5.7 Satisfaction Level on Admin Interface*

Figure 5.7 demonstrates the satisfaction level on admin interface. 1 user felt satisfied and 4 felt satisfied for the intuitive layout, clear icons and buttons, and useful error messages or hints. Furthermore, respondents have good experience with the consistent interface layout with the data that 2 respondents were satisfied and 3 were very satisfied. Finally, the absence of dissatisfied respondents for the question "the overall user-interface design of the app is visually appealing" reveals that respondents find the visual design pleasing, with only 1 neutral response recorded for this interface aspect which does not have effect on positive usability assessment. These results demonstrate that the admin interface delivers high usability effectiveness through well-designed interface elements that prioritize visual appeal, intuitive navigation, clear communication, consistency, and effective error recovery.

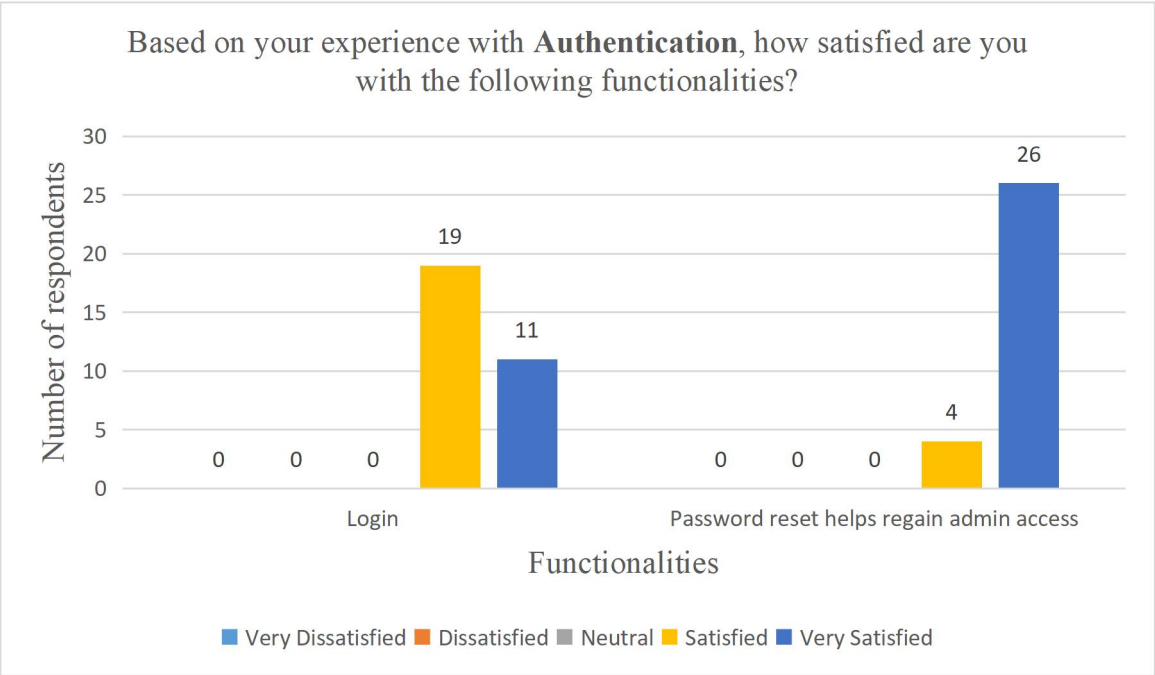



*Figure 5.8 Satisfaction Level on Admin Authentication*

Figure 5.8 demonstrates the satisfaction level on admin authentication module. All functionalities in this module demonstrates a polarized user experience, with 19 respondents felt satisfied, and 11 felt very satisfied with the login process, while 4 respondents felt satisfied and 26 felt very satisfied with the password reset workflow or feature. This positive feedback pattern suggests that the login process and password recovery process implemented are logical and hence gives respondents positive experience.

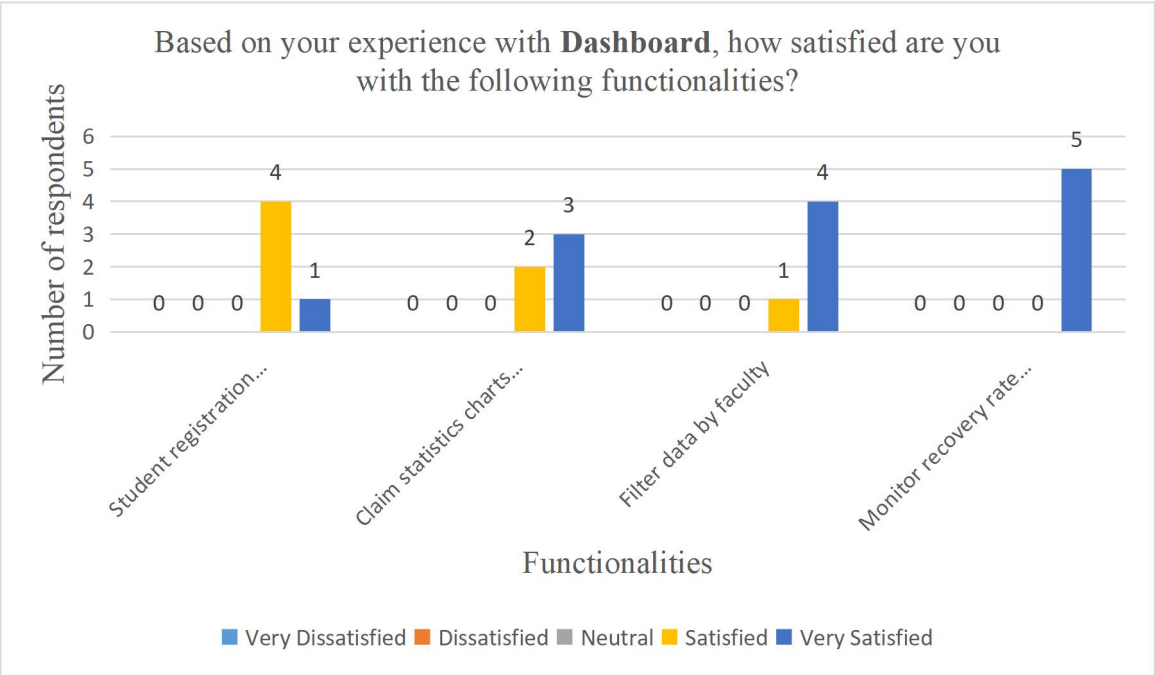



*Figure 5.9 Satisfaction Level on Admin Dashboard*

Figure 5.9 demonstrates the satisfaction level on admin dashboard. The function to monitor recovery rate in dashboard received highest level of positive feedback, with all respondents give response "very satisfied". There are 4, 2, and 1 user were satisfied with the displayed student registration metrics, claim statistics charts and "filter data by faculty" function. Besides, 1 user very satisfied with student registration metrics, 3 very satisfied with claim statistics charts, and 4 very satisfied with "filter data by faculty" function. Overall, the charts reflect positive experience with admin dashboard, indicating that the dashboard is user-friendly and effectively supports administrative needs.

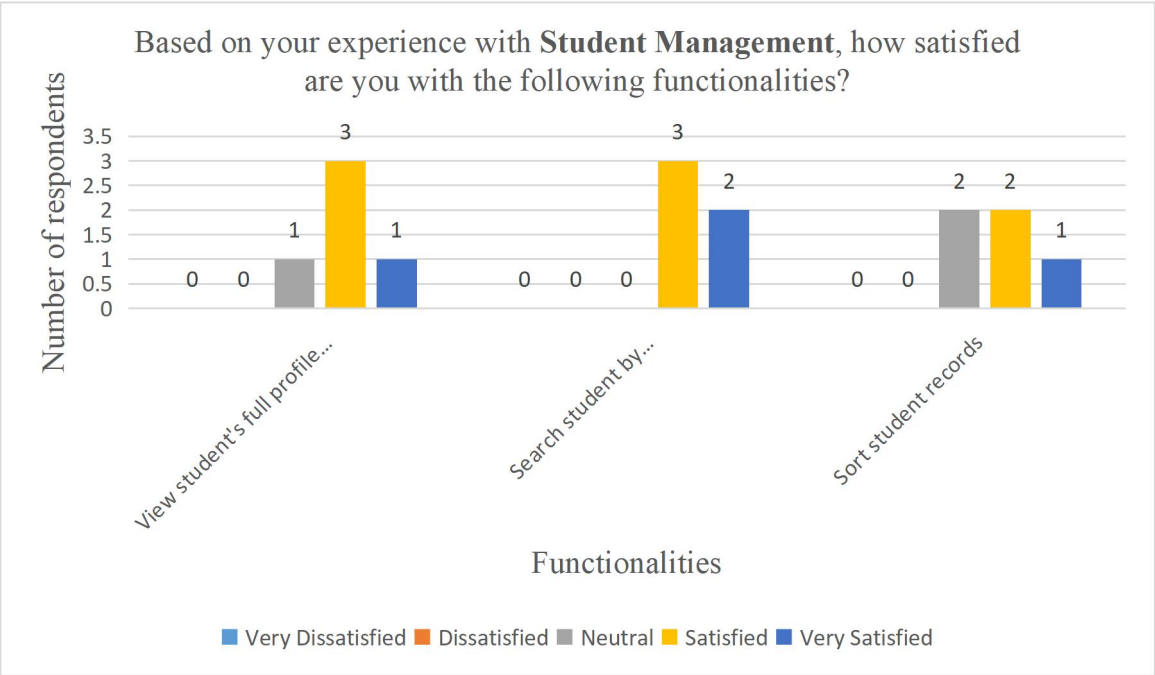



*Figure 5.10 Satisfaction Level on Student Management by Admin*

Figure 5.10 demonstrates the satisfaction level on admin dashboard. For the feature "view student's full profile information", most of the respondents (3) were satisfied with it, 1 was very satisfied, and 1 gave neutral response. Search student functionality had received excellent feedback from the respondents, with 3 users satisfied and 2 very satisfied. Lastly, there was an equal number of respondents (2 each) rate the satisfaction level as 'neutral' and 'satisfied', and 1 was very satisfied. While there were some neutral responses across the features, but the absence of dissatisfied responses across all functionalities highlights that the function is effective and meets user expectation.

