



FINAL REPORT

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

**SearchCare: Patient Smart Mobile Application for Personalized Healthcare
Recommendations**

Maximus Sebastian Bateh Anak Philip Marim

80003

Bachelor's in Computer Science with Honours

(Information Systems)

2025

Abstract

With the rapid growth of mobile technology, healthcare delivery has undergone a significant transformation, enabling new approaches to patient monitoring, medical data management, and personalized care. This project introduces SearchCare, a smart healthcare mobile application developed to serve as a centralized platform that integrates symptom checking, doctor consultation booking, medicine purchasing, and lifestyle health management. Designed for both urban and rural populations, the app leverages cloud infrastructure and artificial intelligence to provide real-time access to verified medical services. Key modules such as role-based dashboards for patients, doctors, and administrators, an AI-powered symptom-disease prediction system, and Firebase-backed authentication and data storage have been successfully implemented. The system also supports secure medical record handling, geolocation-based provider search, and in-app notifications. More than 80% of users rated the SearchCare app's features as simple to use and pertinent to their healthcare needs, according to the system review, which also reveals that most users regarded the software to be visually appealing, effective, and user-friendly. High levels of satisfaction with the app's usability, functionality, and design suggest that users have a satisfying and enjoyable experience. With these completed features, SearchCare demonstrates strong potential in improving healthcare accessibility, enhancing user experience, and contributing to the digital transformation of Malaysia's health sector.

Keywords: Smart healthcare, AI diagnosis, Mobile health application, Centralized platform, Cloud-based medical system

Acknowledgment

I am sincerely grateful to all individuals who have contributed to the successful completion of my final year project. First and foremost, I would like to express my profound gratitude to my project supervisor, Puan Hamizan binti Sharbini, for their invaluable guidance, insights, supports and encouragement throughout my final year project journey. Their insightful feedback and continuous motivation were instrumental in overcoming various challenges and achieving the project's objectives. I also wish my gratitude to all faculty members and staff at Universiti Malaysia Sarawak (UNIMAS), whose guidance and assistance have been a significant source of learning and inspiration.

A heartfelt thanks to my family and friends for their unwavering support and understanding during this project. Their emotional support provided me with their strength and faith to persevere. Finally, I acknowledge the importance of the resources provided by the university, including access to the library archive, which played a crucial role in the project's success. This achievement would not have been possible without the collective effort and support of everyone mentioned above. Thank you everyone and may God bless us all.

Table of Contents

Abstract.....	i
Acknowledgment.....	ii
Chapter 1 INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement.....	2
1.3 Aim and Objectives	2
1.4 Scope.....	3
1.5 Brief Methodology	4
1.6 Significance of Project	6
1.7 Project Schedule.....	6
1.8 Expected Outcome	7
1.9 Project Report and Outline.....	7
1.9.1 Chapter 1: Introduction	7
1.9.2 Chapter 2: Literature Review	8
1.9.3 Chapter 3: Research Methodology	8
1.9.4 Chapter 4: Implementation and Design.....	8
1.9.5 Chapter 5: Test Case and User Testing	8
1.9.6 Chapter 6: Conclusion and Future Work.....	8
1.10 Conclusion	9
Chapter 2 Literature Review	10
2.1 Introduction.....	10
2.2 Review of Smart Healthcare Applications.....	11
2.2.1 BookDoc.....	11
2.2.2 DoctorOnCall	13
2.2.3 MediQ.....	15
2.3 Platform used for Smart Healthcare Applications.....	18
2.4 Software and Programming Language.....	18
2.5 Conclusion	18
Chapter 3 Research Methodology	19
3.1 Introduction.....	19
3.2 Methodology	19

3.3 System design	21
3.3.1 Context Diagram	26
3.3.2 Data Flow Diagram (DFD level – 0)	27
3.3.3 Data Flow Diagram (DFD level – 1)	28
3.3.4 Entity Relationship Diagram (ERD)	33
3.4 Wireframe	37
3.4.1 Splash Screen	37
3.4.2 Log In User Screen	38
3.4.3 Sign Up Register Form Screen	39
3.4.5 Homepage	40
3.4.6 Search Symptoms Page	41
3.5 Information Gathering	42
3.6 Software Requirement	49
3.7 Development	51
3.7.1 Project Flow Using Dart in Flutter	52
3.7.2 Iterative Development Process	52
3.8 Testing	53
3.8.1 Testing Methodology	53
3.9 Documentation	54
3.10 Software and Hardware Requirement	54
3.10.1 Software Requirement	54
3.10.2 Hardware Requirement	55
Chapter 4 Implementation and Design	56
4.1 Introduction	56
4.2 Required Component	56
4.2.1 Android Studio	57
4.2.2 Firebase Storage as a Database	58
4.2.3 Flutter as an open-source UI toolkit	59
4.3 System Flow	60
4.3.1 System Flow for User/Patient	60
4.3.2 System Flow for Doctor	61
4.4 Module Implementation	62

4.4.1 Splash Screen	62
4.4.2 Log In Screen	63
4.4.3 Register Account Type	64
4.4.4 New User/Patient Account Registration	65
4.4.5 New Doctor Account Registration	66
4.4.6 User Dashboard	67
4.4.7 User Profile Setting	68
4.4.8 Buy Medicine Page	69
4.4.9 View Medicine Detail Page	70
4.4.10 Confirm Medicine Payment Page	71
4.4.11 Medicine Payment Confirmation Details	72
4.4.12 Medicine Orders History Page	73
4.4.13 Receipt Summary Details Page	74
4.4.14 Cart Screen	75
4.4.15 Confirm Multiple Medicine Payment Page	76
4.4.16 Multiple Medicine Payment Confirmation Details	77
4.4.17 Search Symptoms Page	78
4.4.18 Book Appointment Page	79
4.4.19 Book Appointment Details Page	80
4.4.20 Appointment Status Page	81
4.4.21 Book Consultations Page	82
4.4.19 Book Consultations Details Page	83
4.4.20 Consultations Status Page	84
4.4.21 View Healthy Tips Page	85
4.4.22 View Healthy Food Tips Page	86
4.4.23 View Healthy Exercise Tips Page	87
4.4.24 View Healthy Facts Tips Page	88
4.4.25 Doctor Dashboard	89
4.4.26 Doctor Setting Page	90
4.4.27 View Requested Appointment	91
4.4.28 Update Appointment Status Confirmation Page	92
4.4.29 View Requested Consultations	93

4.4.30 Consultations Status Confirmation Page.....	94
4.4.31 Admin Login Page.....	95
4.4.32 Admin Dashboard Page.....	96
4.4.33 Add Medicine Page	97
4.4.34 View Medicine Page	98
4.4.35 Update Medicine Details Page	99
4.4.36 Update Healthy Tips Details Page	100
4.4.37 Update Healthy Food Details Page.....	101
4.4.38 Update Healthy Exercise Details Page	102
4.4.39 Update Healthy Facts Details Page	103
4.4.40 Search Symptoms.....	104
4.4.41 Update Symptoms	105
4.4.42 Update Location View.....	106
4.4.43 Location View	107
4.5 Conclusion	108
Chapter 5: Test Case and User Testing	109
5.1 System Checking	109
5.1.1 Test Case	109
5.2 Usability Testing	131
5.2.1 Introduction.....	131
5.2.2 Test Objectives.....	132
5.2.3 Test Participants.....	132
5.2.4 Test Environment.....	132
5.2.5 Task Scenarios	132
5.3 System Evaluation.....	134
5.4 Conclusion	136
Chapter 6: Conclusion and Future Work	138
6.1 Introduction.....	138
6.2 Objective Achievement	138
6.3 Application Constraint	139
6.4 Project Limitation	140
6.5 Future Work	140

6.6 Recommendation	141
6.7 Conclusion	142
Reference	1
Appendices	3

List of Tables

Table 1 Project Schedule Table	6
Table 2 Comparison between application of mobile smart healthcare systems and proposed system: SearchCare	16
Table 3 The phase, activities, task and deliverables table	25
Table 4 Software Requirement	49
Table 5 Development Requirement	52
Table 6 Testing Requirement	53
Table 7 Software Requirements	54
Table 8 Hardware Requirement	55
Table 9 Objective's Achievement	138

List of Figures

Figure 1 Interface of BookDoc	11
Figure 2 Interface of DoctorOnCall	13
Figure 3 Interface of MediQ	15
Figure 4 Waterfall Methodology photo by: Lean, agile, or waterfall? Choose the best product management methodology based on our examples. (2025, April 24). https://www.eleken.co/blog-posts/a-comparative-guide-to-lean-agile-and-waterfal-product-management-methodo	20
Figure 5 The flowchart process user/patient side workflow	22
Figure 6 The flowchart process doctor side workflow	23
Figure 7 The flowchart process admin side workflow.....	24
Figure 8 Context Diagram for SearchCare	26

Figure 9 Level 0 of SearchCare: Patient Smart Mobile Application for Personalized Healthcare Recommendations.....	27
Figure 10 DFD Level 1 for user login process.....	28
Figure 11 DFD Level 1 for register user details process	28
Figure 12 DFD Level 1 for manage profile process	29
Figure 13 DFD Level 1 for set appointment process	29
Figure 14 DFD Level 1 for search and filter symptoms process	30
Figure 15 DFD Level 1 for view healthy life tips process.....	31
Figure 16 DFD Level 1 for pay medicine and service process	31
Figure 17 DFD Level 1 for Search Location process	32
Figure 18 ERD of Entity Patient	33
Figure 19 ERD of Entity Doctor	34
Figure 20 ERD of Entity Emergency Contact	34
Figure 21 ERD of Entity Appointment.....	35
Figure 22 ERD of Entity Consultation.....	35
Figure 23 ERD of Entity Medication.....	36
Figure 24 ERD of Entity Delivery	36
Figure 25 Splash Screen for SearchCare.....	37
Figure 26 Log In screen for SearchCare apps.....	38
Figure 27 Sign Up screen for SearchCare apps	39
Figure 28 Homepage for SearchCare apps	40
Figure 29 Search Symptom page.....	41
Figure 30 Information gathering phase	42
Figure 31 Range age of respondents' pie chart.....	43
Figure 32 Healthcare center recommendation pie chart	44
Figure 33 Method of obtaining healthcare pie chart	44
Figure 34 Online consultation and online diagnosis pie chart	45
Figure 35 Online appointment preference pie chart.....	46
Figure 36 Online medical prescription pie chart.....	46
Figure 37 Preferable method to get medicine pie chart	47
Figure 38 Healthcare notification pie chart.....	47
Figure 39 Daily diet notification pie chart	48
Figure 40 Healthcare tips category pie chart	48

Figure 41 Android Studio workspace	57
Figure 42 Firebase database workspace	58
Figure 43 Installed Flutter	59
Figure 44 App Navigation of the mobile application for user/patient	60
Figure 45 App Navigation of the mobile application for doctor	61
Figure 46 Splash screen	62
Figure 47 Login screen	63
Figure 48 Register Account Type	64
Figure 49 New User/Patient Registration	65
Figure 50 New Doctor Registration	66
Figure 51 User/Patient Dashboard	67
Figure 52 User Profile Setting Page	68
Figure 53 Buy Medicine Page	69
Figure 54 View Medicine Detail Page	70
Figure 55 Confirm Medicine Payment Page	71
Figure 56 Medicine Payment Confirmation Details	72
Figure 57 Medicine Orders History Page	73
Figure 58 Receipt Summary Details Page	74
Figure 59 Cart Screen	75
Figure 60 Confirm Multiple Medicine Payment Page	76
Figure 61 Multiple Medicine Payment Confirmation Details	77
Figure 62 Search Symptoms Page	78
Figure 63 Book Appointment Page	79
Figure 64 Book Appointment Details Page	80
Figure 65 Appointment Status Page	81
Figure 66 Book Consultations Page	82
Figure 67 Book Consultations Details Page	83
Figure 68 Consultations Status Page	84
Figure 69 View Healthy Tips Page	85
Figure 70 View Healthy Food Tips Page	86
Figure 71 View Healthy Exercise Tips Page	87
Figure 72 View Healthy Facts Tips Page	88
Figure 73 Doctor Dashboard Page	89

Figure 74 Doctor Setting Page	90
Figure 75 View Requested Appointment Page	91
Figure 76 Update Appointment Status Confirmation Page	92
Figure 77 View Requested Consultations Page	93
Figure 78 Consultations Status Confirmation Page	94
Figure 79 Admin Login Page.....	95
Figure 80 Admin Dashboard Page	96
Figure 81 Add Medicine Page	97
Figure 82 View Medicine Page.....	98
Figure 83 Update Medicine Details Page	99
Figure 84 Update Healthy Tips Details Page.....	100
Figure 85 Update Healthy Food Details Page.....	101
Figure 86 Update Healthy Exercise Details Page	102
Figure 87 Update Healthy Facts Details Page	103
Figure 88 Search Symptoms Page	104
Figure 89 Update Symptoms Page.....	105
Figure 90 Update Location Page.....	106
Figure 91 Location View	107
Figure 92 Multi Bar Chart of Evaluation on System's Features	134
Figure 93 Multi Bar Chart of Evaluation on Users' Comments	135

Chapter 1 INTRODUCTION

1.1 Background

With developments like telemedicine, electronic health records, and remote monitoring enhancing access and care quality, information and communication technologies (ICTs) are essential to the transformation of healthcare. Among the many issues facing healthcare systems are an ageing population, growing expenses, and a growing need for high-quality care (Wickramasinghe, 2008; Kumar et al., 2024). Early in 2020, the world experienced one of the worst pandemics, COVID-19, which infected more than 250 million people worldwide. To address this global crisis, scientists and technologists from a variety of disciplines banded together and aided. During the pandemic, Language Technology (LT), in particular conversational agents, also known as chatbots, were essential because they allowed for early disease screening (Judson et al., 2020a; Martin et al., 2020b) and accurate information dissemination (Li et al., 2020; Maniou and Veglis, 2020). However, only a few languages currently offer practically helpful chatbots and other LT advantages (Joshi et al., 2020).

Intelligent healthcare recommendation systems use data fusion and machine learning methods to forecast illnesses and offer tailored health recommendations. According to one study, these systems can predict diabetes using ensemble learning with 99.6% accuracy (Baha Ihnaini et al., 2021). In the analysis of healthcare datasets, random forest models have demonstrated encouraging outcomes (Richa Sharma & Shalli Rani, 2020). Recommendation systems in smart city settings can leverage citizen-contributed data and networked devices to enhance healthcare services (Fran Casino et al., 2018). Semantic information extraction and information overload are two issues that web-based healthcare recommendation systems must deal with. For efficient web page recommendations in healthcare contexts, a combination of Bat Agent Optimisation and Weighted Support Vector Machines has been proposed as a solution to these problems (S. Anusuya & N. Mallika, 2022). These developments in intelligent healthcare systems are intended to improve patient quality of life, personalised recommendations, and disease prediction.

While existing platforms such as BookDoc and DoctorOnCall make most of the search for doctors and booking appointments easier for locals in Malaysia, these systems themselves lack intelligent recommendations based on symptoms and real-time checks on availability. Besides, in most cases, they fail to offer optimized results with respect to

location regarding nearby providers. This implies the need for an advanced solution that can make personalized recommendations on healthcare, considering the patient's symptoms, proximity, specialist expertise, and appointment availability.

The primary focus of this project is on the development of a Smart Healthcare Recommendation System, which is a web-based application to identify and recommend the best possible healthcare provider using data-driven algorithms based on the specific need and condition of the patients. This system, therefore, makes the procedure of the matching of providers and patients more efficient with such technologies as machine learning or location-based services, enhances scheduling for appointments, and boosts the patient's overall satisfaction. The far-reaching goal of the venture is to make healthcare more accessible, efficient, and personalized. It should be ensured that, for improved health outcomes and overall satisfaction, patients should receive well-cared services from appropriate specialist.

1.2 Problem Statement

Patients often encounter difficulties when searching for and booking appointments with nearby specialists, which can delay access to necessary medical care. This is because some patients would have difficulties to access transportation even though public transportation is available. Although platforms like BookDoc and DoctorOnCall provide basic features for finding doctors and scheduling appointments, they lack essential functionalities such as real-time availability updates, symptom-based navigation, and advanced filtering options. These shortcomings make it harder for patients to efficiently match with the most appropriate specialist, resulting in a less effective and user-friendly experience.

1.3 Aim and Objectives

The primary objective of this project is to develop a comprehensive mobile application that serves as an essential tool for improving healthcare access and delivery. This application will empower patients by providing a streamlined platform to identify, evaluate, and connect with the most suitable healthcare specialists based on their symptoms, medical history, and doctor expertise. By acting as a central hub, the app will enable patients to effortlessly search for, view, and book appointments with nearby specialists, simplifying and speeding up seeking medical care. This enhanced accessibility will significantly benefit patients, helping them receive timely and effective treatment while reducing unnecessary delays.

Additionally, healthcare providers will be equipped with robust tools to manage their profiles, update availability, and oversee patient appointment request, ensuring the platform remains organized and responsive. The user interface will be designed with both functionality and ease of use in mind, facilitating a seamless experience for all users. Key features of the app will include real-time notifications to keep patients informed about appointment confirmations, reminders, and updates on their healthcare journey. This notification system will ensure that patients stay engaged and never miss critical updates regarding their care.

Furthermore, the application will foster a sense of trust and collaboration within the healthcare ecosystem by enabling better communication and interaction between patients and healthcare providers. By bridging the gap between patients in need of care and specialists with relevant expertise, the app will play a pivotal role in enhancing healthcare accessibility, efficiency, and quality. Through this innovative solution, the project aims to create a sustainable and patient-centred healthcare system that benefits both patients and providers.

The objectives of this project are:

- 1) To gather relevant data, including patient's demographics, medical history, symptoms, and preferences, as well as healthcare provider data such as specialities, rating, availability and location.
- 2) To develop a recommendation algorithm that matches patients with their search and filter inputted symptoms and fetch from the database.
- 3) To integrate the recommendation system with mobile applications for user interaction.

1.4 Scope

The application of this project is defined by the following key functionalities and structure characteristics and limitations.

The application will cater to three primary user categories, each with distinct access levels and functionalities. Patients will be able to create profiles, input symptoms, search for healthcare providers, and book appointments with recommended specialists. Healthcare providers can manage their profiles, update availability, and review appointment requests. The admin will oversee the platform, manage healthcare provider profiles and patient activities, with the authority to approve or reject provider listings and ensure compliance with platform policies.

The functional scope of the app includes various essential features, starting with a secure authentication and authorization system using email and password, complemented by role-based access control to ensure patients, healthcare providers, and administrators have appropriate access levels. Patients can input symptoms to receive tailored recommendations for specialists, filtered by proximity, expertise, and availability. They can book appointments directly through the app, and the system will track the status of bookings for both patients and providers. Healthcare providers can update their availability, manage appointment requests, and respond to patient inquiries. Push notifications will keep users informed about appointment confirmations, reminders, and important updates. Finally, the app will store user profiles, healthcare provider information, and appointment records in a database with robust privacy controls to ensure data security.

The technical scope of the app encompasses several key components. The initial development will focus on the web platform, with plans for future expansion to mobile platforms like Android and iOS. The frontend will prioritize a user-friendly interface with intuitive navigation for both patients and providers. The backend, implemented using Laravel (PHP), will handle patient symptom inputs, recommendation algorithms, provider profiles, and appointment management. A cloud-based MySQL database will securely store user data, including patient profiles, healthcare provider details, and appointment records. REST APIs will connect the frontend to the backend, ensuring seamless communication between components. Security measures will include encrypted authentication, secure storage of sensitive user data, and compliance with relevant healthcare data protection regulations (e.g., HIPAA, GDPR).

1.5 Brief Methodology

The development of this project will follow the Waterfall Methodology, as the project's requirements are well-defined and clearly outlined from the outset. The linear and organized nature of the Waterfall method ensures that each phase is systematically executed, allowing for efficient development and minimizing errors (Alda et al., 2023). The methodology involves sequential stages, including requirements, design, implementation, testing, and maintenance, providing a structured framework for delivering the Smart Healthcare Recommendation System.

First and foremost, the requirements phase. During this phase, the application's functional and technical requirements will be defined to ensure the system meets its

objectives of providing accurate specialist recommendations and efficient appointment scheduling. The key requirements include recommendation algorithm for matching patients with suitable specialists based on symptoms, medical history, and proximity. Next, location-based filtering for prioritizing nearby healthcare providers. Then, real-time appointment scheduling and notifications for patients and healthcare providers. Next, secure data storage for user profiles, healthcare provider information, and appointment records. This phase will also outline security measures, such as encrypted authentication and compliance with privacy regulations (e.g., HIPAA, GDPR).

Second, the design phase. This phase will focus on creating a clear and user-friendly interface for both patients and healthcare providers. The design will include a frontend layout optimized for easy symptom input, specialist search, and appointment booking. Then, a backend architecture for managing user profiles, specialist recommendations, and appointment data. Next, a database schema for securely storing user information, healthcare provider profiles, and appointment details. Lastly, the detailed workflows for patient symptom matching, location-based filtering, and appointment tracking.

Third, the implementation phase. During the implementation phase, each system component will be developed according to the design specifications. The main tasks include developing the recommendation algorithm and integrating it with the symptom input and filtering system. Next, building the real-time scheduling module to manage doctor availability and appointment booking. Hence, implementing the secure authentication system and role-based access control for patients, providers, and admins. Then, connecting the frontend and backend using REST APIs for smooth communication between components.

Subsequently, the testing phase. Comprehensive testing will be conducted to ensure all features function correctly and meet user expectations. Testing will cover the accuracy of the recommendation algorithm in matching patients with appropriate specialists. The testing also includes the functionality of location-based filtering and real-time scheduling. Usability of the user interface for both patients and healthcare providers will be tested for better usage. Lastly, testing the security measures to protect sensitive user data. Bugs and errors identified during testing will be resolved.

Finally, the maintenance phase. After the application is finalized, it will be published on the Google Play Store, making it accessible to patients and healthcare providers on Android devices. Plans for future expansion to iOS devices will also be outlined. After deployment, the application will undergo regular maintenance to address any bugs,

implement updates, and incorporate user feedback for continuous improvement. Ongoing support will ensure the system remains functional, secure, and aligned with user needs.

1.6 Significance of Project

The Smart Healthcare Recommendation System addresses a critical gap in healthcare access and delivery by leveraging technology to simplify and enhance the process of connecting patients with the right specialists. The significance of this project can be outlined in the few areas.

One of it is improving healthcare accessibility. Many patients face challenges in locating suitable healthcare specialists, particularly when they are unsure of which doctor to consult based on their symptoms. This system eliminates guesswork by recommending the most appropriate specialists, reducing the time and effort patients spend searching for care. By including location-based filtering, the platform ensures that patients can access nearby specialists, promoting timely medical intervention and reducing logistical barriers.

Next, educational and research value. For Universiti Malaysia Sarawak (UNIMAS), the project serves as a valuable case study in applying software engineering methodologies, machine learning, and database management in the healthcare domain. It offers opportunities for further research into algorithm optimization, user experience design, and healthcare data security, making it a significant contribution to the academic community.

1.7 Project Schedule

Table 1 Project Schedule Table

Task	Start Date	Finish Date	Duration
Submission of approved brief proposal	10 October 2024	20 October 2024	10 Days
Feedback and comment from reviewer/examiner	20 October 2024	28 October 2024	8 Days
Submission of full proposal	28 October 2024	14 November 2024	17 Days
Submission of Chapter 1	14 November 2024	21 November 2024	8 Days
Submission of Chapter 2	21 November 2024	13 December 2024	20 Days
Submission of Chapter 3	13 December 2024	5 January 2025	19 Days

Submission of FYP 1 final report & paper for assessment	5 January 2025	12 January 2025	16 Days
---	----------------	-----------------	---------

1.8 Expected Outcome

Timely access to medical care is important for proper diagnosis and treatment. But every day, patients struggle to find their nearest specialist and schedule an appointment, delaying care and thus often worsening their symptoms. Indeed, basic functions of doctor search and scheduling are also offered online by platforms such as BookDoc and DoctorOnCall but these systems provide limited functionality for real-time availability, symptom-based navigation, and efficient filtering to quickly match patients to the most appropriate specialist.

This will be an advanced mobile application designed to make the process of finding, assessing, and booking an appointment with a nearby specialist seamless. It will offer real-time scheduling, intelligent symptom-based specialist matching, and location-based filtering for easy access to health care services with reduced waiting times, to improve the experience for patients and further support better health outcomes.

1.9 Project Report and Outline

1.9.1 Chapter 1: Introduction

Introduction is the first chapter for this project, and it comprise of several parts. The first part is background where it explains a short overview of the project that explains why it is required and what problems it will discourse. Next, problem statement will outline the issues of the project with a clear and concise description of the current state and upcoming state and identify the interlude between the two. Then, aim and objective in this project explains is the aim is the overview of the purposes of this project while the objectives are the actions to achieve the aims of the project. Furthermore, project scope is a detailed framework of the project such as the workflow and assigned tasks. A brief methodology of this project is a structure that assists to manage the project by providing a detailed planning by roadmap and meticulous management. The significance of project explains the viability and the aftermath of the project and how it contributes to future research or project. Project schedule illustrates the duration of the completion of the tasks in this project and expected outcome is an expected result and achievement that is obtained from this project.

1.9.2 Chapter 2: Literature Review

Literature review consists of two parts. The comparison of the existing applications or systems or works that are related to this project is the first part of the review. This project will review three related existing systems which consists of the features, strengths and weaknesses of those systems. The complication can be overcome by improvement through analysis. The available tools to develop the mobile application to be used will be in the second part of the review.

1.9.3 Chapter 3: Research Methodology

Chapter 3 will explain a better understanding of the system. Readers will be able to gain better insight of the system and the way it operates. Waterfall Methodology will be discussed in this chapter for methodology. This chapter includes the procedure on collection of related data, the physical and logical design such as user interface design and data flow diagram (DFD).

1.9.4 Chapter 4: Implementation and Design

Chapter 4 provides the information of the development and implementation of this project. Explanation of the design will be described in this project. To ensure that the function in the project is operational, testing will be done, and each function will be elaborated. Some complications will occur during testing and to enhance the performance of the application, the error will be fixed.

1.9.5 Chapter 5: Test Case and User Testing

The procedure and findings of assessing the app's functionality, usability, and dependability should be described in the Test Case and User Testing section. An overview of the main test scenarios (test cases) that were carried out, such as data submission, login/logout, error handling, and particular feature flows, should be included, along with a comparison of the expected and actual results. This part should include discuss the user testing procedure, including how actual users interacted with the application, the feedback gathered, and any usability problems found. The objective is to show that the application satisfies its specifications, is easy to use, and functions effectively under typical usage circumstances.

1.9.6 Chapter 6: Conclusion and Future Work

Chapter 5 explains the summary and achievement of this project. Furthermore, new proposal or suggestions to further improve the application can be provided.

1.10 Conclusion

As a conclusion, this chapter introduces the significant details of this project where the background of this project provides the insight of healthcare development that can be tailored to the recent developments of AI and NLP. The problem statement of this project has been identified where some patients have difficulties to access medical attention due to the transportation issues. The aim of this project is to gather relevant data, including patient's demographic, medical history, symptoms and preferences which the system can develop a recommendation algorithm that matches patients with specialists, clinics, or treatment options based on their profiles and needs as one of the objectives. The scope of this project consists of three user categories- patients, admins and healthcare providers, functional scope which contains various essential features and lastly technical scope of the app encompasses several key components. Next, the brief methodology in this chapter explains the method that will be used which is Waterfall Methodology. Then, the significance of this project consists of three points which are improving healthcare accessibility, enhance the process of connecting patients with the right specialists and for educational and research value. Furthermore, the project schedule has been divided into several dates with their respective start date and end date. Finally, this project is expected to be an advanced mobile application designed to make the process of finding, assessing, and booking an appointment with a nearby specialist seamless which can improve the experience for patients and further support better health outcomes.

Chapter 2 Literature Review

2.1 Introduction

With great promise across a range of medical fields, natural language processing (NLP) has become an essential technique in smart healthcare applications (Zhou et al., 2021). NLP systems have the potential to be used in ophthalmology for therapy monitoring, risk assessment, and illness screening (Yang et al., 2021). With uses in text summarisation, clinical entity recognition, and medical note classification, the integration of natural language processing (NLP) with electronic health records (EHRs) has been thoroughly researched (Hossain et al., 2023). Data imbalance and the requirement for a more thorough model evaluation are two issues that still exist, but (Hossain et al., 2023).

Significant progress has been made in the healthcare industry because of the incorporation of technology, with platforms like BookDoc and DoctorOnCall facilitating convenient appointment scheduling for patients and provide health education through articles. However, these systems lack critical functionalities like real-time availability and symptom-based navigation. Although they confront integration issues, smart healthcare systems that integrate assistive frameworks, wearable technology, and machine learning for diagnosis show promise (Nasr et al., 2021). Concerns about security and privacy in healthcare data sharing may be addressed by federated learning in conjunction with AI and explainable AI (Rahman et al., 2022). While AI provides early disease detection and more effective therapies, edge computing technologies can lower latency and energy consumption in IoT-based healthcare systems (Hayyolalam et al., 2021). Researchers stress the need for standardised approaches to monitor therapeutic outcomes and strategies to promote sustained user engagement in mHealth apps in spite of these developments (Deniz-Garcia et al., 2022).

In this chapter, some smart healthcare applications that have been released in the medical field will be chosen to be reviewed. The reviewing of the applications provides some assistants to develop my project by giving some ideas and insight into the foundation of this project. Several analysis and comparisons of the application will guide me to have better understanding of the feature and design for the development of my project.

2.2 Review of Smart Healthcare Applications

This section will examine several contemporary smart healthcare applications to understand their features, functionality, and impact on healthcare delivery. From aiding with chronic illnesses to offering mental health support, these apps deploy technology like artificial intelligence, the Internet of Things, and big data to fulfill a range of medical needs. More specifically, we want to discuss their strengths and weaknesses in terms of their usability and design. It reviews how these applications comply with data privacy laws and how they can be integrated with existing healthcare systems. We also evaluate their suitability and efficacy in the context of enhancing patient outcomes. These are important insights for the design of smart healthcare applications in the future. They will eventually act as reference points for the design of digital health solutions. There are three smart healthcare applications that have been chosen to be reviewed. There are:

1. BookDoc
2. DoctorOnCall
3. MediQ

2.2.1 BookDoc

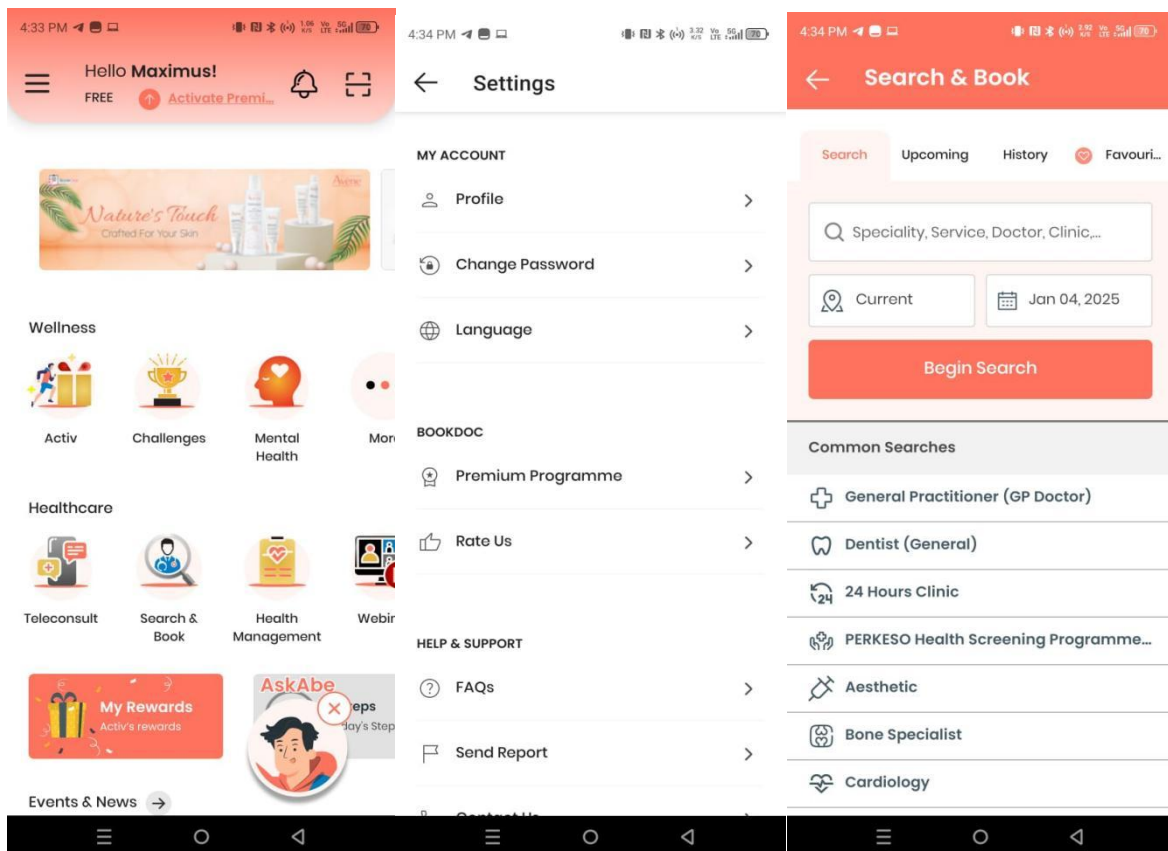


Figure 1 Interface of BookDoc

Figure 2.1 shows the screenshots of the BookDoc application interface. BookDoc has been listed as one of the top smart healthcare applications in Malaysia. Since its inception four years ago, BookDoc has diversified into more products and services. The goal of BookDoc is to be the ultimate health app-to have the solution on everyone's fingertips at anytime and anywhere in the world. There are seven primary features of BookDoc, designed to enhance health management and streamline user experience.

BookDoc offers a comprehensive suite of features designed to enhance health management and streamline the user experience. The application boasts a user-friendly interface that ensures ease of navigation for all users. Key functionalities include appointment booking for seamless access to healthcare professionals, health tracking to monitor personal wellness, and access to a wide range of healthcare services. Additionally, the app promotes a holistic lifestyle through its wellness programs, provides emergency assistance for urgent medical needs, and offers dedicated employee services to support workplace health initiatives.

One of the strengths of BookDoc is the intuitive, user-friendly design reduces the amount of learning required for end-users of differing ages and skill levels. It uses simple, clear designs to increase engagement and build retention. The appointment booking system will save a lot of time wasted in booking a visit to the doctor. Integrating clinic schedules and reminders enhances user convenience. Another strength is that this apps provide emergency assistance that would save many lives with quick access to emergency assistance, especially in urgent situations. It makes life easy for users a peace of mind for patients, especially those managing chronic conditions or attending to elderly family members. The apps also provide health tracking enables users to monitor metrics such as blood pressure, glucose levels, or activity logs and thus take responsibility for their health. Medical history access helps the patients and the doctors during consultations. The application also provides chatbot service to the user.

However, one of the weaknesses of this application is that it may lack personalized options for all types of users, which may include visually challenged or differently able people for whatever other reasons. Another weakness is the chatbot service is needed to be paid for the user to use the service.

