



Faculty of Computer Science and Information Technology

IOT-BASED SMART GARMENT DONATION STATION SYSTEM

YAP WING TOM

Bachelor of Computer Science with Honours
(Network Computing)

2025

IOT-BASED SMART GARMENT DONATION STATION SYSTEM

YAP WING TOM

This project is submitted in partial fulfilment of the
requirements for the Degree of Bachelor of Computer Science with Honors
(Network Computing)

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2025

DECLARATION

I hereby declare that this project together with all the content is my original work where I have not copied and plagiarism from other's work except due to the research – based information and related materials that were adjusted and take out from other resources, which have clearly been cited or stated respectively.

I confirm that the work presented in this thesis is original and has been prepared by me. Any assistance or guidance received in the preparation of this thesis has been properly acknowledged.

Signed,



.....

Yap Wing Tom (81529)

Bachelor of Computer Science with Honours (Network Computing)

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak (UNIMAS)

23 June 2025

ACKNOWLEDGMENT

I would like to express my sincere gratitude to our Final Year Project (FYP) coordinator, Prof. Wang Yin Chai, for providing valuable guidance and insightful advice throughout the course of FYP 2. His lectures and instructions greatly shaped the direction and progress of this project. A special note of appreciation goes to my supervisor, Madam Seleviawati Tarmizi, for her unwavering support, expert suggestions, and constructive feedback at every stage of development. Her guidance helped me refine the project's direction, and her generous loan of Arduino hardware was instrumental in the early prototyping and testing phases. I would also like to thank Associate Professor Dr. Kartinah Zen for kindly lending me additional Arduino hardware components, which were crucial for completing the hardware integration in this project. My heartfelt thanks go to our examiner, Assoc. Prof. Dr. Mohamad Nazim bin Jambli, for his professional evaluation and feedback. I am equally grateful to all reviewers and respondents who provided constructive input and suggestions that helped improve the project. Lastly, I would like to thank my family and friends for their constant encouragement, moral support, and belief in my work during this journey.

Abstract

This paper presents the development and implementation of a smart garment donation monitoring system that leverages the Internet of Things (IoT), Firebase cloud services, and Flutter mobile development. The system is designed to detect donation bin fill levels using ultrasonic sensors and transmit real-time data to a centralized platform accessible by both users and administrators. It integrates embedded hardware, cloud-based storage, and cross-platform mobile user interfaces to enable real-time monitoring, user engagement, and efficient donation event management. Comprehensive testing ensured the functionality, usability, and reliability of the system. The paper also discusses limitations and proposes enhancements for future versions.

Keywords: IoT, smart donation system, garment donation, real-time monitoring, user engagement

Abstrak

Kertas ini membentangkan pembangunan dan pelaksanaan sistem pemantauan pendermaan pakaian pintar yang menggunakan Internet Kebendaan (IoT), perkhidmatan awan Firebase, dan pembangunan mudah alih Flutter. Sistem ini direka untuk mengesan tahap isian tong pendermaan menggunakan sensor ultrasonik dan menghantar data masa nyata ke platform berpusat yang boleh diakses oleh pengguna dan pentadbir. Ia menggabungkan perkakasan terbenam, penyimpanan berasaskan awan, dan antara muka pengguna mudah alih rentas platform untuk membolehkan pemantauan masa nyata, penglibatan pengguna, dan pengurusan acara pendermaan yang cekap. Ujian menyeluruh telah dijalankan bagi memastikan fungsi, kebolehgunaan dan kebolehpercayaan sistem. Kertas ini juga membincangkan had sistem dan mencadangkan penambahbaikan untuk versi akan datang.

Kata kunci: IoT, sistem pendermaan pintar, pendermaan pakaian, pemantauan masa nyata, penglibatan pengguna.

Table of Contents

DECLARATION.....	I
ACKNOWLEDGMENT	II
ABSTRACT	III
ABSTRAK	IV
CHAPTER 1.....	1
INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Problem Statement	2
1.3 Scope.....	3
1.4 Aim and Objectives.....	3
1.5 Brief Methodology	4
1.5.1 Requirement Gathering Phase	4
1.5.2 System Design Phase.....	4
1.5.3 Development Phase	4
1.5.4 Testing and Deployment Phase.....	4
1.6 Significant of Project	5
1.7 Project Schedule.....	6
1.8 Expected Outcomes	8
1.9 Chapter Outline	8
CHAPTER 2	9
LITERATURE REVIEW	9
2.1 Introduction.....	9
2.2 IoT in Donation Systems.....	10
2.3 Practical Applications in Donation Systems	11
2.4 Analysis of Comparable Systems.....	12
2.4.1 Smart Donation Machine Using Arduino	12
2.4.2 Android-Based Application for Food, Clothes, and Books Donation.....	13
2.4.3 Application for Donation Centers.....	15
2.4.4 IoT-Based Smart Donation Box.....	17