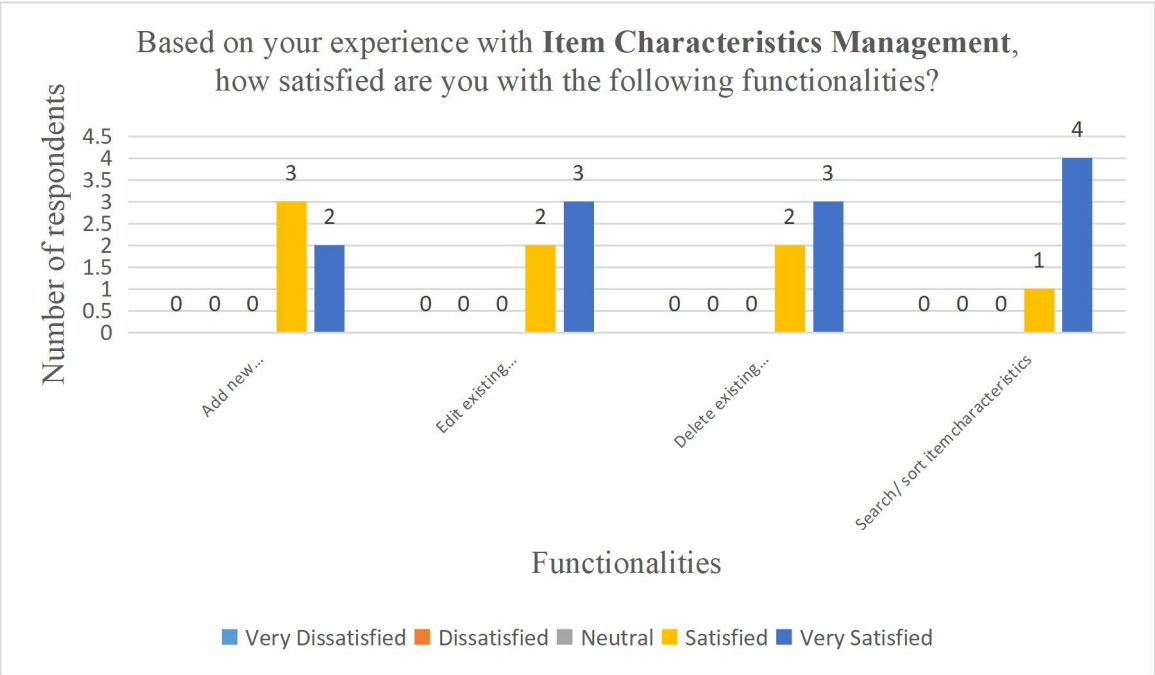



*Figure 5.11 Satisfaction Level on Item Characteristics Management by Admin*

Figure 5.11 demonstrates the satisfaction level on item characteristics management module. For the features "Add new categories/colours/locations", "Edit existing categories/colours/locations", "Delete existing categories/colours/locations" and "Search/sort item characteristics", there were 3, 2, 2, and 1 respondent who were satisfied respectively, while 2, 3, 3, and 4 respondents were very satisfied respectively. The high satisfaction levels suggest that the system is well-designed to support efficient item management tasks for administrators.

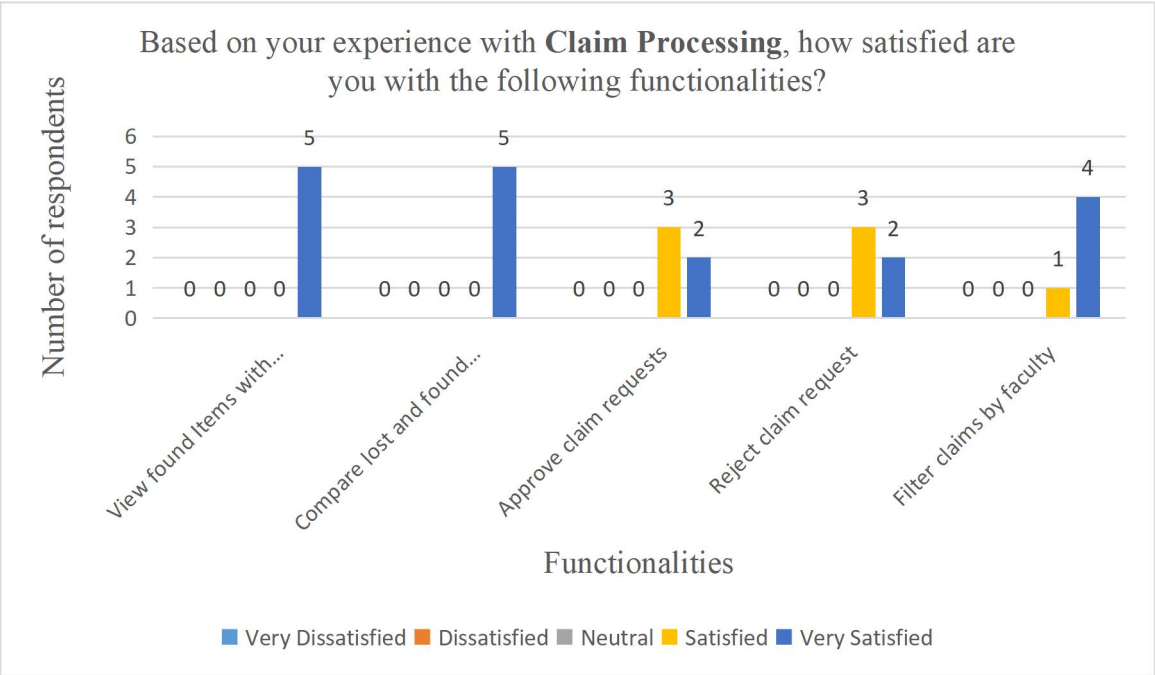



*Figure 5.12 Satisfaction Level on Claim Processing by Admin*

Figure 5.12 demonstrates the satisfaction level on claim processing module. The features "view found items with pending claims" and "compare lost and found items" depicts the highest satisfaction levels, with both receiving 5 "very satisfied" responses. For features "approve claim request" and "reject claim request", they show similar pattern with 3 satisfied responses and 2 very satisfied responses. Additionally, 1 respondent was satisfied and 4 were satisfied to be able to filter claims by faculty. These positive feedback from respondents indicate these features are performing well and meeting user's expectation.

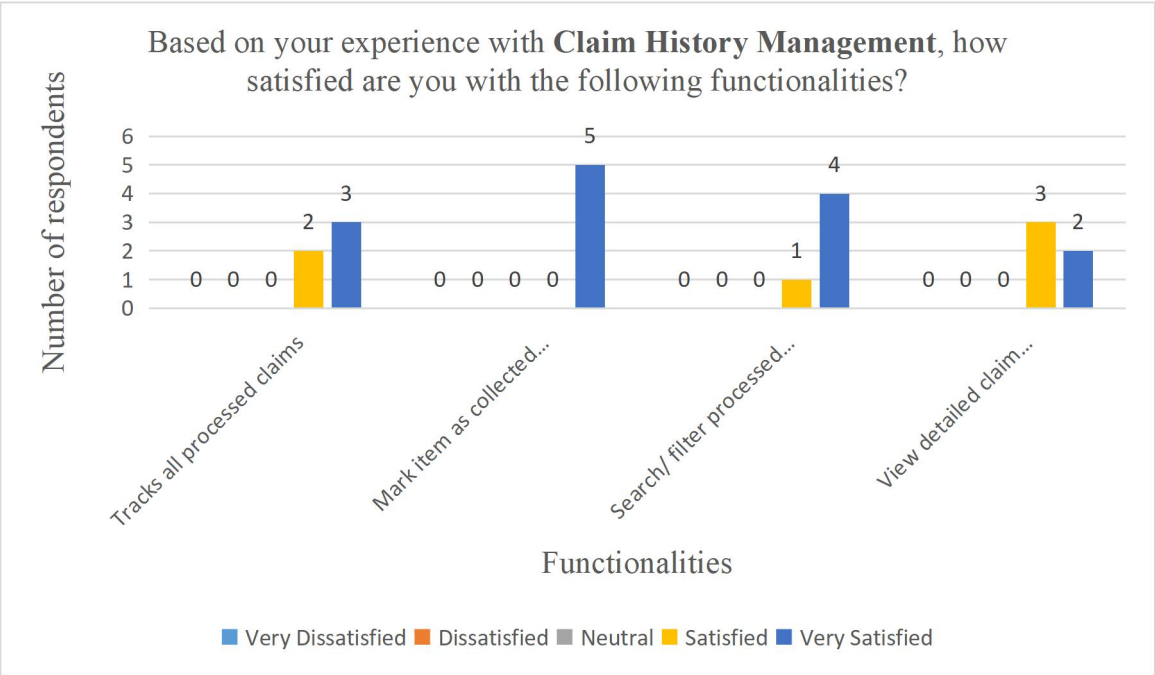



*Figure 5.13 Satisfaction Level on Claim History Management by Admin*

Figure 5.13 demonstrates the satisfaction level on claim history management module. The features "mark item as collected" depicts the highest satisfaction levels, with 5 very satisfied respondents. The feature to track all processed claims perform well with 2 satisfied responses and 3 very satisfied responses collected. Similarly, the "search/ filter processed claims" works well with 1 satisfied response and 4 satisfied responses gathered from respondents. Lastly, 3 respondents were satisfied and 2 were very satisfied when viewing detailed claim information. Notably, across all four functionalities, there were zero negative responses (very dissatisfied, dissatisfied), which suggests that the claim history management module has achieved excellent usability design. The result reveals the features provide positive experience to the respondents and support efficient claim management workflows for administrators.

# Chapter 6: Conclusion

## 6.1 Introduction

This chapter concludes the FindIt project by reflecting the overall achievement. It evaluates the actual project outcomes and compares them with the initial objectives set to ensure project's success. Furthermore, this chapter discusses about the limitations and constraints faced in the current system, and the corresponding enhancements can be carried out in future work to expand the system capabilities.

## 6.2 Project Achievements

The comparison between the initial objectives defined in chapter 1 and the project achievements is shown in the table below:

*Table 6.1 Project objectives and achievements*

| Objectives | Achievements |
|---|---|
| To design and develop the FindIt: Simplifying Campus Item Recovery with AI-driven Lost and Found app. | The mobile app and web interface were successfully built using Flutter and Laravel respectively. The mobile application is built for student users to easily report lost or found items, search for lost or found items, and receive real-time notifications about matching items and claim status. On the other hand, the web interface is built for administrator to view user account, manage item characteristic, approve or reject item claims, view claim history and mark the item as collected. By combining these technologies and delivering these features, a robust system is built. |

| | |
|---|---|
| To integrate AI-driven features for automating item matching processes within the app. | The successful integration of an AI model for image feature extraction (i.e. DINOv2) and the utilization of comparison algorithm (i.e. cosine similarity) automates the process of identifying potential matches between reported lost and found items. When a new item is reported by the students, the extracted features of the item is compared against existing images in the database to identify visually similar items. The student/item owner benefits from this feature to reduce the reliance on manually searching of the found item periodically. The image matching mechanism also considers item metadata such as category, color, and location to further improve matching accuracy. |
| To evaluate the usability and functionality of the lost and found app. | To ensure the system met its functional and user-centric objectives, functional and usability testing were conducted. Functional testing validated that all system modules performed correctly as defined in the requirements. The result of usability testing revealed high levels of user satisfaction with the software system in terms of interface design and the ease of navigation. |

## 6.3 Limitations and Constraints

Despite the project's success, limited project duration and computer resources available had caused several limitations and constraints in the system. They are listed below:

- The current version of the mobile app is available only on Android, while iOS users have no access to the mobile app currently.
- The process of reporting item with image attached (involves extract image feature, query database and compare similarity between image) takes slightly longer time to complete because cloud computing resources is limited and hence CPU is used to running the AI model, DINOv2.
- Items with text-based identifiers such as student IDs or labels are not yet detectable through image matching.
- Direct communication between item owners and finders in the mobile application is not available and have to rely solely on admin's verification.

## 6.4 Conclusion

In a nutshell, this project successfully achieved its objectives by developing a functional mobile and web-based system, FindIt, that streamlines the process of recovering lost items within a university campus. Our main goal of making campus item recovery simpler via using image matching with AI was realized. The mobile application enables students to report lost or found items, while the web-based system allows administrators to manage user claims and item listings efficiently.