2.2.2 DoctorOnCall

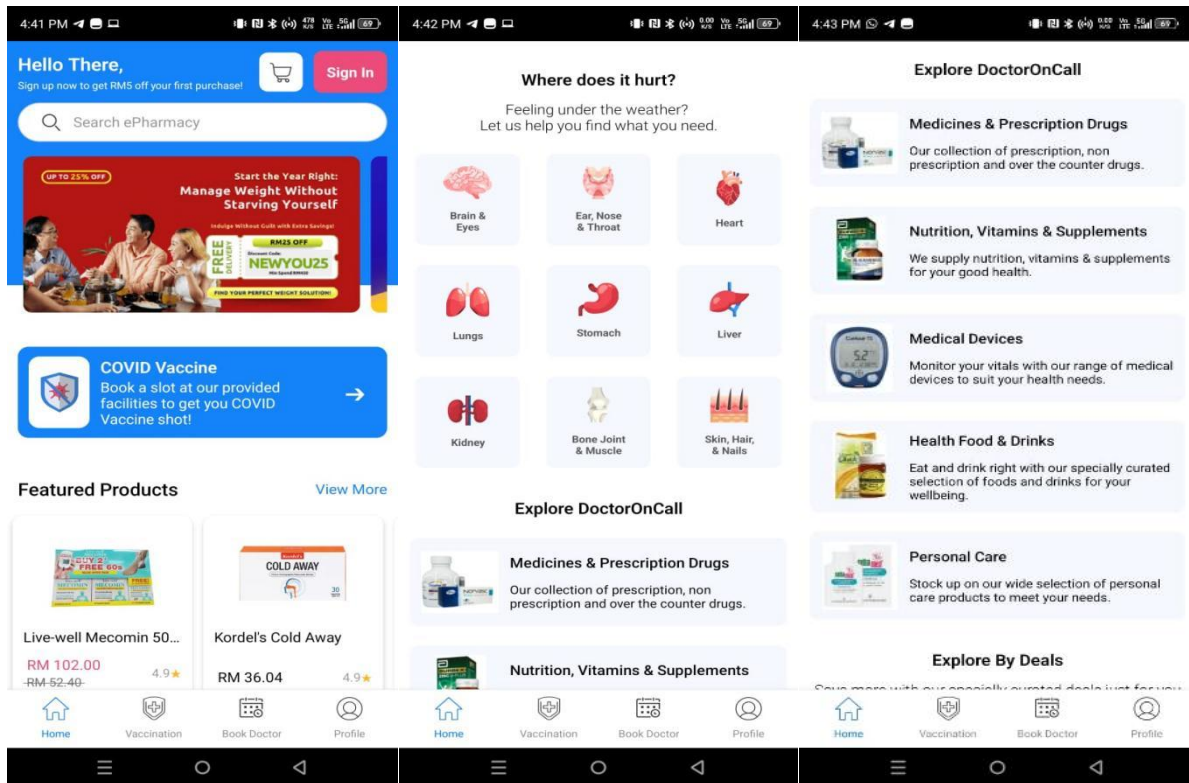


Figure 2 Interface of DoctorOnCall

Figure 2.2 shows the screenshots of the DoctorOnCall application interface. DoctorOnCall is one of the top listed smart healthcare in Malaysia. DoctorOnCall is a pioneering telehealth service, offering its mobile application users access to healthcare services with ease of access. Patients can consult licensed doctors, order medication, and even book health screenings from the comfort of their home. Its seamless interface with the assurance of reliability in the services provided has helped assure timely, convenient care for its patients.

The strength that can be elaborated from this app is DoctorOnCall provides universal access to healthcare. This is because DoctorOnCall is one of the most acknowledged telehealth platforms that enable patients, through mobile apps, to gain easy access to healthcare services. It allows patients to consult licensed doctors, order their medicines, and book health screenings from the convenience of their homes.

The DoctorOnCall application offers a range of features that support its goal of providing universal access to healthcare. These features include remote diagnostics that allow doctors to assess patients without physical visits, home-based services that deliver medical support directly to users' residences, home-based blood testing for convenient health monitoring, and online consultation services that connect patients with licensed doctors virtually. Together,

these features ensure that users receive timely and efficient healthcare from the comfort of their homes.

DoctorOnCall is one of the most acknowledged telehealth platforms that enable users, through mobile apps, to gain easy access to healthcare services. It allows users to consult licensed doctors, order their medicines, and book health screenings from the convenience of their homes. Its seamless interface combined with reliable services makes sure timely care is availed with much convenience to the user. Furthermore, this app possesses teleconsultation benefits which ensures that doctors and specialists are reached in minutes for timely treatments, something very essential during an emergency or for the treatment of chronic ailments. This feature decreases barriers to healthcare access arising from factors like distance, mobility, or transportation problems, besides saving lives with medical intervention at timely opportunities. Subsequently, DoctorOnCall enable easy and smooth integration between remote and physical consultations, thereby granting flexibility in either doctor or patients. For instance, being hybrid enables continuity of care, thereby guaranteeing high quality of care; thus, application obviates physical travel and congestions into the clinic but enables prescriptions and consultations in their own comforts to reach all end consumers.

The highly convenient service of DoctorOnCall at Home might suffer from limitations to its geography and logistics, such as not being available throughout a region, delayed scheduling, or shortage of staff, which might impact user satisfaction. Just as the doctor-patient interaction in a hybrid model can be seamless, reduced quality because of a lack of familiarity with technology might erode a critical feeling of personal touch that is integral to care. Moreover, home blood testing entirely depends on whether the sample was collected, transported, and analysed correctly, while any mistake or delay might affect test results and, correspondingly, either mislead doctors or disappoint consumers. Meeting such interrelated challenges is crucial for consistent positive patients' experiences.

2.2.3 MediQ

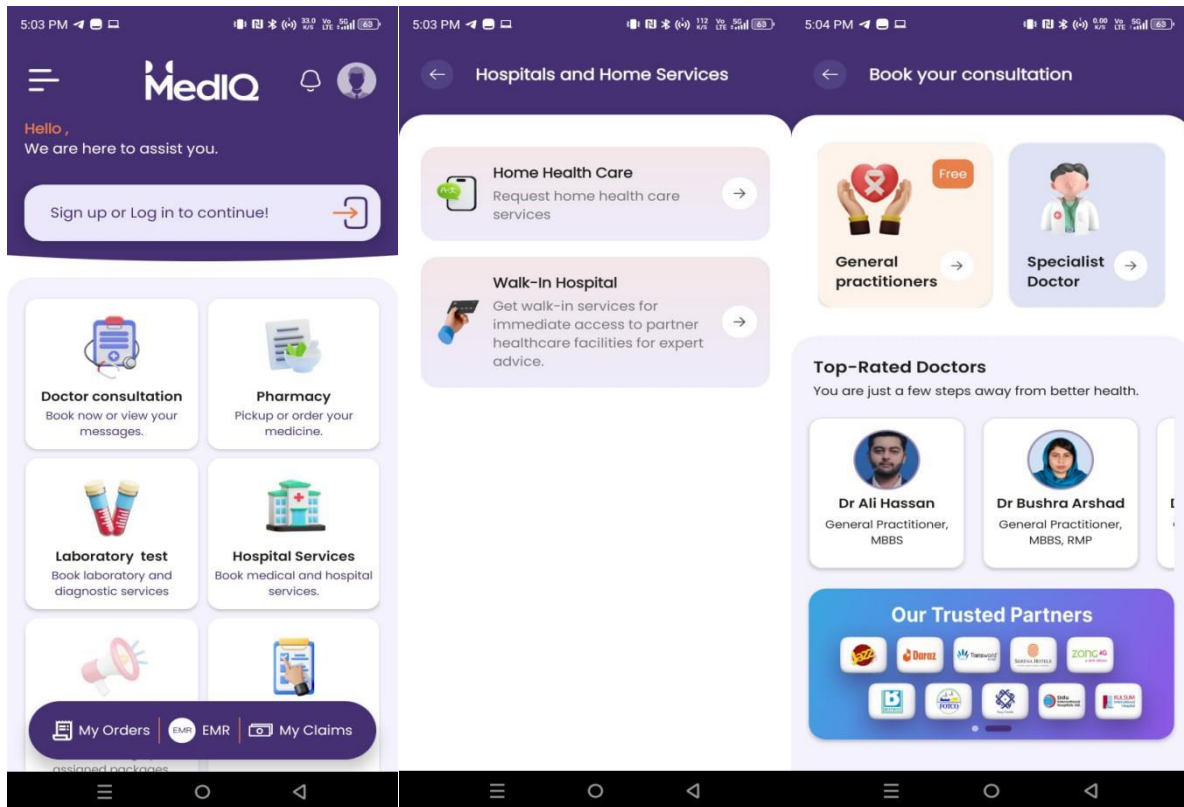


Figure 3 Interface of MediQ

Figure 2.3 shows the screenshots of the MediQ application interface. MediQ Smart Healthcare is Pakistan's First Digitally Integrated Hybrid Healthcare Platform, changing the dynamics of healthcare delivery through its physical panel and inclusive easy-to-use digital platform by offering convenience on-demand and cashless walk-ins. Equipped with a bouquet of integrated end-to-end technology platforms, integrated devices, and operation support services, the doctors MediQ strive for delivering compassionate care. MediQ are committed to offering healthcare that is accessible, affordable, and accountable, thus enabling unique experiences of our customers.

(<https://mediq.com.pk/about-us-best-corporate-healthcare-services/>)

There are few features that are presented in the MediQ application which are online booking system, automated notifications, online video consultation, access to service and e-prescription service. The strength that this app possesses through the app's features together enhances healthcare efficiency and patient satisfaction at key pain points within the care process. This can be displayed by the features of the appointment booking decreases the administrative workload, allowing patients the convenience of booking anytime, while proper time

management ensures workflows are optimized with shorter waiting times, hence increasing productivity and satisfaction. Timely automated notifications reminders help improve treatment adherence and continuity of care for better patient outcomes, while the clinic flow goes uninterrupted. Easy access to profiles which enables patients to access the services empowers patients to make informed decisions based on provider profile details, their availability, and reviews for the purposes of transparency and building trust. E-prescription digital health records allow for convenience and access in that medical information is kept safe for retrieval by both patients and providers when needed. Therefore, this makes coordination easier and increases the continuity of care. Features like this, all put together, contribute to streamlining and making the overall experience in healthcare more efficient and patient centred.

This application is flawless, save that video consultations wholly depend on stable internet connectivity, which to the disadvantaged areas, becomes an uphill task. On the same token, video consultations have a limited scope of diagnosis in instances where physical examinations or other diagnostic tests may be needed, conditions that virtual consultations cannot investigate.

2.2.4 Comparison between application of mobile smart healthcare systems and proposed systems

Table 2 Comparison between application of mobile smart healthcare systems and proposed system: SearchCare

Characteristic	BookDoc	DoctorOnCall	MedIQ	SearchCare
Tracking System	/	/	X	/
Feedback Response	/	/	/	/
Report Summary	/	/	/	/
Chatbot Service	/	X	X	/
Database Storage	/	/	/	/
GPS Capability	/	/	X	/
Colour Used	Light toned Red	Light blue	Purple	Light Blue

User Interface Design	X	/	X	/
Auto notifications	/	/	/	/
Appointment Tracking	/	/	X	/
Remote Consultation	X	/	/	/
Symptoms Search	X	/	X	/
Healthcare Tips And Wellness Program	/	/	/	/
Emergency Assistance Service	/	/	X	X
Delivery Service	X	/	X	/
Admin Privileges	/	/	/	/

From table 2.1, we can conclude that from four healthcare application system including the proposed system, there are slight differences between them. This is because each healthcare system is different from one another. For instance, BookDoc application does not provide remote consultation service which may be essential for patient that do not have access to transportation. BookDoc also does not provide delivery service. Application such as DoctorOnCall and SearchCare offers delivery services, making it easier for users to access medications or health products. The other function such as symptom checker is not implemented in the system. Application such as MediQ and DoctorOnCall does not provide chatbot service while BookDoc and SearchCare provided it so it may assist user/patient to surf for information in more flexible time as will not pack up the doctor schedule through consultation session. BookDoc and DoctorOnCall provide emergency service but MediQ and SearchCare does not provide it. Emergency service is encouraged in the smart healthcare

mobile application because nearest pharmacies, clinics or hospitals can provide emergency medical attention to user that request the service. In conclusion, each application has their own strength that have the same purpose. Some application does not certain unique features that may assist user or patient but it can be covered by implement the same features from different application.

2.3 Platform used for Smart Healthcare Applications

The cross-platform development applied in Flutter will be used. This consequently permits the manufacture of the application to be made for both Android and iOS, hence reducing development time and costs. This is important in health care because healthcare applications must be made available on various devices so that patients and healthcare professionals can access them easily. It will, in turn, mean that the deployment of Flutter offers a very innovative development phase for Smart Healthcare Applications. These will respond to a wide array of diversified patients' needs and healthcare service providers.

2.4 Software and Programming Language

Android Studio is very highly recommended by Grgurina et al (201) for the development of smart healthcare mobile applications. These applications can be designed using an Integrated Development Environment, Android Studio IDE that works corporately with the Android Development Tools, ADT plug-in and Android Software Development Kit SDK. There is no need to use an emulator from SDK instead; the test of the android application can be directly done on a physical device. Besides that, it can make the development of the application faster and straight forward. Thus, Android SDK has provided some useful material for developers such as extensive documentation, tutorials, samples, best practice guidance, and an array of tools for numerous development purposes.

2.5 Conclusion

As a conclusion, by analyzing the existing application can help the developer to search for the strengths and weaknesses. From the analysis, the GPS tracking feature, chatbot service, emergency assistance services, delivery services, appointments tracking and symptoms checkers are some very important features for a smart healthcare application in this era. The strengths and weaknesses of the existing application will be considered in mobile app development. The features include an attractive interface that will suit some vision impairment patients, multiple healthcare tip notifications and some animation to make the app more engaging.

Chapter 3 Research Methodology

3.1 Introduction

The system design for the project is developed within this chapter. It contains the methods and algorithm pseudocode for this project. In addition, this section provides a description of the entire project development in flowchart form. Different methodologies adopted and applied for the development of the test of a smart healthcare mobile application project are presented. The Waterfall methodology has been chosen from several software engineering process models. This would, henceforth, act as the methodology of approach in the research process of this project. The most characteristic features of this process model are its linearity and well-defined phases. Advantages for choosing this methodology: The sequential and linearity of the process model ensure that the project progresses in an organized manner. Also, detailed documentation is done at every stage, which acts as a reference for the team and stakeholders in minute detail. This chapter also mentions the software and hardware requirements along with the operating system environment for the project.

3.2 Methodology

Methodology is the implemented system method that is being used to construct the project according to the project objective need. There are many types of methodologies that have been used to develop a system or application. In this project, the methodology that is being used is Waterfall Methodology. This method being chosen because it is a suitable and perfect match to be used to develop this project, which is a mobile-based application.

Waterfall methodology could be used in the final year project of a smart healthcare mobile-based application, enabling the exact path in an organized way, working effectively with academic expectations. This kind of methodology ensures that the project will flow sequentially through distinct stages, which may comprise requirements analysis, design, development, testing, and finally deployment; hence, better planning is observed. Since most healthcare applications have specific and stable requirements, this model will enable addressing all the essential features, such as patient data management, appointment scheduling, and health tracking, without frequent changes. Also, the emphasis on documentation at each stage complements the need for a detailed project report, helping to present a clear workflow and maintain a professional standard.

Additionally, predictability and the organization of the Waterfall methodology are ideal in controlling time-bound academic projects. You focus on one phase at a time with the assurance of its thorough execution without the prospect of oversight. The sequential nature makes sure that adherence to regulations-for instance, data privacy standards-is considered beforehand in a project. This methodology will let you create a well-designed and tested application, while you will be able to show that you can follow an industry-standard methodology, which is important for your academic evaluation but also for future professional projects.

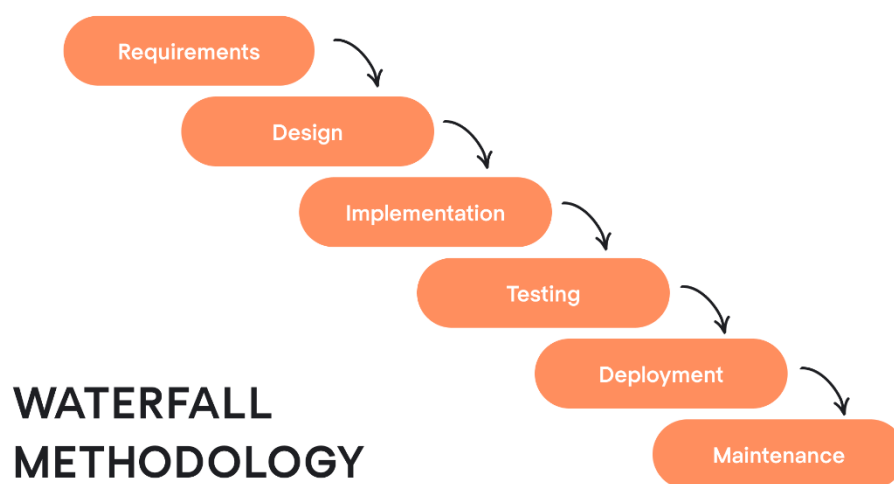
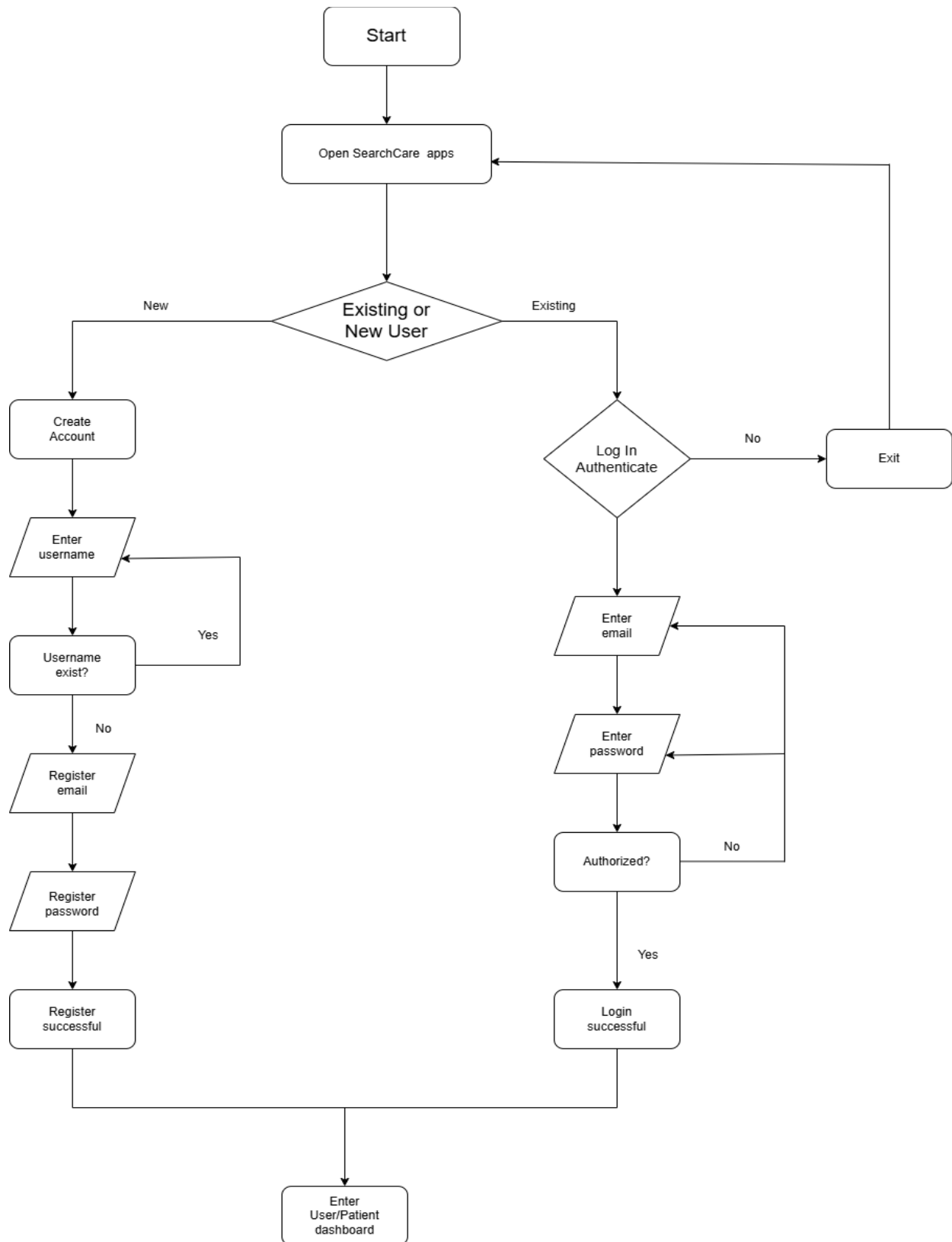


Figure 4 Waterfall Methodology photo by: Lean, agile, or waterfall? Choose the best product management methodology based on our examples. (2025, April 24). <https://www.eleken.co/blog-posts/a-comparative-guide-to-lean-agile-and-waterfal-product-management-methodo>

3.3 System design

The figures below displays the flowcharts process of SearchCare mobile application:



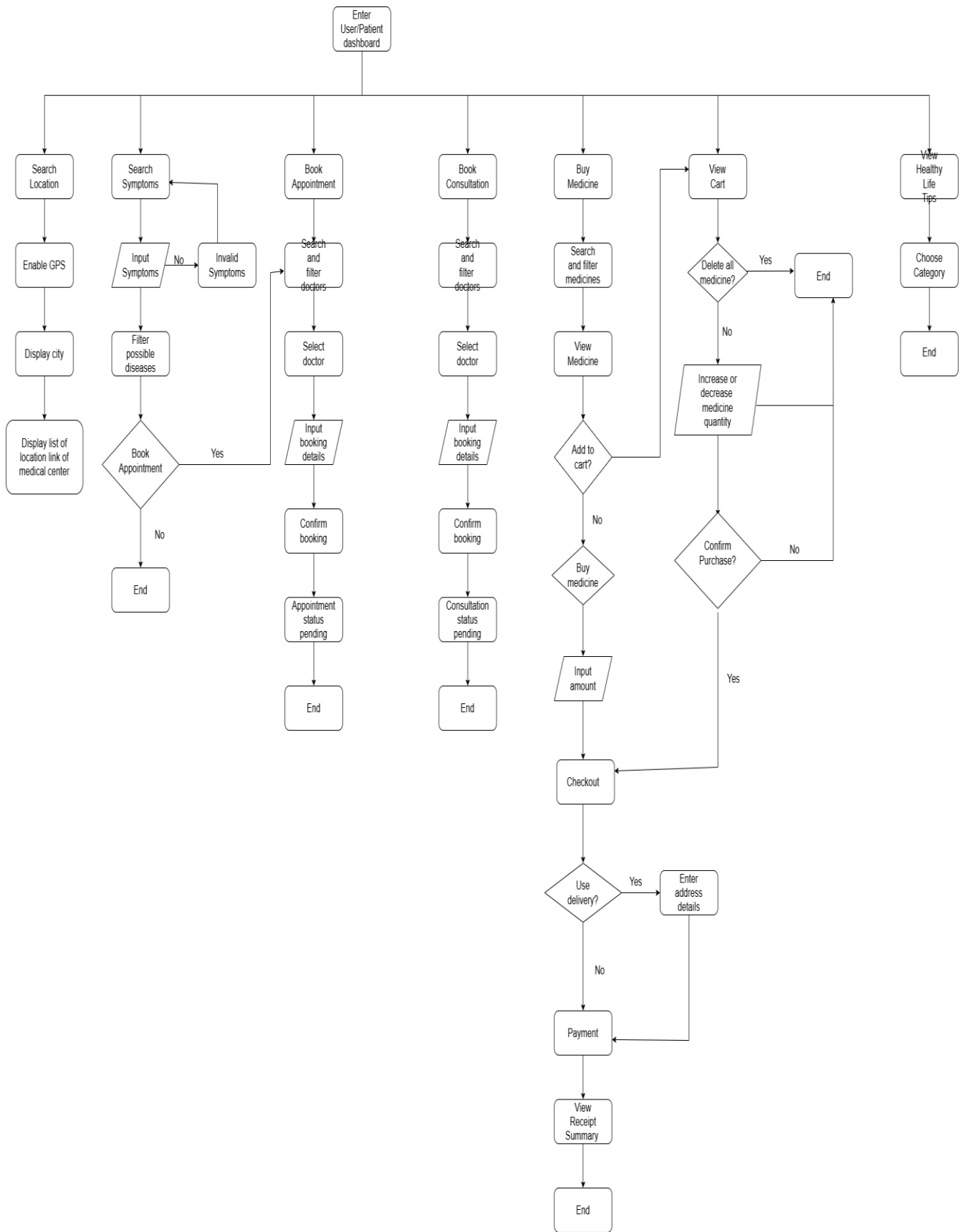


Figure 5 The flowchart process user/patient side workflow

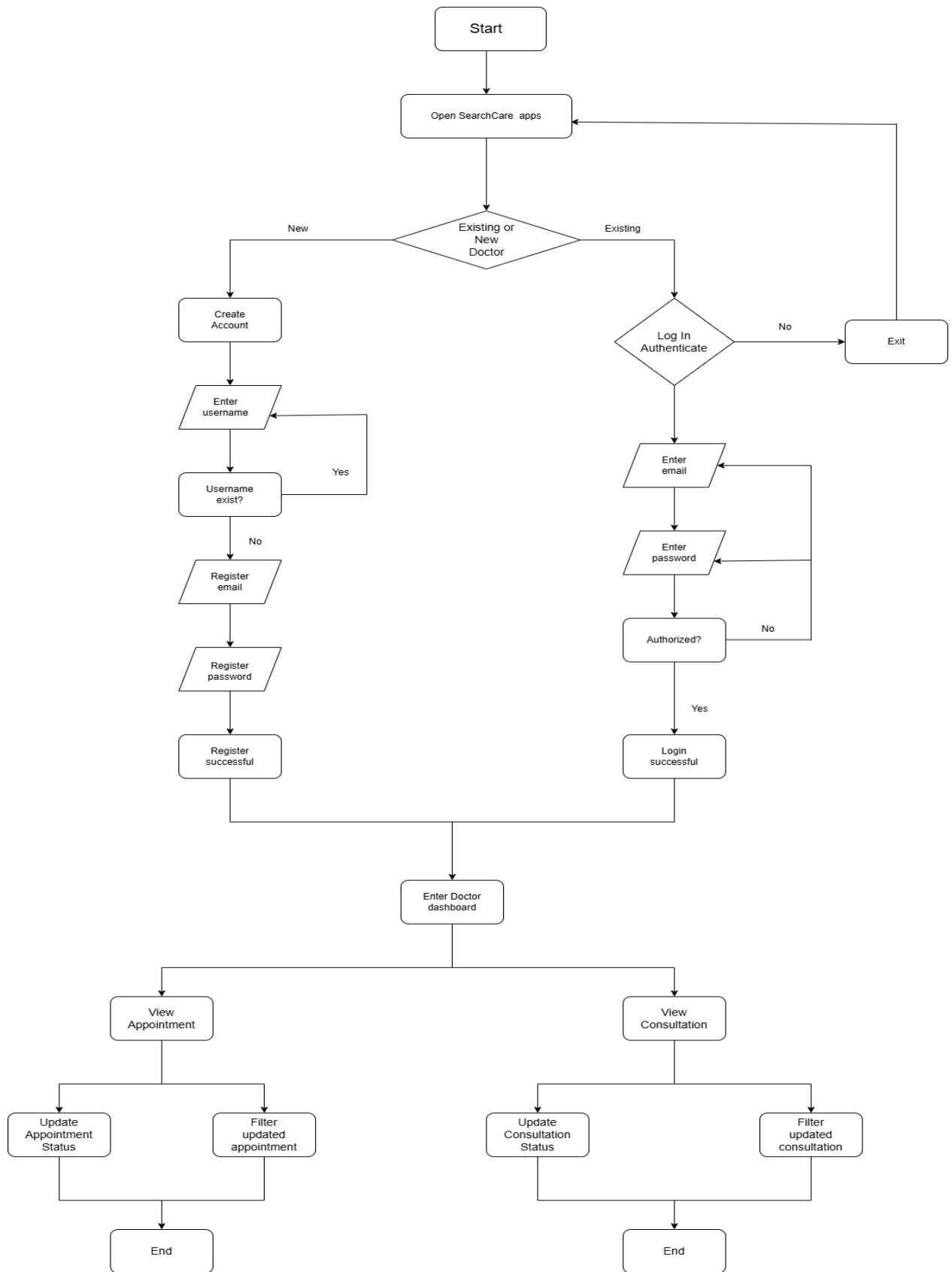


Figure 6 The flowchart process doctor side workflow

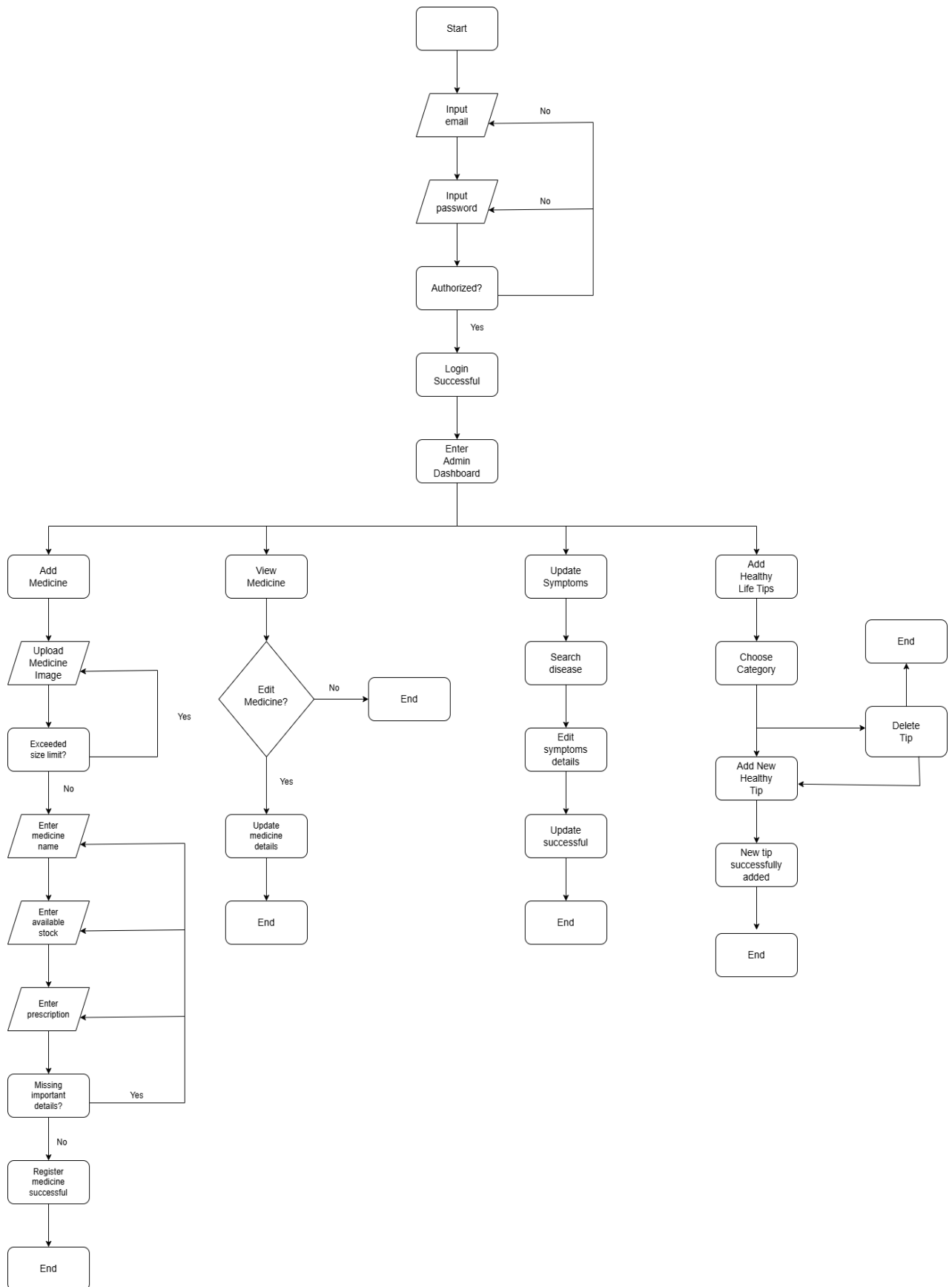


Figure 7 The flowchart process admin side workflow

The phase, activities, task and deliverables are shown below.

Table 3 The phase, activities, task and deliverables table

PHASE	ACTIVITIES	TASK	DELIVERABLES
Information Gathering	Gather info through online survey form, reading and research related articles.	Conduct feasibility study about smart healthcare applications and mobile applications.	The project objectives, problem statement, scope and significance of project.
Design	Design the system interface and process flow.	Determine how the system will work	System physical interface process and flow diagram.
Development	Use Flutter to develop the application.	Develop a prototype of the mobile application.	Prototype samples of the application.
Testing	Test the application on android operating system and test it to users.	Identify defects and fix appeared bugs.	The system should be able to collect patient data, appoint online consultation, book appointment, provide auto notifications, provide e-prescriptions, do payment and billing, and chatbot assistance.
Documentation	Write the report of the application.	Write the report of the application.	Final year project full report.

3.3.1 Context Diagram

The context diagram is the highest level of data flow diagram. A context diagram illustrates the data flow diagram between the entities and contains one main process. The process in the context diagram is process 0, which represents the apps, SearchCare, smart healthcare mobile application, and illustrates the functionality and flow in the entire apps. The context diagram of the proposed apps shows interaction between the user/ patient and the doctor.

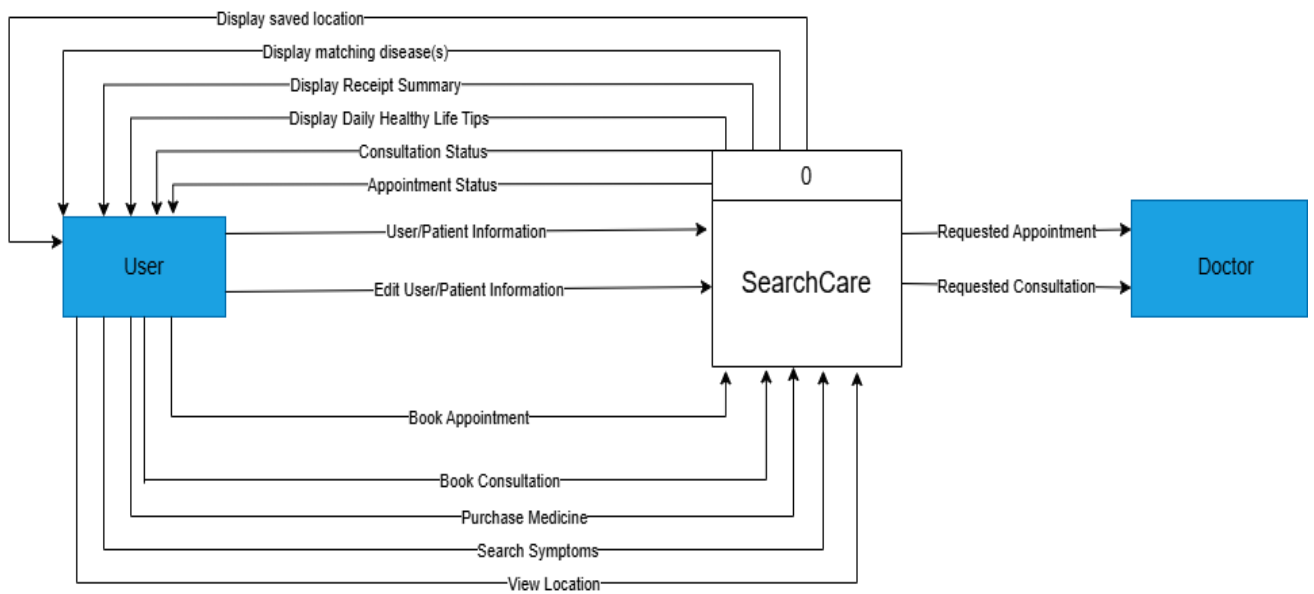


Figure 8 Context Diagram for SearchCare

3.3.2 Data Flow Diagram (DFD level – 0)

Level 0 Diagram illustrates the process that occurs in the SearchCare smart healthcare mobile application system. There are 8 main processes which are user login, register user detail, manage user profile, search and filter symptoms, apply online booking appointment, apply online consultation, payment and receipt for medicine purchase and view healthy life tips.