2.5 Comparison of Existing Systems	19
2.6 Current Gaps in Research	21
2.6.1 Garment-Specific Features	21
2.6.2 Advanced Security	21
2.6.3 Enhanced User Engagement	21
2.6.4 Scalability	22
2.7 Future Trends and Research Directions	23
2.8 Summary	24
CHAPTER 3.....	25
Methodology	25
3.1 Introduction.....	25
3.2 System Requirements.....	25
3.3 Tools and Technologies.....	26
3.3.1 Hardware Tools.....	26
3.3.2 Software Tools.....	28
3.4 System Design	29
3.4.1 Use Case Diagram	29
3.4.2 Flow Chart of the Proposed System	30
3.4.3 Hardware System Architecture Design.....	32
3.4.4 Architecture Design	34
3.4.4.1 IoT Layer.....	34
3.4.4.2 Application Layer.....	35
3.4.4.3 Cloud Layer.....	35
3.4.5 Prototype (UI Design)	36
3.4.6 Database Design	40
3.5 Development Phase.....	42
3.5.1 Hardware Development	42
3.5.2 Software Development	43
3.5.3 Testing.....	44
3.5.3.1 Unit Testing	44
3.5.3.2 Integration Testing.....	44
3.5.3.3 Real-Device Testing	44
3.6 Summary	45
CHAPTER 4.....	46
Implementation	46
4.1 Introduction.....	46
4.2 Hardware Components.....	47
4.2.1 Arduino Nano 33 IoT board.....	47
4.2.2 HC-SR04 ultrasonic sensor	47

4.2.3 LED Indicators	47
4.2.4 Breadboard and Jumper Wires.....	47
4.3 Hardware Components Connection	48
4.3.1 First Ultrasonic Sensor (HC-SR04) and LED Indicators.....	48
4.3.2 Second Ultrasonic Sensor (HC-SR04) and LED Indicators	49
4.4 Software Components.....	51
4.4.1 Embedded System Integration.....	51
4.4.1.1 PlatformIO with Arduino Configuration and Firebase Realtime Database Integration ...	51
4.4.2 Flutter	52
4.4.2.1 UI of Flutter.....	52
4.4.3 Firebase.....	68
4.4.3.1 Firestore Database	69
4.4.3.2 Firebase Authentication.....	74
4.4.3.3 Firebase Realtime Database	75
4.5 Key Features	77
4.5.1 Dual-Bin Monitoring with Ultrasonic Sensors	77
4.5.2 Real-Time, Per-Bin Data Streams to Firebase	77
4.5.3 On-Site LED Indicators	78
4.5.4 Secure & Scalable Cloud Connectivity	78
4.5.5 Instant Feedback in the Flutter Application.....	78
4.5.6 Flutter Application Features	79
4.5.6.1 Locate Donation Bins and Stations	79
4.5.6.2 View and Participate in Event Promotions	79
4.5.6.3 Submit and Promote Your Event	80
4.6 Summary	81
CHAPTER 5.....	82
Testing.....	82
5.1 Introduction.....	82
5.2 Functional Testing.....	83
5.3 Non-Functional Testing.....	84
5.3.1 Using Real Phone Device – OPPO Reno5 F	84
5.3.2 Firebase Test Lab (Cloud-Based Non-Functional Testing).....	85
5.3.3 Firebase Security Rules Simulator (Security Testing)	86
5.3.4 Dart Analyze	87
5.4 Test Case	89
5.4.1 Test Cases for SplashScreen and IntroScreen.....	89
5.4.2 Test Cases for Sign-In.....	90
5.4.3 Test Cases for Sign-Up	91
5.4.4 Test Cases for User dashboard.....	92
5.4.5 Test Cases for User Profile	93
5.4.6 Test Cases for Locate Donation Station and Bins Page.....	94
5.4.7 Test Cases for Event Promotion Page.....	95
5.4.8 Test Cases for Submit and Promote Your Event Page	97
5.4.9 Test Cases for admin Dashboard	99
5.4.10 Test Cases for Notifications bins	100