## 6.5 Future Work

To overcome the current system's limitations and constraints faced, the following future work is recommended:

- Expand mobile application support to iOS using Flutter's cross-platform capabilities.
- Utilize GPU in cloud to run the model and integrate a vector similarity search system (e.g., FAISS) to optimize match retrieval speed and image matching process.
- A separate OCR module (e.g., Tesseract, EasyOCR) can be added alongside DINOv2 to improve system performance in identifying ID cards, books, or labels.
- Incorporate an in-app chat system to facilitate direct communication between students.

## References


*About Us*. (2024). Lost & Found Network. https://lostandfoundnetworks.com/page/About%20Us

Anas, M., Saini, S., Agarwal, N., Khanna, A., Yadav, A., & Kaur, K. (2023). *An Assistive Tool for Finding Lost Items: A Review, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)*, 12(11). https://doi.org/10.17577/IJERTV12IS110077

App Store. (2023, November 11). *Lost'NFound*. App Store. https://apps.apple.com/my/app/lostnfound/id6470925549?platform=ipad

Atlassian, B. (n.d.). Waterfall Methodology for Project Management | Atlassian. Atlassian. https://www.atlassian.com/agile/project-management/waterfall methodology

Banoula, M. (2024, September 3). *What is Tensorflow? Deep Learning Libraries & Program Elements*. Simplilearn.com. https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflow

*Dinov2 Base · Models · Dataloop*. (2025). https://dataloop.ai/library/model/facebook_dinov2-base/

*Download and install Android Studio | Android Developers*. (2024). Android Developers. https://developer.android.com/codelabs/basic-android-kotlin- compose-install-android-studio#0

Fastfwd. (2022, March 14). *What is Laravel? A Beginner's Introduction - fastfwd*. Fastfwd. https://www.fastfwd.com/what-is-laravel/

GeeksforGeeks. (2023, December 29). Requirements Gathering Introduction, processes, benefits and tools. GeeksforGeeks. https://www.geeksforgeeks.org/requirements-gathering introduction- processes-benefits-and-tools/

GeeksforGeeks. (2024b, August 5). *Bootstrap Tutorial*. GeeksforGeeks. https://www.geeksforgeeks.org/bootstrap/

Habersham, L. (2021, June 16). Lost and Found: 4 Tips for Managing Commonly Misplaced Items. T-mobile. https://www.t-mobile.com/news/devices/lost-and-found-4-tips-for-managing-commonly-misplaced-items

Hoffman, H. (2024, July 22). *Hugging Face Transformers: Leverage Open-Source AI in Python*. https://realpython.com/huggingface-transformers/

Hoory, L., & Bottorff, C. (2024, October 15). What is waterfall methodology? Here's how it can help your project management strategy. Forbes Advisor. https://www.forbes.com/advisor/business/what-is-waterfall-methodology/

*Hugging face free tier*. (n.d.). FreeTier.co. https://freetier.co/directory/products/hugging-face

Jones, H. (2024, May 9). *Improving cross-platform compatibility enhances digital accessibility*. Baseline. https://www.baselinemag.com/news/improving-cross-platform-compatibility-enhances-digital-accessibility/

Karunaratne, A. (2022, September 17). *Laragon is a modern, simple, and flexible Windows Development Environment that provides support for multiple PHP, Apache, and MySQL versions, with quick HTTPS setup, Quick-app templates, Composer, npm, and more*. PHP.Watch. https://php.watch/articles/laragon- windows-php

Laoyan, S. (2024, April 26). Guide to Waterfall Methodology: Free Template & Examples [2024] • Asana. Asana. https://asana.com/resources/waterfall -project-management methodology

*Lost and Found - apps on Google Play*. (2024). https://play.google.com/store/apps/details?hl=en&id=com.techju.developer.lotandfound

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021.

*Meet Android Studio*. (n.d.). Android Developers. https://developer.android.com/studio/intro

Nikitin, V. (2024, April). *Software Development Methodologies: Types and comparison*. Software Development Methodologies: Types and comparison. (n.d.). https://www.itransition.com/software-development/methodologies

Onipe, D. T. (2023, May 7). *Introduction to Flutter: Getting Started with Cross- Platform Development*. DEV Community. https://dev.to/bigdexter/introduction-to-flutter-getting-started-with-cross-platform-development-mmg

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., & Bojanowski, P. (2024). DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Petkovic, J., Duench, S., Trawin, J., Dewidar, O., Pardo, J. P., Simeon, R., DesMeules, M., Gagnon, D., Roberts, J. H., Hossain, A., Pottie, K., Rader, T., Tugwell, P., Yoganathan, M., Presseau, J., & Welch, V. (2021). Behavioural interventions delivered through interactive social media for health behaviour change, health outcomes, and health equity in the adult population. *Cochrane Library*, *2021*(6). https://doi.org/10.1002/14651858.cd012932.pub2

RemoteScout. (2023, September 13). Das Wasserfallmodell in der Softwareentwicklung. *RemoteScout24*. https://remotescout24.com/en/blog/1217- waterfall-model-in-software- development

*Run apps on the Android Emulator*. (2024). Android Developers. https://developer.android.com/studio/run/emulator

Shizuya, Y. (2025, March 12). Vision Embedding Comparison for Image Similarity Search: EfficientNet vs. ViT vs. VINO vs. CLIP vs. BLIP2. *Medium*. https://pub.towardsai.net/vision-embedding-comparison-for-image-similarity-search-efficientnet-vs-4eac6bf553c4

Simplilearn. (2024, July 24). *What is FastAPI: The future of Modern Web development*. Simplilearn.com. https://www.simplilearn.com/what-is-fastapi-article

Stewart, W. (2025, June 10). *What is PyTorch? A deep dive for engineers (and how to deploy it)*. Northflank. https://northflank.com/blog/what-is-pytorch#:~:text=PyTorch%20is%20an%20open%2Dsource,Python%20integration%2C%20and%20broad%20ecosystem.

*Transfer learning and fine-tuning*. (2024). TensorFlow. https://www.tensorflow.org/tutorials/images/transfer_learning

*Visual Studio vs Visual Studio Code - What's Best In 2024?* (2024). https://www.turing.com/kb/ultimate-guide-visual-studio-vs-visual-studio-code

Yasar, K., & Lewis, S. (2022, November 16). *PyTorch*. Search Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/PyTorch

# Appendix A

| ID | Task Name | Duration | Start |
|---|---|---|---|
| 1 | **Phase 1: Final Year Project 1** | **103 days** | **10/7/24** |
| 2 | **1.1 Brief Proposal** | **14 days** | **10/7/24** |
| 3 | Identify Project Title | 2 days | 10/7/24 |
| 4 | Identify Basic Skill Requirements | 1 day | 10/9/24 |
| 5 | Write Background of the Project / Project Introduction | 5 days | 10/10/24 |
| 6 | Determine Problem Statement | 3 days | 10/15/24 |
| 7 | Determine Project Objectives | 2 days | 10/18/24 |
| 8 | Determine Project Outcomes | 1 day | 10/20/24 |
| 9 | Milestone: Brief Proposal Approved | 0 days | 10/20/24 |
| 10 | **1.2 Full Proposal** | **18 days** | **10/21/24** |
| 11 | Refine Background of the Project | 5 days | 10/21/24 |
| 12 | Write Project Scope | 4 days | 10/26/24 |
| 13 | Write Brief Methodolog | 4 days | 10/30/24 |
| 14 | Write Significance of Project | 3 days | 11/3/24 |
| 15 | Planning the project schedule | 2 days | 11/6/24 |
| 16 | Milestone: Full Proposal Approved | 0 days | 11/7/24 |
| 17 | **1.3 Project Introduction** | **7 days** | **11/7/24** |
| 18 | Write Project Outlline | 6 days | 11/7/24 |
| 19 | Compile Full Proposal with Project Outline | 1 day | 11/13/24 |
| 20 | Milestone: Project Introduction Approved | 0 days | 11/13/24 |

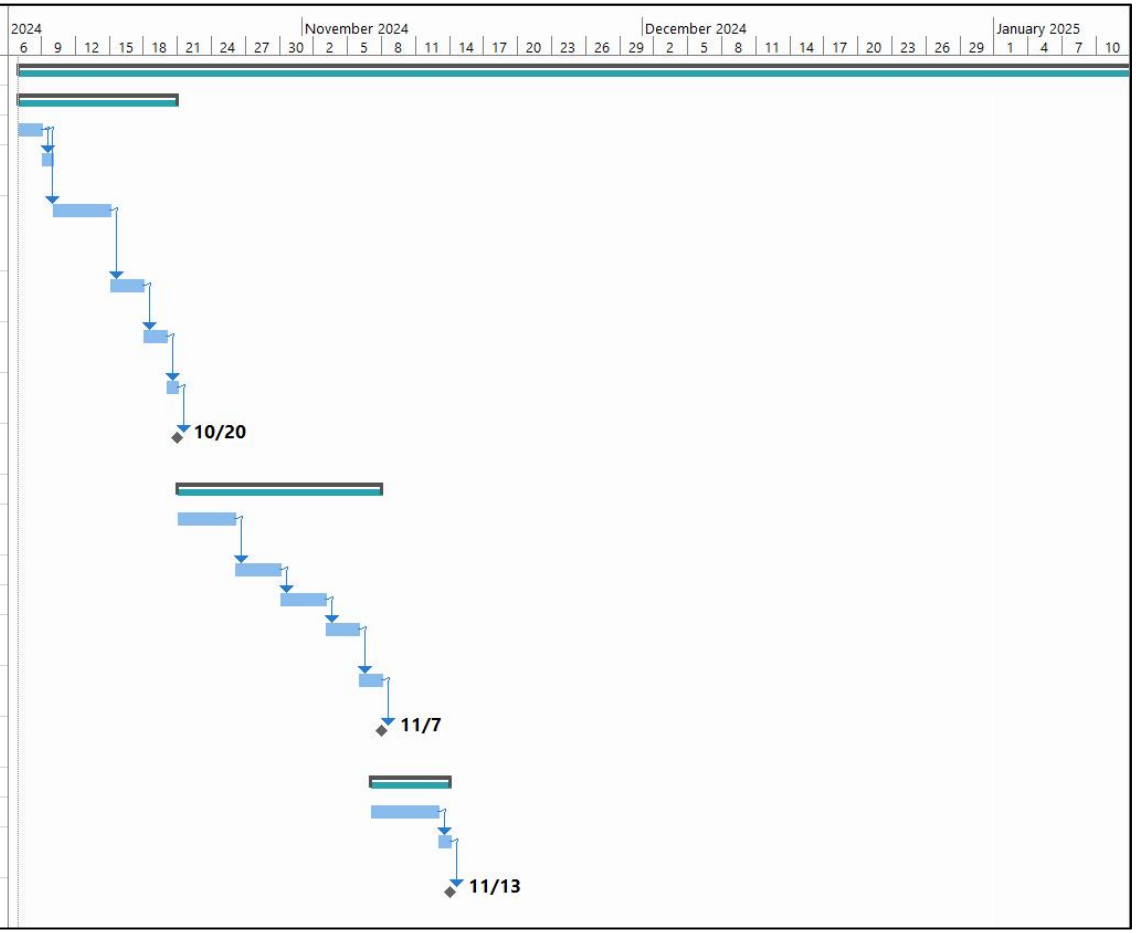



*Figure A.1 Project Gantt Chart for FYP1.*

| ID | Task Name | Duration | Start |
|---|---|---|---|
| 21 | **1.4 Literature Review** | **17 days** | **11/14/24** |
| 22 | Make a Review on Existing System | 9 days | 11/14/24 |
| 23 | Compare the Features between Existing and Proposed System | 5 days | 11/23/24 |
| 24 | Review Technologies and Tools used to Develop the Proposed | 3 days | 11/28/24 |
| 25 | Milestone: Literature Review Approved | 0 days | 11/30/24 |
| 26 | **1.5 Requirement Analysis and Design** | **36 days** | **12/1/24** |
| 27 | Collect and Analyze Requirements | 7 days | 12/1/24 |
| 28 | Document Requirement | 5 days | 12/8/24 |
| 29 | Design System | 7 days | 12/13/24 |
| 30 | Design Database | 7 days | 12/20/24 |
| 31 | Design User Interface | 10 days | 12/27/24 |
| 32 | Milestone: Requirement Analysis and Design Approved | 0 days | 1/5/25 |
| 33 | **1.6 FYP 1 Full Report** | **12 days** | **1/6/25** |
| 34 | Finalize and Compile all Chapters | 8 days | 1/6/25 |
| 35 | Prepare FYP 1 Presentat | 4 days | 1/14/25 |
| 36 | Milestone: FYP 1 Full Report Approved | 0 days | 1/17/25 |