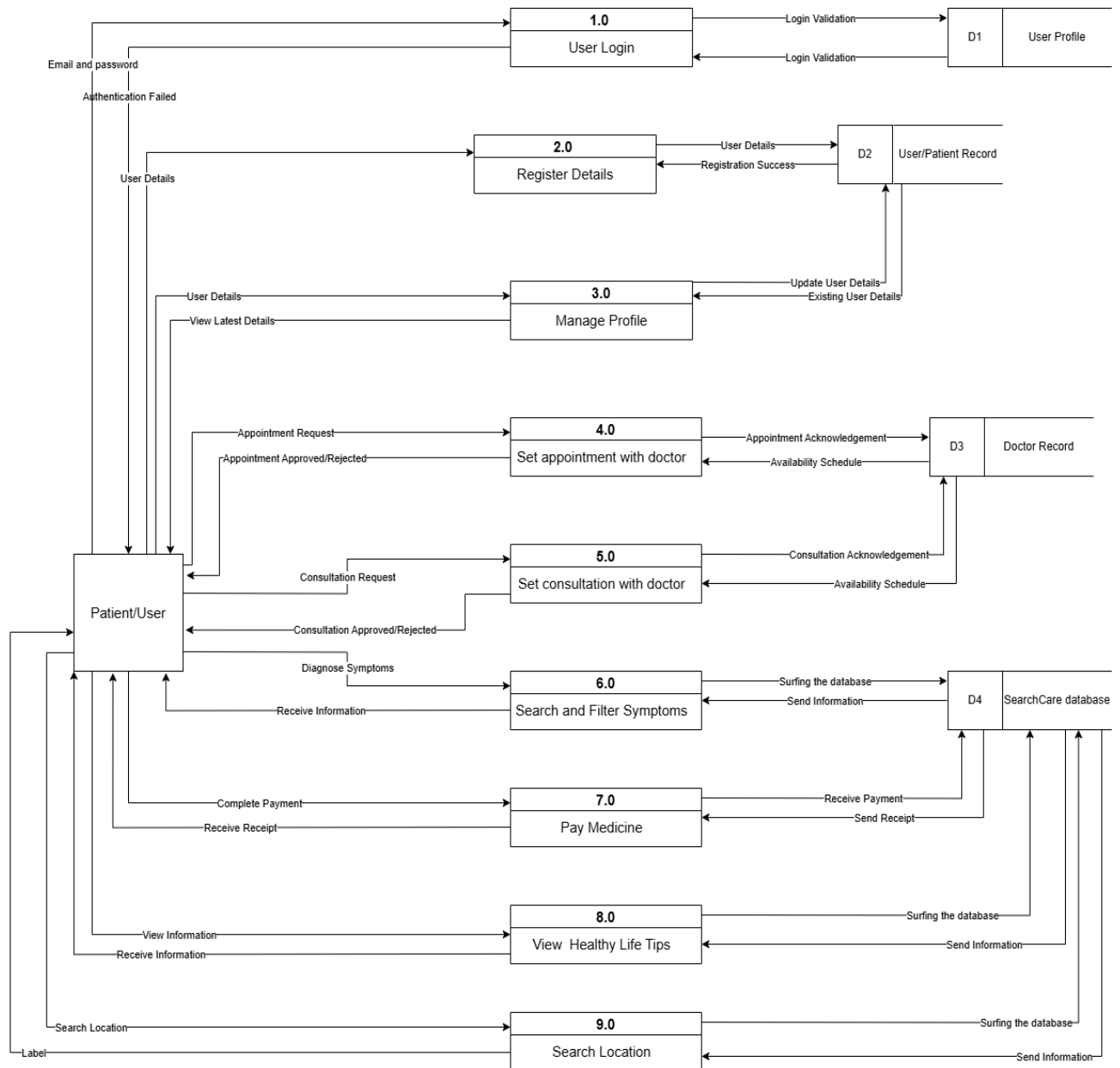


Figure 9 Level 0 of SearchCare: Patient Smart Mobile Application for Personalized Healthcare Recommendations

3.3.3 Data Flow Diagram (DFD level – 1)

Level 1 Diagram is the breakdown of all processes in the application. It is an overview of the full application system. The main processes are broken down into sub processes that will be performed in each of the main processes of the level 0 diagram.

3.3.3.1 User Login Process

There are processes done on user login. For the applications to be used, a preliminary process is followed where the user has to log in. These applications do the validation of that logging in and verify if a particular user can use it or not.

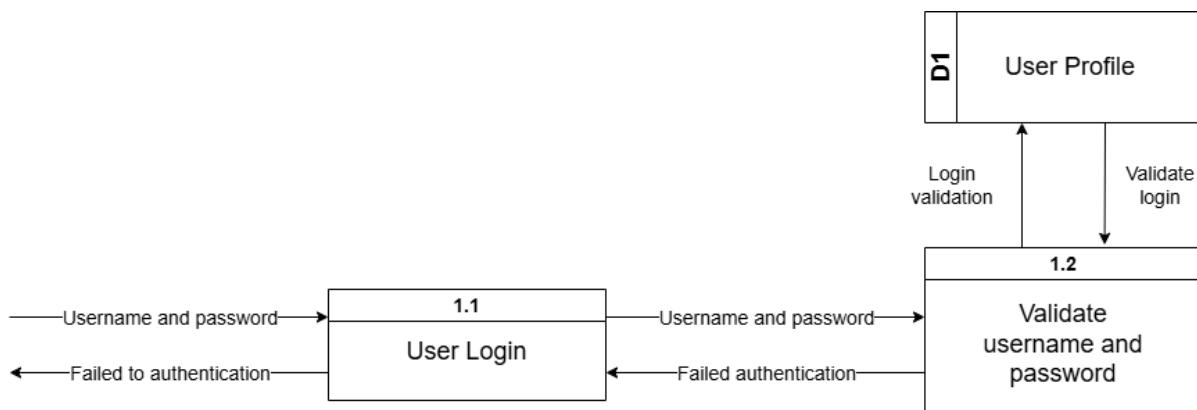


Figure 10 DFD Level 1 for user login process

3.3.3.2 Register User Details Process

This process is basically for the patient's registration. Since the user inputs their information into the user database, then the login information such as a username and password used for logging in will also be used in their profile for information. It will cross-check if the information has been entered correctly and further validate the same before storing into the patient's database.

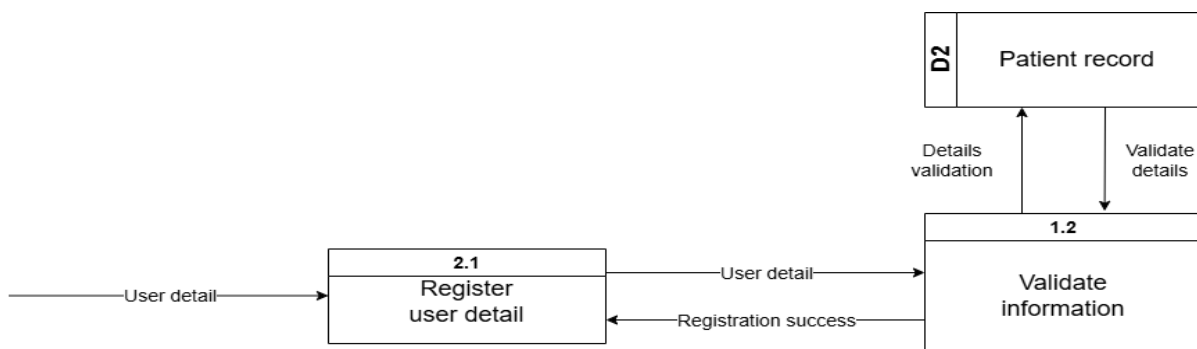


Figure 11 DFD Level 1 for register user details process

3.3.3.3 Manage Profile Process

Manage profile process is where the user able to update their profile details or delete certain information to be tailored with their daily lifestyle or preferences. For example, user want to update their address or their closest family.

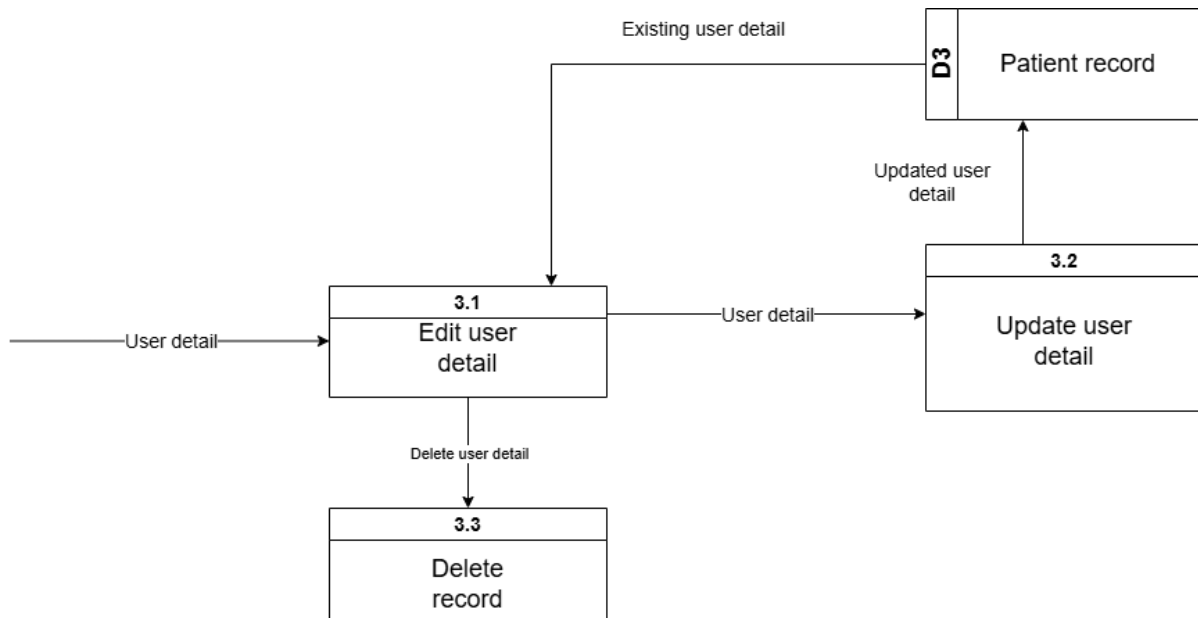


Figure 12 DFD Level 1 for manage profile process

3.3.3.4 Set Appointment Process

A DFD Level 1 diagram for the process "Set Appointment with Doctor" in a smart healthcare mobile application shows the relationship between the patient, the doctor, and the system. The process involves the patient selecting a doctor, the available time slot, and confirmation of the appointment; it is then stored in the database. Data flows include appointment requests, doctor availability checks, and notifications to both the patient and doctor upon confirmation.

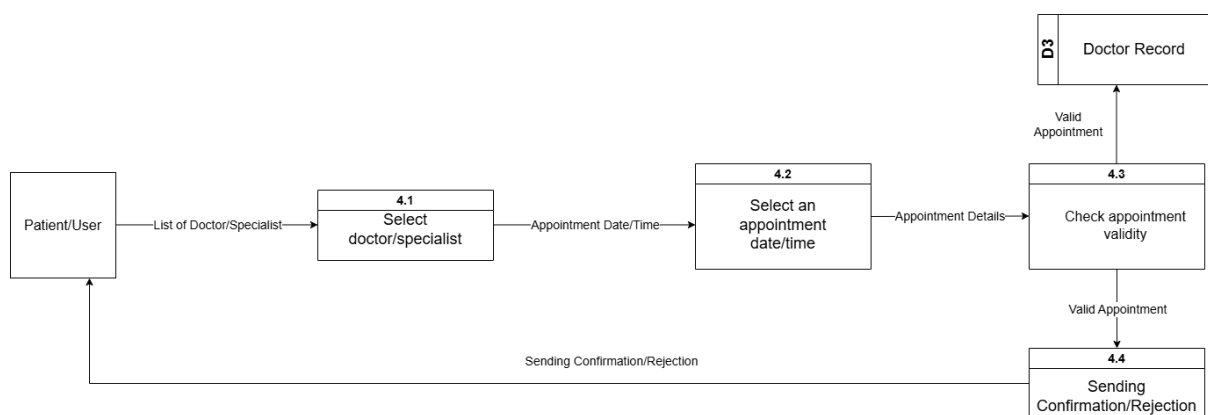


Figure 13 DFD Level 1 for set appointment process

3.3.3.5 Set Consultation Process

A DFD Level 1 of the process "Set Consultation with Doctor" in a smart healthcare mobile application shows the interaction of a patient, doctor, and system in setting up a consultation. It involves a process where the patient selects a doctor, specifies his preference for consultation, either video call or chat, confirms the consultation, which gets recorded in the database. Data flows comprise consultation requests, checks on availability by doctors, and confirmation messages to the patients and the concerned doctor.

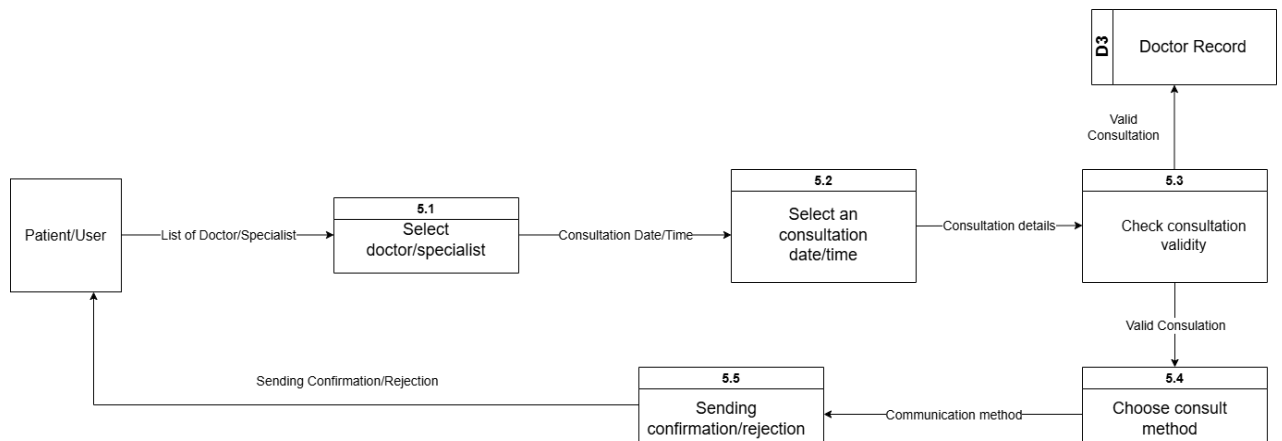


Figure 3.11 DFD Level 1 for set consultation process

3.3.3.6 Search and Filter Symptoms Process

A DFD Level 1 diagram for the process of "Search and Filter Symptoms" for a smart healthcare mobile application describes how the user interacts with the system to ascertain certain health problems. The user provides the symptoms he feels, which are matched with a medical database that returns possible conditions or diagnosis, while enabling further filtering by category, duration, and other aspects.

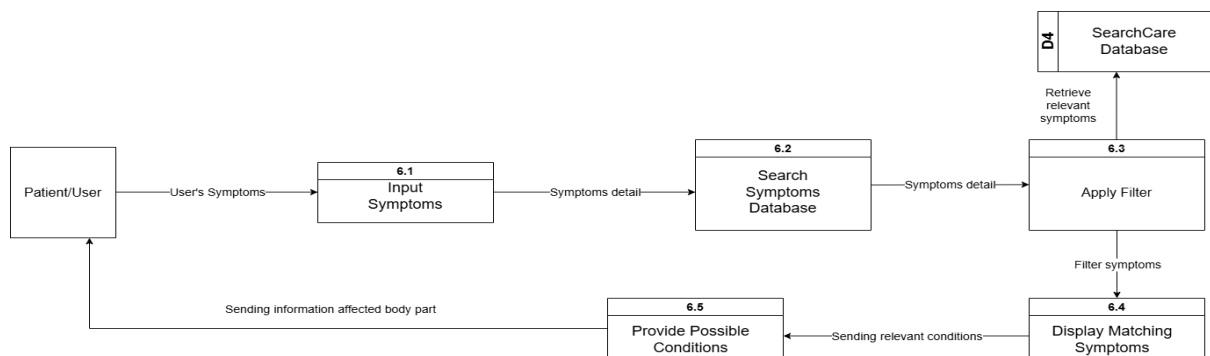


Figure 14 DFD Level 1 for search and filter symptoms process

3.3.3.7 View Healthy Life Tips

A DFD Level 1 diagram for the process "View Healthy Life Tips" in a smart healthcare mobile application outlines the flow of data between users and SearchCare database. It involves the user selecting lifestyles category they wish to view and gain information about healthy tips.

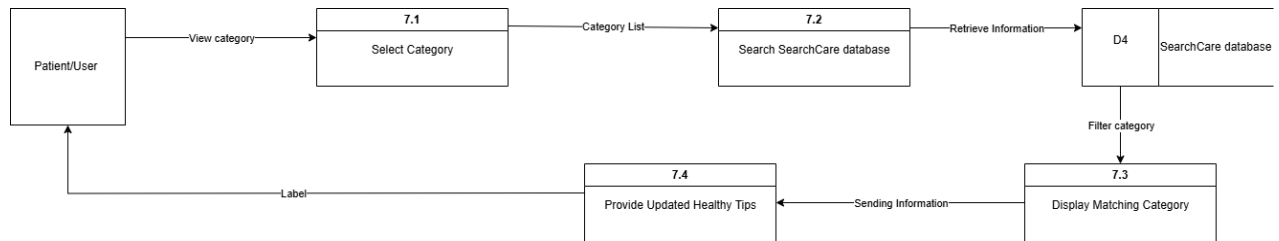


Figure 15 DFD Level 1 for view healthy life tips process

3.3.3.8 Pay medicine and service Process

A DFD Level 1 diagram for the process "Pay Medicine and Delivery" in a smart healthcare mobile application outlines the flow of data between users, pharmacies, and payment systems. It involves the user selecting prescribed medicines, confirming the payment through a secure gateway, and providing the delivery details to be forwarded to the pharmacy for processing. It covers data flow of payments, confirmation of an order, or status about dispatch to the user to have a smooth transaction and tracking experience.

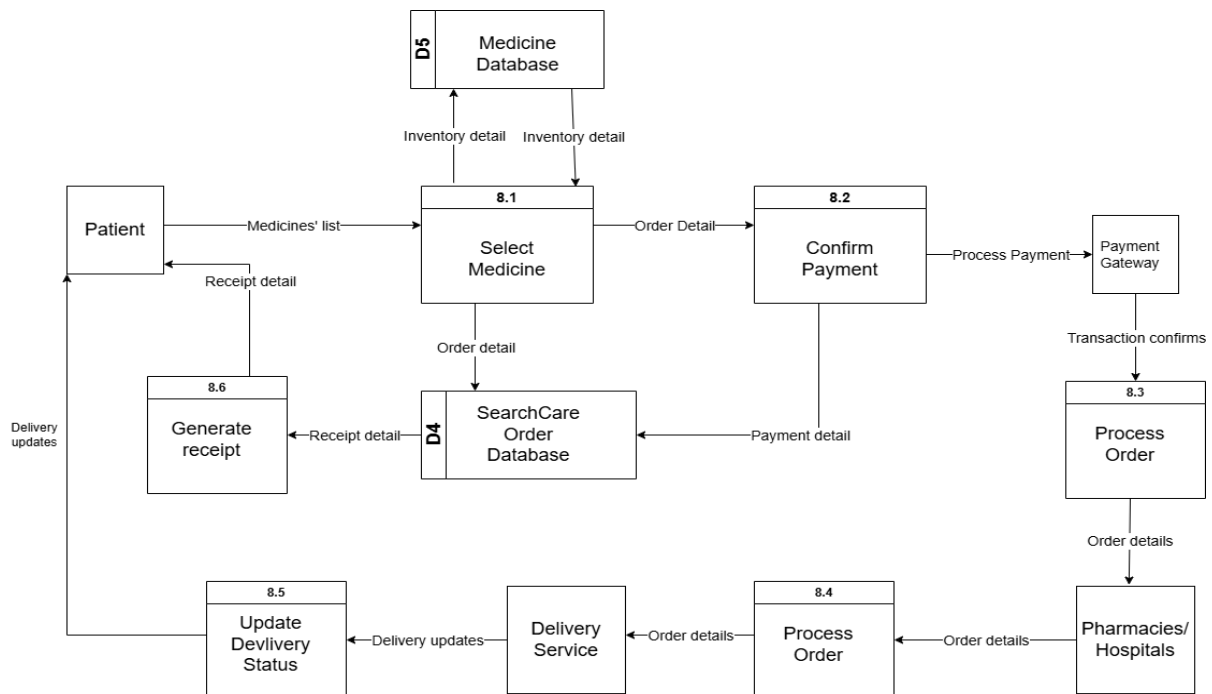


Figure 16 DFD Level 1 for pay medicine and service process

3.3.3.9 Search Location

A DFD Level 1 diagram for the process "Search Location" in a smart healthcare mobile application outlines the flow of data between users and SearchCare database. It involves the user search and filter location of clinics and hospitals that are listed and located in the city or regions they are living in.

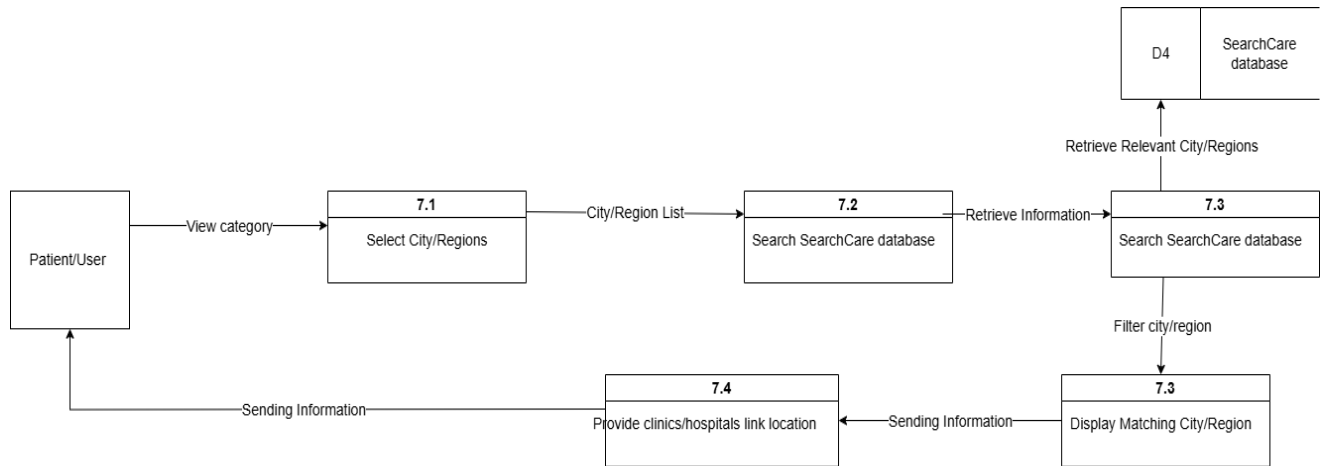


Figure 17 DFD Level 1 for Search Location process

3.3.4 Entity Relationship Diagram (ERD)

The Entity-Relationship Diagram (ERD) for a smart healthcare mobile application, SearchCare includes entities, attributes, and relationships that represent the data model of the system. Entities that are listed in the ERD are patient, doctor, pharmacy, hospital, emergency contact, appointment, consultation, medication, notification and delivery.

3.3.4.1 Entity Patient

Patient	
PK	<u>Pt_id int NOT NULL</u>
	Pt_firstName char(100)
	Pt_lastName char(100)
	DateOfBirth date NOT NULL
	Gender char(10)
	Address char (50)
	Address char (50)
	City char(10)
	Country char(10)
	Email char(10)
	Mobile int(12)

Figure 18 ERD of Entity Patient

Entity Patient consists of 11 attributes. Attribute Pt_id represent the patient ID in the application database which serves as the primary key in the entity. Pt_firstName represented the patient's first name while Pt_lastName represented the patient's last name. DateOfBirth represent the patient's birthday date. Gender represents the patient's gender. Address represents the patient's house address. Mobile and email represent the patient's contact number and ways to contact.

3.3.4.2 Entity Doctor

Doctor	
PK	<u>DocID int NOT NULL</u>
	DocName char(100)
	Specialization char(100)
	ContactInfo char(20)
	PlaceOfWork char(30)
	License Num int(50)

Figure 19 ERD of Entity Doctor

Entity Doctor represented by six attributes which will be stored in the database. Attribute DocID serves as the primary key in entity Doctor. Attribute DocName represent the name of the doctor. Specialization is an attribute that represent medical field the doctor specializes for. Attribute ContactInfo is the attribute in the entity that represent the contact information of the doctor. PlaceOfWork is the attribute to represent the location of the workplace of the doctor. License_Num is the attribute that represent the license number of a certified doctor in their career.

3.3.4.3 Entity Emergency Contact

Emergency Contact	
PK	<u>ContactID int NOT NULL</u>
	EmeConName char(100)
	Relationship char(50)
	EmeConNum char(20)
FK	PatientID

Figure 20 ERD of Entity Emergency Contact

Entity Emergency Contact consist of one primary key and four attribute where one of the attributes is a foreign key. The primary key for the entity is ContactID and the foreign key is PatientID where ContactID will be tailored with respective PatientID. EmeConName represent the emergency contact name, Relationship attribute represent the connection between patient and emergency contact person and EmeConNum is the contact number of the emergency contact.

3.3.4.4 Entity Appointment

Appointment	
PK	<u>AppointmentID int NOT NULL</u>
	Appdate date not NULL
	Time char(10)
	Status char(10)
FK	PatientID
FK	DoctorID

Figure 21 ERD of Entity Appointment

Entity Appointment consist of primary key of AppointmentID and five attributes where two of the attributes are foreign key, PatientID and DoctorID. Appdate is the attribute that represent appointment date, Time represent time of the appointment being held and Status represent either the appointment is confirmed or rejected.

3.3.4.5 Entity Consultation

Consultation	
PK	<u>ConsultationID int NOT NULL</u>
	Consultdate date not NULL
	ConsultTime char(10)
	ConsultStatus char(10)
FK	PatientID
FK	DoctorID

Figure 22 ERD of Entity Consultation

Entity Consultation consist of primary key of ConsultationID and five attributes where two of the attributes are foreign key, PatientID and DoctorID. Consultdate is the attribute that represent consultation date, ConsultTime represent time of the consultation being held and ConsultStatus represent either the consultation is confirmed or rejected.

3.3.4.6 Entity Medication

Medication	
PK	<u>MedicationID int NOT NULL</u>
	MedicineName char(100)
	Dosage char(10)
	Frequency char(20)
	Instruction char(50)
FK	PatientID
FK	DoctorID

Figure 23 ERD of Entity Medication

Entity Medication have primary key of MedicationID and two foreign keys, PatientID and DoctorID. Attribute MedicineName represent the name of medicine, Dosage represent dosage needed to be taken for the medicine, Frequency is the period needed to take the medicine, Instruction represent the action for patient on how to take the medicine.

3.3.4.7 Entity Delivery

Delivery	
PK	<u>DeliveryID int NOT NULL</u>
	DeliveryAddress char(50)
	DeliveryStatus char(50)
	DeliveryDate Date NOT NULL
	DeliveryTime Time NOT NULL
	Courier Info char(50)
FK	PatientID
FK	OrderID

Figure 24 ERD of Entity Delivery

Entity Delivery consists of eight attributes. Primary key is represented by DeliveryID and two foreign keys, PatientID and OrderID being the attributes as well. DeliveryAddress, the address of the patient to be delivered to, DeliveryStatus, to keep up the updates of the delivery, DeliveryDate and DeliveryTime to record the date and time of before and arrival of the

delivery, and Courier_Info to represent the deliverer name and vehicle used to deliver the purchased medicine.

3.4 Wireframe

Recent research has investigated multiple ways to create and utilize wireframes in the design process for UI. WireGen utilizes large language models for the automated creation of mid-fidelity wireframes from short design descriptions, setting a new performance baseline (Feng et al., 2023). For 3D applications, WireframeNet generates complete wireframe structures from point clouds with the help of geometric transformations and connectivity analysis by Cao et al. (2023), while PC2WF presents an end-to-end trainable network that can convert 3D point clouds to wireframe models through a multistep process introduced by Liu et al. (2021). The UI design retrieval through a deep learning-based approach using the wireframe image autoencoder was meant to find relevant high-fidelity UI designs based on the input of sketched wireframes, outperforming state-of-the-art in UI design research by Chen et al. (2020). These studies enhance both efficiency and effectiveness in the generation and use of wireframes across a wide spectrum of domains.

3.4.1 Splash Screen

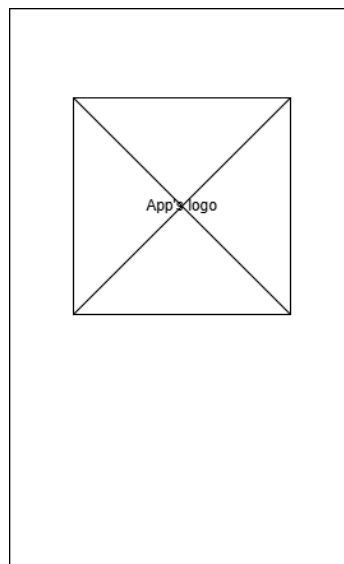


Figure 25 Splash Screen for SearchCare

Figure 25 represent the splash screen for opening screen of SearchCare apps. User/Patient will know that the apps belongs to SearchCare.

3.4.2 Log In User Screen

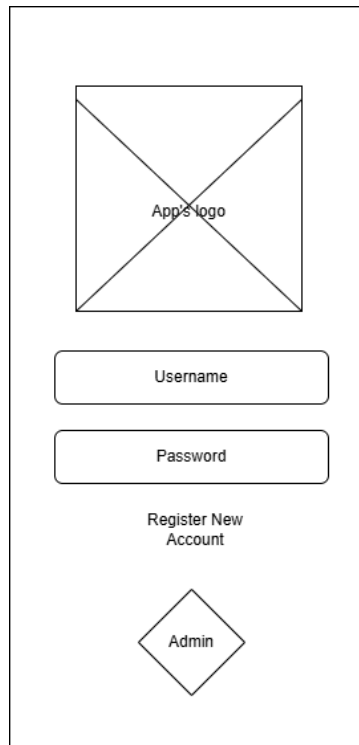
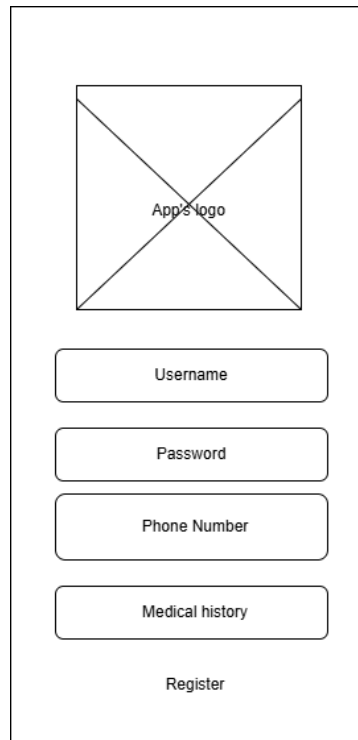


Figure 26 Log In screen for SearchCare apps

Figure 26 shows the log in screen for Search Care apps. User/Patient will input their username and password to sign in into the app. The page also has function for new user to register account. This page also have function for admin to log in.

3.4.3 Sign Up Register Form Screen



The image shows a vertical rectangular form for signing up. At the top is a square placeholder for the app's logo, with a large 'X' drawn across it and the text 'App's logo' in the center. Below the logo are four rounded rectangular input fields, each containing a label: 'Username', 'Password', 'Phone Number', and 'Medical history'. At the bottom of the form is a 'Register' button.

Figure 27 Sign Up screen for SearchCare apps

Figure 3.31 shows the sign up screen for SearchCare application. The function of this page is to register new user that want to use this application. User can input their detail in the provided space so it can be stored inside the database of the application.

3.4.5 Homepage

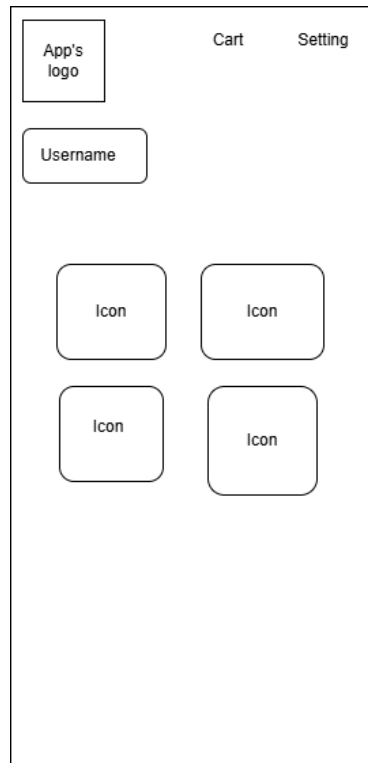


Figure 28 Homepage for SearchCare apps

Figure 3.32 shows the wireframe screen for homepage for SearchCare apps. The homepage will include main features for SearchCare application during implementation phase.

3.4.6 Search Symptoms Page

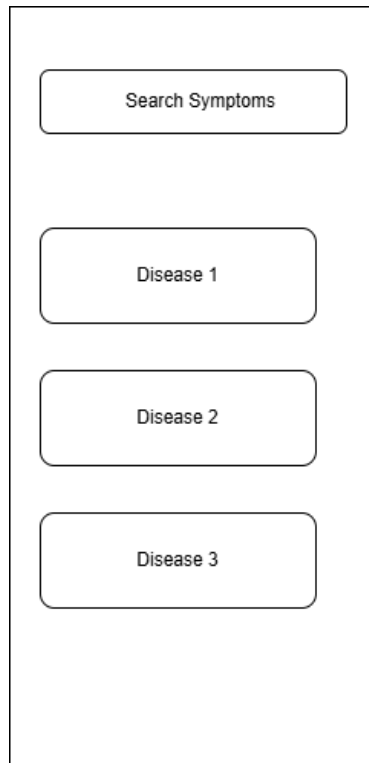


Figure 29 Search Symptomts page

Figure 29 illustrates the wireframe of search symptoms page for SearchCare application. This will be implemented for the achievement of second objectives.

3.5 Information Gathering

A feasibility study regarding the project has been conducted to obtain precise objectives, scope, and problem statement of a proposed project that needs to be developed so that simple indication of the proposed project can be obtained. Figure 3.3 elaborates the idea in gathering phase.

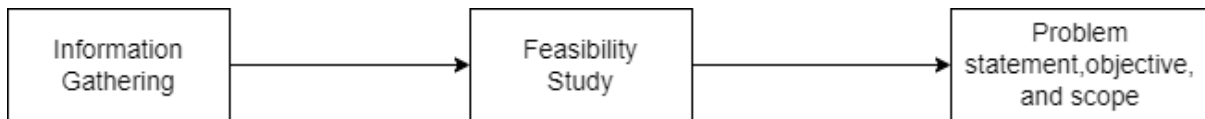


Figure 30 Information gathering phase

This is the stage where data gathering begins by collecting information, such as the project of the mobile application and techniques used in the processing of data from different types of resources including online survey forms, websites, and articles. Such information obtained from these various sources plays an important role in gaining more consideration for what should be done and just to have a simple idea about the process flow. All this important resource will be interpreted to identify the objective, scope, and problem statement needed and crucial for this entire project.

In this section, an online survey questionnaire has been conducted among Kota Samarahan residents. The participants consist of 51 adults, which are 51 questionnaires in the Kota Samarahan area. These were then collected, analyzed, and finalized into pie-chart form to see the difference in results across different categories of adult age easily. Then, the overall of each factor was summarized to show the overall healthcare factors of the society in Kota Samarahan toward healthcare.