5.4.11 Test Cases for Add Donation Station Page	102
5.4.12 Test Cases for Add Donation Bins Page	103
5.5 Summary	107
CHAPTER 6.....	108
Conclusion and Future Work	108
6.1 Introduction.....	108
6.2 Achievement of Objectives	109
6.3 Limitations	111
6.3.1 Internet Dependency for IoT and Mobile App.....	111
6.3.2 Image Loading Issues	111
6.3.3 Lack of Offline Support.....	111
6.3.4 Limited Validation and User Feedback.....	111
6.3.5 Simplified Admin Workflow	112
6.3.6 Limited User Profile Features.....	112
6.3.7 Firebase Plan Limitations (FCM and Storage Not Implemented)	112
6.3.8 Limited Adoption & Community Engagement.....	112
6.3.9 Justification of Hardware Use in Certain Areas.....	112
6.4 Future Improvements	114
6.4.1 Enhanced Arduino-Based Hardware Setup.....	114
6.4.2 Robust Image and Asset Handling.....	114
6.4.3 Offline Mode and Local Caching	114
6.4.4 Improved Admin Dashboard with Analytics	115
6.4.5 Push Notification Integration (FCM).....	115
6.4.6 Advanced User Profile Features	115
6.4.7 Event Categorization and Scheduling Tools.....	115
6.4.8 Enhanced Map and Location Features.....	115
6.4.9 Accessibility and Multilingual Support	115
6.4.10 Security Enhancements and Role-Based Access	116
6.4.11 Data Analytics and Reporting Tools	116
6.4.12 Adoption Campaigns and Outreach.....	116
6.5 Summary	117
References.....	118

List of Figures

Figure 1.1: Project Schedule FYP 1	6
Figure 1.2: Project Schedule FYP 2	7
Figure 2.1: Hardware Integration Of The Smart Donation Machine (Krishna Et Al., 2017).....	12
Figure 2.2: Use Case Diagram Of Communication Between Clients And System (Pandey & Kumar, 2022).....	13
Figure 2.3: Hci-Based Design Phases Of The Seva App (Pandey & Kumar, 2022).....	14
Figure 2.4: Entity Relationship Diagram (Satam Et Al., 2021)	15
Figure 2.5: System Design (Satam Et Al., 2021).....	16
Figure 2.6: Block Diagram Of Smart Donation Box (Setiawan & Hariono, 2024).....	17
Figure 3.1: Iot Device For Hardware Tools	26
Figure 3.2: Use Case Diagram	29
Figure 3.3: Flow Chart Of The Proposed Iot-Based Smart Donation Bin	30
Figure 3.4: System Architecture Design showing the connection of all the hardware	32
Figure 3.5: Architecture of the IoT-Based Smart Garment Donation Station System	34
Figure 3.6: UI Design for splash screen 1 and 2	36
Figure 3.7: UI Design for Sign-in and sign-up page	37
Figure 3.8: UI Design for home page	37
Figure 3.9: UI Design for Locate Station Page, Locate Bins Page and Event Promotion Page	38
Figure 3.10: UI Design for Profile Page	39
Figure 3.11: Database Design	40
Figure 3.12: Phased Development Workflow for an IoT-Based Smart Garment Donation Station.....	43

Figure 4.1: Connection of All of Hardware Components	48
Figure 4.2: UI for While User Loading the Apps.....	52
Figure 4.3: UI for User Sign In and Sign Up	53
Figure 4.4: UI for User Dashboard and Profile Station	54
Figure 4.5: UI for Locate Donation Stations and Bins.....	55
Figure 4.6: UI for Search Feature of Locate Donation Station and Bins	56
Figure 4.7: UI for Event Promotion.....	57
Figure 4.8: UI for Event Promotion History	58
Figure 4.9: UI for Corporate Event Promotion	59
Figure 4.10: UI for Admin Dashboard	60
Figure 4.11: UI for Notification Bins.....	62
Figure 4.12: UI for Add Donations Station and Bins.....	63
Figure 4.13: UI for Approve Event Promotions.....	64
Figure 4.14: UI for Approve Event Promotions of Approved Events and Rejected Events	66
Figure 4.15: Firestore Database of Collections for Users Collection	69
Figure 4.16: Firestore Database of Collections for Donation Station Collection	70
Figure 4.17: Firestore Database of Collections for Donation Bins Collection	71
Figure 4.18: Firestore Database of Collections for Event Promotion Collection	72
Figure 4.19: Firebase Authentication	74
Figure 4.20: Firebase Realtime Database	75
Figure 5.1: Firebase Test Lab about Robo Test of Pixel 5, API level 30	85

Figure 5.2: Firebase Test Lab about Robo Test of Test Issues	85
Figure 5.3: Results of Test Run Rules Playground in Rules of Real-Time Database	86
Figure 5.4: Results of Dart Analyze.....	87