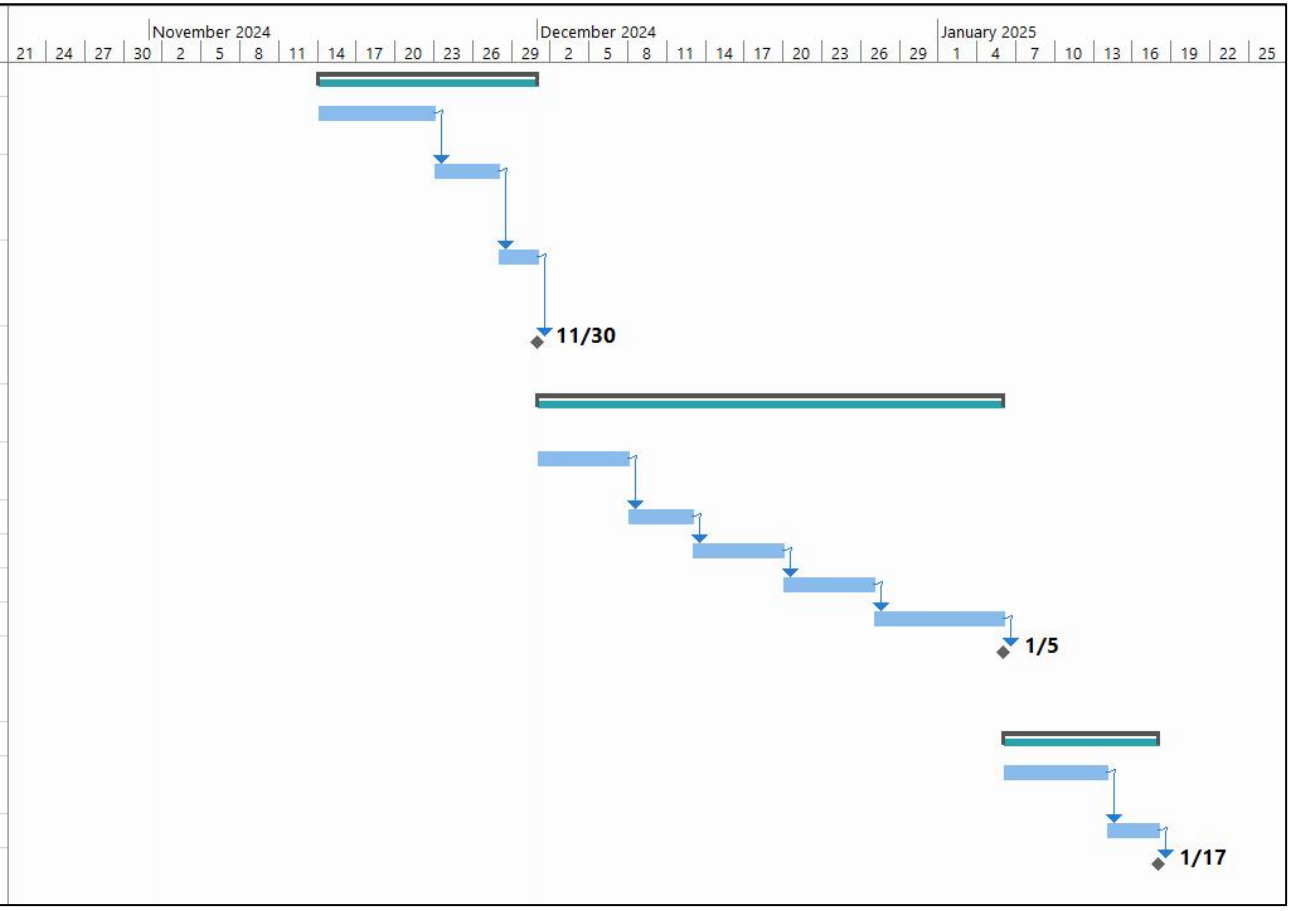



*Figure A.2 Project Gantt Chart for FYP1.*

| ID | Task Name | Duration | Start |
|---|---|---|---|
| 37 | **Phase 2: Final Year Project 2** | **120 days** | **2/24/25** |
| 38 | **2.1 Implementation** | **81 days** | **2/24/25** |
| 39 | Develop System | 70 days | 2/24/25 |
| 40 | Write Environment Setu | 4 days | 5/5/25 |
| 41 | Document User Interface Design and System Features | 7 days | 5/9/25 |
| 42 | Milestone: Implementation | 0 days | 5/15/25 |
| 43 | **2.2 Testing** | **12 days** | **5/16/25** |
| 44 | Develop Test Cases | 5 days | 5/16/25 |
| 45 | Conduct Functional Test | 2 days | 5/21/25 |
| 46 | Conduct Usability Testin | 2 days | 5/23/25 |
| 47 | Fix Bugs | 3 days | 5/25/25 |
| 48 | Milestone: Testing Completed | 0 days | 5/27/25 |
| 49 | **2.3 Conclusion and Future Works** | **4 days** | **5/28/25** |
| 50 | Conclude Project Achievement and Project Limitation | 2 days | 5/28/25 |
| 51 | Identify Future Work | 2 days | 5/30/25 |
| 52 | Milestone: Conclusion and Future Works | 0 days | 5/31/25 |
| 53 | **2.4 FYP 2 Full Report & Pa** | **10 days** | **6/1/25** |
| 54 | Finalize and Compile all Chapters | 4 days | 6/1/25 |
| 55 | Write Paper | 6 days | 6/5/25 |
| 56 | Milestone: FYP 2 Final Report and Paper Completed | 0 days | 6/10/25 |
| 57 | **2.5 FYP Final Report & Documentation** | **13 days** | **6/11/25** |
| 58 | Finalize and Compile all Chapters from FYP 1 & FYP 2 Report | 4 days | 6/11/25 |
| 59 | Prepare Installation Kits | 3 days | 6/15/25 |
| 60 | Prepare User Manual | 6 days | 6/18/25 |
| 61 | Milestone: FYP 2 Final Report and Documentation | 0 days | 6/23/25 |

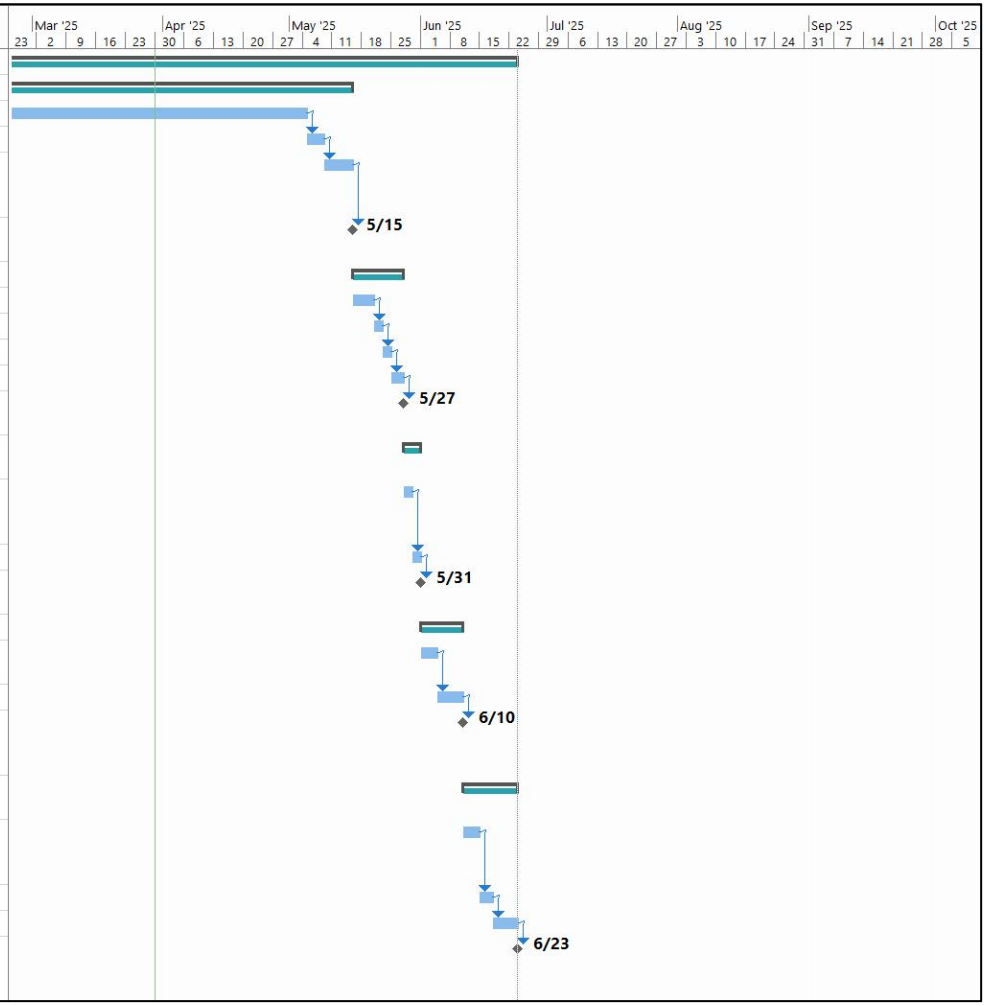
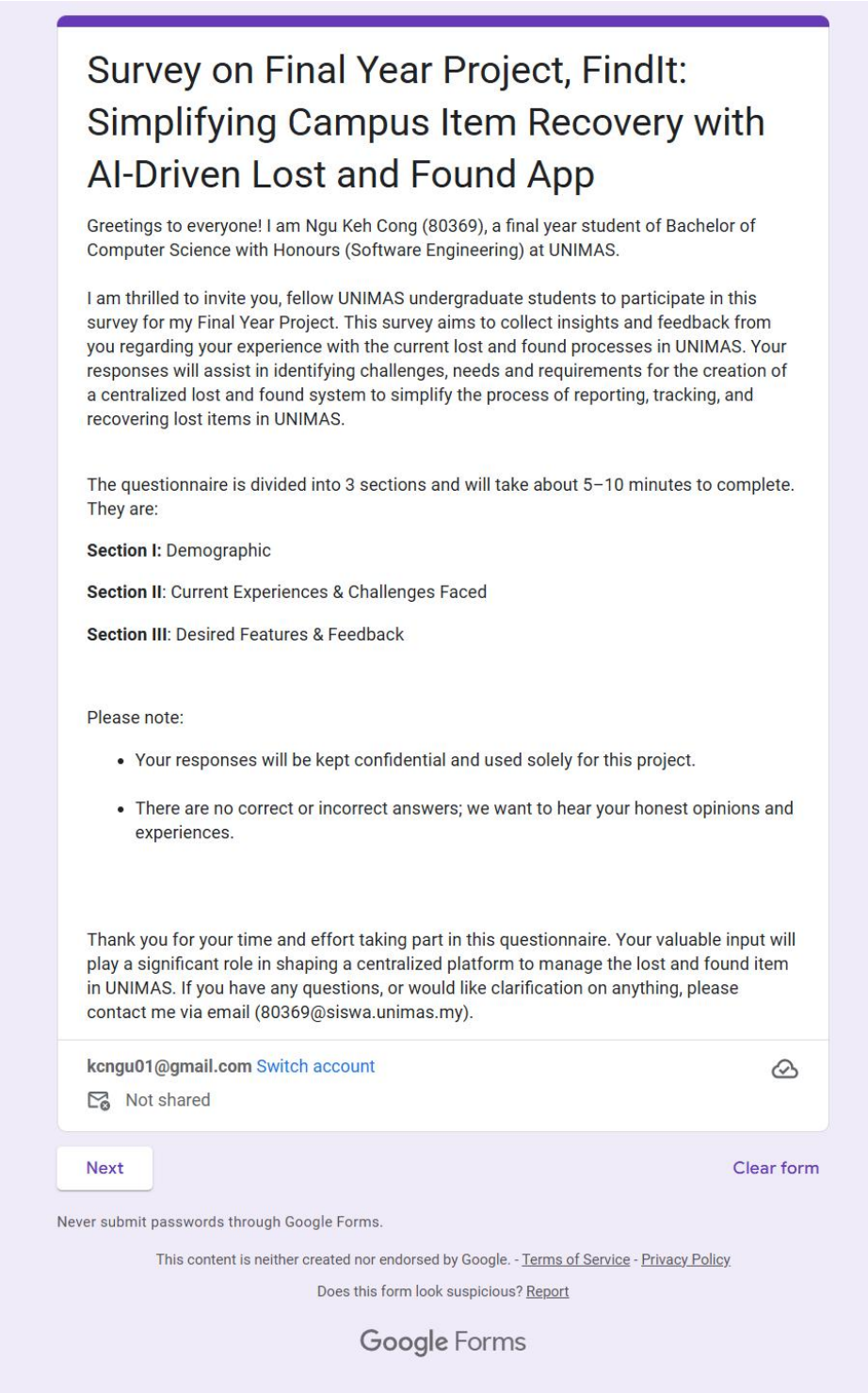