Count of range of age

Age (years old)

51 responses

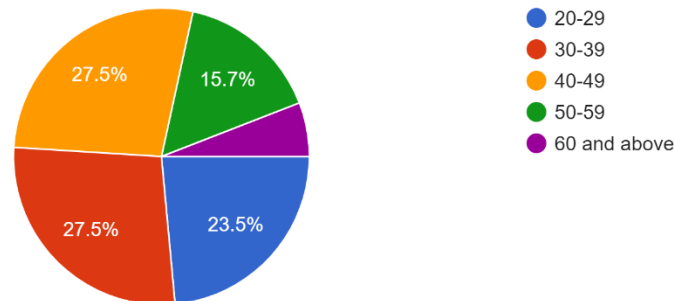


Figure 31 Range age of respondents' pie chart

Figure 31 above shows that there are two type of range age with the same percentage of respondents which is 27.5% which is the highest percentage among others which consist of the age range of 30 – 39 years old and 40 – 49 years old which means around 14 adults respectively. Next, the second highest percentage that is recorded is 23.5% for the range age of 20 – 29 years old which makes them 12 adults. The lowest percentage recorded is 5.9% for the range age 60 and above that makes 3 adults while the second lowest percentage recorded is 50 – 59 years old that makes 8 adults which is 15.7%.

Count of preferable healthcare center type

Which healthcare center type would be preferable during visit?

51 responses

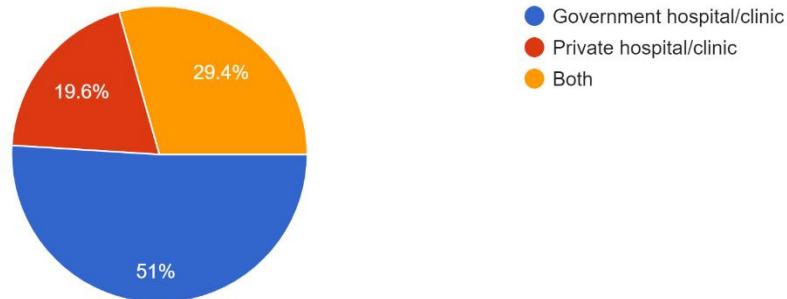


Figure 32 Healthcare center recommendation pie chart

Count of preferable method of obtaining healthcare

Preference on getting source of healthcare

51 responses

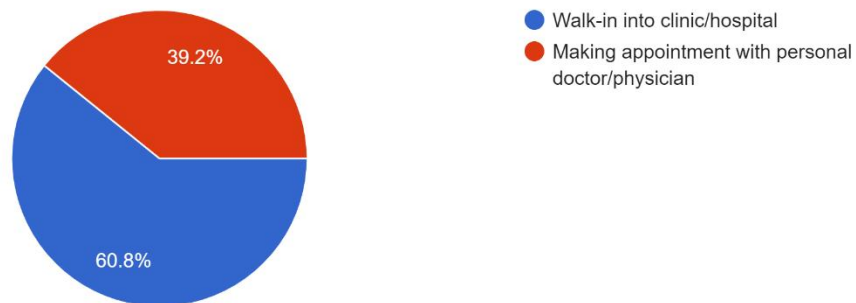


Figure 33 Method of obtaining healthcare pie chart

Based on Figure 32, the most percentage respondent count recorded for preferable healthcare center is 51% which makes 26 adults that prefer to go to government clinic or hospital meanwhile it is stated that 29.4% of adult respondent would prefer to go both for government and private healthcare center which makes them 15 adults. Lastly, only 19.4% is recorded for adults that prefer to go to private clinic or hospital which is only 10 adults.

As shown in Figure 33, during the surveys there are 51 respondents which consist of adult respondents. 31 out of 51 respondents would prefer to go to walk-in clinics or hospital while

20 respondents prefer to make appointments with their personal doctor or physician. Thus, the percentage of respondents recorded is 60.8% and 39.2% respectively.

Would you prefer to get treatment through online consultation and online diagnosis just to save time during waiting session during walk-in session?

51 responses

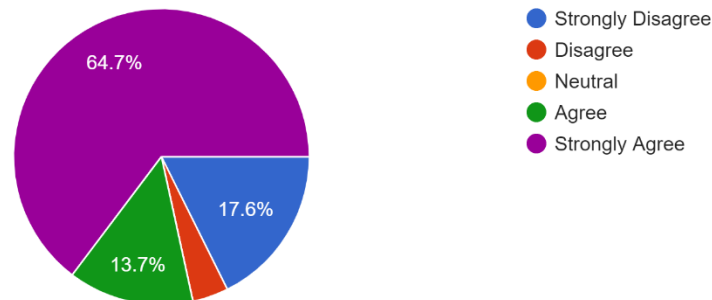


Figure 34 Online consultation and online diagnosis pie chart

Based on Figure 34 above, 64.7% of the respondents strongly agree to get treatment through online consultation and online diagnosis to save time which makes them 33 adults. However, 17.6% or 9 respondents strongly disagree to get treatment through online consultation and online diagnosis. This is because those respondents may not want to risk getting treatment online with the risk of getting wrong diagnosis. Meanwhile, 13.7% of the respondents agree with online consultation and online diagnosis to get treatment and there are 3.9% of the respondents disagree to get online diagnosis which are 9 and 2 adults respectively.

Would you prefer to do online appointment with your doctor at the nearest clinic/hospital to avoid traffic on the road?

51 responses

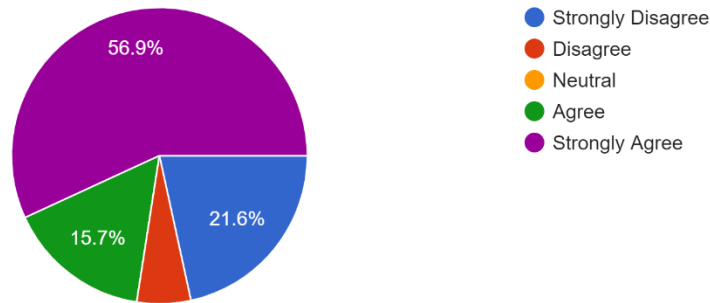


Figure 35 Online appointment preference pie chart

Figure 35 displays that 56.9% of the respondents strongly agree to do online appointments with their doctor at their nearest clinic or hospital to avoid traffic while 21.6% of the respondents strongly disagree with it. Thus, 29 adults strongly agree to do online appointments, and 11 adults strongly disagree to do online appointments. Next, 15.7% of the respondents agree to do online appointments while 5.9% of the respondents disagree to do online appointments which are 8 adults and 3 adults respectively.

Would you prefer to get your medical prescriptions through online consultation from your nearest clinic/doctor?

50 responses

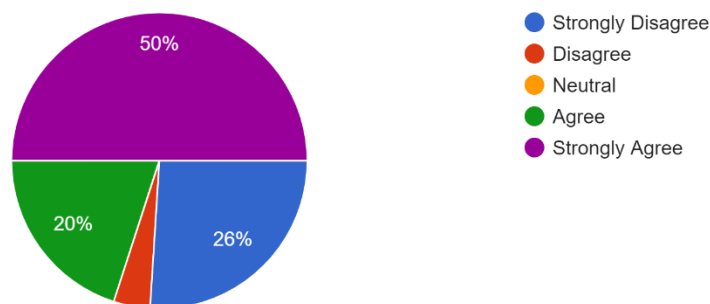


Figure 36 Online medical prescription pie chart

Based on Figure 36, the highest percentage recorded is 50% which is half of the respondents strongly agree to get medical prescriptions through online consultation from their nearest clinic or doctor which is 25 adults. Meanwhile, 26% of the respondents strongly disagree with getting their medical prescriptions through online consultation which makes them 13 adults. Then,

20% of the respondents (10 adults) agree with getting their medical prescriptions through online consultation and 4% of the respondents (2 adults) disagree with it.

Which method would you prefer to get your medicine after an online diagnosis or online consultation?

51 responses

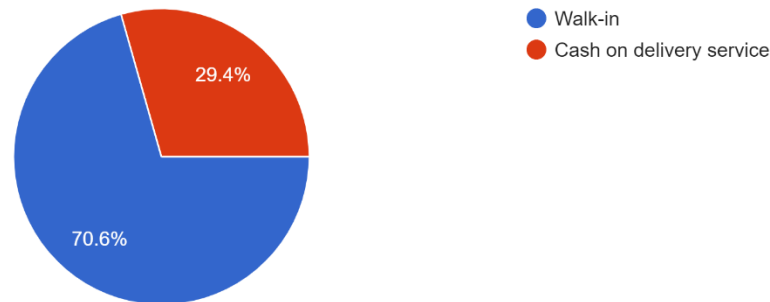


Figure 37 Preferable method to get medicine pie chart

Figure 37 shows that 70.6% of the respondents prefer to walk-in method to get their medicine after an online diagnosis or online consultation with their doctor which makes them 36 adults while 29.4% of the respondents prefer to use cash on delivery (COD) method to get their medicine after their online session with their doctor which makes them 15 adults.

Would you prefer to receive daily notifications about healthcare tips everyday?

51 responses

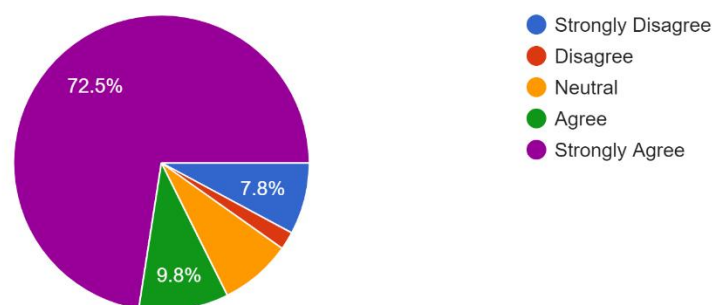


Figure 38 Healthcare notification pie chart

Based on Figure 38, 72.5% of the respondents which makes them 37 adults strongly agree to receive daily notification about healthcare tip while 7.8% of the respondents which is 4 adults strongly disagree to receive daily notification about healthcare tip. 9.8% of respondents agree to receive daily notification about healthcare tips which makes them 5 adults while 7.8% of

respondents which makes them 4 adults neutral to receive daily notification about healthcare tip.

Would you prefer to receive feedback about your daily diet intake if you want to manage your diet efficiently?

51 responses

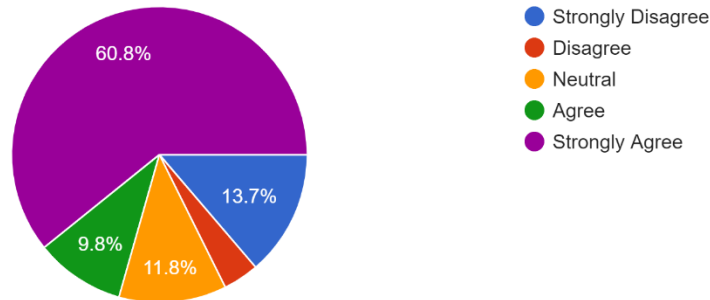


Figure 39 Daily diet notification pie chart

Figure 39 above shows that the highest percentage recorded for respondents strongly agree to prefer to receive feedback about their daily diet intake is 60.8% which makes them 31 adults while the highest percentage recorded strongly disagree to prefer to receive feedback about their daily diet intake is 13.7% which makes them 7 adults. There are 9.8% of the respondents (5 adults) agree that their daily diet intake to receive feedback while 11.8% of the respondents (6 adults) neutral that their daily diet intake to receive feedback and 3.9% of the respondents disagree which makes them 2 adults.

Would you like to know there are different type of category of healthcare tips?

51 responses

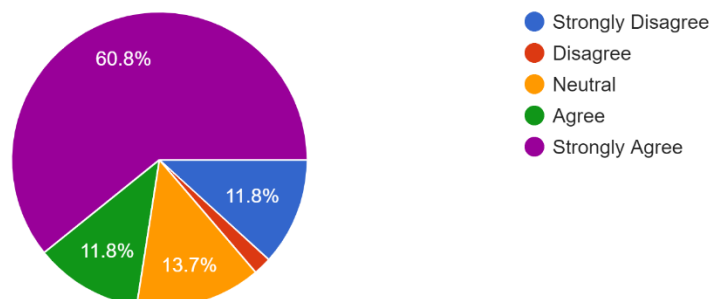


Figure 40 Healthcare tips category pie chart

Based on Figure 40, 60.8% of the respondents strongly agree to discover more about different types of categories of healthcare tips which makes them 31 adults. Meanwhile 11.8% of respondents strongly disagree with discovering more about different types of categories of healthcare tips which makes them 6 adults. 13.7% of respondents feels neutral to discover more about different types of categories of healthcare tips which makes them 7 adults and 11.8% agree that they want to discover more about different types of categories of healthcare tips which makes them 6 adults.

3.6 Software Requirement

Before developing the mobile application, there are several requirements that are needed to implement in the app.

Table 4 Software Requirement

Feature	Behaviour
Online Function	<ul style="list-style-type: none"> • Most application on Google Play or Play Store use online features to handle and manage the data. • SearchCare mobile application use online database to store the patient and doctor data on database store.
Database	<ul style="list-style-type: none"> • Most mobile application use SQLite Database libraries or Shared preferences as local data storage. • This application use Firebase because SearchCare, as a smart healthcare mobile application, prioritizes real-time synchronization, secure data handling, and scalability—requirements that are better met by Firebase rather than SQLite. • Firebase is a cloud-hosted NoSQL database, which means it can instantly sync data across multiple devices and platforms (iOS, Android, web). • For SearchCare, where physicians, users, and administrators want real-time access to consultations, appointments, medication inventory, etc., this is crucial. • In contrast, SQLite is local to the device. It is unable to update or sync automatically across several devices.
Data Integrity	<ul style="list-style-type: none"> • The accuracy, consistency, and dependability of user and medical data over the course of its lifecycle are referred to as data integrity in the context of

	<p>SearchCare, a clever healthcare mobile application.</p> <ul style="list-style-type: none"> • Firebase’s design, technologies, and security guidelines all contribute to high data integrity. • Rules or Firebase Functions can be used to validate input fields such as email, phone number, ID number, etc. both at the client and Firestore levels. • Prevents inaccurate or distorted data from entering the database.
Build approached	<ul style="list-style-type: none"> • The software requirement phase’s build strategy describes how the system is planned, created, and tested to satisfy both functional and non-functional needs. The build methodology for SearchCare, a clever healthcare mobile application, guarantees that the app is safe, scalable, easy to use, and designed to manage private medical information and interactions. • In terms of software requirements, the SearchCare build methodology guarantees: <ol style="list-style-type: none"> 1. Architecture that is scalable and modular 2. Secure access based on roles 3. Integration of cloud databases in real time 4. Effective development using Firebase and Flutter 5. Adherence to both functional and non-functional objectives is crucial, particularly in a hospital setting. 6. A mobile health application that is professional, safe, and easy to use is the result of this methodical process.
Responsive Design	<ul style="list-style-type: none"> • The capacity of an application to fluidly adjust to various screen sizes, orientations, and resolutions is known as responsive design, and it guarantees a consistent user experience on all devices. Responsive design is essential to the usability,

	<p>accessibility, and general professionalism of SearchCare, a healthcare mobile application that caters to both patients and physicians.</p> <ul style="list-style-type: none"> • A key component of SearchCare’s mobile development approach is responsive design. It guarantees smooth interaction between the app and patients, physicians, and administrators on a range of screens, devices, and orientations. With the help of Flutter’s responsive widgets and layout features, the design successfully makes SearchCare scalable for future improvements, accessible, and easy to use.
<p>Suitable IDE</p>	<ul style="list-style-type: none"> • For a mobile application like SearchCare to be developed, tested, and deployed effectively, selecting the appropriate Integrated Development Environment (IDE) is essential. Because SearchCare is developed with Flutter and interfaces with Firebase, the IDE needs to handle cloud integration, UI design, cross-platform development, and real-time testing. • Android Studio is ideal for SearchCare as it offers all the necessary tools to build, test, and deploy a secure, scalable, and responsive mobile healthcare application using Flutter and Firebase.

3.7 Development

The Flutter framework and the Dart programming language were used in the methodical development of the SearchCare mobile application. Android Studio, the main IDE and compiler for the project, will be used to develop and test the app.

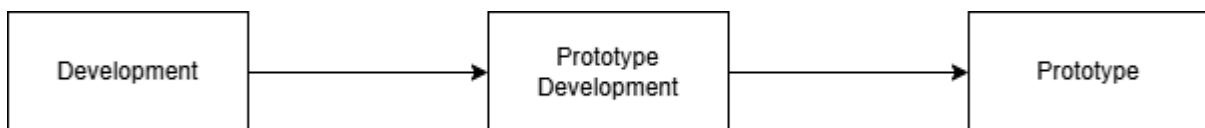


Figure 3.36 Development Phase

Using Android Studio’s powerful Flutter and Dart integration, the first prototype was created. In order to envision how patients, physicians, and administrators would utilise the application, the prototype concentrated on the main user interface (UI) structure and screen navigation. Included in this prototype were:

- Main menu, registration, and login wireframes
- A simple navigation system with Flutter routes
- Placeholder user interface for bookings, preferences, and medication lists
- Basic dummy data to mimic database communication

Table 5 Development Requirement

Component	Tool/Technology Used
IDE/Compiler	Android Studio
Programming Language	Dart
Framework	Flutter
Database	Firebase Firestore
Authentication	Firebase Auth
File Storage	Firebase Storage
Notifications	Firebase Cloud Messaging
Development Environment	Tool/Technology Used

3.7.1 Project Flow Using Dart in Flutter

- StatefulWidget used for screens that require dynamic UI (e.g., cart, booking forms).
- FirebaseFirestore.instance.collection() used to retrieve and manipulate Firestore data.
- Navigator.push() and Navigator.pop() handled screen transitions.
- FutureBuilder and StreamBuilder used for real-time database updates.

3.7.2 Iterative Development Process

The application was developed using an **incremental and iterative approach**:

1. UI development and layout creation.
2. Backend logic and database integration.
3. Real-time data testing and bug fixing.
4. User testing for usability and functionality.
5. Final implementation and polish of features.

The development of SearchCare will be carried out using Android Studio as the development environment, Dart as the programming language, and Firebase as the backend platform. By combining Flutter’s UI flexibility with Firebase’s real-time capabilities, the project will be able to deliver a scalable and responsive healthcare application that meets the requirements of patients, doctors, and administrators.

3.8 Testing

A crucial stage in the creation of the SearchCare mobile application is testing. It guarantees that the program works as intended, has no serious defects, and offers a seamless user experience. 30 varied testers, including students, working adults, and general users with and without technical expertise, will be participate in the SearchCare testing procedure, which combined functional, usability, and user acceptability testing (UAT).

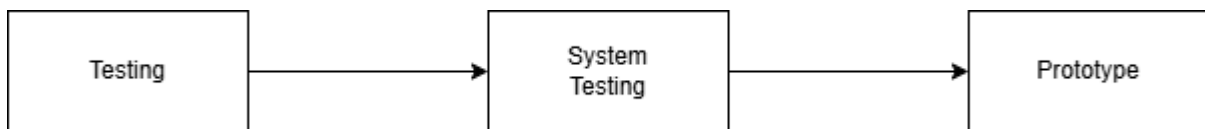


Figure 3.37 Testing Phase

3.8.1 Testing Methodology

Table 6 Testing Requirement

Test Type	Description
Functional Testing	Ensured that every feature works as intended (login, book appointments, etc).
Usability Testing	Assessed how intuitive and user-friendly the app interface is.
User Acceptance Test	Gathered feedback from real users to validate the app meets its objectives.

The testing phase for SearchCare will engage **30 real users** in validating and refining the application. Their feedback will help improve UI/UX, fix hidden bugs, and ensure core functionalities met user expectations. This real-world testing contributed to delivering a **stable, user-friendly, and reliable healthcare app**, ready for broader deployment and future enhancements.

3.9 Documentation

An essential component of the SearchCare mobile application development process is documentation. It guarantees that future developers, testers, and stakeholders fully comprehend the application’s structure, logic, features, and procedures. Throughout the project, the documentation was methodically produced to support the phases of development, testing, deployment, and maintenance.

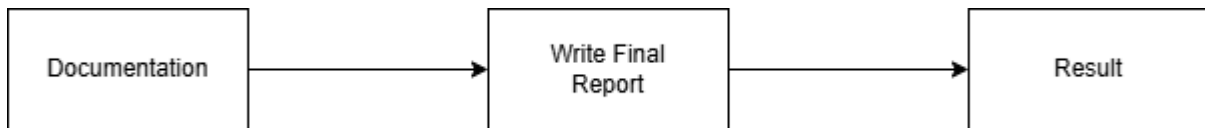


Figure 3.38 Testing Phase

The documentation of **SearchCare** will be prepared with clarity, detail, and accessibility in mind. It serves as a reference not only for maintenance and future upgrades but also as a valuable learning resource for anyone exploring the codebase, structure, or logic of the application. Proper documentation will enhance the app’s sustainability, scalability, and professionalism in a real-world healthcare solution context.

3.10 Software and Hardware Requirement

3.10.1 Software Requirement

Table 7 Software Requirements

Component	Specification/Tool	Purpose
Operating System	Windows 10/11 or macOS (latest version)	To run Android Studio and Flutter SDK
IDE	Android Studio	App development and testing
Programming Language	Dart	Core language for Flutter framework
Framework	Flutter SDK (v3.0 or above)	UI and mobile app development
Database	Firebase Firestore	Cloud-hosted NoSQL database for app data
Authentication Service	Firebase Authentication	Secure user login, registration, and role-based access
Cloud Storage	Firebase Storage	Upload and store profile pictures and assets

Dependency Manager	pubspec.yaml (Flutter package manager)	Manages third-party packages and plugins
Emulator/Testing Tool	Android Emulator / Physical Device	App preview and testing
Design Tool (optional)	Figma / Adobe XD	UI/UX wireframes and prototype design
Web Browser	Chrome / Firefox	Firebase Console access, documentation
Version Control (optional)	Git + GitHub	Source code management and collaboration

3.10.2 Hardware Requirement

Table 8 Hardware Requirement

Component	Minimum Specification	Recommended Specification
Processor (CPU)	Intel Core i3 / AMD Ryzen 3	Intel Core i5 or higher / AMD Ryzen 5 or higher
RAM	8 GB	16 GB or more
Storage	20 GB free disk space	SSD with at least 100 GB free
Graphics Card	Integrated Graphics	Dedicated GPU (for faster emulation)
Display	13" or larger with 1366×768 resolution	Full HD (1920×1080) or higher
Internet Connection	Required	Stable connection for Firebase and updates
Android Test Device (Phone)	Android 8.0+ device with camera, 2 GB RAM	Android 10+ device with 4 GB RAM or more

Chapter 4 Implementation and Design

4.1 Introduction

Implementation is a phase that apply what had been discussed, analysed and designed in the previous chapter. It begins with building the user interface and user experience, typically involving the services of experts (Saeidnia et al., 2023). The production process will then employ the appropriate programming languages and platforms, i.e., Java and Android Studio (Saeidnia et al., 2023; Ramakrishnam Raju et al., 2022). Integration of hardware units and sensors may be needed for certain applications (Ramakrishnam Raju et al., 2022). While being implemented, performance factors measurement such as client satisfaction, profitability of projects, and quality is essential (Zamim, 2021). The application needs to be usable by users, subsequently, assessed through the utilization of tools like questionnaires (Saeidnia et al., 2023). The integration of real-time data collection and management systems could be provided as additional functionalities (Intawong et al., 2020). Finally, the efficiency of the application has to be improved by user engagement and feedback throughout the implementation period to cater to individual requirements (Intawong et al., 2020).

4.2 Required Component

To develop a mobile application using Flutter, I need to download several pieces of software in order to have full functionality. I first download Android Studio, which contains the Android SDK, emulator, and debugging tools necessary to construct and test my application on Android. I then download the Flutter SDK, which is actually the framework that enables you to develop cross-platform apps using a common codebase written in Dart. As my app grows to include features like user authentication, real-time databases, and cloud storage—like in a drug delivery app—I add Firebase by introducing Firebase CLI tools and FlutterFire packages like `firebase_core`, `firebase_auth`, and `cloud_firestore`. These tools work together to create an integrated development experience where device deployment is handled by Android Studio, the UI and logic are driven by Flutter, and backend services like login and data storage are handled by Firebase.

4.2.1 Android Studio

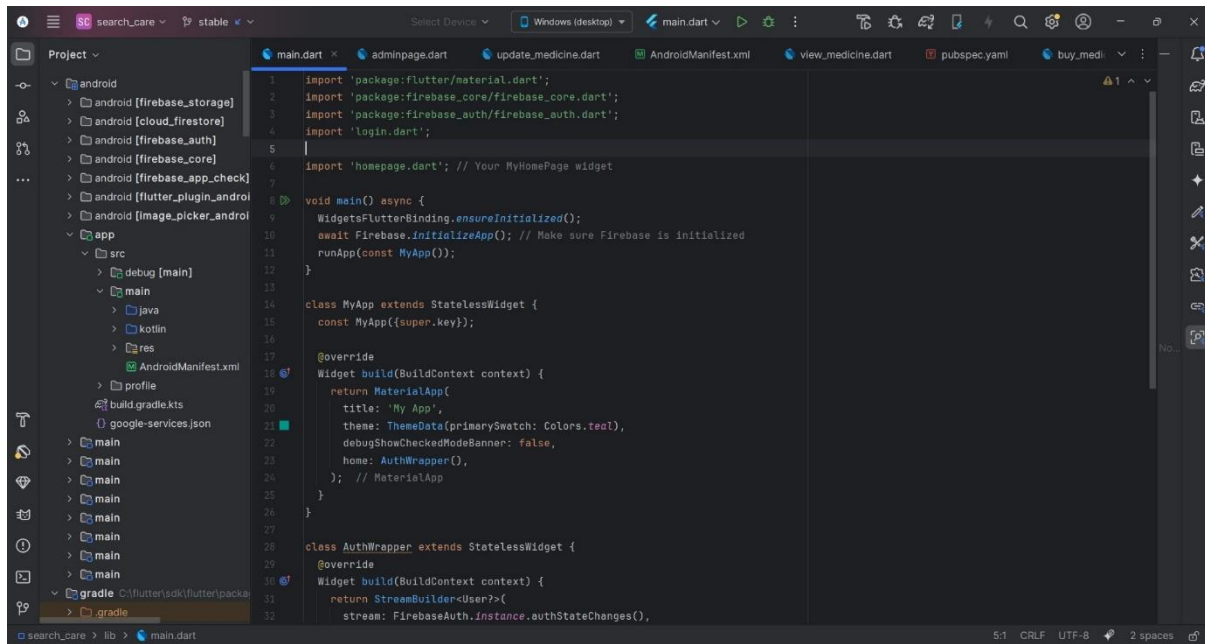


Figure 41 Android Studio workspace

Android Studio is the official Integrated Development Environment (IDE) for developing Android applications, designed and maintained by Google. It is a collection of tools used to test, debug, and develop Android applications. Android Studio includes the Android SDK (Software Development Kit), which contains libraries, build tools, and APIs that are required to compile and run Android apps. It also features an app emulator to test an app without physically having a device, a smart code editor with intelligent completion, and Gradle, an automated build framework that handles dependencies and project configurations. It also features support for version control tools and integration with performance profilers, layout editors, and debuggers, which make it a robust native Android and cross-platform development environment when combined with tools like Flutter.

4.2.2 Firebase Storage as a Database

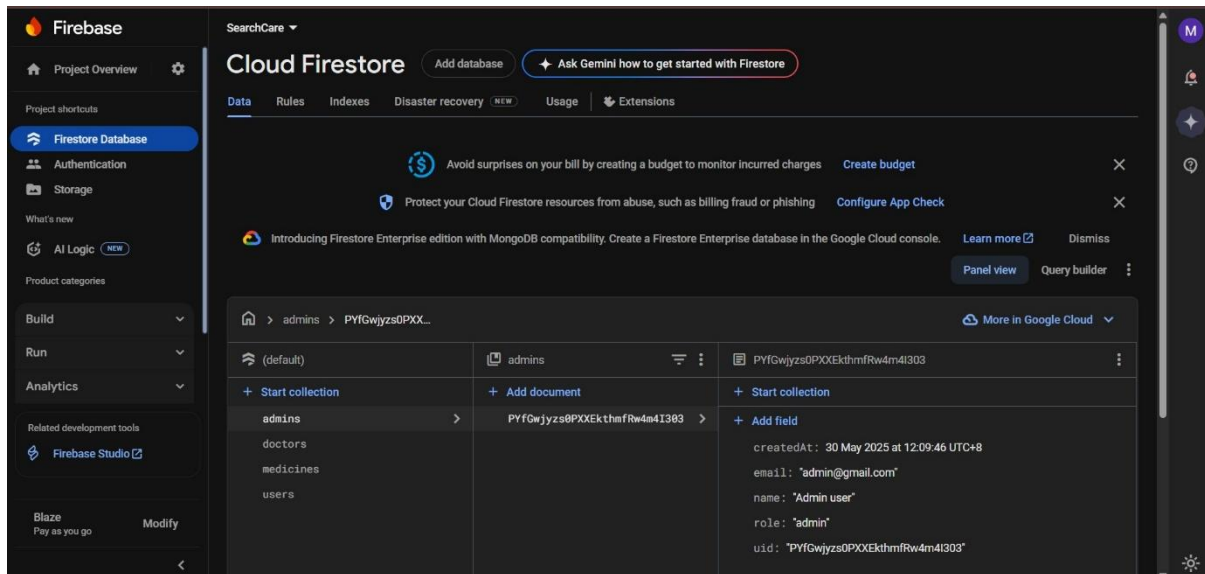


Figure 42 Firebase database workspace

Firebase is a powerful backend solution with real-time database and cloud storage features, and hence it is the best option for storing and managing various types of data in a medical app. On user registration by patients and doctors via the app, their information like names, contact numbers, role, and other useful details is securely stored in Cloud Firestore, the NoSQL cloud database offered by Firebase. The application also saves comprehensive medicine records, including names, prices, quantities, and prescription needs, in Firestore. Firebase also saves healthcare advice, which can be read and updated in real time, and maintains online consultation and appointment schedules, availability of doctors, appointment times, and history of consultations. This centralized and real-time database allows synchronized data among users and devices effortlessly, delivering constant, real-time data throughout the application.

4.2.3 Flutter as an open-source UI toolkit

```
Microsoft Windows [Version 10.0.26100.4202]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jason>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.32.0, on Microsoft Windows [Version 10.0.26100.4202], locale en-MY)
[✓] Windows Version (11 Home Single Language 64-bit, 24H2, 2009)
[✓] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
[✓] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Build Tools 2019 16.11.47)
    X The current Visual Studio installation is incomplete.
      Please use Visual Studio Installer to complete the installation or reinstall Visual Studio.
[✓] Android Studio (version 2024.3.2)
[✓] VS Code (version 1.100.2)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 1 category.
```

Figure 43 Installed Flutter

Flutter is an open-source UI library developed by Google that allows developers to build high-performance, cross-platform mobile applications from a shared codebase. It runs on the Dart language and supports a rich set of pre-configured widgets that allow for rapid and expressive UI development across Android and iOS platforms. In mobile application development, Flutter considerably accelerates the process with tools such as hot reload, allowing developers to see changes immediately without having to restart the app. It is also capable of integrating with several backend services such as Firebase and thus facilitating simple implementation of services such as authentication, database interaction, and storage in the cloud. With Flutter, app developers can build beautiful, highly responsive apps that provide a native-like user experience while having platform consistency, lower development time, and effort.

4.3 System Flow

4.3.1 System Flow for User/Patient

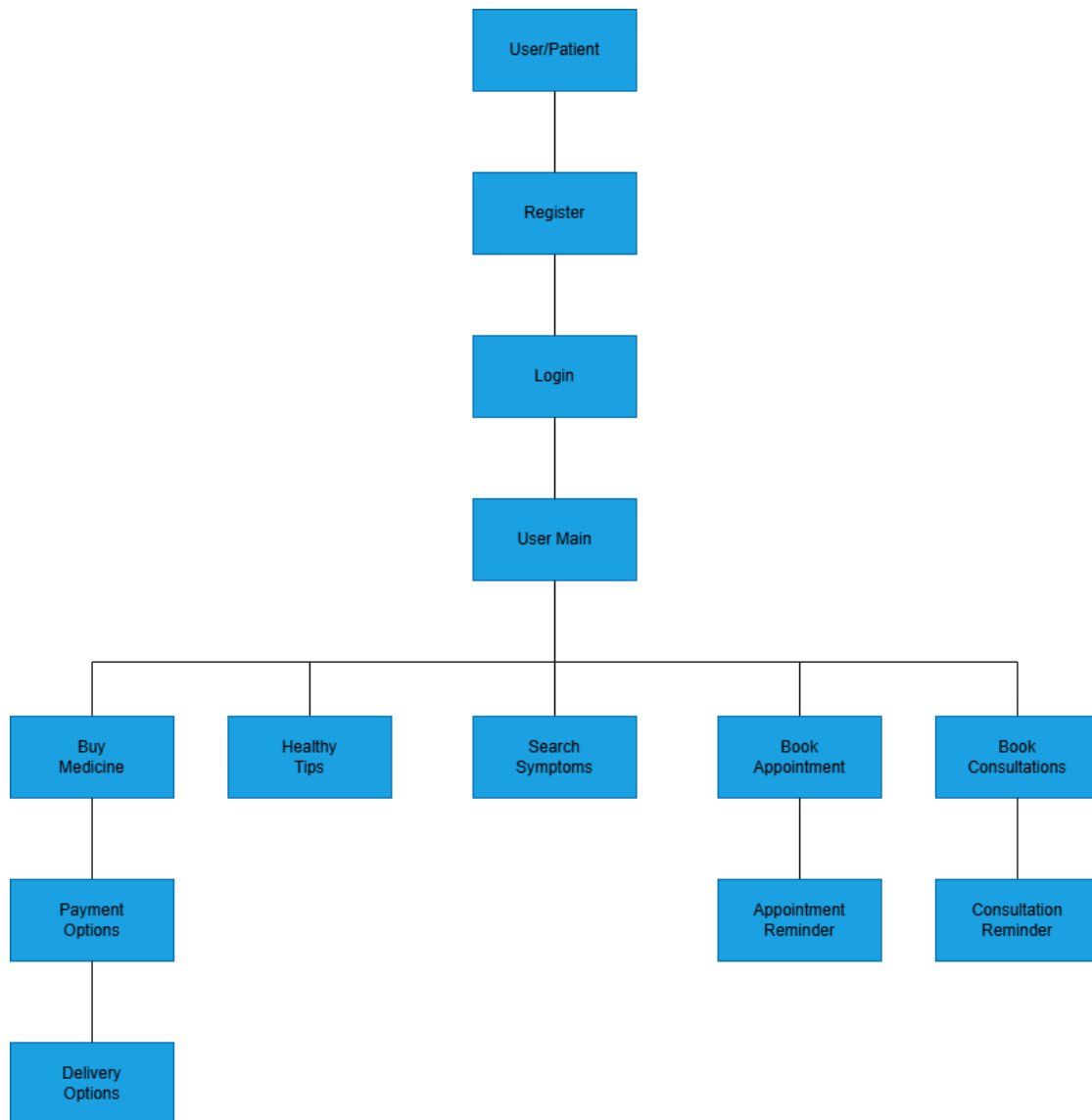


Figure 44 App Navigation of the mobile application for user/patient

4.3.2 System Flow for Doctor

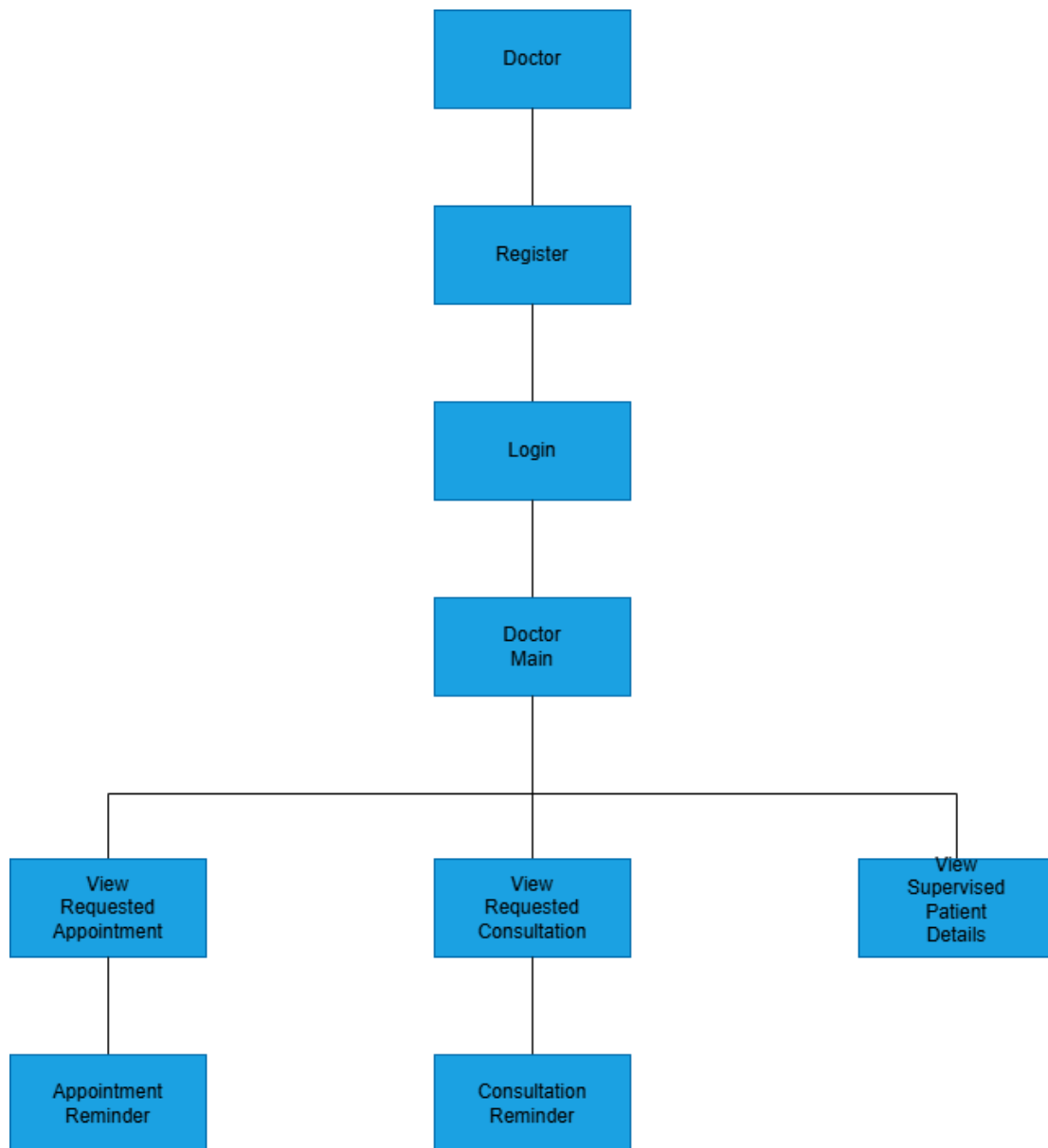


Figure 45 App Navigation of the mobile application for doctor

4.4 Module Implementation

Module deployment under the auspices of a smart healthcare mobile application implies the systematic installation and integration of various functional modules (or elements) after the initial development phase is complete. Having developed and tested the application, every module that relates to specific features such as user authentication, medicine management, scheduling of appointments, doctor-patient interaction, and health guidance has to be configured, integrated, and deployed into the live environment. This is to make sure all the components are working together and securely in the live application.