List of Table

Table 1.1: Comparison Of Existing Systems with Proposed System	19
Table 2.1: Connection of Arduino Nano 33 IoT with First Ultrasonic Sensor (HC-SR04) and Led ...	48
Table 2.2: Connection of Arduino Nano 33 IoT with Second Ultrasonic Sensor (HC-SR04) and Led	49
Table 2.3: Visual and Informational Differences Between Donation Stations and Donation Bins	55
Table 3.1: Test Cases for SplashScreen and IntroScreen	89
Table 3.2: Test Cases for Sign-In	90
Table 3.3: Test Cases for Sign-Up	91
Table 3.4: Test Cases for User dashboard	92
Table 3.5: Test Cases for User Profile	93
Table 3.6: Test Cases for Locate Donation Station and Bins Page	94
Table 3.7: Test Cases for Event Promotion Page	95-96
Table 3.8: Test Cases for Submit and Promote Your Event Page	97-98
Table 3.9: Test Cases for admin Dashboard	99
Table 3.10: Test Cases for Notifications bins	100-101
Table 3.11: Test Cases for Add Donation Station Page	102
Table 3.12: Test Cases for Add Donation Bins Page	103
Table 4.13: Test Cases for Approve Event Promotion Page	104-106
Table 4.1: Achievement of Objectives	109

CHAPTER 1

INTRODUCTION

1.1 Introduction

The traditional approach to garment donation often encounters numerous logistical and security challenges, particularly with unmonitored donation stations. Without real-time monitoring, these stations frequently experience issues like overflowing bins and mismanaged donations, resulting in inefficiencies and possible misuse of resources. With an increasing awareness of sustainability and social responsibility, communities are exploring innovative ways to improve donation practices, especially through the use of technology (Zanella et al., 2014). IoT-based solutions, which have already demonstrated success in various sectors, offer an effective means to monitor and manage donation stations (Krishna et al., 2017).

By integrating Internet of Things (IoT) technology and a mobile application, the Smart Garment Donation Station System aims to provide a solution that enhances the entire donation process. This system will enable real-time monitoring, allowing staff to track bin capacity and receive timely notifications when bins require emptying (Prajitno et al., 2019). Additionally, it incorporates features for secure staff verification and provides donors with push notifications about donation events, ensuring the system is interactive and convenient for users (Krishna et al., 2017).

IoT innovations have shown potential for improving donation management, making it more secure and convenient for both staff and donors (Zanella et al., 2014). Similar implementations in other fields have demonstrated that real-time monitoring reduces labor costs, minimizes waste, and improves overall operational efficiency (Zafar et al., 2018). By adopting these practices within garment donation systems, we aim to reduce the burdens on administrative staff and promote increased public engagement through accessible, technology-driven solutions.

1.2 Problem Statement

Traditional garment donation stations face significant limitations that hinder their effectiveness. A lack of real-time monitoring capabilities often leads to overflowing bins, which are not only unsightly but can deter potential donors and create additional work for staff. This inefficiency in managing bin capacity also increases the likelihood of theft and improper use, as unsupervised stations are more vulnerable to unauthorized access (Howell, 2013). Moreover, thieves frequently exploit these vulnerabilities by pretending to be volunteers or authorized personnel to steal items from donation stations or bins, further compromising the integrity of the donation process (Most Charity Fraud Perpetrated by Staff or Volunteers, Report Suggests, n.d.). The conventional donation systems struggle with security, leaving donated items at risk of theft or vandalism, particularly in public spaces.

Additionally, donors frequently encounter difficulties locating nearby donation stations, resulting in fewer contributions and missed opportunities for community engagement. Traditional donation stations rarely offer interactive features or real-time updates, which limits communication with potential donors and reduces awareness of donation events or promotions (Prajitno et al., 2019). These limitations highlight the need for a more efficient and secure solution that leverages modern technology to streamline the donation process and foster stronger community participation.

By implementing an IoT-based solution, this project addresses these key issues, offering secure, real-time monitoring, location tracking, and interactive features that improve convenience and safety for both donors and staff. The Smart Garment Donation Station System aims to transform the traditional donation process, making it more accessible, reliable, and efficient for all involved.

1.3 Scope

The project focuses on integrating IoT sensors for real-time monitoring of bin levels, notifying staff when bins are full. It includes developing a user-friendly mobile app for locating donation stations, checking bin statuses, and receiving notifications. To enhance security, the system implements staff verification mechanisms to prevent unauthorized access. Additionally, it fosters community engagement through push notifications, keeping users informed about donation events, promotions, and updates. This project will primarily serve donation organizations, administrators, and individual donors, ensuring secure, convenient, and efficient garment donation.