*Figure A.3 Project Gantt Chart for FYP2.*

# Appendix B



## Survey on Final Year Project, FindIt: Simplifying Campus Item Recovery with AI-Driven Lost and Found App

Greetings to everyone! I am Ngu Keh Cong (80369), a final year student of Bachelor of Computer Science with Honours (Software Engineering) at UNIMAS.

I am thrilled to invite you, fellow UNIMAS undergraduate students to participate in this survey for my Final Year Project. This survey aims to collect insights and feedback from you regarding your experience with the current lost and found processes in UNIMAS. Your responses will assist in identifying challenges, needs and requirements for the creation of a centralized lost and found system to simplify the process of reporting, tracking, and recovering lost items in UNIMAS.

The questionnaire is divided into 3 sections and will take about 5–10 minutes to complete. They are:

**Section I:** Demographic

**Section II**: Current Experiences & Challenges Faced

**Section III**: Desired Features & Feedback

Please note:

- Your responses will be kept confidential and used solely for this project.
- There are no correct or incorrect answers; we want to hear your honest opinions and experiences.

Thank you for your time and effort taking part in this questionnaire. Your valuable input will play a significant role in shaping a centralized platform to manage the lost and found item in UNIMAS. If you have any questions, or would like clarification on anything, please contact me via email (80369@siswa.unimas.my).

**kcngu01@gmail.com** Switch account

Not shared

Next

Clear form

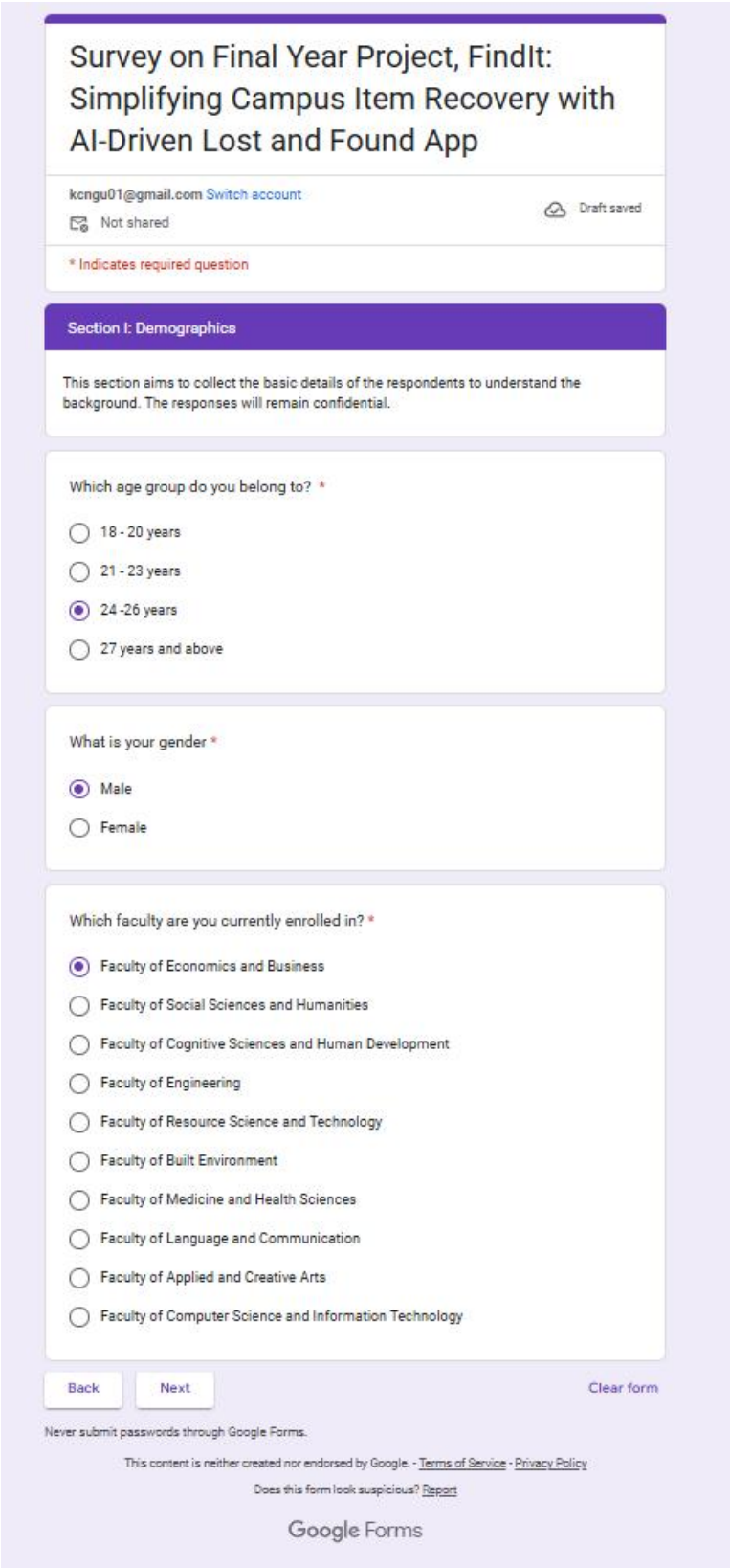
Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. - Terms of Service - Privacy Policy

Does this form look suspicious? Report

Google Forms

*Figure B.1 Survey Form Description.*



Survey on Final Year Project, FindIt: Simplifying Campus Item Recovery with AI-Driven Lost and Found App

kcngu01@gmail.com Switch account

Not shared

Draft saved

* Indicates required question

Section I: Demographics

This section aims to collect the basic details of the respondents to understand the background. The responses will remain confidential.

Which age group do you belong to? *

- 18 - 20 years
- 21 - 23 years
- 24 -26 years
- 27 years and above

What is your gender *

- Male
- Female

Which faculty are you currently enrolled in? *

- Faculty of Economics and Business
- Faculty of Social Sciences and Humanities
- Faculty of Cognitive Sciences and Human Development
- Faculty of Engineering
- Faculty of Resource Science and Technology
- Faculty of Built Environment
- Faculty of Medicine and Health Sciences
- Faculty of Language and Communication
- Faculty of Applied and Creative Arts
- Faculty of Computer Science and Information Technology

Back Next Clear form

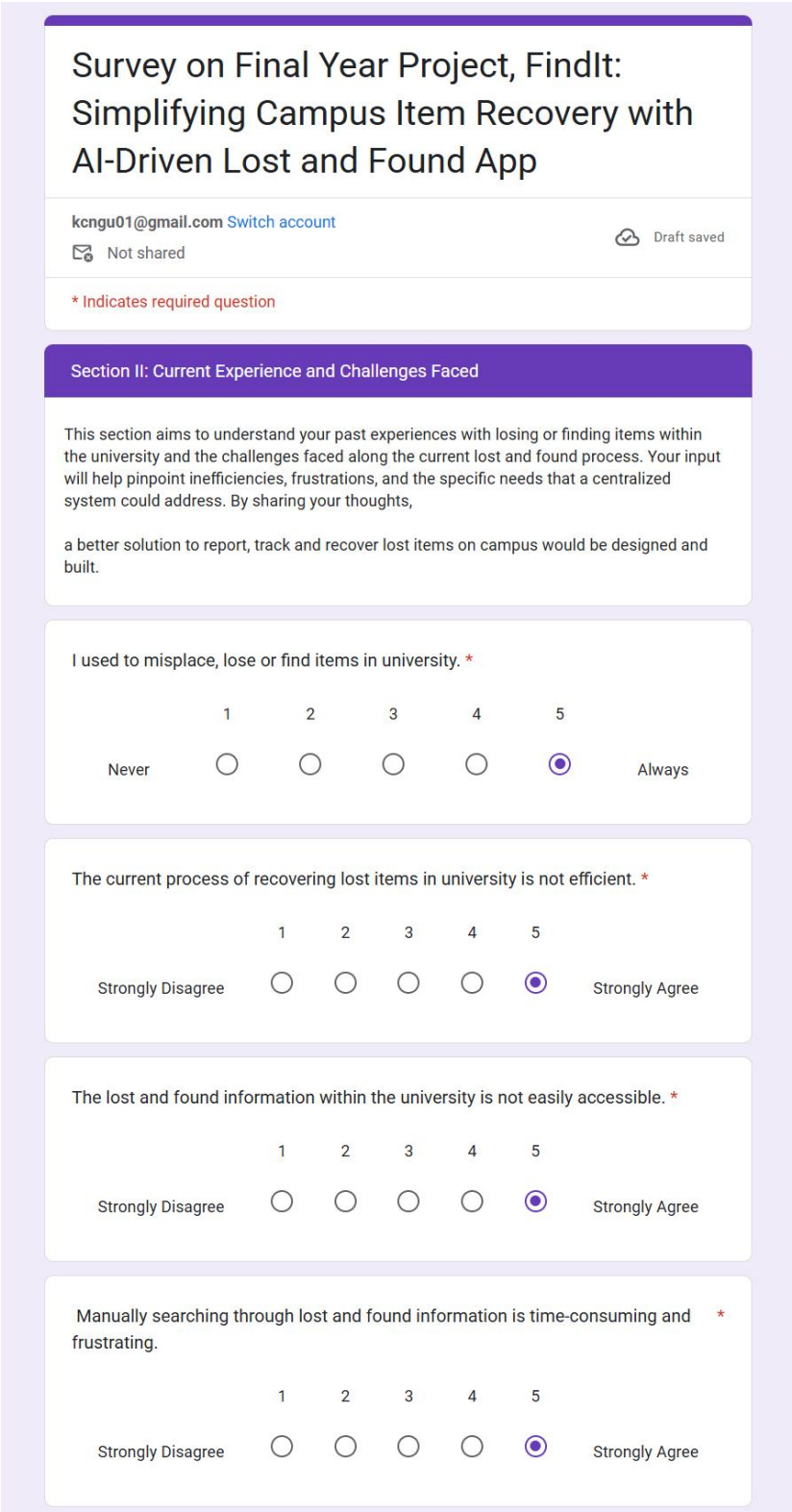
Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. - Terms of Service - Privacy Policy