4.4.1 Splash Screen

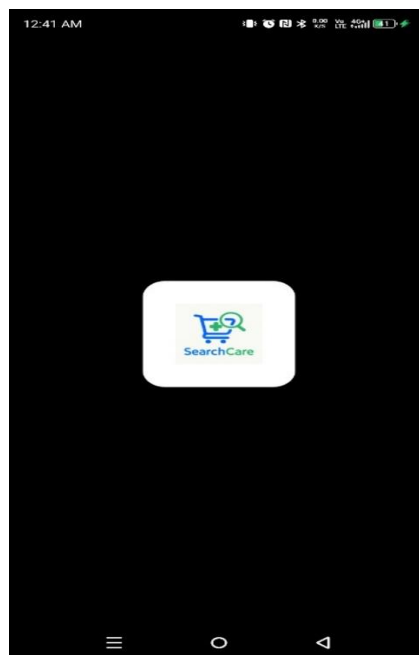


Figure 46 Splash screen

The first screen that appears when the SearchCare application is activated is called the Splash Screen. It fulfils the following functions:

Branding: To strengthen brand identity, the app's name and logo are displayed.

Loading Resources: Starts up essential services including Storage,Firestore, and Firebase Authentication.**Navigation Decision:**

Directs the user to the relevant module after determining whether they are already authenticated:

Authenticated user → Role-based home screen (admin, doctor, or user)

Not verified → Welcome/Login screen

4.4.2 Log In Screen

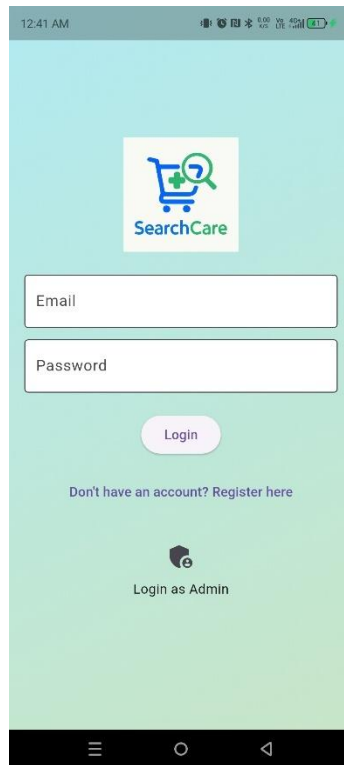


Figure 47 Login screen

The login screen provides dedicated functionality to support and facilitate specific healthcare workflows. It authenticates users securely using Firebase Authentication and redirects them to their respective dashboards based on role (patient, doctor, or admin).

4.4.3 Register Account Type

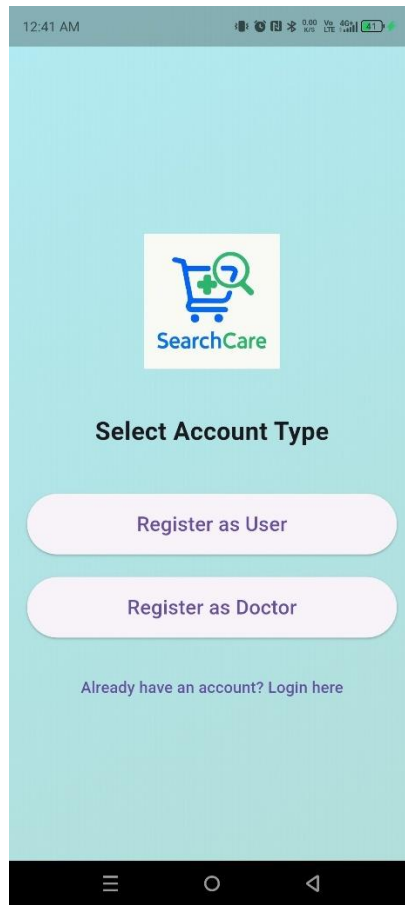


Figure 48 Register Account Type

The choose registration account type provides dedicated functionality to support and facilitate specific healthcare workflows. This screen integrates tightly with Firebase to support its specific functionality and enhance healthcare delivery for users.

4.4.4 New User/Patient Account Registration

The screenshot displays the 'User Registration' screen in the SearchCare mobile application. The interface is light blue and contains the following elements:

- SearchCare Logo:** Located at the top left, featuring a shopping cart icon with a green cross and a magnifying glass.
- Registration Fields:** A series of input boxes for 'Email', 'Password', 'Phone Number', 'City', 'Country', 'Username', 'Full Name', 'ID Number', and 'Address'.
- Gender Selection:** A dropdown menu labeled 'Gender: Male' is positioned at the bottom left and top right of the form.
- Medical History Checkboxes:** Two checkboxes are located on the right side: 'Have criminal record' and 'Have medical history', both currently unchecked.
- Registration Button:** A prominent purple 'Register' button is centered at the bottom of the form.
- Navigation:** A link labeled 'Back to account type selection' is located at the bottom right of the form.

Figure 49 New User/Patient Registration

The new user/patient registration provides dedicated functionality to support and facilitate specific healthcare workflows. This screen captures personal details, contact information, and medical background for new patients to ensure proper onboarding into the healthcare system.

4.4.5 New Doctor Account Registration

The screenshot displays the 'Doctor Registration' screen in the SearchCare mobile application. The interface is light blue and features the SearchCare logo at the top left. The title 'Doctor Registration' is centered above a purple camera icon. The registration form consists of several input fields: Email, Password, Phone Number, City, and Country on the left side; and Phone Number, City, Country, Gender (set to 'Male'), Full Name, License Number, Expertise, and Workplace on the right side. A prominent purple 'Register' button is located at the bottom right, with a blue link 'Back to account type selection' positioned below it. The bottom of the screen shows the standard Android navigation bar.

Figure 50 New Doctor Registration

The new doctor registration provides dedicated functionality to support and facilitate specific healthcare workflows. It allows new medical professionals to register by submitting credentials, specialization, and verification documents, integrating role-based access for doctors.

4.4.6 User Dashboard

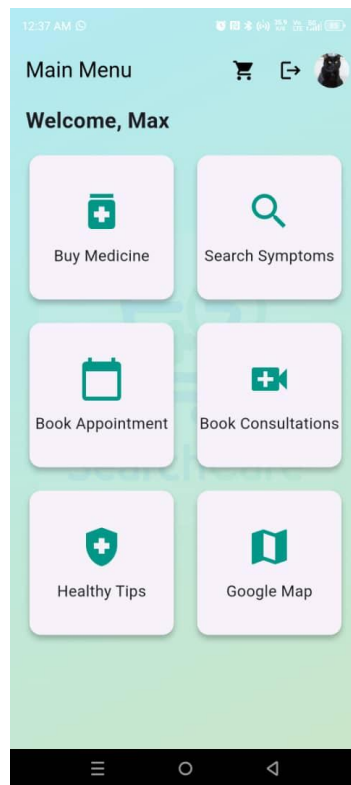


Figure 51 User/Patient Dashboard

The user dashboard provides dedicated functionality to support and facilitate specific healthcare workflows. It displays an overview of key user interactions, such as appointment status, consultation history, medicine orders, and health tips.

4.4.7 User Profile Setting



Figure 52 User Profile Setting Page

The user profile setting provides dedicated functionality to support and facilitate specific healthcare workflows. It allows users to update personal information, medical preferences, and profile photo, ensuring accurate and personalized care delivery.

4.4.8 Buy Medicine Page



Figure 53 Buy Medicine Page

The buy medicine page provides dedicated functionality to support and facilitate specific healthcare workflows. Users can browse a list of available medicines, apply filters by category, and add selected items to their cart for purchase.

4.4.9 View Medicine Detail Page



Figure 54 View Medicine Detail Page

The view medicine detail page provides dedicated functionality to support and facilitate specific healthcare workflows. Displays comprehensive details such as medicine composition, dosage, usage instructions, price, and stock availability.

4.4.10 Confirm Medicine Payment Page

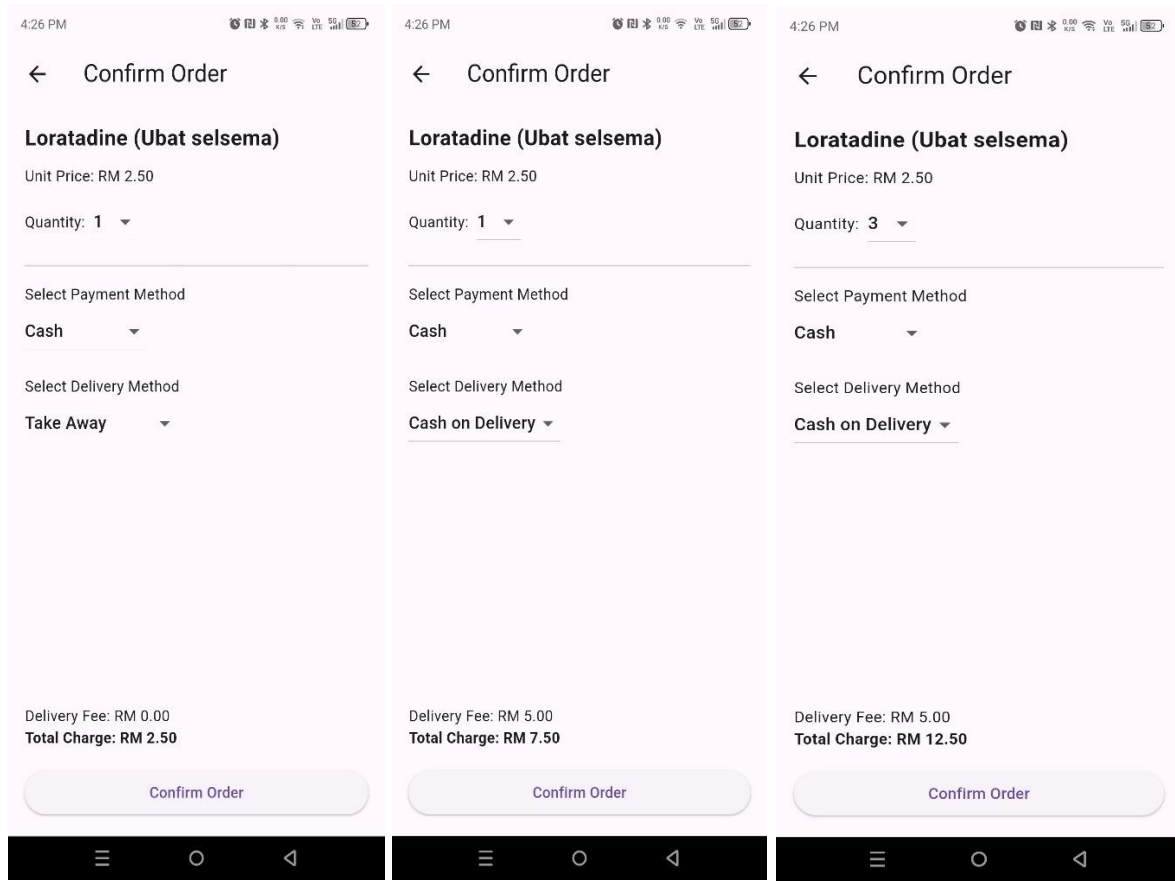


Figure 55 Confirm Medicine Payment Page

The confirm medicine payment page provides dedicated functionality to support and facilitate specific healthcare workflows. Facilitates secure payment transactions through integrated gateways, displaying cost breakdown and delivery estimates. It also has feature for user to choose increase medicine quantity to buy and choose delivery method either cash on delivery (COD) or take away.

4.4.11 Medicine Payment Confirmation Details

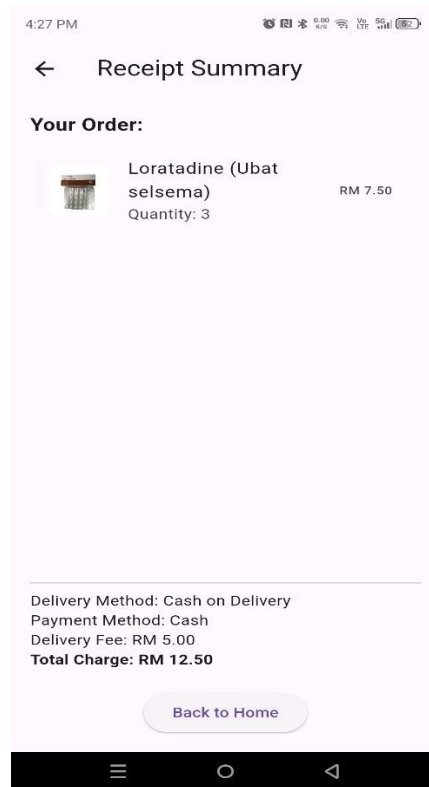


Figure 56 Medicine Payment Confirmation Details

The medicine payment confirmation details provide dedicated functionality to support and facilitate specific healthcare workflows. Facilitates secure payment transactions through integrated gateways, displaying cost breakdown and delivery estimates.

4.4.12 Medicine Orders History Page

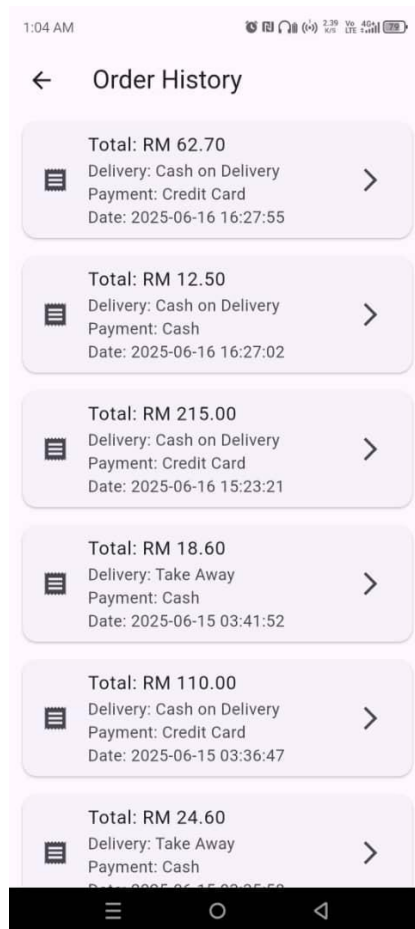


Figure 57 Medicine Orders History Page

The medicine orders history page provides dedicated functionality to support and facilitate specific healthcare workflows. Enables users to track past medicine purchases, download invoices, and monitor delivery statuses.

4.4.13 Receipt Summary Details Page

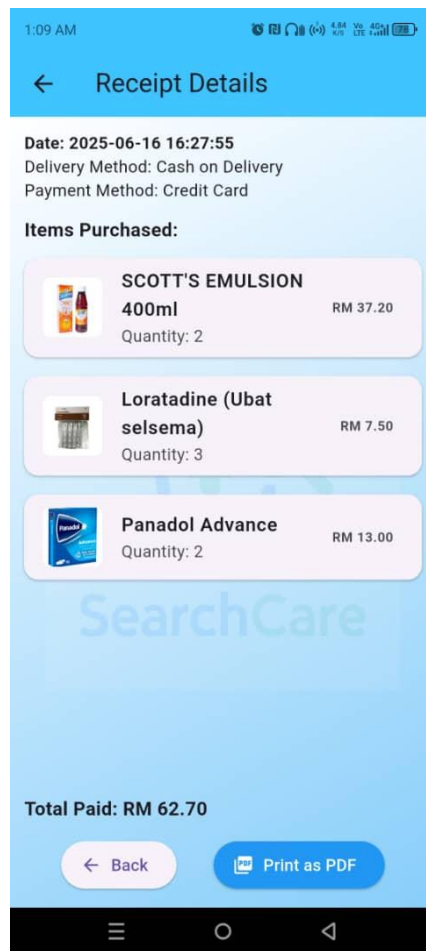


Figure 58 Receipt Summary Details Page

The receipt summary details page provides dedicated functionality to support and facilitate specific healthcare workflows. This screen integrates tightly with Firebase to support its specific functionality and enhance healthcare delivery for users.

4.4.14 Cart Screen

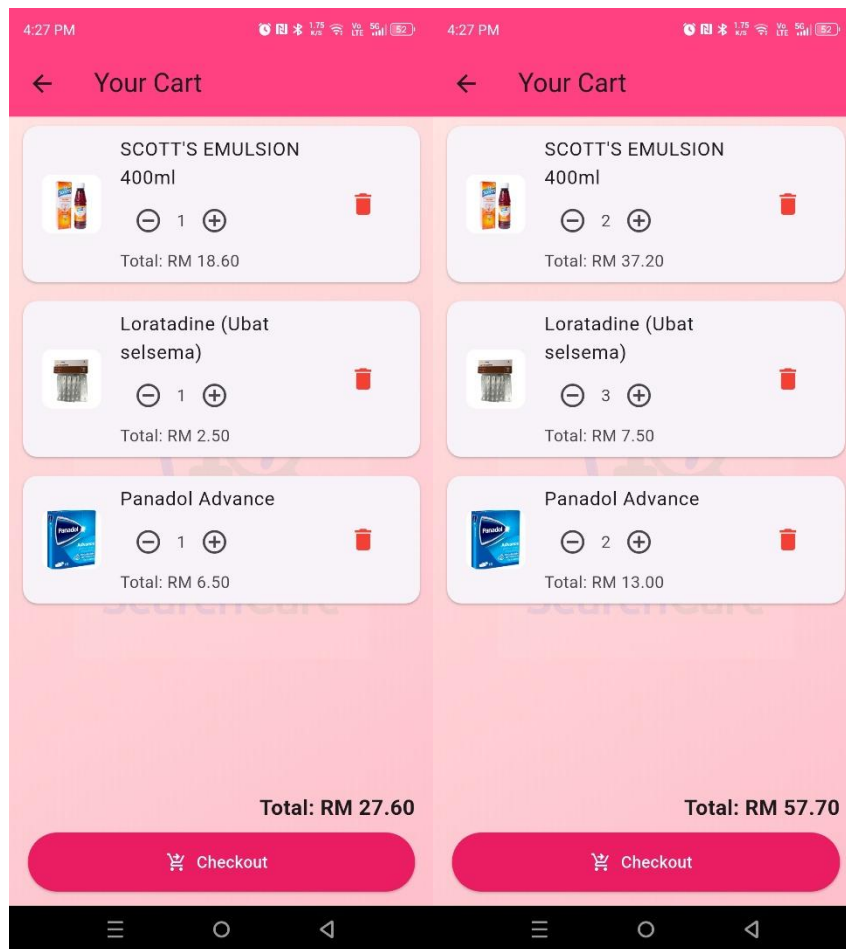


Figure 59 Cart Screen

The cart screen provides dedicated functionality to support and facilitate specific healthcare workflows. Manages selected medicines prior to checkout, allows quantity updates or deletions, and provides total cost summary.

4.4.15 Confirm Multiple Medicine Payment Page

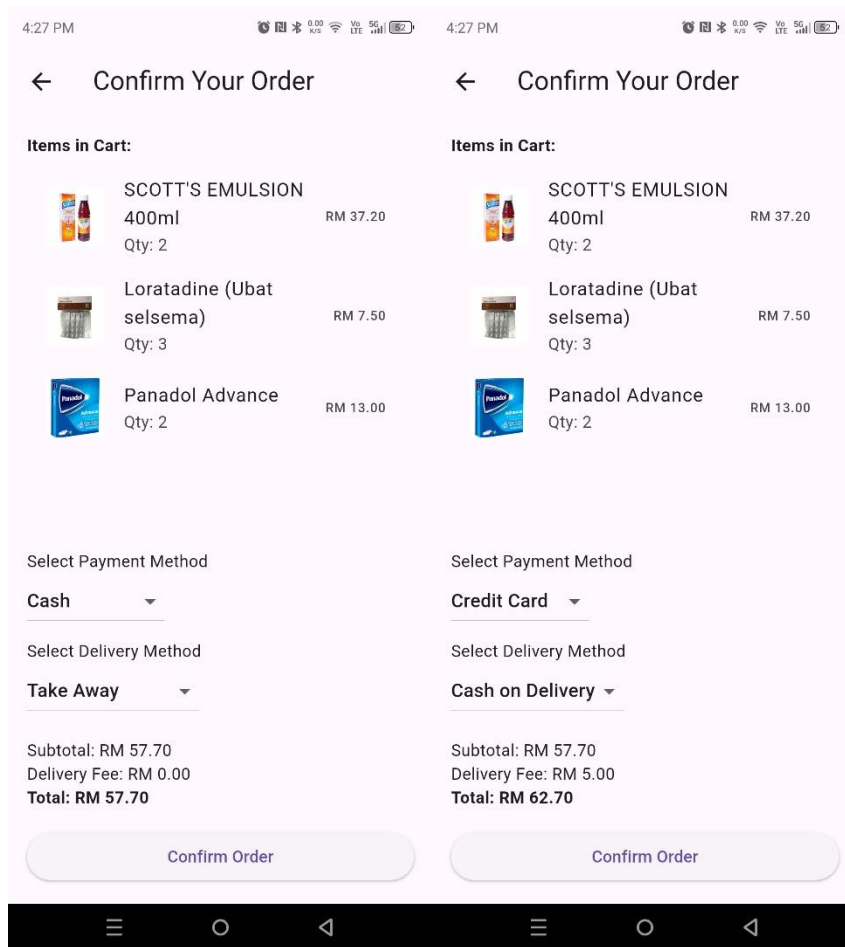


Figure 60 Confirm Multiple Medicine Payment Page

The confirm multiple medicine payment confirmation details provides dedicated functionality to support and facilitate specific healthcare workflows. Facilitates secure payment transactions through integrated gateways, displaying cost breakdown and delivery estimates.

4.4.16 Multiple Medicine Payment Confirmation Details

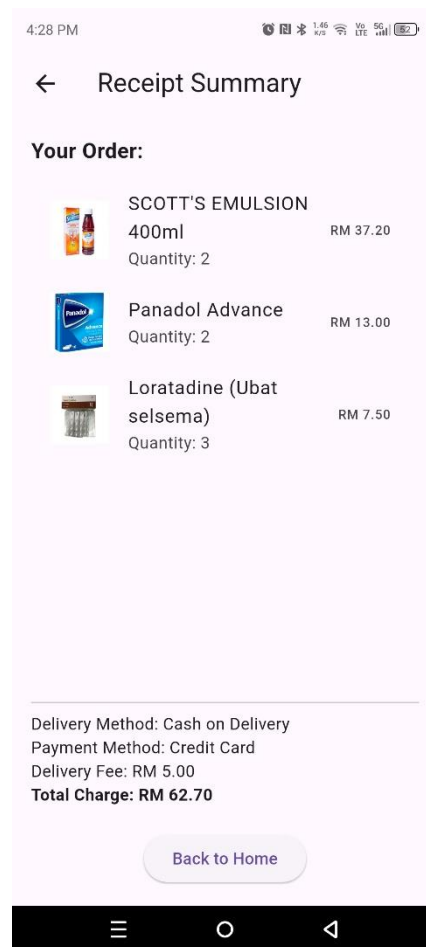


Figure 61 Multiple Medicine Payment Confirmation Details

The confirm multiple medicine payment receipt confirmation details provides dedicated functionality to support and facilitate specific healthcare workflows. The multiple medicine payment receipt summary details page provides dedicated functionality to support and facilitate specific healthcare workflows. This screen integrates tightly with Firebase to support its specific functionality and enhance healthcare delivery for users.

4.4.17 Search Symptoms Page

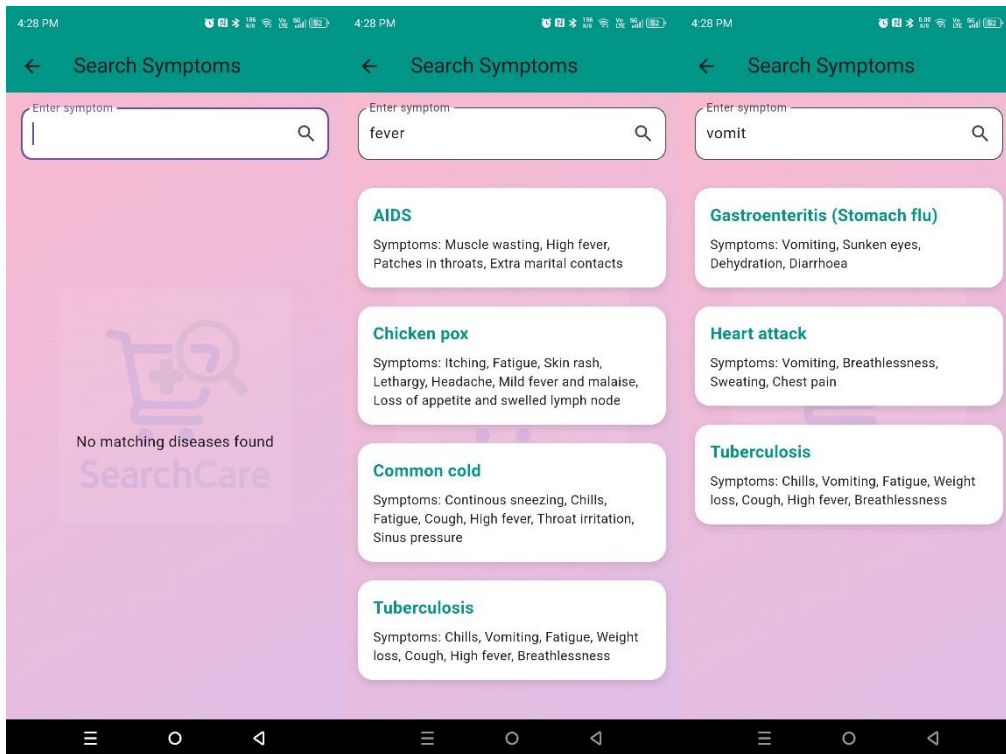


Figure 62 Search Symptoms Page

The search symptoms page provides dedicated functionality to support and facilitate specific healthcare workflows. Enables symptom-based input and returns suggested causes or matching consultations using Firebase-powered health logic.

4.4.18 Book Appointment Page

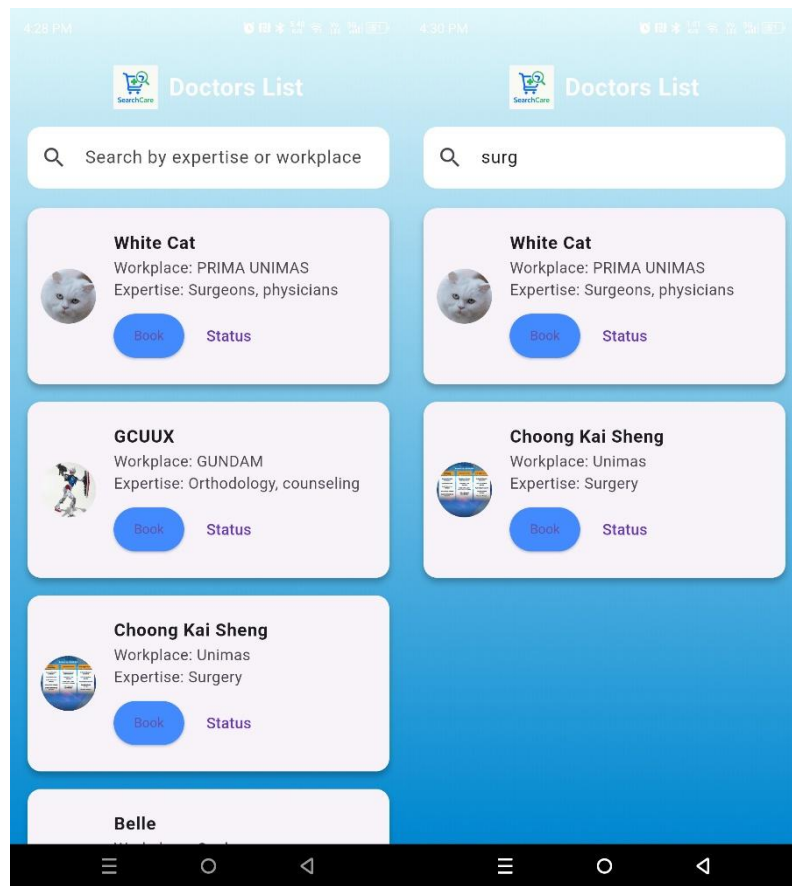


Figure 63 Book Appointment Page

The book appointment page provides dedicated functionality to support and facilitate specific healthcare workflows. Allows patients to choose date, time, and available doctors for clinical appointments, while tracking approval and rescheduling status.

4.4.19 Book Appointment Details Page

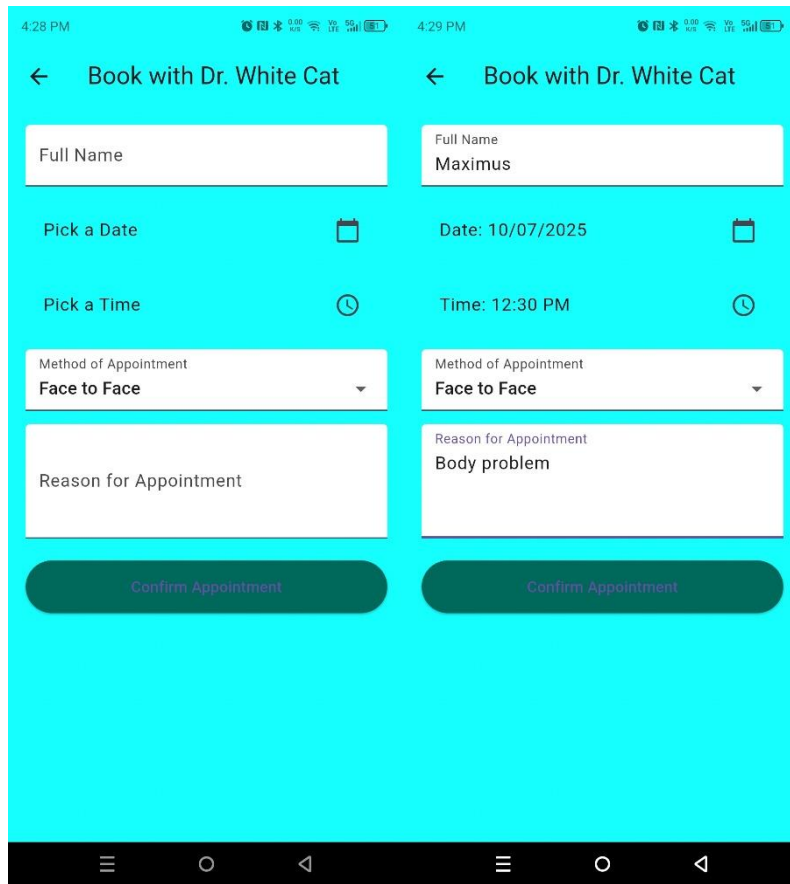


Figure 64 Book Appointment Details Page

The book appointment details page provides dedicated functionality to support and facilitate specific healthcare workflows. Allows patients to choose date, time, and available doctors for clinical appointments, while tracking approval and rescheduling status. It serves as a platform for patients to key-in the appointments booking process.

4.4.20 Appointment Status Page



Figure 65 Appointment Status Page

The appointment status provides dedicated functionality to support and facilitate specific healthcare workflows. Allows patients to choose date, time, and available doctors for clinical appointments, while tracking approval and rescheduling status.

4.4.21 Book Consultations Page

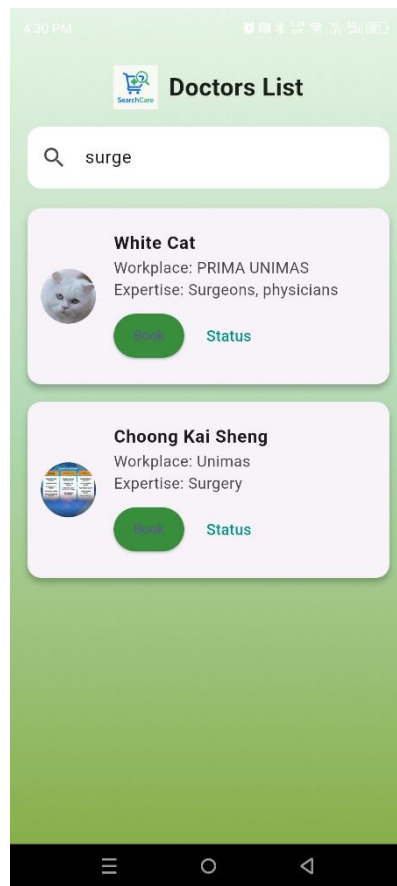


Figure 66 Book Consultations Page

The book consultation page provides dedicated functionality to support and facilitate specific healthcare workflows. Supports virtual health interactions by letting patient book and manage real-time or asynchronous consultations.