Does this form look suspicious? Report

Google Forms

*Figure B.2 Section I for Demographics.*



Survey on Final Year Project, FindIt: Simplifying Campus Item Recovery with AI-Driven Lost and Found App

kcngu01@gmail.com Switch account

Not shared

Draft saved

* Indicates required question

Section II: Current Experience and Challenges Faced

This section aims to understand your past experiences with losing or finding items within the university and the challenges faced along the current lost and found process. Your input will help pinpoint inefficiencies, frustrations, and the specific needs that a centralized system could address. By sharing your thoughts,

a better solution to report, track and recover lost items on campus would be designed and built.

I used to misplace, lose or find items in university. *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Never | ○ | ○ | ○ | ○ | ◉ | Always |

The current process of recovering lost items in university is not efficient. *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Strongly Disagree | ○ | ○ | ○ | ○ | ◉ | Strongly Agree |

The lost and found information within the university is not easily accessible. *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Strongly Disagree | ○ | ○ | ○ | ○ | ◉ | Strongly Agree |

Manually searching through lost and found information is time-consuming and frustrating. *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Strongly Disagree | ○ | ○ | ○ | ○ | ◉ | Strongly Agree |

*Figure B.3 Section II for Current Experience and Challenges Faced.*

There is a need for a centralized system to manage the lost and found items in the university. *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Strongly Disagree | ○ | ○ | ○ | ○ | ◉ | Strongly Agree |

I felt more worried if I cannot track the status of the lost items. *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Strongly Disagree | ○ | ○ | ○ | ○ | ◉ | Strongly Agree |

The update on lost and found item is not timely. *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Strongly Disagree | ○ | ○ | ○ | ○ | ◉ | Strongly Agree |

The process of requesting updates on the lost items in the social media group (WhatsApp, Telegram) or from the staff repeatedly is tiring and troublesome. *

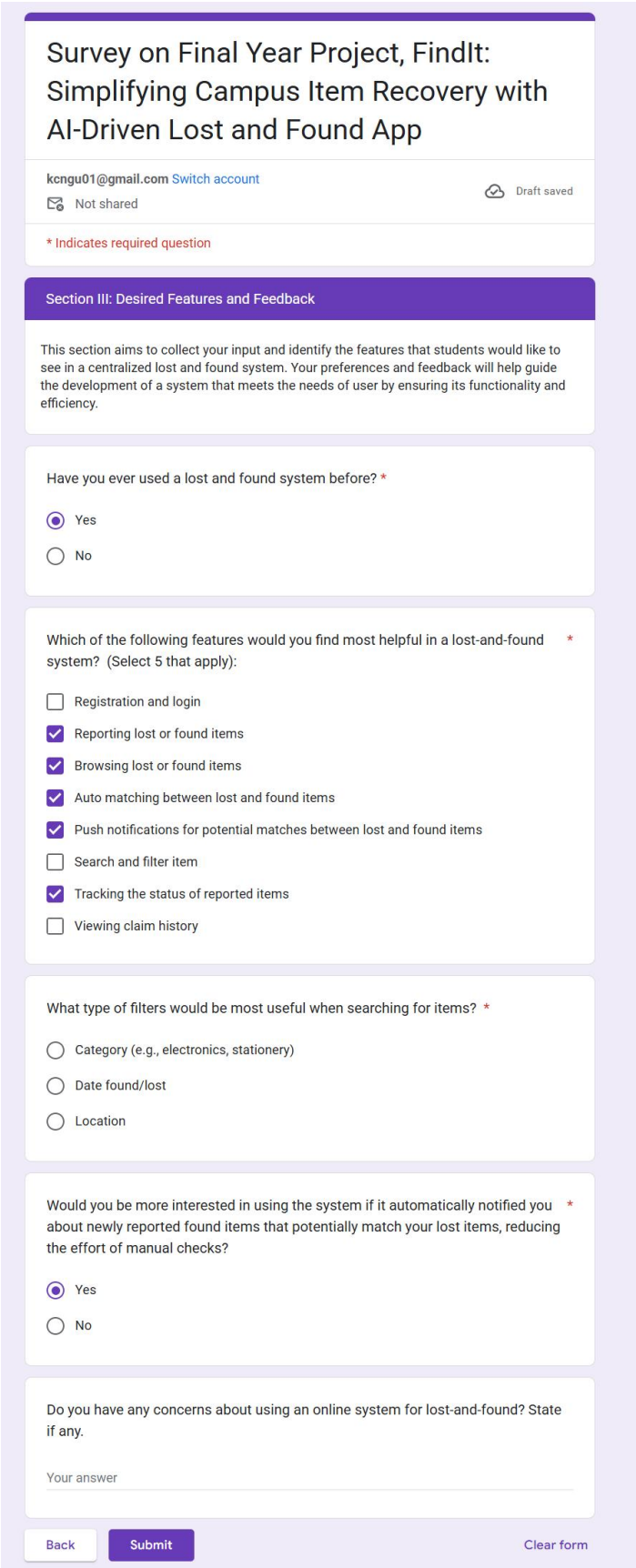
| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Strongly Disagree | ○ | ○ | ○ | ○ | ◉ | Strongly Agree |

The current process of reporting lost or found items is unclear. *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Strongly Disagree | ○ | ○ | ○ | ○ | ◉ | Strongly Agree |

Back Next Clear form

*Figure B.4 Section II for Current Experience and Challenges Faced.*



Survey on Final Year Project, FindIt: Simplifying Campus Item Recovery with AI-Driven Lost and Found App

kcngu01@gmail.com Switch account

Not shared

Draft saved

* Indicates required question

Section III: Desired Features and Feedback

This section aims to collect your input and identify the features that students would like to see in a centralized lost and found system. Your preferences and feedback will help guide the development of a system that meets the needs of user by ensuring its functionality and efficiency.

Have you ever used a lost and found system before? *

- (●) Yes
- ( ) No

Which of the following features would you find most helpful in a lost-and-found system? (Select 5 that apply): *

- [ ] Registration and login
- [x] Reporting lost or found items
- [x] Browsing lost or found items
- [x] Auto matching between lost and found items
- [x] Push notifications for potential matches between lost and found items
- [ ] Search and filter item
- [x] Tracking the status of reported items
- [ ] Viewing claim history

What type of filters would be most useful when searching for items? *

- ( ) Category (e.g., electronics, stationery)
- ( ) Date found/lost
- ( ) Location

Would you be more interested in using the system if it automatically notified you about newly reported found items that potentially match your lost items, reducing the effort of manual checks? *

- (●) Yes
- ( ) No

Do you have any concerns about using an online system for lost-and-found? State if any.

Your answer

Back Submit Clear form

*Figure B.5 Section III for Desired Features and Feedback.*

# Appendix C

## FindIt Mobile App - User Experience Feedback

Greetings to everyone! I am Ngu Keh Cong (80369), a final year student of Bachelor of Computer Science with Honours (Software Engineering) at UNIMAS.

This questionnaire aims to evaluate the usability of the FindIt mobile application. Before proceeding with this questionnaire, please visit FindIt to download the APK file and install the mobile app on your Android device. Spend a few minutes exploring the app's features and functionality to familiarize yourself with the system. As a **UNIMAS student**, you are our primary target user for this application.

The questionnaire is divided into 2 sections and will take about 5–10 minutes to complete. They are:

**Section I:** Satisfaction Level on User Interfaces

**Section II**: Satisfaction Level on Functionalities

Please note:

- Your responses will be kept confidential and used solely for this project.
- Answer based on your experience during today's testing session.
- There are no correct or incorrect answers; we want to hear your honest opinions and experiences.

Thank you for your time and effort taking part in this questionnaire. Your valuable input will play a significant role in shaping a centralized platform to manage the lost and found item in UNIMAS. If you have any questions, or would like clarification on anything, please contact me via email (80369@siswa.unimas.my).

**kcngu01@gmail.com** Switch account

Not shared

Next

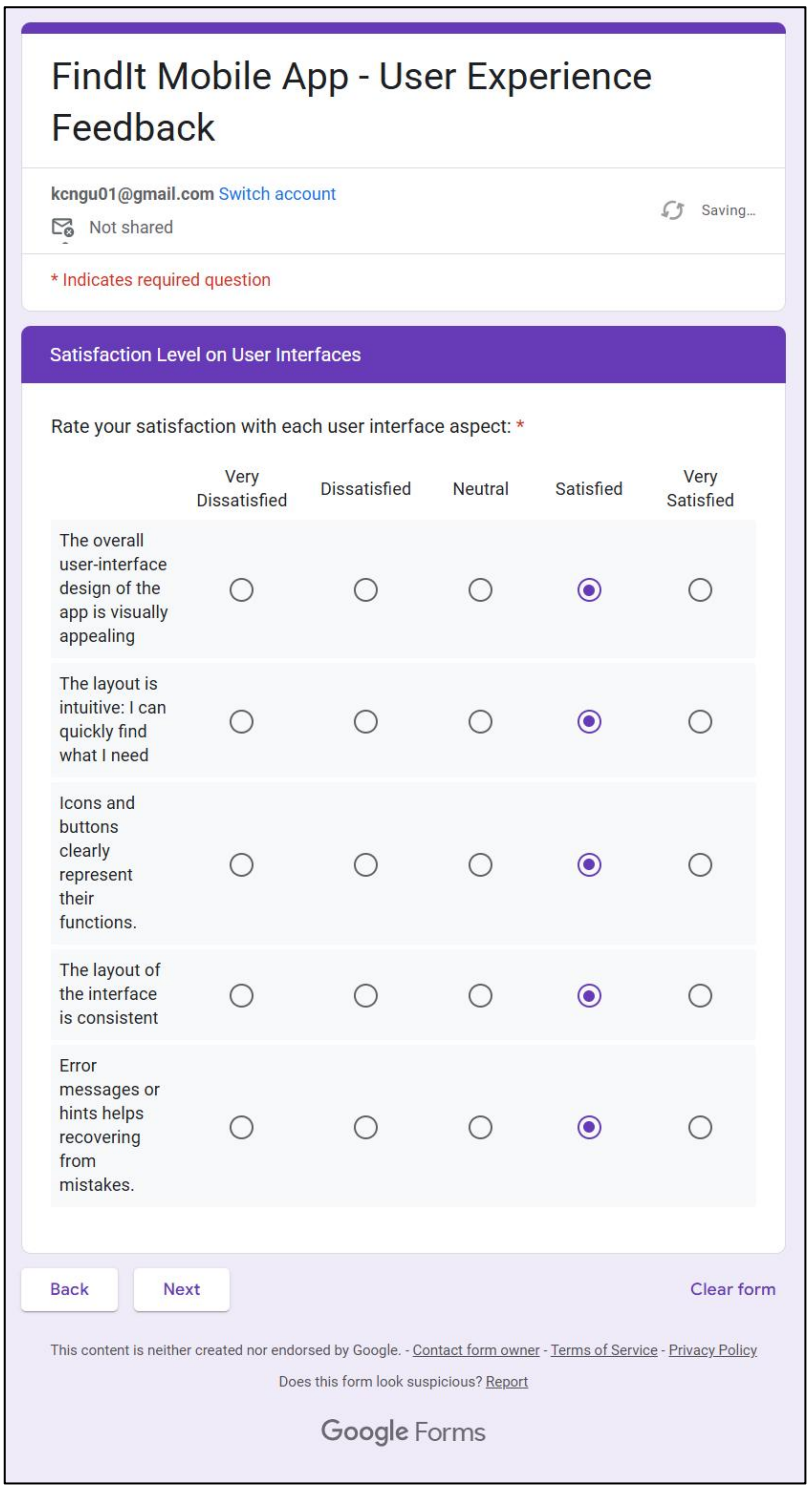
Clear form

This content is neither created nor endorsed by Google. - Contact form owner - Terms of Service - Privacy Policy