4.4.19 Book Consultations Details Page

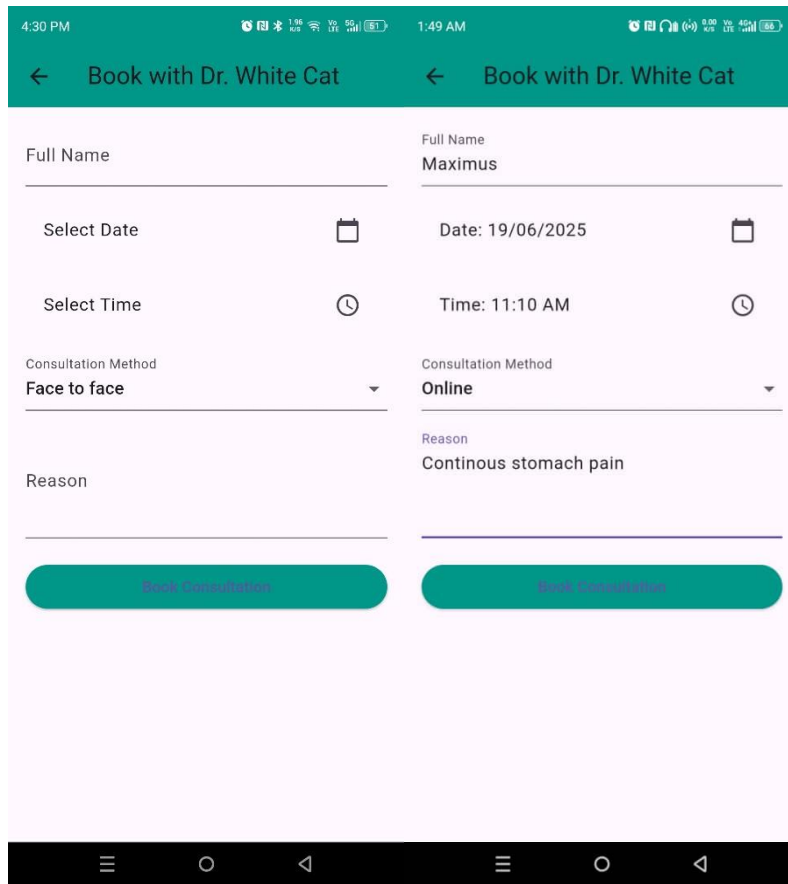


Figure 67 Book Consultations Details Page

The book consultation details page provides dedicated functionality to support and facilitate specific healthcare workflows. Supports virtual health interactions by letting patients book and manage real-time or asynchronous consultations. It serves as a platform for patients to key-in the consultations booking process.

4.4.20 Consultations Status Page

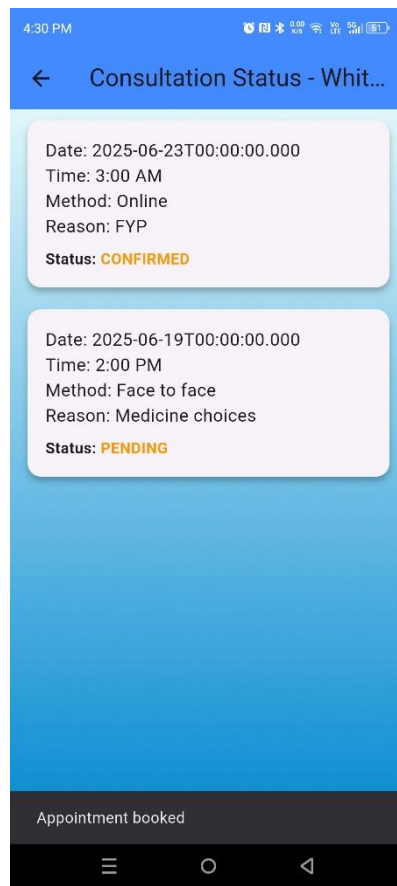


Figure 68 Consultations Status Page

The consultation status provides dedicated functionality to support and facilitate specific healthcare workflows. Supports virtual health interactions by letting patients book and manage real-time or asynchronous consultations.

4.4.21 View Healthy Tips Page

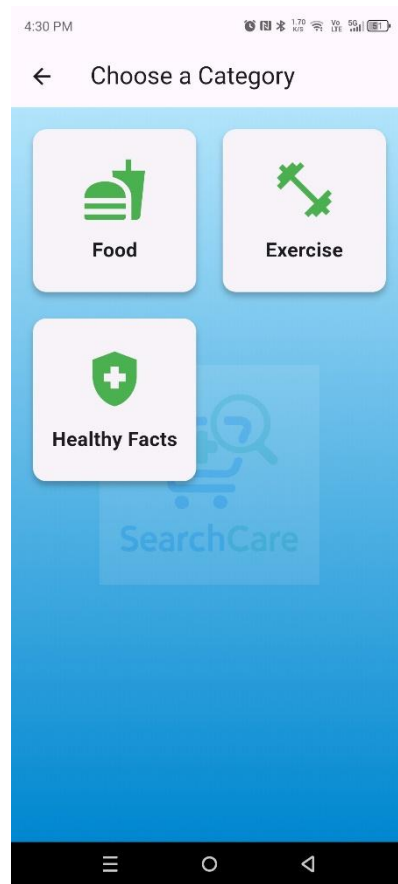


Figure 69 View Healthy Tips Page

The view healthy tips provide dedicated functionality to support and facilitate specific healthcare workflows. Presents categorized lifestyle advice including nutrition, exercise, and wellness facts, curated by medical experts.

4.4.22 View Healthy Food Tips Page

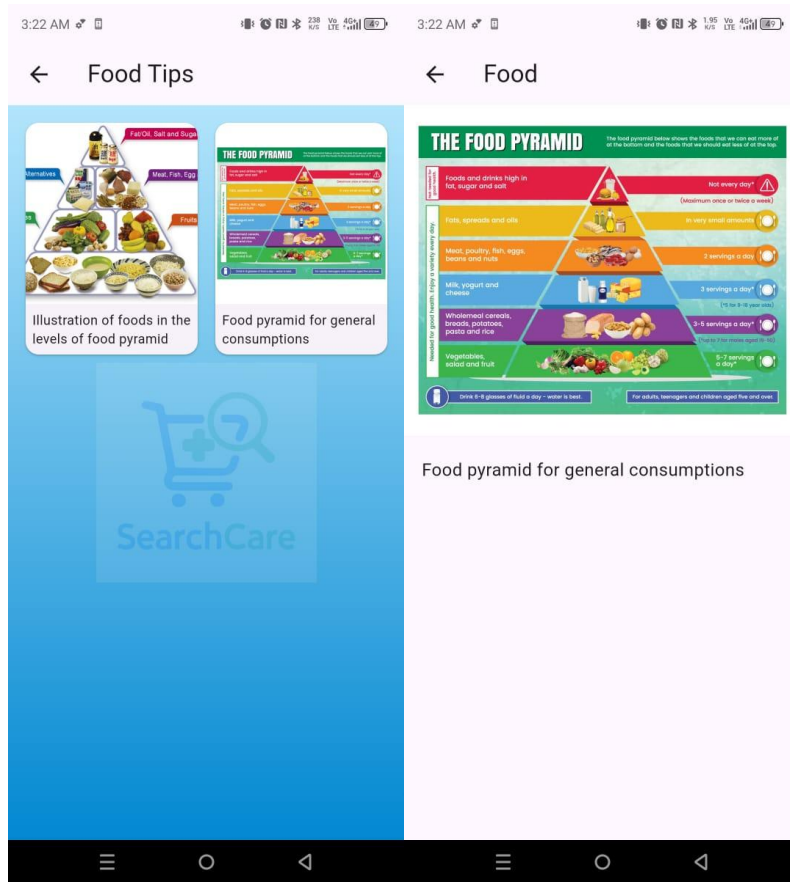


Figure 70 View Healthy Food Tips Page

The view healthy food provides dedicated functionality to support and facilitate specific healthcare workflows. Presents categorized lifestyle advice including nutrition, exercise, and wellness facts, curated by medical experts.

4.4.23 View Healthy Exercise Tips Page

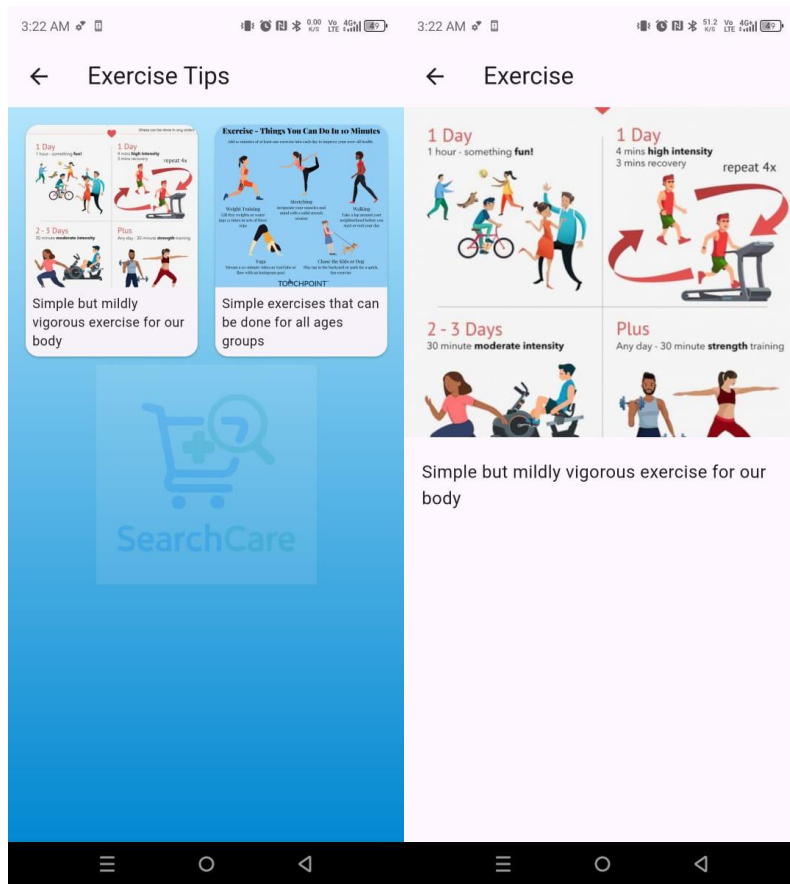


Figure 71 View Healthy Exercise Tips Page

The view healthy exercise tips page provides dedicated functionality to support and facilitate specific healthcare workflows. Presents categorized lifestyle advice including nutrition, exercise, and wellness facts, curated by medical experts.

4.4.24 View Healthy Facts Tips Page

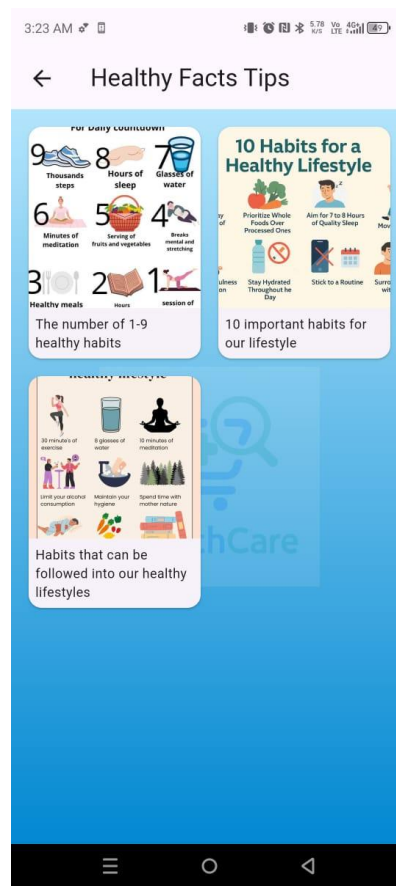


Figure 72 View Healthy Facts Tips Page

The view healthy facts provides dedicated functionality to support and facilitate specific healthcare workflows. Presents categorized lifestyle advice including nutrition, exercise, and wellness facts, curated by medical experts.

4.4.25 Doctor Dashboard

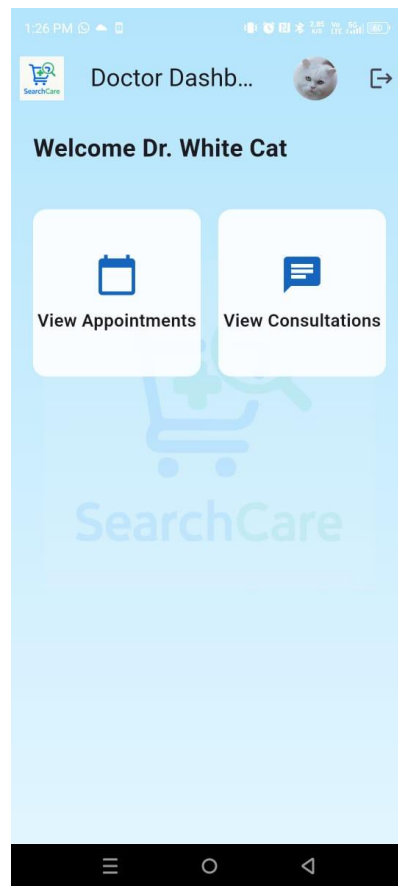


Figure 73 Doctor Dashboard Page

The doctor dashboard provides dedicated functionality to support and facilitate specific healthcare workflows. Provides doctors with tools to manage appointments and respond to consultations.

4.4.26 Doctor Setting Page

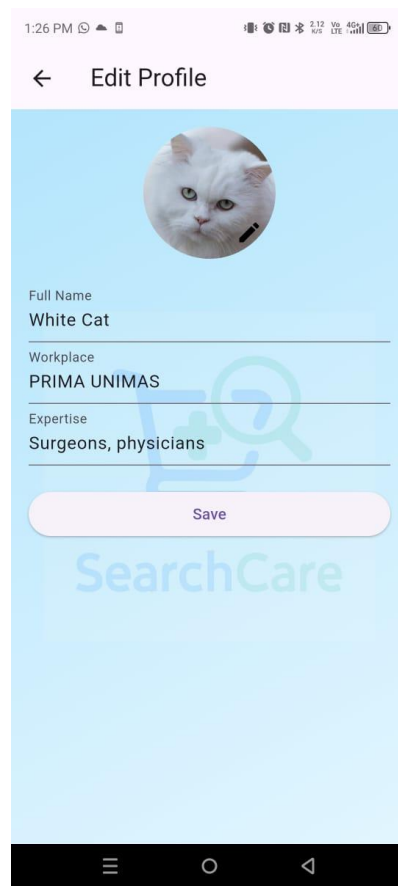


Figure 74 Doctor Setting Page

The doctor setting page provides dedicated functionality to support and facilitate specific healthcare workflows. Doctors or admins can update their personal or professional info, adjust availability, and manage notification settings.

4.4.27 View Requested Appointment

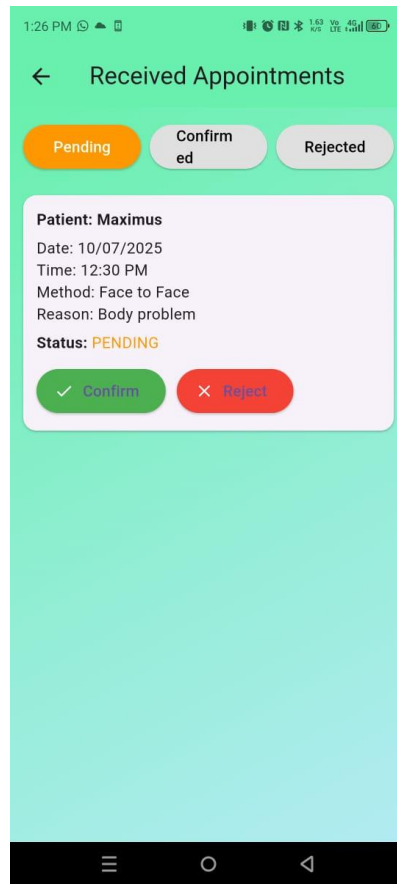


Figure 75 View Requested Appointment Page

The view requested appointment page provides dedicated functionality to support and facilitate specific healthcare workflows. Allows patients to choose date, time, and available doctors for clinical appointments, while tracking approval and rescheduling status.

4.4.28 Update Appointment Status Confirmation Page

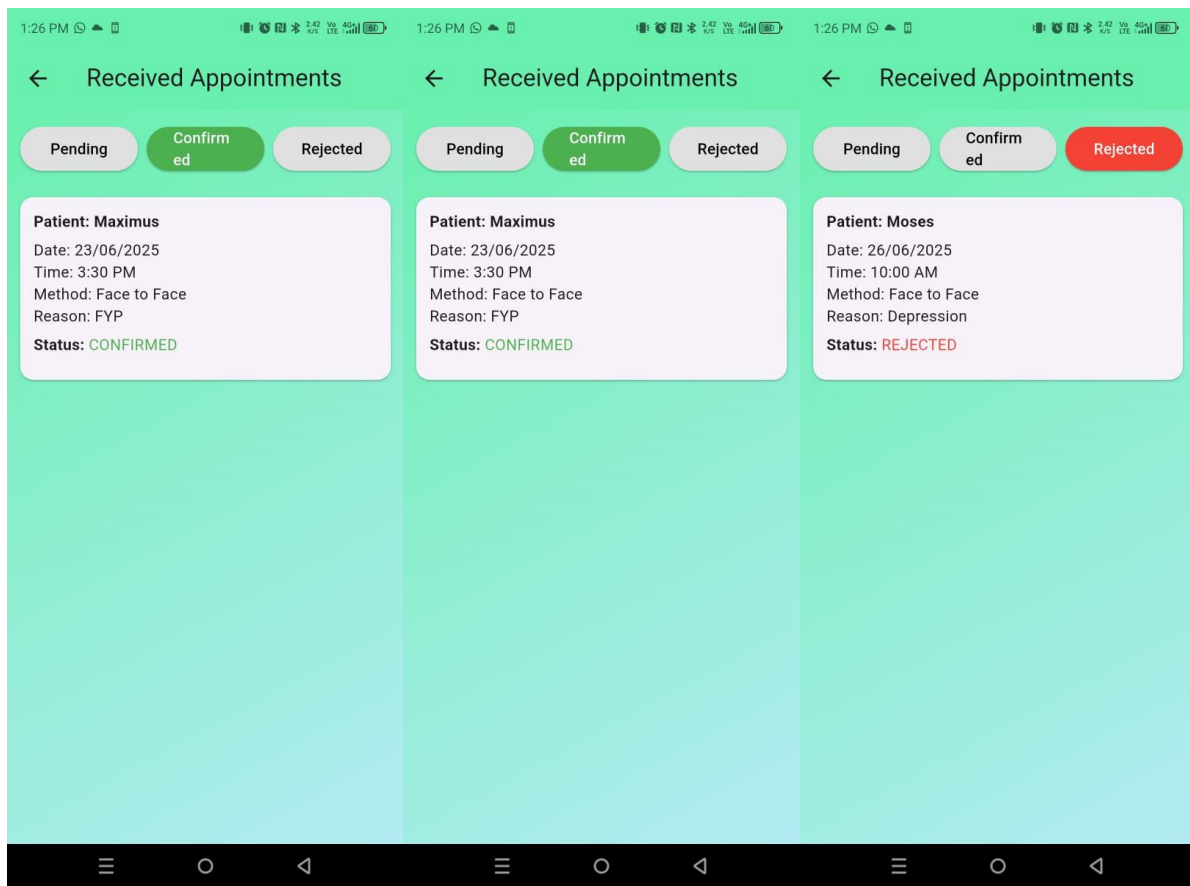


Figure 76 Update Appointment Status Confirmation Page

The appointment status confirmation page provides dedicated functionality to support and facilitate specific healthcare workflows. Informing patients that the doctor accepts the appointments of the requested date, time, and availability of the doctor for clinical appointments, while tracking approval and rescheduling status.

4.4.29 View Requested Consultations

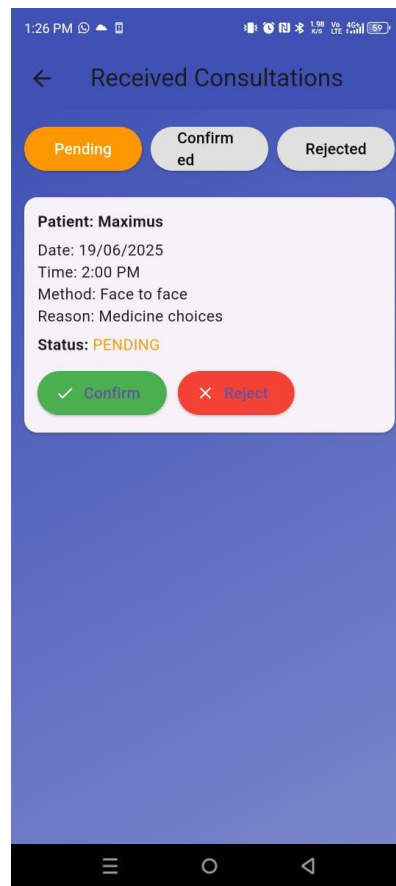


Figure 77 View Requested Consultations Page

The view requested consultation provides dedicated functionality to support and facilitate specific healthcare workflows. Supports virtual health interactions by letting patients book and manage real-time or asynchronous consultations.

4.4.30 Consultations Status Confirmation Page

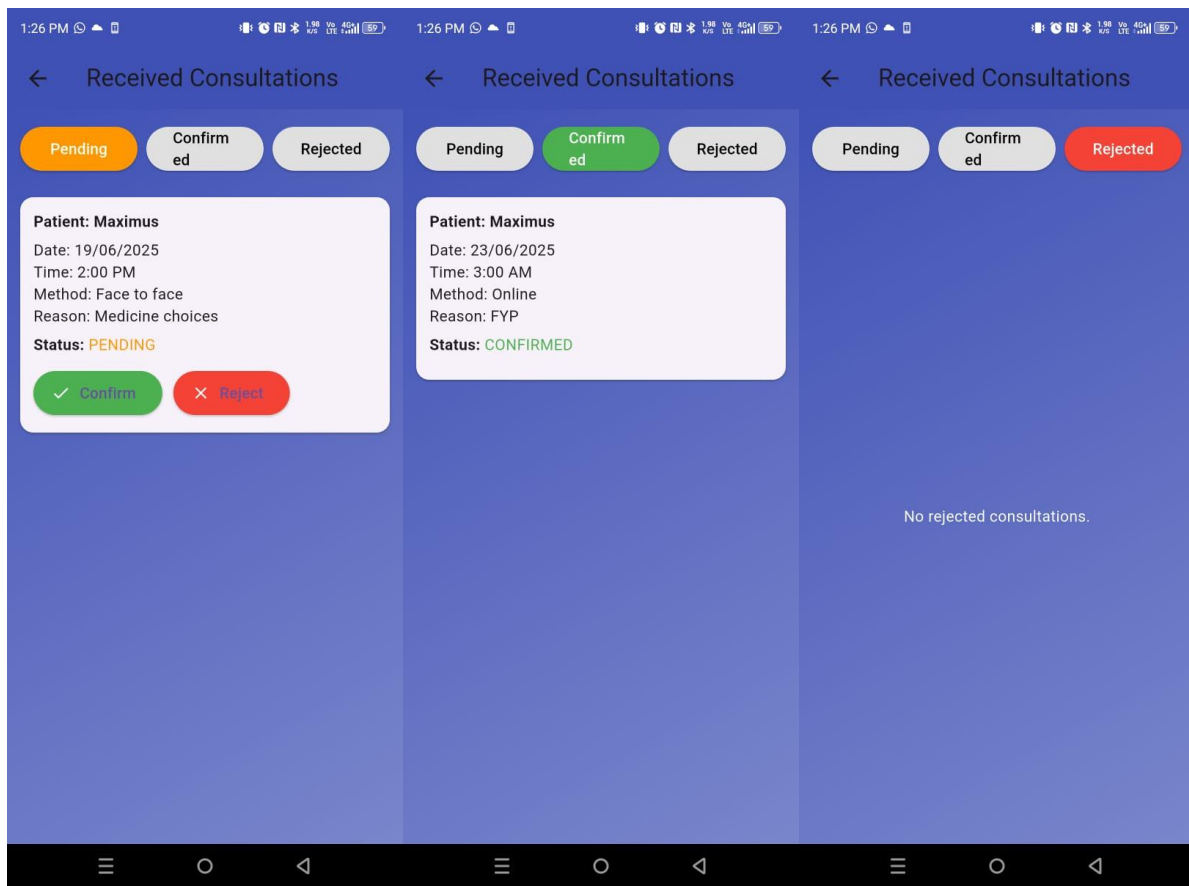


Figure 78 Consultations Status Confirmation Page

The consultation status confirmation provides dedicated functionality to support and facilitate specific healthcare workflows. Supports virtual health interactions by letting patients book and manage real-time or asynchronous consultations.

4.4.31 Admin Login Page

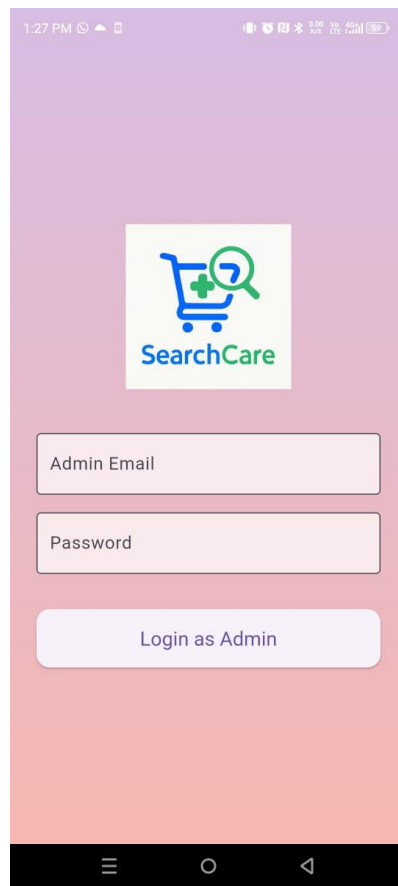


Figure 79 Admin Login Page

The admin login page provides dedicated functionality to support and facilitate specific healthcare workflows. It authenticates users securely using Firebase Authentication and redirects them to their respective dashboards based on role (patient, doctor, or admin).

4.4.32 Admin Dashboard Page



Figure 80 Admin Dashboard Page

The admin dashboard provides dedicated functionality to support and facilitate specific healthcare workflows. Allows admins to oversee platform activities, manage medicine inventory, verify doctor registrations, and update health content.

4.4.33 Add Medicine Page

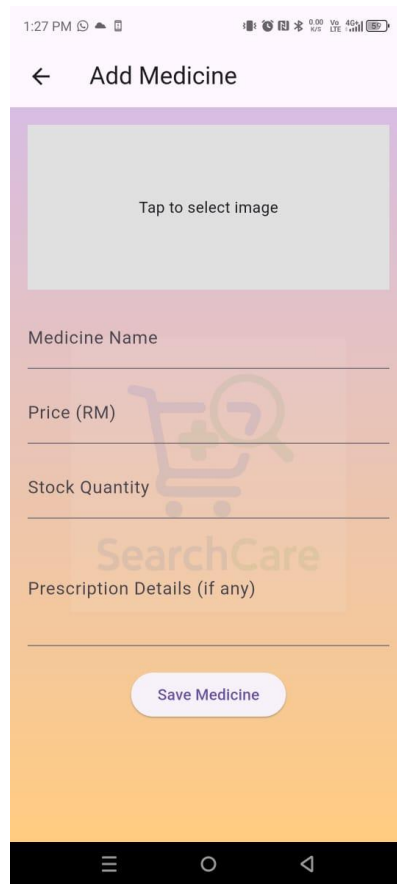


Figure 81 Add Medicine Page

The add medicine provides dedicated functionality to support and facilitate specific healthcare workflows. Admins can input new medicines with descriptions, categories, pricing, and image uploads to update the online pharmacy.

4.4.34 View Medicine Page

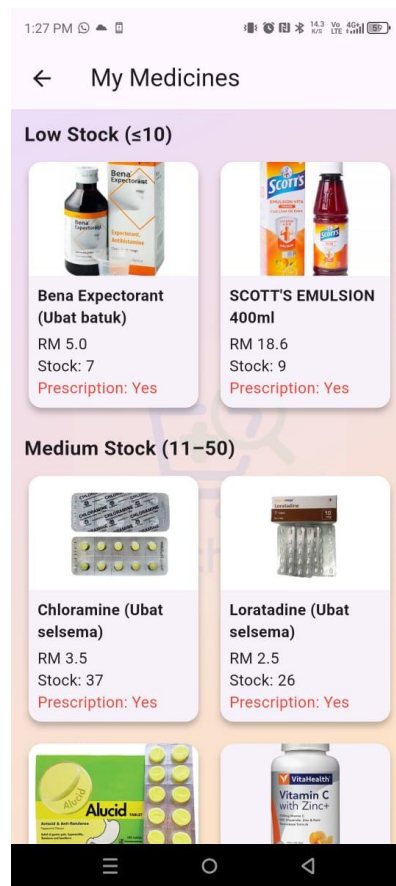


Figure 82 View Medicine Page

The view medicine provides dedicated functionality to support and facilitate specific healthcare workflows. Displays all added medicines with sorting and filtering options for easier inventory management.

4.4.35 Update Medicine Details Page

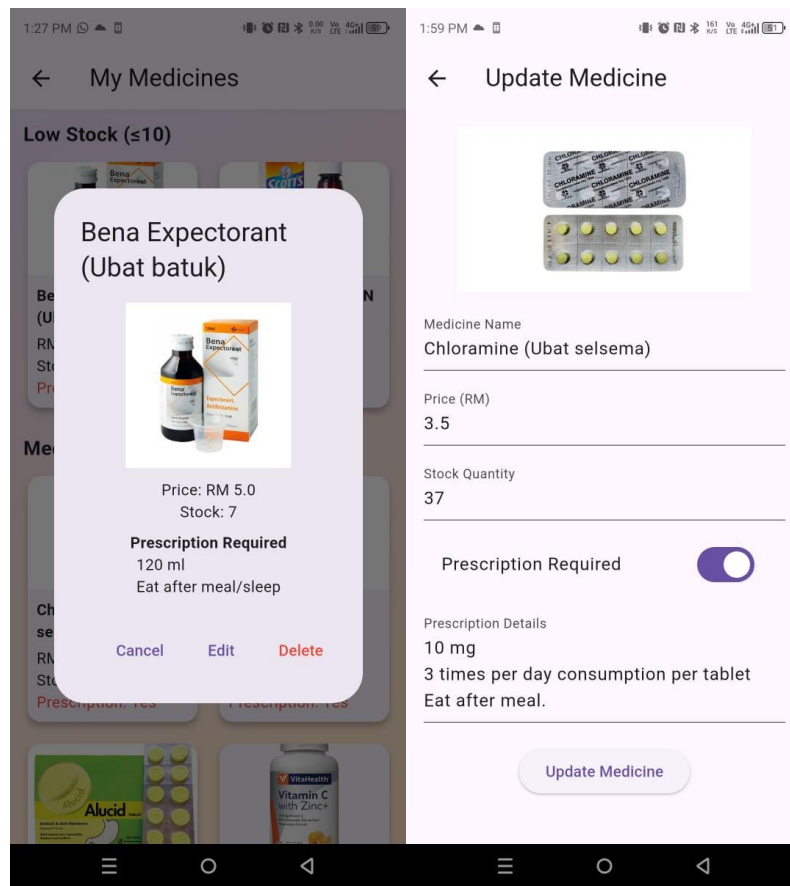


Figure 83 Update Medicine Details Page

The update medicine details provides dedicated functionality to support and facilitate specific healthcare workflows. Displays comprehensive details such as medicine composition, dosage, usage instructions, price, and stock availability.

4.4.36 Update Healthy Tips Details Page

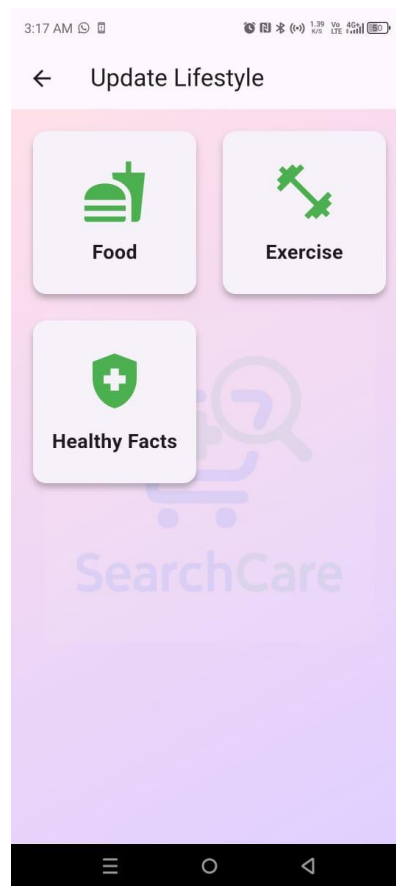


Figure 84 Update Healthy Tips Details Page

The update healthy tips details page provides dedicated functionality to support and facilitate specific healthcare workflows. Presents categorized lifestyle advice including nutrition, exercise, and wellness facts, curated by medical experts.

4.4.37 Update Healthy Food Details Page

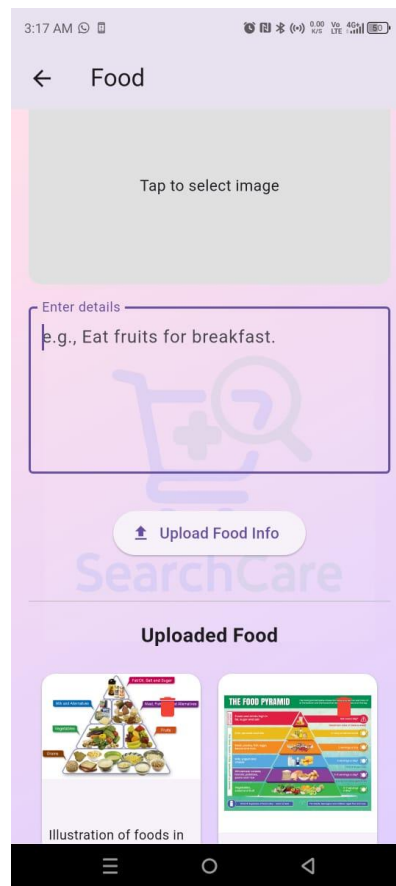


Figure 85 Update Healthy Food Details Page

The update healthy food provides dedicated functionality to support and facilitate specific healthcare workflows. Presents categorized lifestyle advice including nutrition, exercise, and wellness facts, curated by medical experts.

4.4.38 Update Healthy Exercise Details Page

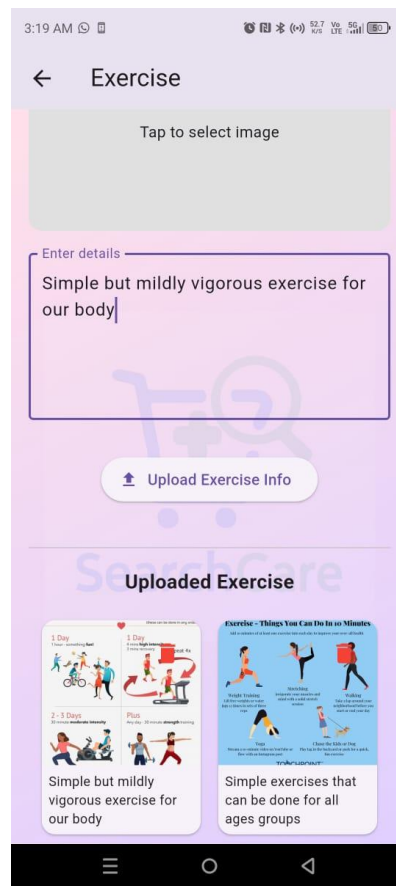


Figure 86 Update Healthy Exercise Details Page

The update healthy exercise details provides dedicated functionality to support and facilitate specific healthcare workflows. Presents categorized lifestyle advice including nutrition, exercise, and wellness facts, curated by medical experts.