Does this form look suspicious? Report

Google Forms

*Figure C.1 FindIt App User Experience Survey Form Description*



FindIt Mobile App - User Experience Feedback

kcngu01@gmail.com Switch account

Not shared

Saving...

* Indicates required question

Satisfaction Level on User Interfaces

Rate your satisfaction with each user interface aspect: *

| | Very Dissatisfied | Dissatisfied | Neutral | Satisfied | Very Satisfied |
|---|---|---|---|---|---|
| The overall user-interface design of the app is visually appealing | ○ | ○ | ○ | ◉ | ○ |
| The layout is intuitive: I can quickly find what I need | ○ | ○ | ○ | ◉ | ○ |
| Icons and buttons clearly represent their functions. | ○ | ○ | ○ | ◉ | ○ |
| The layout of the interface is consistent | ○ | ○ | ○ | ◉ | ○ |
| Error messages or hints helps recovering from mistakes. | ○ | ○ | ○ | ◉ | ○ |

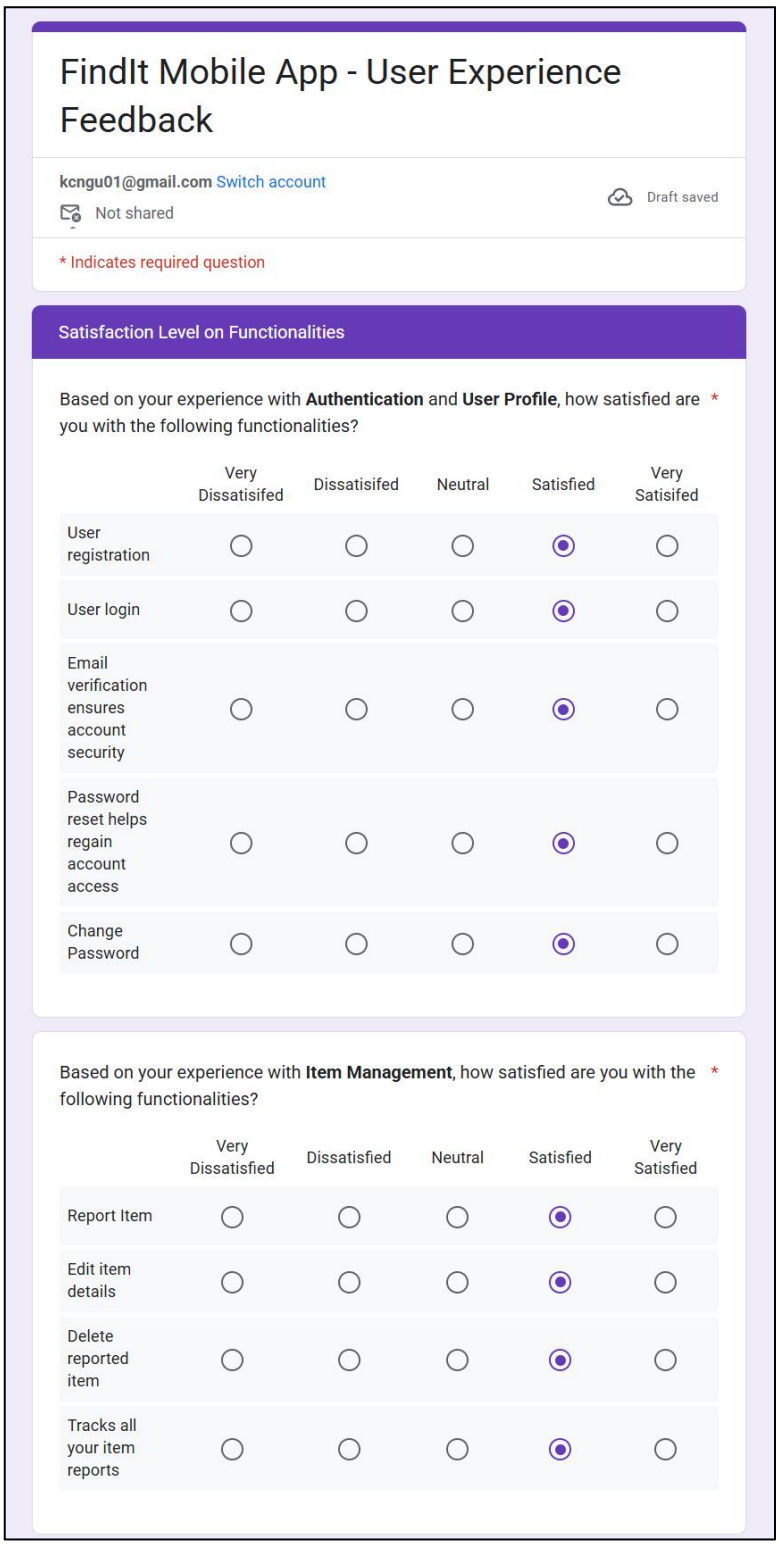
Back Next Clear form

This content is neither created nor endorsed by Google. - Contact form owner - Terms of Service - Privacy Policy

Does this form look suspicious? Report

Google Forms

*Figure C.2 Student's Satisfaction Level on User Interface*



FindIt Mobile App - User Experience Feedback

kcngu01@gmail.com Switch account

Not shared

Draft saved

* Indicates required question

Satisfaction Level on Functionalities

Based on your experience with **Authentication** and **User Profile**, how satisfied are you with the following functionalities? *

| | Very Dissatisifed | Dissatisifed | Neutral | Satisfied | Very Satisifed |
|---|---|---|---|---|---|
| User registration | ○ | ○ | ○ | ◉ | ○ |
| User login | ○ | ○ | ○ | ◉ | ○ |
| Email verification ensures account security | ○ | ○ | ○ | ◉ | ○ |
| Password reset helps regain account access | ○ | ○ | ○ | ◉ | ○ |
| Change Password | ○ | ○ | ○ | ◉ | ○ |

Based on your experience with **Item Management**, how satisfied are you with the following functionalities? *

| | Very Dissatisified | Dissatisified | Neutral | Satisfied | Very Satisified |
|---|---|---|---|---|---|
| Report Item | ○ | ○ | ○ | ◉ | ○ |
| Edit item details | ○ | ○ | ○ | ◉ | ○ |
| Delete reported item | ○ | ○ | ○ | ◉ | ○ |
| Tracks all your item reports | ○ | ○ | ○ | ◉ | ○ |

*Figure C.3 Student's Satisfaction Level on Functionalities*

Based on your experience with **Search and Discovery**, how satisfied are you with the following functionalities? *

| | Very Dissatisfied | Dissatisfied | Neutral | Satisfied | Very Satisfied |
|---|---|---|---|---|---|
| Search items by keywords | ○ | ○ | ○ | ◉ | ○ |
| Filter items by category/colour/location | ○ | ○ | ○ | ◉ | ○ |
| Browse items based on item types | ○ | ○ | ○ | ◉ | ○ |
| View item details | ○ | ○ | ○ | ◉ | ○ |

Based on your experience with **Claim Management**, how satisfied are you with the following functionalities? *

| | Very Dissatisfied | Dissatisfied | Neutral | Satisfied | Very Satisfied |
|---|---|---|---|---|---|
| Submit claim for found items | ○ | ○ | ○ | ◉ | ○ |
| View potential matches for lost items | ○ | ○ | ○ | ◉ | ○ |
| Track claim request status | ○ | ○ | ○ | ◉ | ○ |

Based on your experience with **Notifications**, how satisfied are you with the following functionalities? *

| | Very Dissatisfied | Dissatisfied | Neutral | Satisfied | Very Satisfied |
|---|---|---|---|---|---|
| Receive notifications alert about potential item matches | ○ | ○ | ○ | ◉ | ○ |
| Receive claim update notifications | ○ | ○ | ○ | ◉ | ○ |
| Mark notifications as read | ○ | ○ | ○ | ◉ | ○ |
| View notification history | ○ | ○ | ○ | ◉ | ○ |

Back Submit Clear form

This content is neither created nor endorsed by Google. - Contact form owner - Terms of Service - Privacy Policy

Does this form look suspicious? Report

Google Forms

*Figure C.4 Student's Satisfaction Level on Functionalities*

# FindIt Web-based Administrative Interface: User Experience Feedback

Greetings to everyone! I am Ngu Keh Cong (80369), a final year student of Bachelor of Computer Science with Honours (Software Engineering) at UNIMAS.

This questionnaire aims to evaluate the usability of the FindIt web-based administrative interface system. Before proceeding, please visit FindIt and explore the system to familiarize yourself with its features and functionality. As a **UNIMAS administrative staff member**, you are our primary target user for this platform.

The questionnaire is divided into 2 sections and will take about 5–10 minutes to complete. They are:
**Section I:** Satisfaction Level on User Interfaces

**Section II**: Satisfaction Level on Functionalities

Please note:

- Your responses will be kept confidential and used solely for this project.
- Answer based on your experience during the testing session.
- There are no correct or incorrect answers; we want to hear your honest opinions and experiences.

Thank you for your time and effort taking part in this questionnaire. Your valuable input will play a significant role in shaping a centralized platform to manage the lost and found items in UNIMAS. If you have any questions, or would like clarification on anything, please contact me via email (80369@siswa.unimas.my).

**kcngu01@gmail.com** Switch account

Not shared

Next

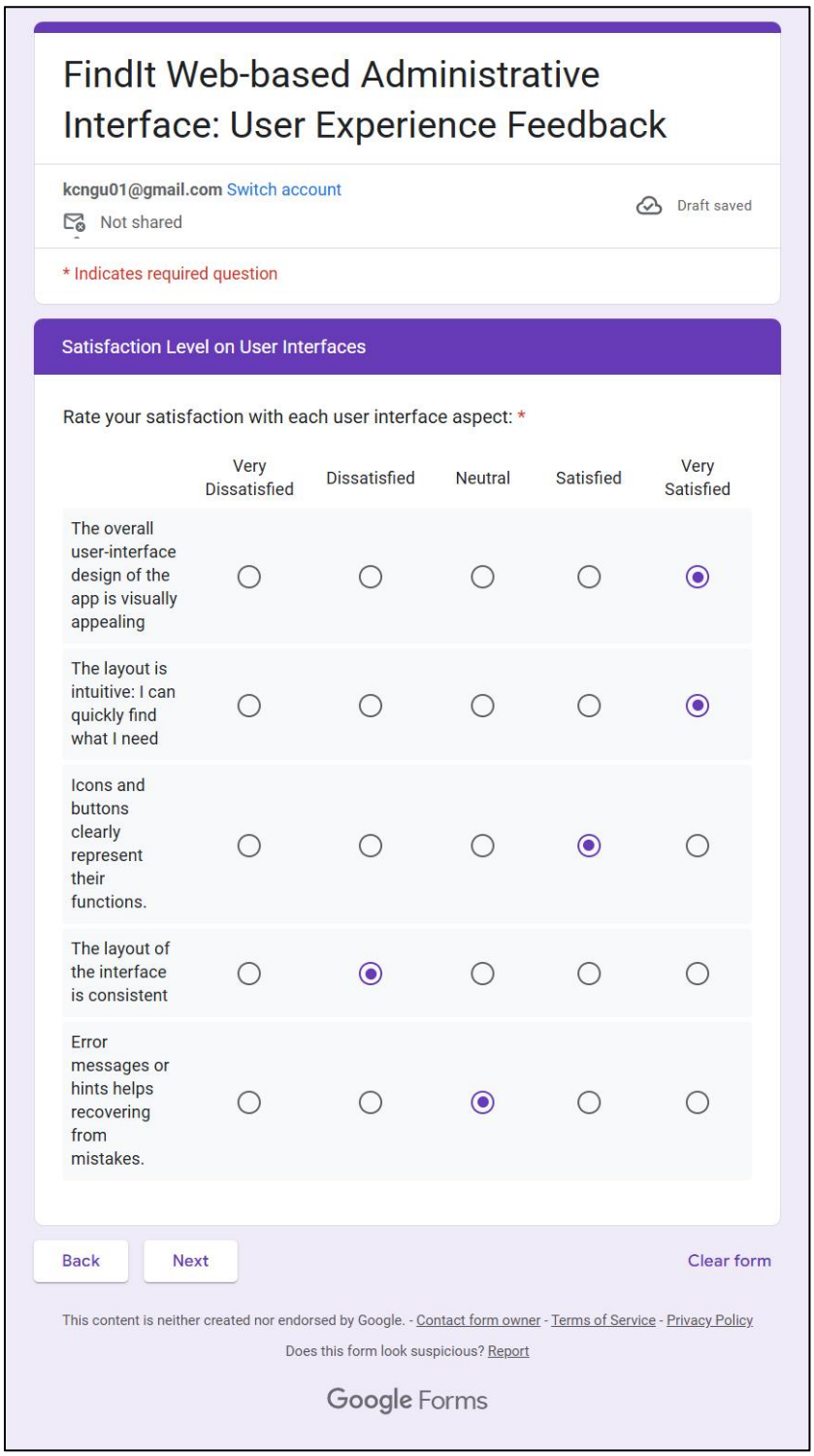
Clear form

This content is neither created nor endorsed by Google. - Contact form owner - Terms of Service - Privacy Policy

Does this form look suspicious? Report

Google Forms

*Figure C.5 FindIt Admin Interface User Experience Survey Form Description*



FindIt Web-based Administrative Interface: User Experience Feedback

kcngu01@gmail.com Switch account

Not shared

Draft saved

* Indicates required question

Satisfaction Level on User Interfaces

Rate your satisfaction with each user interface aspect: *

| | Very Dissatisfied | Dissatisfied | Neutral | Satisfied | Very Satisfied |
|---|---|---|---|---|---|
| The overall user-interface design of the app is visually appealing | ○ | ○ | ○ | ○ | ◉ |
| The layout is intuitive: I can quickly find what I need | ○ | ○ | ○ | ○ | ◉ |
| Icons and buttons clearly represent their functions. | ○ | ○ | ○ | ◉ | ○ |
| The layout of the interface is consistent | ○ | ◉ | ○ | ○ | ○ |
| Error messages or hints helps recovering from mistakes. | ○ | ○ | ◉ | ○ | ○ |

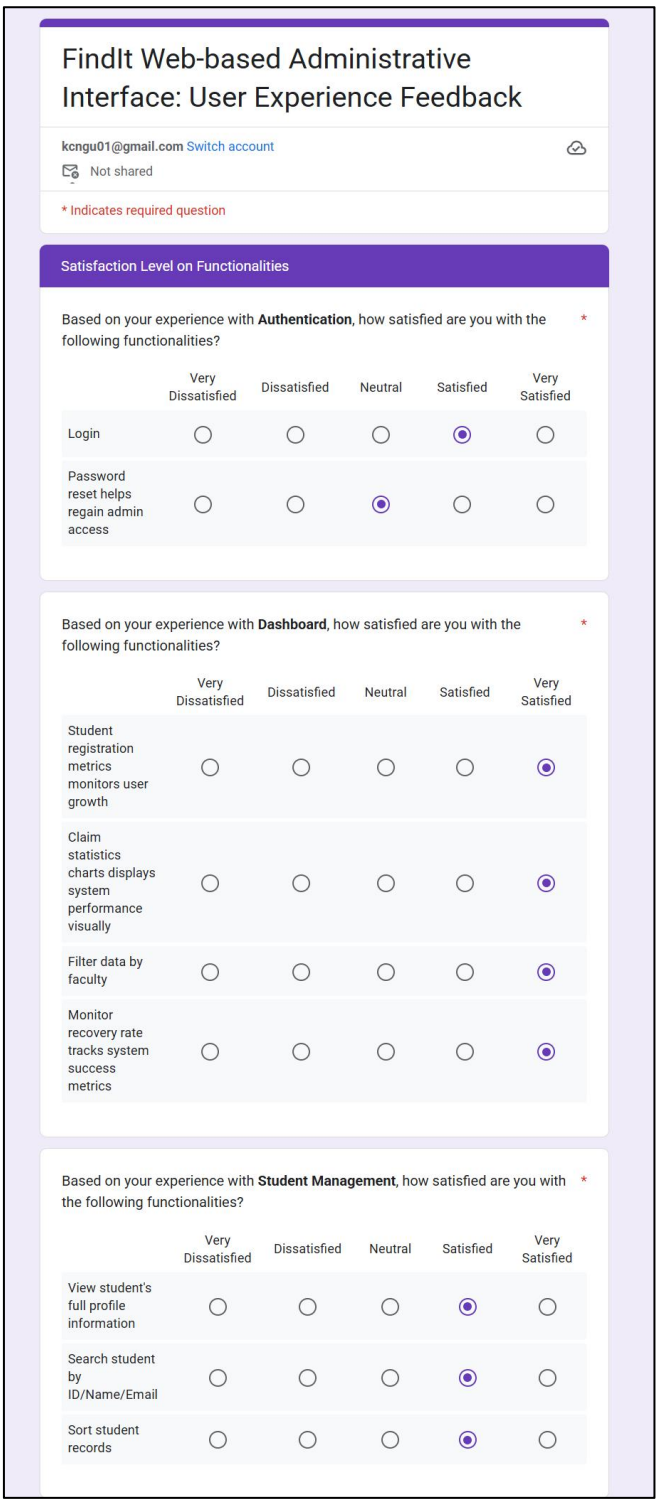
Back Next Clear form

This content is neither created nor endorsed by Google. - Contact form owner - Terms of Service - Privacy Policy

Does this form look suspicious? Report

Google Forms

*Figure C.6 Admin's Satisfaction Level on User Interface*



FindIt Web-based Administrative Interface: User Experience Feedback

kcngu01@gmail.com Switch account

Not shared

* Indicates required question

Satisfaction Level on Functionalities

Based on your experience with **Authentication**, how satisfied are you with the following functionalities? *

| | Very Dissatisfied | Dissatisfied | Neutral | Satisfied | Very Satisfied |
|---|---|---|---|---|---|
| Login | ○ | ○ | ○ | ◉ | ○ |
| Password reset helps regain admin access | ○ | ○ | ◉ | ○ | ○ |

Based on your experience with **Dashboard**, how satisfied are you with the following functionalities? *

| | Very Dissatisfied | Dissatisfied | Neutral | Satisfied | Very Satisfied |
|---|---|---|---|---|---|
| Student registration metrics monitors user growth | ○ | ○ | ○ | ○ | ◉ |
| Claim statistics charts displays system performance visually | ○ | ○ | ○ | ○ | ◉ |
| Filter data by faculty | ○ | ○ | ○ | ○ | ◉ |
| Monitor recovery rate tracks system success metrics | ○ | ○ | ○ | ○ | ◉ |

Based on your experience with **Student Management**, how satisfied are you with the following functionalities? *

| | Very Dissatisfied | Dissatisfied | Neutral | Satisfied | Very Satisfied |
|---|---|---|---|---|---|
| View student's full profile information | ○ | ○ | ○ | ◉ | ○ |
| Search student by ID/Name/Email | ○ | ○ | ○ | ◉ | ○ |
| Sort student records | ○ | ○ | ○ | ◉ | ○ |

*Figure C.7 Admin's Satisfaction Level on Functionalities*

Based on your experience with **Item Characteristics Management**, how satisfied are you with the following functionalities? *

| | Very Dissatisfied | Dissatisfied | Neutral | Satisfied | Very Satisfied |
|---|---|---|---|---|---|
| Add new categories/colours/locations | ○ | ○ | ○ | ◉ | ○ |
| Edit existing categories/colors/locations | ○ | ○ | ○ | ◉ | ○ |
| Delete existing categories/colours/locations | ○ | ○ | ○ | ◉ | ○ |
| Search/ sort item characteristics | ○ | ○ | ○ | ◉ | ○ |

Based on your experience with **Claim Processing**, how satisfied are you with the following functionalities? *

| | Very Dissatisfied | Dissatisfied | Neutral | Satisfied | Very Satisfied |
|---|---|---|---|---|---|
| View found Items with pending claims | ○ | ○ | ○ | ◉ | ○ |
| Compare lost and found items | ○ | ○ | ○ | ◉ | ○ |
| Approve claim requests | ○ | ○ | ○ | ◉ | ○ |
| Reject claim requests | ○ | ○ | ○ | ◉ | ○ |
| Filter claims by faculty | ○ | ○ | ○ | ◉ | ○ |

Based on your experience with **Claim History Management**, how satisfied are you with the following functionalities? *

| | Very Dissatisfied | Dissatisfied | Neutral | Satisfied | Very Satisfied |
|---|---|---|---|---|---|
| Tracks all processed claims | ○ | ○ | ○ | ◉ | ○ |
| Mark item as collected confirms successful item recovery | ○ | ○ | ○ | ◉ | ○ |
| Search/ filter processed claims | ○ | ○ | ○ | ◉ | ○ |
| View detailed claim information | ○ | ○ | ○ | ◉ | ○ |

Back Submit Clear form

This content is neither created nor endorsed by Google. - Contact form owner - Terms of Service - Privacy Policy

Does this form look suspicious? Report

Google Forms

*Figure C.8 Admin's Satisfaction Level on Functionalities*