4.4.39 Update Healthy Facts Details Page

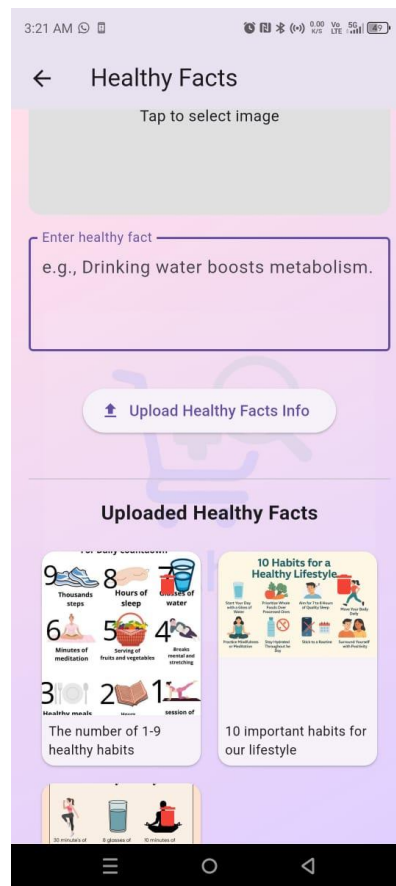


Figure 87 Update Healthy Facts Details Page

The update healthy facts page provides dedicated functionality to support and facilitate specific healthcare workflows. Presents categorized lifestyle advice including nutrition, exercise, and wellness facts, curated by medical experts.

4.4.40 Search Symptoms

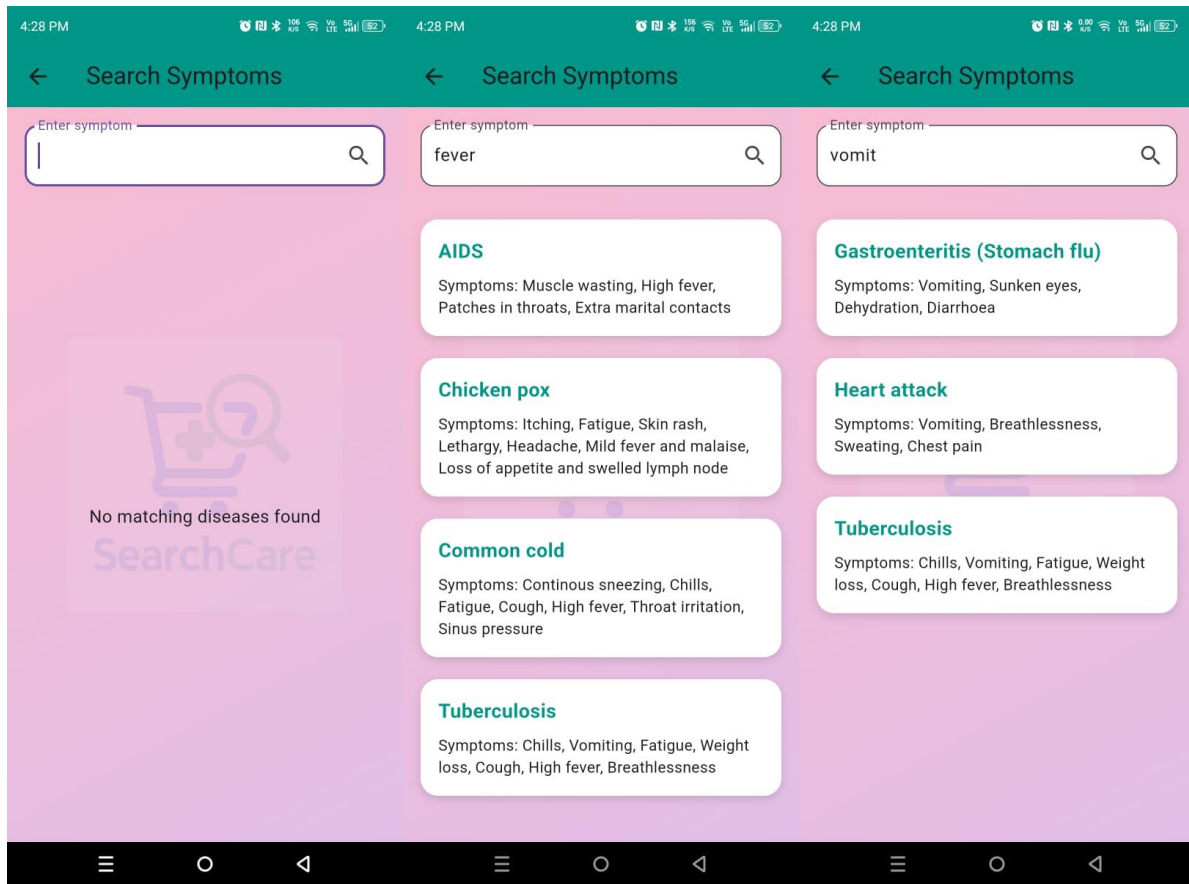


Figure 88 Search Symptoms Page

The search symptoms page provides dedicated functionality to support and facilitate specific healthcare workflows. Users experiences the symptom-based input and receives suggested causes or matching disease based in symptoms input using Firebase-powered health logic.

4.4.41 Update Symptoms

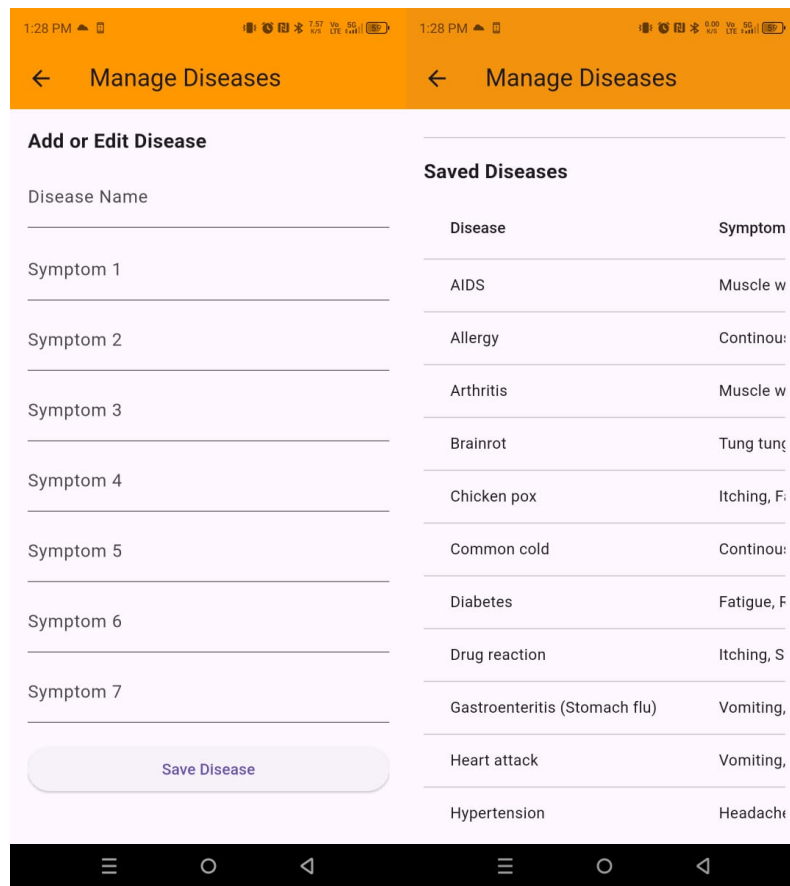


Figure 89 Update Symptoms Page

The update symptoms details page provides dedicated functionality to support and facilitate specific healthcare workflows. Enables symptom-based input and returns suggested causes or matching consultations using Firebase-powered health logic.

4.4.42 Update Location View

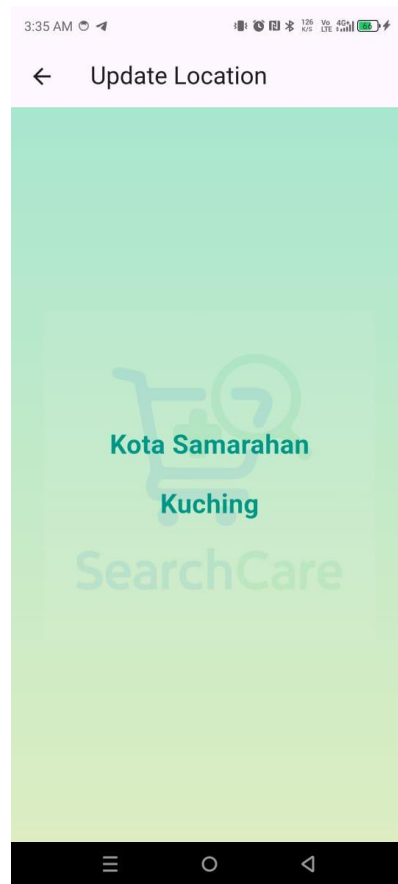


Figure 90 Update Location Page

The update location view page provides dedicated functionality to support and facilitate specific healthcare workflows. Enables location-based input and returns suggested matching city/regions using Firebase-powered health logic.

4.4.43 Location View

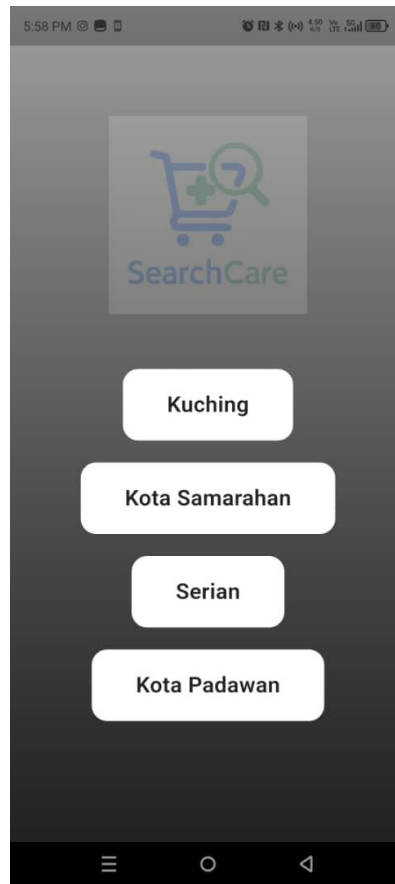


Figure 91 Location View

The location view page provides dedicated functionality to support and facilitate specific healthcare workflows. Users experiences the location-based input and receives matching city/regions they are currently located so they can locate clinics/hospitals that are located in the area.

4.5 Conclusion

The implementation and design of the SearchCare application reflect a comprehensive approach to developing a smart healthcare solution tailored to modern needs. The system was structured with well-defined components and modules to ensure scalability, usability, and functionality. At its core, the application integrates essential components such as Firebase for authentication and cloud data storage, Firestore for real-time database management, and Flutter as the cross-platform development framework. These technologies work together to ensure secure user access, responsive interface, and seamless data handling.

The modular design includes distinct functional units such as user and doctor registration, appointment booking, medicine purchasing, lifestyle content viewing, and disease-symptom prediction. Each module was designed to serve a specific purpose while maintaining a cohesive flow within the app. For instance, the doctor module enables profile management and appointment handling, while the admin module manages content, medicines, and approvals. Meanwhile, the user module empowers patients to manage their health through features like booking consultations, viewing order history, and accessing wellness content. Through this modular and component-driven design, SearchCare successfully delivers a holistic and efficient healthcare experience that meets both user and system requirements.

Chapter 5: Test Case and User Testing

5.1 System Checking

System testing, which covers a variety of topics to guarantee operation, usability, and security, is an essential stage in the development of mobile applications. As evidenced by the creation of a diet and dental health app, usability testing assesses the app's efficacy and user-friendliness (Muhamat et al., 2021). For apps that need to handle information accurately and quickly, real-time data collection and performance testing are crucial (Azadi et al., 2020). According to Huang et al. (2022), the development process frequently consists of several layers and modules, such as user registration, data management, and integration with external systems. It serves to show that the application is operating in accordance with the requirements specified in the application. Additionally, users who have been chosen to participate in the examination of the applicants perform the usability test. Every test and result obtained from application testing has been recorded and succinctly explained in a table that has been organised in a certain way.

5.1.1 Test Case

Test Case 1: Login Process for both user/patient Module: Login Functionality: Email and Password Verification		
Test 1: Valid Login Description: Verify user can successfully log in with valid credentials. Preconditions: User must already be registered in the system. Test Steps: 1. Open login screen 2. Enter valid email and password 3. Tap "Login" button Input : <table border="1"><tr><td>Email: max@gmail.com</td></tr><tr><td>Password: 88888888</td></tr></table> Expected Result: User is redirected to home/dashboard screen with success message Actual Result: As expected. Status: Pass	Email: max@gmail.com	Password: 88888888
Email: max@gmail.com		
Password: 88888888		
Test 2: Invalid Email Format Description: Check validation when email format is incorrect. Preconditions: None Test Steps: 1. Enter invalid email format (e.g., missing '@') 2. Enter password 3. Tap "Login" button Input : <table border="1"><tr><td>Email: maxgmail.com</td></tr><tr><td>Password: 88888888</td></tr></table>	Email: maxgmail.com	Password: 88888888
Email: maxgmail.com		
Password: 88888888		

Expected Result: System shows "Invalid email format" error and blocks login

Actual Result: As expected.

Status: Pass

Test 3: Wrong Password

Description: Verify system rejects incorrect password

Precondition: User must exist in database

Test Steps: 1. Enter valid email
2. Enter wrong password
3. Tap "Login" button

Input : Email: max@gmail.com

Password: WrongPass

Expected Result: System shows "Incorrect password" message

Actual Result: As expected.

Status: Pass

Test 4: Empty email and password

Description: Check system behaviour when login fields are empty

Preconditions: None

Test Steps: 1. Leave email and password empty
2. Tap "Login" button

Input : Email:

Password:

Expected Result: Validation message: "Please enter email and password"

Actual Result: As expected.

Status: Pass

Test 5: Unregistered email

Description: Check system behaviour when logging in with unregistered email

Preconditions: Email does not exist in Firebase/Auth

Test Steps: 1. Enter unregistered email
2. Enter any password
3. Tap "Login" button

Input : Email:

meowmeow@gmail.com

Expected Result: Error: "No user found with this email"

Actual Result: As expected.

Status: Pass

Test Case 2: User Registration Process

Module: Register Module

Functionality: New User Registration, Field Validations, and Media Upload

Test 1: Successful Registration with Valid Data

Description: Verify user can register successfully with valid input and profile picture

Precondition: Device has internet connection

Test Steps: 1. Fill all fields correctly
2. Upload valid profile picture
3. Tap "Register"

Input: Profile Picture: Uploaded image (JPEG, <2MB)

Full Name: Maximus Bateh

Email: max@gmail.com

Password: 88888888

Phone: 0123456789

City: UNIMAS

Country: UNIMAS

Gender: Male

Username: Maxy

ID Number: 1111111

Address: 123, Jalan Example

Criminal Record: No (checkbox)

Medical History: None

Expected Result: User is successfully registered, redirected to login/home screen, and data is stored in Firebase Auth and Storage and Firestore Database.

Actual Result: As expected.

Status: Pass

Test 2: Missing Required Fields

Description: Check if the form blocks submission when required fields are empty

Precondition: None

Test Steps: 1. Leave some fields blank
2. Tap "Register"

Input: Leave email or password empty

Expected Result: Show validation messages like "Email is required", prevent submission.

Actual Result: As expected.

Status: Pass

Test 3: Invalid Email Format

Description: Validate incorrect email format is not accepted

Precondition: None

Test Steps: 1. Enter email without "@"
2. Try to submit

Input: Email - maxgmail.com

Expected Result: Show "Invalid email format" message

<p>Actual Result: As expected. Status: Pass</p>
<p>Test 4: Password too short Description: Check validation for weak/short password Test Steps: 1. Enter password less than 6 characters 2. Try to submit Input: Password – 12345 Expected Result: Show error: "Password must be at least 6 characters" Actual Result: As expected. Status: Pass</p>
<p>Test 5: Upload Invalid Profile Picture Description: Test image upload validation Test Steps: 1. Upload file that's not an image or too large Input: File type: .pdf or image >5MB Expected Result: Show error: "Only image files under 5MB are allowed" Actual Result: As expected. Status: Pass</p>
<p>Test 6: Duplicate email registration Description: Check system prevents registration using same email Test Steps: 1. Enter email that's already registered 2. Try to submit Input: Email – max@gmail.com Expected Result: Show error: "Email already in use" Actual Result: As expected. Status: Pass</p>
<p>Test 7: Criminal Record Checkbox Handling Description: Verify system correctly saves criminal record status based on checkbox selection Precondition: All other fields are filled correctly Test Steps: 1. Fill out the registration form 2. Tick or leave the criminal record checkbox 3. Tap "Register" Input variant A: <input checked="" type="checkbox"/> Checkbox ticked (user has a criminal record) Expected Result variant A: criminal_record field saved as "Yes" in Firestore Actual Result variant A: As expected. Status variant A: Pass Input variant B: <input type="checkbox"/> Checkbox not ticked (user has no record) Expected Result variant B: criminal_record field saved as "No" or not present (based on logic) Actual Result variant B: As expected. Status variant B: Pass</p>

Test 8: Store data in Firestore after registration

Description: Confirm all registration data (including profile picture URL) is stored in Firestore

Precondition: User successfully registered

Test steps: 1. Register new user
2. Check Firestore users/{uid}

Input:

Profile Picture: Uploaded image (JPEG, <2MB)
Full Name: Maximus Bateh
Email: max@gmail.com
Password: 88888888
Phone: 0123456789
City: UNIMAS
Country: UNIMAS
Gender: Male
Username: Maxy
ID Number: 1111111
Address: 123, Jalan Example

Expected Result: All fields appear correctly in Firestore, including picture URL

Actual Result: As expected.

Status: Pass

Test Case 3: User Menu Functional Setting

Module: User Menu

Functionality: Ensure all buttons and navigation features on the user menu screen work correctly.

Test 1: Access user settings (Edit profile)

Description: Verify user can navigate to the settings page to edit their profile

Precondition: User is logged in

Test steps: 1. Open User Menu
2. Tap "Settings"

Expected Result: App navigates to the Edit Profile page, existing user details are prefilled

Actual Result: As expected.

Status: Pass

Test 2: Buy Medicine

Description: Verify user can open the Buy Medicine screen from menu

Precondition: User is logged in

Test steps: 1. Tap "Buy Medicine" in the menu

Expected Result: App navigates to medicine listing page, showing available medicines

Actual Result: As expected.

Status: Pass

Test 3: Search Symptom Feature
Description: Check if tapping "Search Symptom" opens symptom search functionality
Precondition: User is logged in
Test steps: 1. Tap "Search Symptom" from the user menu
Expected Result: User is directed to symptom input/ search screen
Actual Result: As expected.
Status: Pass

Test 4: Book Appointment
Description: Verify user can access the appointment booking page
Precondition: Doctor list and appointment are set up
Test steps: 1. Tap "Book Appointment"
Expected Result: Navigates to doctor list or booking form screen
Actual Result: As expected.
Status: Pass

Test 5: Book Consultation
Description: Ensure user can access consultation screen
Precondition: Consultation module is available
Test steps: 1. Tap "Book Consultation"
Expected Result: App opens consultation form or doctor consultation list
Actual Result: As expected.
Status: Pass

Test 6: View Healthy Facts
Description: Verify healthy facts page opens correctly
Precondition: Healthy facts are uploaded in Firestore
Test steps: 1. Tap "View Healthy Facts"
Expected Result: Navigates to a page listing healthy tips (exercise, food, facts)
Actual Result: As expected.
Status: Pass

Test 7: View Location Feature
Description: Ensure user can view different medical centre location links based on a selected city from a pre-set list configured by admin
Precondition: Admin has pre-set several cities (e.g., Kuching, Kota Samarahan, Serian) with corresponding medical centre links in the app
Test steps: 1. Open the User Menu
2. Tap "View Location"
3. Select a city from the list
4. View the medical location link
Expected Result: App opens map showing nearby hospitals or health centers
Actual Result: As expected.
Status: Pass

<p>Test Case 4: Buy Medicine (User Menu)</p> <p>Module: Buy Medicine</p> <p>Functionality: Ensure that the user can browse, add, and purchase medicines through a functional, intuitive buying process.</p>
<p>Test 1: View Medicines One by One</p> <p>Description: Verify user can view the list of medicines and tap each item to see full details</p> <p>Precondition: User is logged in and medicines are available in the database</p> <p>Test steps: 1. Open User Menu 2. Tap 'Buy Medicine' 3. Scroll and tap a medicine</p> <p>Expected Result: App navigates to the medicine detail page, showing full info including name, price, availability</p> <p>Actual Result: As expected</p> <p>Status: Pass</p>
<p>Test 2: Add Medicine to Cart</p> <p>Description: Ensure user can add a medicine to their cart</p> <p>Precondition: User is logged in and has accessed medicine detail page</p> <p>Test steps: 1. Tap 'Add to Cart' on a medicine page</p> <p>Expected Result: Medicine is added to user's cart in Firestore</p> <p>Actual Result: As expected</p> <p>Status: Pass</p>
<p>Test 3: View Cart</p> <p>Description: Verify that the user can view items added to cart</p> <p>Precondition: User has at least one item in cart</p> <p>Test steps: 1. Tap 'Cart' icon in User Menu</p> <p>Expected Result: Cart screen shows item list with name, quantity, price, and total</p> <p>Actual Result: As expected</p> <p>Status: Pass</p>
<p>Test 4: Update Quantity in Cart</p> <p>Description: Ensure user can update the quantity of items in the cart</p> <p>Precondition: Cart has at least one item</p> <p>Test steps: 1. Tap '+' or '-' icon beside an item in the cart</p> <p>Expected Result: Quantity updates in real time and total recalculates</p> <p>Actual Result: As expected</p> <p>Status: Pass</p>
<p>Test 5: Confirm Buy Medicine – Takeaway</p> <p>Description: Verify user can proceed with buying medicines using Takeaway option</p> <p>Precondition: User has added items to cart and is ready to checkout</p> <p>Test steps:</p> <ol style="list-style-type: none"> 1. Go to Cart 2. Tap 'Checkout' 3. Choose payment method

4. Select 'Takeaway'
5. Confirm order
Expected Result: Order is saved with 'Takeaway' method under user's orders collection
Actual Result: As expected
Status: Pass

Test 6: Confirm Buy Medicine – Delivery
Description: Verify user can proceed with buying medicines using Delivery option
Precondition: User has added items to cart and is ready to checkout
Test steps:
1. Go to Cart
2. Tap 'Checkout'
3. Choose payment method
4. Select 'Delivery'
5. Confirm order
Expected Result: RM5 fee added and order saved with 'Delivery' method under user's orders collection
Actual Result: As expected
Status: Pass

Test 7: Choose Payment Method
Description: Ensure user can select different payment methods during checkout
Precondition: User is on the checkout page
Test steps:
1. On Checkout screen
2. Select Cash / FPX / Credit Card
3. Confirm payment
Expected Result: Selected method is recorded in Firestore under 'paymentMethod'
Actual Result: As expected
Status: Pass

Test 8: View Order History
Description: Verify user can view their past medicine purchases
Precondition: User has placed at least one order
Test steps:
1. Tap 'Order History' in User Menu
Expected Result: List of past orders is shown with medicine name, date, price, and status
Actual Result: As expected
Status: Pass

Test Case 5: Book Appointment (User menu)
Module: Book Appointment
Functionality: Ensure that users can successfully search for, schedule, and track medical appointments with registered doctors through the app.

Test 1: Search and Filter Doctor

Description: Verify user can search for doctors based on name or filter by expertise

Precondition: Doctors are registered and stored in Firestore with expertise field

Test steps:

1. Open User Menu
2. Tap 'Book Appointment'
3. Enter doctor's name or select expertise

Expected Result: Filtered list of doctors appears based on search or selected expertise

Actual Result: As expected

Status: Pass

Test 2: Book Requested Doctor

Description: Ensure user can tap on a listed doctor to book an appointment

Precondition: Doctor list is visible after search/filter

Test steps:

1. Select a doctor from the list
2. Tap 'Book Appointment'

Expected Result: Appointment form opens with the selected doctor's name prefilled

Actual Result: As expected

Status: Pass

Test 3: Fill Appointment Details

Description: Ensure user can fill in all required details for appointment booking

Precondition: User is on the appointment booking form

Test steps:

1. Fill in Full Name
2. Pick a Date from calendar
3. Select a Time (24-hour format)
4. Choose method (Online or Face-to-Face)
5. Write the Reason for appointment
6. Tap 'Submit'

Expected Result: Booking is saved to Firestore under user's appointments with status = pending

Actual Result: As expected

Status: Pass

Test 4: View Booking Status

Description: Verify that user can view the status of booked appointments

Precondition: User has made at least one appointment request

Test steps:

1. Open User Menu
2. Tap 'Appointment Status'
3. View booking card

Expected Result: Each appointment shows doctor name, date, time, reason, and status (pending, accepted, or rejected)

Actual Result: As expected

Status: Pass

<p>Test Case 6: Book Consultation (User menu)</p> <p>Module: Book Consultation</p> <p>Functionality: Ensure that users can successfully schedule and manage one-on-one consultations with registered doctors through the mobile app.</p>
<p>Test 1: Search and Filter Doctor for Consultation</p> <p>Description: Verify user can search for doctors based on name or filter by expertise</p> <p>Precondition: Doctors are registered and stored in Firestore with expertise field</p> <p>Test steps:</p> <ol style="list-style-type: none"> 1. Open User Menu 2. Tap 'Book Consultation' 3. Enter doctor's name or select expertise <p>Expected Result: Filtered list of doctors appears based on search or selected expertise</p> <p>Actual Result: As expected</p> <p>Status: Pass</p>
<p>Test 2: Book Requested Doctor for Consultation</p> <p>Description: Ensure user can select a doctor and proceed to book a consultation</p> <p>Precondition: Doctor list is visible after search/filter</p> <p>Test steps:</p> <ol style="list-style-type: none"> 1. Select a doctor from the list 2. Tap 'Book Consultation' <p>Expected Result: Consultation form opens with the selected doctor's name prefilled</p> <p>Actual Result: As expected</p> <p>Status: Pass</p>
<p>Test 3: Fill Consultation Details</p> <p>Description: Ensure user can fill in all required details for consultation booking</p> <p>Precondition: User is on the consultation booking form</p> <p>Test steps:</p> <ol style="list-style-type: none"> 1. Fill in Full Name 2. Pick a Date from calendar 3. Select a Time (24-hour format) 4. Choose method (Online or Face-to-Face) 5. Write the Reason for consultation 6. Tap 'Submit' <p>Expected Result: Consultation request is saved to Firestore under user's consultations with status = pending</p> <p>Actual Result: As expected</p> <p>Status: Pass</p>
<p>Test 4: View Consultation Status</p> <p>Description: Verify that user can view the status of booked consultations</p> <p>Precondition: User has made at least one consultation request</p>

Test steps:

1. Open User Menu
2. Tap 'Consultation Status'
3. View consultation card

Expected Result: Each consultation shows doctor name, date, time, reason, and status (pending, accepted, or rejected)

Actual Result: As expected

Status: Pass

Test Case 7: View Healthy Tips (User menu)

Module: View Healthy Tips

Functionality: Ensure that users can successfully view different healthy tips that has been uploaded by admin.

Test 1: Access Healthy Tips Section

Description: Ensure user can navigate to the Healthy Tips section from the user menu.

Precondition: User is logged in.

Test steps:

1. Open User Menu
2. Tap "View Healthy Tips"

Expected Result: App navigates to the Healthy Tips screen with visible category options (Food, Exercise, Healthy Facts).

Actual Result: As expected.

Status: Pass

Test 2: Select Food Category

Description: Verify that user can view healthy food tips when selecting the Food category.

Precondition: Admin has uploaded content under the Food category in Firestore.

Test steps:

1. From the Healthy Tips screen, tap "Food" category
2. View the list of food-related tips

Expected Result: A list of healthy food tips is displayed with images, titles, and descriptions.

Actual Result: As expected.

Status: Pass

Test 3: Select Exercise Category

Description: Verify that user can view exercise tips when selecting the Exercise category.

Precondition: Admin has uploaded exercise-related tips in Firestore.

Test steps:

1. From the Healthy Tips screen, tap "Exercise" category
2. View the list of exercise-related tips

Expected Result: A list of healthy exercise tips is displayed with titles and images.

Actual Result: As expected.

Status: Pass

Test 4: Select Healthy Facts Category

Description: Verify that user can view general health facts when selecting the Healthy Facts category.

Precondition: Admin has uploaded general health facts in Firestore.

Test steps:

1. From the Healthy Tips screen, tap "Healthy Facts" category
2. View the list of general health facts

Expected Result: A list of healthy facts is displayed, including tips or trivia about wellness, habits, or body systems.

Actual Result: As expected.

Status: Pass

Test 5: Handle No Tips Available

Description: Confirm that a proper message is shown when no healthy tips are available in the selected category.

Precondition: Selected category has no data in Firestore.

Test steps:

1. Tap any category that has no uploaded content

Expected Result: A message appears: "No tips available."

Actual Result: As expected.

Status: Pass

Test Case 8: Search Symptoms

Module: Search Symptoms

Functionality: User can search or filter symptoms that they are experiencing or curious about

Test 1: Access Symptom Search Page

Description: Verify user can navigate to the symptom search page from the main menu.

Precondition: User is logged in

Test steps:

1. Open User Menu
2. Tap "Search Symptom"

Expected Result: App navigates to the symptom search screen.

Actual Result: As expected.

Status: Pass

Test 2: Search Symptom with Keyword

Description: Verify that user can input a symptom and receive relevant health tips or recommendations.

Precondition: User is on the Search Symptom screen

Test steps:

1. Type "fever" in the search bar
2. Tap "Search"

Expected Result: App displays results related to "fever" with relevant health tips or diseases.

Actual Result: As expected.

Status: Pass

Test Case 9: Search Location

Module: Location View

Functionality: User can identify clinics that are located around the cities they are in.

Test 1: Access Location Page

Description: Verify user can navigate to the location information page.

Precondition: User is logged in

Test steps:

1. Open User Menu
2. Tap "View Location"

Expected Result: App navigates to the list of cities with preset clinic/medical center links.

Actual Result: As expected.

Status: Pass

Test 2: View Location Links by City

Description: Verify user can see the location links based on pre-set cities.

Precondition: Location data has been set by the admin in the database.

Test steps:

1. On Location page, scroll through list of cities
2. Tap a city

Expected Result: User sees clinic/medical center links specific to the selected city.

Actual Result: As expected.

Status: Pass

Test Case 10: Doctor Registration Process

Module: Register Module

Functionality: New User Registration, Field Validations, and Media Upload

Test 1: Access Doctor Registration Page

Description: Verify that user can navigate to the Doctor Registration form from account type selection.

Precondition: User is on the account type selection screen.

Test steps:

1. Open app
2. Tap "Register"
3. Choose "Doctor"

Expected Result: App navigates to the doctor registration form.

<p>Actual Result: As expected. Status: Pass</p>
<p>Test 2: Input Email and Password Description: Ensure doctor can enter valid email and password. Precondition: User is on the doctor registration form. Test steps: 1. Enter a valid email (e.g., catdoctor@gmail.com) 2. Enter a valid password (e.g., 88888888) Expected Result: No error shown; input fields accept the data. Actual Result: As expected. Status: Pass</p>
<p>Test 3: Input Contact and Location Description: Ensure doctor can fill in phone number, city, and country. Precondition: User has filled email and password. Test steps: 1. Enter phone number (e.g., 0123456789) 2. Enter city (e.g., KUCHING) 3. Enter country (e.g., Malaysia) Expected Result: Data is accepted without error. Actual Result: As expected. Status: Pass</p>
<p>Test 4: Select Gender Description: Verify gender selection dropdown or radio buttons are functional. Precondition: User is on the registration form. Test steps: 1. Tap on gender field 2. Select either Male, Female, or Other Expected Result: Gender is correctly selected and displayed. Actual Result: As expected. Status: Pass</p>
<p>Test 5: Input Personal and Professional Details Description: Ensure doctor can enter their full name, medical license number, expertise, and workplace. Precondition: User is on registration form. Test steps: 1. Enter full name (e.g., Dr. Aiman Noor) 2. Enter license number (e.g., MD-198732) 3. Enter expertise (e.g., Cardiology) 4. Enter workplace (e.g., Pantai Medical Centre) Expected Result: All fields accept input and show no validation error. Actual Result: As expected. Status: Pass</p>
<p>Test 6: Submit Doctor Registration</p>

Description: Verify that tapping the register button validates and submits all input data.

Precondition: All required fields are filled.

Test steps:

1. Tap “Register” button

Expected Result: Account is created in Firebase Auth, and profile details are saved in Firestore under doctors collection.

Actual Result: As expected.

Status: Pass

Test 7: Navigate Back to Account Type Selection

Description: Ensure user can navigate back to choose a different registration type.

Precondition: User is on the doctor registration form.

Test steps:

1. Tap “Back” or “Already have an account?”

Expected Result: App returns to the screen for selecting registration as User or Doctor.

Actual Result: As expected.

Status: Pass

Test Case 11: Doctor Menu Functional Setting

Module: Doctor Dashboard

Functionality: Ensure all buttons and navigation features on the user menu screen work correctly.

Test 1: Access Doctor Menu

Description: Ensure doctor can access the Doctor Menu after successful login.

Precondition: Doctor is logged in.

Test steps:

1. Login with a valid doctor account
2. App navigates to Doctor Menu

Expected Result: Doctor Menu is displayed with available options (Settings, View Appointment, View Consultation, Sign Out).

Actual Result: As expected.

Status: Pass

Test 2: Access Settings (Edit Doctor Profile)

Description: Verify doctor can navigate to settings to edit their profile details.

Precondition: Doctor is on the Doctor Menu screen.

Test steps:

1. Tap on “Settings”
2. View pre-filled profile information
3. Edit and save changes (e.g., workplace, expertise)

Expected Result: Profile details are updated in Firestore, and confirmation is shown.

Actual Result: As expected.

Status: Pass

Test 3: View Appointment Requests

Description: Ensure doctor can view a list of appointment requests from patients.

Precondition: Appointments are available in Firestore under appointments collection linked to the doctor.

Test steps:

1. Tap “View Appointment”
2. See list of appointment requests with patient details and status (e.g., pending)

Expected Result: Appointment details such as full name, date, time, method, and reason are displayed.

Actual Result: As expected.

Status: Pass

Test 4: Accept or Reject Appointment

Description: Verify doctor can update the appointment status to either accepted or rejected.

Precondition: Doctor is viewing a pending appointment.

Test steps:

1. Tap “Accept” or “Reject” on the appointment
2. Confirm the action

Expected Result: The status in Firestore is updated and the user is notified.

Actual Result: As expected.

Status: Pass

Test 5: View Consultation Requests

Description: Ensure doctor can view a list of consultation requests.

Precondition: Consultation records exist in Firestore linked to the doctor.

Test steps:

1. Tap “View Consultation”
2. See list of consultation requests with patient details and current status

Expected Result: Consultation details (date, time, method, reason) are displayed.

Actual Result: As expected.

Status: Pass

Test 6: Accept or Reject Consultation

Description: Verify doctor can change consultation request status.

Precondition: Doctor is viewing a pending consultation request.

Test steps:

1. Tap “Accept” or “Reject”
2. Confirm action

Expected Result: Firestore status field is updated accordingly and reflected in the patient's app.

Actual Result: As expected.

Status: Pass

Test 7: Sign Out

Description: Ensure doctor can sign out of the app successfully.

Precondition: Doctor is on the Doctor Menu screen.

1. Test steps:

Tap "Sign Out"

Expected Result: User is logged out and navigated back to the login or welcome screen.

Actual Result: As expected.

Status: Pass

Test Case 12: Admin login page

Module: Admin Login

Functionality: Admin's email and Password Verification

Test 1: Access Admin Login Page

Description: Ensure user can navigate to the Admin Login page from the account type selection screen.

Precondition: User is on the main login or account selection screen.

Test steps:

1. Launch the app
2. Tap "Login"
3. Choose "Admin"

Expected Result: App navigates to a dedicated Admin Login screen with email and password fields.

Actual Result: As expected.

Status: Pass

Test 2: Enter Correct Admin Credentials

Description: Verify that admin can login using valid predefined credentials stored in Firestore.

Precondition: Email and password match admin@gmail.com and admin123, which are validated against Firestore records.

Test steps:

1. Enter email: admin@gmail.com
2. Enter password: admin123
3. Tap "Login"

Expected Result: Credentials are authenticated, and the admin is redirected to the Admin Dashboard.

Actual Result: As expected.

Status: Pass

Test 3: Enter Incorrect Admin Credentials

Description: Ensure incorrect admin credentials are rejected.

Precondition: User enters invalid email or password.

Test steps:

1. Enter email: admin123@gmail.com
2. Enter password: wrongpass
3. Tap "Login"

Expected Result: Login fails; error message appears: "Invalid admin credentials".

Actual Result: As expected.

Status: Pass

Test 4: Leave Fields Empty

Description: Validate that login is not permitted if email or password is missing.

Precondition: Fields are left empty.

Test steps:

1. Leave email and/or password blank
2. Tap "Login"

Expected Result: App shows validation error: "Please enter email and password"

Actual Result: As expected.

Status: Pass

Test 5: Verify Admin Authentication in Firestore

Description: Ensure that Firebase Authentication properly checks the credentials against Firestore data for admin verification.

Precondition: Admin credentials are entered and login is attempted.

Test steps:

1. Tap the "Login" button
2. The system checks Firebase Authentication for the admin's email and password
3. Firestore is queried for the admin data (e.g., checking the "role" field in the admin collection)

Expected Result:

- Admin is authenticated successfully if credentials match.
- If credentials don't match, authentication fails.

Actual Result: As expected.

Status: Pass

Test 6: Submit Admin Login Form

Description: Verify that tapping the login button performs authentication using Firebase Authentication and Firestore.

Precondition: Admin credentials have been entered correctly.

Test steps:

1. Tap the "Login" button

Expected Result:

- Firebase Authentication checks the entered credentials against the admin record in Firestore.
- If the credentials match, the user is authenticated and logged in.

<ul style="list-style-type: none"> ○ The app navigates to the Admin Dashboard or Admin Menu. <p>Actual Result: As expected. Status: Pass</p>
<p>Test 6: Successful Admin Login Redirection</p> <p>Description: Verify that the user is redirected to the appropriate admin dashboard or menu after successful login.</p> <p>Precondition: Admin has been authenticated successfully.</p> <p>Test steps:</p> <ol style="list-style-type: none"> 1. After successful login, app should navigate to the Admin Dashboard or Admin Menu screen. <p>Expected Result:</p> <ul style="list-style-type: none"> ○ User is redirected to the Admin Dashboard or Menu. ○ The correct content for the admin role is displayed (e.g., manage users, view reports, etc.). <p>Actual Result: As expected. Status: Pass</p>

<p>Test Case 13: Admin menu page</p> <p>Module: Admin Dashboard</p> <p>Functionality: Ensure all buttons and navigation features on the admin dashboard screen work correctly.</p>
<p>Test 1: Access Admin Dashboard Menu</p> <p>Description: Ensure that the authenticated admin can access the Admin Dashboard after login.</p> <p>Precondition: Admin is logged in with valid credentials.</p> <p>Test steps:</p> <ol style="list-style-type: none"> 1. Login with admin@gmail.com and admin123 2. App redirects to Admin Dashboard <p>Expected Result: Admin Dashboard menu is displayed with all functions (Add Medicine, View Medicine, Update Tips, Update Location).</p> <p>Actual Result: As expected. Status: Pass</p>
<p>Test 2: Add New Medicine</p> <p>Description: Verify admin can add new medicine with image, name, price, availability, stock, and prescription requirements.</p> <p>Precondition: Admin is on the Admin Dashboard screen.</p> <p>Test steps:</p> <ol style="list-style-type: none"> 1. Tap “Add Medicine” 2. Fill in all required fields: name, price, stock, availability, prescription

3. Upload image
4. Tap “Save”

Expected Result:

- Medicine is added to Firestore under admins/{adminId}/medicines
- Image is uploaded to Firebase StorageConfirmation message shown

Actual Result: As expected.

Status: Pass

Test 3: View Added Medicines

Description: Verify admin can view a list of previously added medicines.

Precondition: Admin has added at least one medicine.

Test steps:

1. Tap “View Medicine”
2. Observe grid or list display of medicines

Expected Result:

- Medicines are listed with image, name, price, stock, and prescription status
- Tapping one allows for editing or deleting

Actual Result: As expected.

Status: Pass

Test 4: Update or Delete Medicine

Description: Ensure that admin can update or delete an existing medicine.

Precondition: Admin is viewing medicine list.

Test steps:

1. Tap on a medicine item
2. Tap “Edit” to update details or “Delete” to remove the record

Expected Result:

- Updates are saved to Firestore
- Delete removes medicine from Firestore and deletes image from Storage

Actual Result: As expected.

Status: Pass

Test 5: Update Healthy Tips

Description: Verify that admin can update healthy tips under categories like food, exercise, or facts.

Precondition: Admin is on Admin Dashboard screen.

Test steps:

1. Tap “Update Healthy Tips”
2. Choose category (e.g., Food, Exercise, Healthy Facts)
3. Enter or edit content
4. Tap “Update”

Expected Result:

- Tip content is saved in Firestore under correct category
- Confirmation is shown

Actual Result: As expected.

Status: Pass

Test 6: Update Location of Medical Centers or Clinics

Description: Ensure that admin can update a list of locations for medical centers and clinics with links or addresses.

Precondition: Admin is on Admin Dashboard screen.

Test steps:

1. Tap "Update Location"
2. Add new location or edit existing ones
3. Fill in city name and link or address
4. Tap "Save"

Expected Result:

- Location list is updated in Firestore under locations collection
- Users can later view city-based location info from their app

Actual Result: As expected.

Status: Pass

Test 7: Navigation between Admin Features

Description: Verify admin can navigate smoothly between dashboard functions.

Precondition: Admin is logged in and inside the dashboard.

Test steps:

1. Tap each option in dashboard one by one
2. Navigate back using app bar or system back button

Expected Result:

- Each function opens correctly
- App navigates back to dashboard without crash

Actual Result: As expected.

Status: Pass

Test Case 14: Checkout multiple medicine

Module: Checkout medicines

Functionality:

Test 1: Access Checkout Page with Multiple Cart Items

Description: Ensure user can access the checkout page with multiple items in the cart.

Precondition: User is logged in and has added two or more medicines to cart.

Test Steps:

1. Go to Cart Page
2. Tap "Checkout"

Expected Result: User is navigated to Checkout Page with all cart items listed, showing name, price, quantity, and subtotal.

Actual Result: As expected.

Status: Pass

Test 2: Adjust Quantity for Each Medicine

Description: Ensure user can increase or decrease the quantity of each medicine before checkout.

Precondition: User is on Checkout Page.

Test Steps:

1. Tap “+” or “-” on quantity selector beside each item

Expected Result: Quantity updates instantly and total price recalculates correctly.

Actual Result: As expected.

Status: Pass

Test 3: Calculate Total Price with Delivery Fee

Description: Confirm correct total is calculated based on items and delivery method.

Precondition: User has selected items and is on Checkout Page.

Test Steps:

1. Select Delivery (COD)

2. Verify total = sum of item prices + RM5.00

Expected Result: Delivery fee of RM5.00 is added once per order, not per item.

Actual Result: As expected.

Status: Pass

Test 4: Select Payment Method

Description: Ensure user can select one payment method (Cash, FPX, or Credit Card).

Precondition: User is on Checkout Page.

Test Steps:

1. Tap on payment options dropdown or radio buttons

2. Select one method

Expected Result: Selection is saved and displayed. Cannot select more than one at a time.

Actual Result: As expected.

Status: Pass

Test 5: Select Delivery Method

Description: Ensure user can choose Take Away or Delivery (COD).

Precondition: User is on Checkout Page.

Test Steps:

1. Tap delivery options

2. Select one method

Expected Result: UI updates to show or hide delivery fee accordingly.

Actual Result: As expected.

Status:Pass

Test 6: Confirm Order and Save to Firestore

Description: Confirm the order processes correctly and stores each medicine as a separate order.

Precondition: User has selected all required options.

<p>Test Steps:</p> <ol style="list-style-type: none"> 1. Tap “Confirm Order” <p>Expected Result: Each item is stored as a separate document in users/{uid}/orders/ with status set to 'pending'.</p> <p>Actual Result: As expected.</p> <p>Status:Pass</p>
<p>Test 7: Clear Cart After Checkout</p> <p>Description: Ensure the cart is cleared after a successful checkout.</p> <p>Precondition: Order is successfully confirmed.</p> <p>Test Steps:</p> <ol style="list-style-type: none"> 1. After confirming order, return to Cart Page <p>Expected Result: Cart is empty; user receives confirmation or success screen.</p> <p>Actual Result: As expected.</p> <p>Status: Pass</p>
<p>Test 8: Show Receipt Summary</p> <p>Description: Display final receipt to user after order is confirmed.</p> <p>Precondition: Order has been processed.</p> <p>Test Steps:</p> <ol style="list-style-type: none"> 1. App navigates to Receipt Summary Page <p>Expected Result: Shows itemized list, payment method, delivery method, and total.</p> <p>Actual Result: As expected.</p> <p>Status: Pass</p>

5.2 Usability Testing

5.2.1 Introduction

When developing mobile apps, user testing is especially important for applications pertaining to health. It entails using a variety of techniques, including field testing, surveys, and interviews, to assess usability features including satisfaction, utility, and simplicity of use (Johnson et al., 2022). A study on baby fall prevention showed that app development can be improved by using a behaviour theory-informed approach (Cooray et al., 2021). App evaluations and developer conversations can reveal human-centric issues that developers should take into account, such as app usage, inclusivity, and user reactions (Khalajzadeh et al., 2022). As demonstrated by the creation of MS Energise, a fatigue management app for people with multiple sclerosis, iterative development combined with usability and resonance testing is successful in producing user-friendly apps (Kessel et al., 2021). This usability test was conducted to evaluate the effectiveness, efficiency, and user satisfaction of the SearchCare

mobile application. The test focused on real users performing common tasks within the app such as registering, booking appointments, purchasing medicine, and accessing health information.

5.2.2 Test Objectives

The objectives for the usability testing are included:

- Assess the ease of navigation and task completion for key features.
- Identify any usability issues or bottlenecks.
- Gather user feedback on interface design, language, layout, and speed.
- Improve the overall user experience based on findings.

5.2.3 Test Participants

The total participant that are involved during the application testing are 30 participants. They are consists of age group 20-25 years old because most of the participant are UNIMAS student and a small group of adults in the range of age 30-45 years old. Most of the participants have a high proficiency in using technology.

5.2.4 Test Environment

The participants were provided suitable environment to execute the testing:

- Platform: Android smartphones (and emulators for remote)
- Duration: 30–45 minutes per session
- Tools: Google Forms, Screen recording tools, Task sheets

5.2.5 Task Scenarios

Table 5.0 Usability Testing

Task No	Task Description	Succes Criteria
T1	Register as a new patient and complete full profile setup	Profile saved without error, all fields filled
T2	Search symptoms and get suggestions for conditions	Related symptom content appears
T3	Book an appointment with a doctor using expertise filter	Appointment sent and confirmed in status list

T4	Buy medicine, increase quantity, and check out with delivery and payment options	Payment and delivery method successfully selected
T5	View appointment and consultation status	Able to check if doctor has confirmed/rejected
T6	Able to check if doctor has confirmed/rejected	No confusion or misclicks during navigation
T7	Doctor logs in using email and password.	Doctor is successfully redirected to the Doctor Dashboard.
T8	Doctor accesses the appointment list from the dashboard.	List of appointments is displayed with correct patient details.
T9	Doctor confirms or rejects an appointment request.	Status is updated in Firestore and reflected in user view.
T10	Doctor views the consultation requests submitted by patients.	Details of each consultation are correctly loaded.
T11	Doctor accepts or rejects the consultation.	Status changes in Firestore and user gets updated view.
T12	Doctor edits their name, specialization, or uploads a new profile picture.	Updated information is saved and shown on dashboard.
T13	Admin enters credentials and logs in.	Admin dashboard is loaded with options to manage content.
T14	Admin fills medicine form and uploads image.	Medicine is saved in Firestore and shown in medicine list.
T15	Admin updates stock, price, or name of a medicine.	Changes are reflected instantly in Firestore and UI.
T16	Admin edits or deletes an existing tip.	Updated tip is shown to users correctly.
T17	Admin adds a new symptom used in diagnosis module.	Symptom is added to Firestore and searchable.
T18	Admin accesses list of registered users and doctors.	Data is accurately loaded and viewable.

5.3 System Evaluation

In mobile app development, system evaluation entails evaluating different app features to guarantee efficacy and quality. Customisability, content, and evidence are important evaluation domains (Mackey et al., 2021). Effectiveness, efficiency, and user pleasure are all aspects of usability that are crucial (Dutta et al., 2022). To score apps, evaluation frameworks frequently include scoring tools like summary statistics and Likert scales (Hensher et al., 2021). Uncertainty in app assessment results from the wide variation in evaluation methodologies. To investigate user experiences and difficulties, evaluation techniques may involve both qualitative focus groups and quantitative surveys (Gambo & Shakir, 2022).

As a result, the responses and results are compiled and evaluated. A system evaluation has been carried out to assess the proposed system with the target users of this proposed app, which are among UNIMAS students and public workers within UNIMAS. A total of 30 participants, comprising first-year students through fourth-year students from various programs and faculties as well as several adults, are to be chosen at random to evaluate the proposed application, SearchCare. The evaluation's feedback and results are gathered, analysed, and presented graphically.

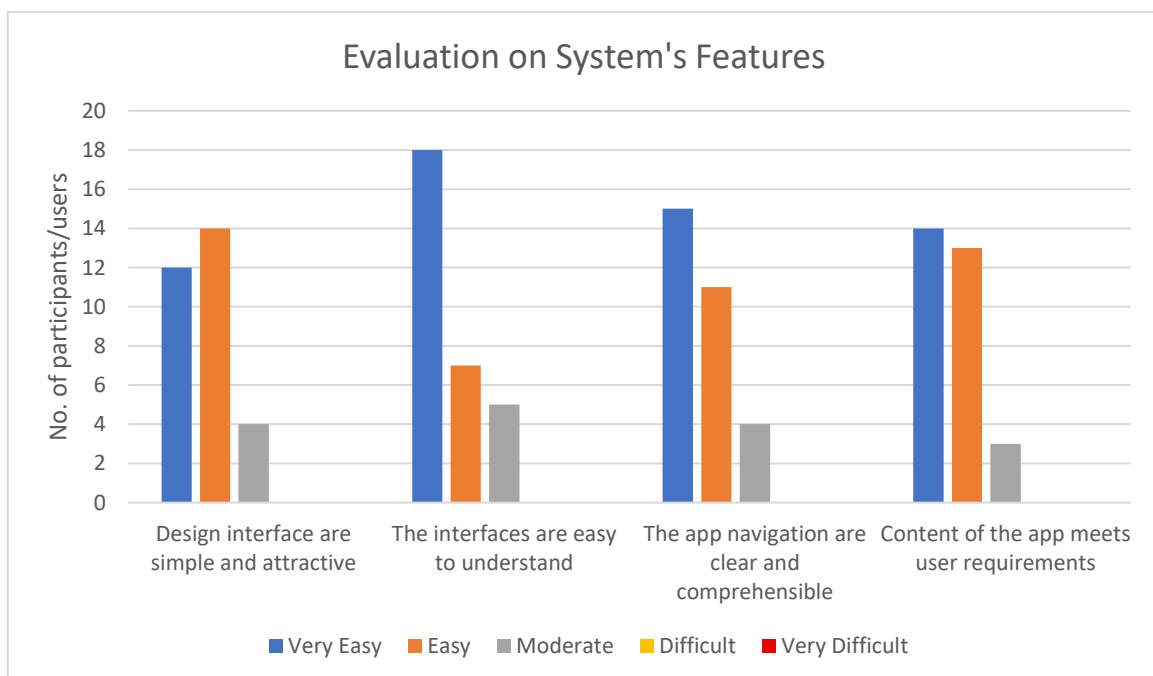


Figure 92 Multi Bar Chart of Evaluation on System's Features

Figure 92, the multi-bar graph titled “*Evaluation on System’s Features*” presents feedback from 30 participants on four core aspects of the SearchCare app. For the feature “*Design interface is simple and attractive*”, 12 participants (40%) rated it as Very Easy, 14 (46.7%) as Easy, and 4 (13.3%) as Moderate. This indicates that over 86% of users found the design either *Very Easy* or *Easy*, showing high satisfaction with the interface simplicity and attractiveness. Next, regarding “*The interfaces are easy to understand*”, 18 (60%) chose Very Easy, 7 (23.3%) Easy, and 5 (16.7%) Moderate. Therefore, a strong majority (83.3%) found the interface easy to understand, highlighting good usability and intuitiveness. For “*The app navigation is clear and comprehensible*”, 15 (50%) found it Very Easy, 11 (36.7%) Easy, and 4 (13.3%) Moderate. Thus, 86.7% of users rated navigation as *Very Easy* or *Easy*, reflecting a well-structured and intuitive navigation flow. Lastly, for “*Content of the app meets user requirements*”, 14 (46.7%) rated it Very Easy, 13 (43.3%) Easy, and 3 (10%) Moderate. Furthermore, an overwhelming 90% believe the content meets their requirements (Very Easy or Easy), showing the content is relevant and appropriate. The chart illustrates high user satisfaction across all measured system features, with at least 83% of participants finding each feature *Very Easy* or *Easy*. This indicates that SearchCare's usability and interface design are well-received, with minimal difficulty reported.

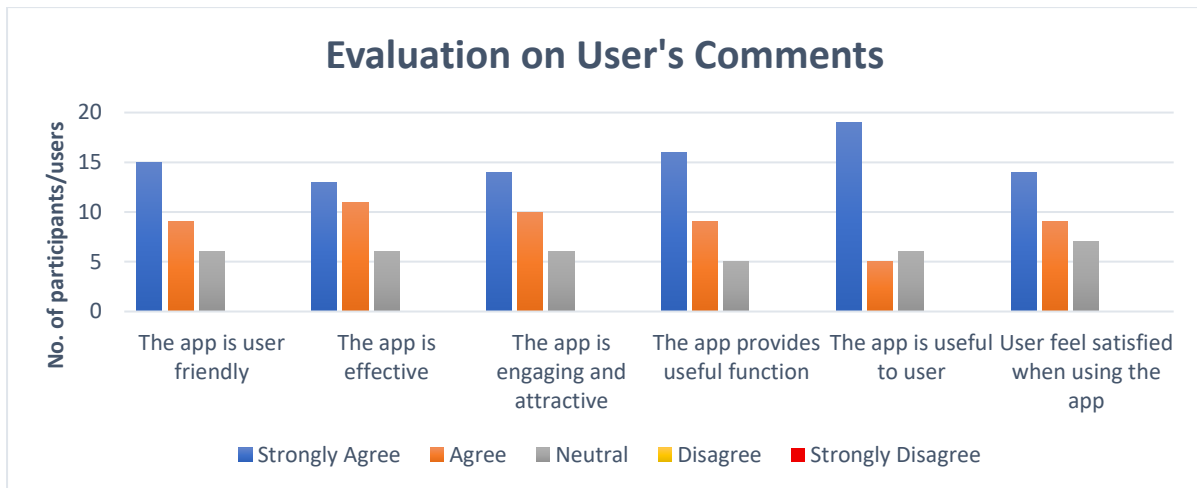


Figure 93 Multi Bar Chart of Evaluation on Users’ Comments

Figure 93 illustrates the multi-bar graph titled “*Evaluation on System’s Features*”, responses from 30 participants across six key user experience statements about the SearchCare app. For “*The app is user friendly*”, 15 participants (50.0%) strongly agreed, 9 (30%) agreed, and 6 (20.0%) were neutral. This is because most users found the app intuitive and easy to use. The high percentage of agreement reflects that users could navigate and interact with the system

comfortably without requiring much guidance. Next, *“The app is effective”*, 13 (43.3%) strongly agreed, 11 (36.7%) agreed, and 6 (20.0%) remained neutral. This is due to the fact that the majority of users believe that the app performs its intended functions well. It achieves its goals in helping users manage their healthcare-related tasks, which enhances its perceived usefulness. Then, regarding *“The app is engaging and attractive”*, 14 (46.7%) strongly agreed, 10 (33.3%) agreed, and 6 (20.0%) were neutral. Nearly half of the participants felt the app was visually appealing and kept them interested. This suggests that the app’s design and content presentation are effective in maintaining user attention and interaction. For *“The app provides useful function”*, 16 (53.3%) strongly agreed, 9 (30%) agreed, and 5 (16.7%) were neutral. A strong half of the users recognized the practical utility of the app’s features. The functions provided are considered relevant and beneficial to users’ healthcare needs, indicating successful feature implementation. In the statement *“The app is useful to user”*, a notable 19 participants (63.3%) strongly agreed, 5 (16.7%) agreed and 6 (20.0%) were neutral. This is the most positively rated statement, showing that users clearly acknowledge the overall value the app brings to them. Finally, for *“User feel satisfied when using the app”*, 14 participants (46.7%) strongly agreed, 9 (30.0%) agreed, and 7 (23.3%) were neutral. Half of the participants experienced high satisfaction using the app, and one-third agreed with the sentiment. This reflects a strong emotional and practical approval of the app, affirming a successful user experience.

5.4 Conclusion

With the tests comprising system tests, usability tests, and in-depth system tests for the SearchCare app, the findings are that the system operates effectively and consistently in accordance with its stated purposes. The system tests validated that all the modules like user registration, consultation with physicians, medication procurement, and content navigation operate seamlessly without any significant errors. The screen navigation, responsiveness of operations, and compatibility with Firebase services were guaranteed to fulfill the specified technical requirements. This means that the app is stable with well-developed features that facilitate health-related communications between users and medical professionals.

The usability test and assessment validate the system's effectiveness on the part of the users. The participants collectively rated the application as user-friendly, efficient, interactive, and useful, with most of them indicating "Very Easy" or "Strongly Agree" in the feedback. The most important aspects of interface usability, navigation ease, and content relevance were all positively achieved, with no users indicating significant dissatisfaction. This is an attestation

that the application is fulfilling the purpose it was meant to achieve while being efficient and satisfactory in terms of user experience. SearchCare shows good performance both in technical stability and usability to render it a practical, user-centered solution for the healthcare industry.

Chapter 6: Conclusion and Future Work

6.1 Introduction

The development of the suggested system will be summarised and discussed in this chapter. It will be explained whether it can accomplish all of the goals outlined in Chapter 1 as well as the applicants' constraints, since the implementation phase already demonstrates the accomplishments and advancements made. Additionally, this chapter clarifies suggestions for future work and enhancements for the suggested applications.

6.2 Objective Achievement

The objective of the SearchCare: Patient Smart Mobile Application for Personalized Healthcare Recommendations mobile application has been successfully accomplished and achieved from the SearchCare apps are stated in the table below:

Table 9 Objective's Achievement

Objective	Description	Achievement
Objective 1	To gather relevant data, including patient's demographics, medical history, symptoms, and preferences, as well as healthcare provider data such as expertise, rating, availability, and location.	<ul style="list-style-type: none"> - User registration collects full name, gender, city, country, phone and ID. - Medical history and criminal record stored in Firebase. - Doctors register with license number, expertise, workplace. - Doctor availability and admin-set location data are integrated.
Objective 2	To develop a recommendation algorithm that matches patients with their search and filter inputted symptoms and fetch from the database.	<ul style="list-style-type: none"> - Filter/search functions for doctors by name or expertise. - Symptom search returns relevant facts or conditions. - Categorized medicine listings support guided choices.

Objective 3	To integrate the recommendation system with mobile applications for user interaction.	<ul style="list-style-type: none"> - Integrated UI using Flutter and Firebase backend. - User Menu enables access to search, booking, and location features. - Booking and consultation requests flow directly to doctors. - Medicine module includes purchase, delivery, and payment options. - Firebase Authentication ensures secure role-based access.
----------------	---	---

6.3 Application Constraint

There were a small number of constraints occurred throughout the implementation of the proposed application. During the development of the application, several technical and operational constraints were encountered that affected the progress and overall efficiency. One of the primary challenges was the need to learn Dart, the programming language used with Flutter in Android Studio. As Dart was a new language for the development, time had to be allocated to understand its syntax, state management, widget structure, and asynchronous programming patterns. This learning curve slightly delayed early stages of development, especially in UI design and integration with backend services. Additionally, integrating Firebase services like Authentication and Firestore into the Flutter app required careful configuration in Android Studio, where missteps in linking packages, setting up google-services.json, or applying correct permissions could break app functionality.

Moreover, authenticating and managing two distinct user roles — normal users and doctors — within Firebase posed challenges in database structure and access control. Ensuring that each type of user only had access to appropriate data required the use of custom claims, role-based Firestore rules, and secure document organization. This complexity was further compounded by delays in conducting interviews with actual users, which limited insight into real-world use cases and user experience expectations. Another constraint involved ensuring the accuracy of medicine orders, as it was critical to match the user’s selection with the correct item from the database. This required meticulous validation, precise UI interaction, and reliable

syncing between UI state and Firestore data. Any mismatch could lead to incorrect purchases, affecting both user satisfaction and operational integrity.

6.4 Project Limitation

During the implementation of the project, several limitations were encountered that impacted the integration of key features. One major challenge was the inaccessibility of Google Maps services. This was primarily due to API restrictions, billing account issues, and platform compatibility, which prevented the app from displaying dynamic maps or leveraging geolocation features for delivery tracking and user address suggestions. As a result, users were unable to view real-time delivery routes or pickup points, reducing the app's usability and interactivity. In cases where map access was partially functional, reliance on internet connectivity and device permissions further limited consistent user experience. To overcome this, static maps or alternative solutions like OpenStreetMap were considered, albeit with reduced functionality.

Another significant limitation was the inability to integrate real-time delivery service functions and secure QR code payment methods. Most local delivery providers do not offer accessible or affordable public APIs for real-time logistics, and partnerships or business-level accounts are often required. This restricted the app to basic delivery status updates without live tracking. Additionally, implementing QR-based payment methods such as DuitNow or FPX was not feasible due to security and compliance concerns. Handling sensitive payment data requires strict adherence to security standards like PCI-DSS, which were beyond the project's current scope. As a workaround, manual payment confirmation and secure third-party gateways were explored to facilitate user transactions safely while maintaining project viability.

6.5 Future Work

To enhance the reliability and performance of the smart healthcare mobile application, several future improvements are proposed. One key enhancement involves optimizing backend performance by implementing Cloud Functions and Firestore triggers, which would automate tasks such as appointment status updates, email notifications, and medicine stock adjustments in real time. Additionally, real-time analytics and crash monitoring tools like Firebase Performance Monitoring and Crashlytics will be integrated to track usage patterns and automatically detect bugs or performance bottlenecks, improving the app's stability and responsiveness. Enhancing authentication with multi-factor authentication (MFA) could also add an extra layer of security, especially for sensitive user and doctor data.

From a UI/UX perspective, the app will evolve to include more interactive and user-friendly features such as voice-based input for appointment booking, dark mode for accessibility, in-app chat or virtual assistant, and personalized dashboards that show appointment history, medicine orders, and health tips. These additions aim to make the app more intuitive, especially for elderly patients or first-time users. For deeper domain alignment, collaboration with medical professionals and lecturers from PRIMA UNIMAS will be sought to review, validate, and guide the medical accuracy and features of the app. Their inspection will ensure the app adheres to real-world clinical workflows and ethics, strengthening its credibility. Furthermore, future updates will include printable or scannable QR code-based appointment approvals, or integration with national healthcare systems for verifying genuine bookings. This feature will help prevent misuse of appointment slots, ensuring that only approved patients can access consultations, which is vital for managing medical resources efficiently.

6.6 Recommendation

To ensure the safety and integrity of the smart healthcare application, it is highly recommended to implement advanced cybersecurity measures, especially for sensitive user roles like doctors. One of the most critical improvements would be the adoption of Multi-Factor Authentication (MFA) for doctor accounts. This added layer of security—combining a password with a verification code sent via SMS, email, or an authenticator app—would significantly reduce the risk of account breaches. Preventing unauthorized access is crucial, as any compromise could lead to misuse of doctor profiles, potentially deceiving patients or leaking confidential medical information.

In addition to authentication, enhanced data protection strategies should be applied to the medicine database. This includes role-based access control (RBAC), encryption at rest and in transit, and real-time logging of all access to medicine records. These measures will protect against unauthorized modifications and ensure the reliability of medicine-related data. Furthermore, the app should integrate automated notifications via SMS or WhatsApp for doctors, reminding them of upcoming appointments. This improves appointment adherence and ensures timely preparation for consultations. Another valuable feature would be the implementation of a secure online chat system between doctors and patients, allowing real-time communication for follow-up questions or advice, especially for virtual consultations. Lastly, expanding the app's symptom database with a wider variety of medical conditions and common complaints will enrich the symptom search functionality, enabling users to better

identify potential health issues and receive more accurate recommendations for appointments or treatments. These features collectively enhance the app's usability, security, and clinical relevance.

6.7 Conclusion

By providing a strong and user-focused healthcare platform, the SearchCare: Patient Smart Mobile Application for Personalised Healthcare Recommendations has effectively achieved its intended goals. The application effortlessly handles user registration, doctor consultation booking, medication purchases, and healthcare recommendations by integrating Firebase for real-time data management and Flutter for effective cross-platform UI development. Through modular features like symptom search, appointment scheduling, and personalised medication recommendations, the program seamlessly supports two different user roles—patients and doctors—while maintaining user privacy, data integrity, and ease of use.

Notwithstanding these successes, the idea ran into a number of obstacles and limits when it was put into action. Development was momentarily hampered by technical issues like early inexperience with the Dart programming language, Firebase role-based access management, and Flutter's asynchronous features. The scope of real-time interactivity and convenience was further limited by external factors, such as the absence of secure QR code payment systems and Google Maps. To preserve essential functionality, the team used innovative workarounds including third-party gateways and static maps. Additionally, schedule restrictions made it difficult to do real user testing, which created small obstacles to obtaining empirical input on the app's performance and usability.

The SearchCare system has a lot of room to grow and improve in the future. Future research will concentrate on integrating interactive elements like voice input and chat-based communication, implementing multi-factor authentication (MFA) for secure authentication, automating tasks in real-time with Cloud Functions, and improving medical validation in conjunction with healthcare professionals. The symptom database should be expanded, and secure QR-based workflows should be used to improve the medication and appointment systems. Enhancing overall patient outcomes, bringing the system closer to real-world clinical operations, and boosting system credibility all depend on these initiatives. As a result, the SearchCare app provides a useful and scalable platform for the provision of individualised, digital healthcare in Malaysia and beyond.

Reference

- About us - Best Corporate Healthcare Services. (2023, November 8). MediIQ Smart Healthcare. <https://mediq.com.pk/about-us-best-corporate-healthcare-services/>
- Anusuya, S., & Mallika, N. M. (2022). Web page recommendation system using bat optimization and weighted support vector machine algorithm for health care service. *International Journal of Health Sciences*, 5296–5317. <https://doi.org/10.53730/ijhs.v6ns2.6333>
- Casino, F., Patsakis, C., Batista, E., Postolache, O., Martínez-Ballesté, A., & Solanas, A. (2018, September 1). Smart Healthcare in the IoT Era: A Context-Aware Recommendation Example. *IEEE Xplore*. <https://doi.org/10.1109/ISSI.2018.8538106>
- Hayyolalam, V., Aloqaily, M., Ozkasap, O., & Guizani, M. (2021). Edge Intelligence for empowering IoT-Based healthcare systems. *IEEE Wireless Communications*, 28(3), 6–14. <https://doi.org/10.1109/mwc.001.2000345>
- Hossain, E., Rana, R., Higgins, N., Soar, J., Barua, P. D., Pisani, A. R., & Turner, K. (2023). Natural Language Processing in Electronic Health Records in relation to healthcare decision-making: A systematic review. *Computers in Biology and Medicine*, 155, 106649. <https://doi.org/10.1016/j.combiomed.2023.106649>
- Ihnaini, B., Khan, M. A., Khan, T. A., Abbas, S., Daoud, M. Sh., Ahmad, M., & Khan, M. A. (2021). A Smart Healthcare Recommendation System for Multidisciplinary Diabetes Patients with Data Fusion Based on Deep Ensemble Learning. *Computational Intelligence and Neuroscience*, 2021, 1–11. <https://doi.org/10.1155/2021/4243700>

- Joshi, P., Santy, S., Budhiraja, A., Bali, K., & Choudhury, M. (2020). The state and fate of linguistic diversity and inclusion in the NLP world. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6282–6293.
- Kumar, R., Rani, N., Singla, S., & Bansal, J. (2024). Enhance Healthcare. *INTERNATIONAL JOURNAL of SCIENTIFIC RESEARCH in ENGINEERING and MANAGEMENT*, 08(07), 1–6. <https://doi.org/10.55041/ijrem36512>
- Maniou, T. A., & Veglis, A. (2020). Employing a chatbot for news dissemination during crisis: Design, implementation and evaluation. *Future Internet*, 12(7).
- Nasr, M., Islam, M. M., Shehata, S., Karray, F., & Quintana, Y. (2021). Smart healthcare in the age of AI: recent advances, challenges, and future prospects. *IEEE Access*, 9, 145248–145270. <https://doi.org/10.1109/access.2021.3118960>
- Rahman, A., Hossain, M. S., Muhammad, G., Kundu, D., Debnath, T., Rahman, M., Khan, M. S. I., Tiwari, P., & Band, S. S. (2022). Federated learning-based AI approaches in smart healthcare: concepts, taxonomies, challenges and open issues. *Cluster Computing*, 26(4), 2271–2311. <https://doi.org/10.1007/s10586-022-03658-4>
- Sharma, R., & Rani, S. (2020). A Novel Approach for Smart-Healthcare Recommender System. *Advances in Intelligent Systems and Computing*, 503–512. https://doi.org/10.1007/978-981-15-3383-9_46
- Wickramasinghe, N., & Geisler, E. (Eds.). (2008). *Encyclopedia of Healthcare Information Systems*. IGI Global. <https://doi.org/10.4018/978-1-59904-889-5>
- Zhou, B., Yang, G., Shi, Z., & Ma, S. (2022). Natural language processing for smart healthcare. *IEEE Reviews in Biomedical Engineering*, 17, 4–18. <https://doi.org/10.1109/rbme.2022.3210270>

Appendices

Chapter 3

Information Gathering Questionnaire

Section A:

Question 1: Age: 20-29, 30-39, 40-49, 50-59, 60 and above

Question 2: Government medical centre or private medical centre

Question 3: Walk-in into medical centre or Online appointment/consultation with doctors

Section B and C:

Strongly agree, Agree, Neutral, Disagree, Strongly disagree

Smart Healthcare Application: User Experience Questionnaire	
Section A	Age (years old)
	Which healthcare center type would be preferable during visit?
	Preference on getting source of healthcare
Section B	Would you prefer to get treatment through online consultation and online diagnosis just to save time during waiting session during walk-in session?
	Would you prefer to do online appointment with your doctor at the nearest clinic/hospital to avoid traffic on the road?
	Would you prefer to get your medical prescriptions through online consultation from your nearest clinic/doctor?
	Which method would you prefer to get your medicine after an online diagnosis or online consultation?
Section C	Would you prefer to receive daily notifications about healthcare tips everyday?

	Would you prefer to receive feedback about your daily diet intake if you want to manage your diet efficiently?
	Would you like to know there are different type of category of healthcare tips?

Chapter 5

System Evaluation

Section A:

Very easy, Easy, Moderate, Difficult, Very difficult

Section B:

Strongly agree, Agree, Neutral, Disagree, Strongly disagree

Feedback survey about SearchCare usability	
Section A	Design interface are simple and attractive
	The interfaces are easy to understand
	The app navigation are clear and comprehensible
	Content of the app meets user requirements
Section B	Content of the app meets user requirements
	The app is user friendly
	The app is effective
	The app is engaging and attractive
	The app provides useful function