1.4 Aim and Objectives

- To implement IoT sensors that monitor donation bin capacity in real time and notify administrators when bins are full or require attention.
- To develop a mobile app that allows users to locate nearby donation stations, view operating hours, and receive staff verification details.
- To enable the app to push notifications to users about upcoming donation events or promotional activities.

1.5 Brief Methodology

1.5.1 Requirement Gathering Phase

To successfully implement the IoT-Based Smart Garment Donation Station System, the first phase involves requirement gathering. This includes collecting inputs from key stakeholders such as donation station managers, staff, and potential donors to understand the functional needs. These requirements focus on essential features like real-time bin monitoring, location tracking for donation stations, and event notification capabilities. By engaging stakeholders early, the system can be tailored to address their specific challenges and enhance overall functionality.

1.5.2 System Design Phase

The system design phase focuses on creating both hardware and software components. For hardware, IoT-enabled sensors are designed to monitor bin capacity and will be integrated into donation bins to provide real-time updates. On the software side, a mobile application with a user-friendly interface will be developed to include features such as station location tracking, real-time monitoring of bin status, and push notifications. This design ensures a seamless connection between hardware and software, enhancing user engagement and operational efficiency.

1.5.3 Development Phase

In the development phase, the mobile application will be coded to incorporate core functionalities like station locating and real-time notifications. IoT sensors will be configured to capture and transmit data on bin status, enabling continuous monitoring. Security measures, such as staff verification features, will also be implemented to prevent unauthorized access to donation bins. These elements work together to ensure the system operates securely and reliably.

1.5.4 Testing and Deployment Phase

The testing and deployment phase ensures that the system is fully functional and user-ready. Unit tests and integration tests will validate the hardware-software interaction, particularly between the IoT sensors and the mobile app. User acceptance testing (UAT) will be conducted to confirm usability and functionality. Finally, the system will be deployed, with the mobile app made publicly available, allowing users to locate donation stations and securely contribute to the donation process. This phased approach ensures the project is robust, secure, and user-friendly.

1.6 Significant of Project

The IoT-Based Smart Garment Donation Station System brings significant benefits to both the community and managing organizations. By incorporating real-time monitoring and staff verification, the system enhances security and reduces risks of unauthorized access and theft. The integration of IoT sensors allows for efficient tracking of bin levels, preventing overflow and optimizing collection schedules, ultimately reducing staff workload. For the public, the mobile app provides a user-friendly way to locate nearby donation stations, view bin availability, and make contributions conveniently at any time. Moreover, the inclusion of push notifications about donation events and promotions encourages greater community participation, boosts donation rates, and raises awareness about charitable initiatives. This system not only improves operational efficiency but also strengthens public trust and engagement in the donation process.

1.7 Project Schedule

				Week															
				Oct				Nov				Dec				Jan			
Project Activities (FYP1)	Duration (days)	Start Date	End Date	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Selection of Project Title																			
Identify Project Title.	3	10/10/2024	13/10/2024																
Select and confirm Supervisor.	6	14/10/2024	20/10/2024																
Brief Proposal																			
Prepare Brief Project Description	2	14/10/2024	16/10/2024																
Submit to Supervisor for review and approval	2	17/10/2024	19/10/2024																
Submit to FYP System	0	20/10/2024	20/10/2024																
Full Proposal																			
Prepare Full Project Description	9	29/10/2024	7/11/2024																
Submit to Supervisor for review and approval	4	8/11/2024	12/11/2024																
Submit to FYP System	1	13/11/2024	14/11/2024																
CHAPTER 1: INTRODUCTION																			
Define Project Background and Motivation	1	15/11/2024	16/11/2024																
Identify Problem Statement	1	15/11/2024	16/11/2024																
Set Objectives and Goals	1	17/11/2024	18/11/2024																
Outline Scope of Project	1	17/11/2024	18/11/2024																
Significance of the Project	1	17/11/2024	18/11/2024																
Submit to Supervisor for review and approval	1	19/11/2024	20/11/2024																
Submit to FYP System	0	21/11/2024	21/11/2024																
CHAPTER 2: BACKGROUND/LITERATURE REVIEW/STATE OF ART																			
Conduct Literature Review	4	22/11/2024	26/11/2024																
Analyze Similar Systems and Solutions	3	27/11/2024	30/11/2024																
Identify Knowledge Gaps	3	1/12/2024	4/12/2024																
Review Relevant Technologies and Tools	4	5/12/2024	9/12/2024																
Submit to Supervisor for review and approval	2	10/12/2024	12/12/2024																
Submit to FYP System	0	13/12/2024	13/12/2024																
CHAPTER 3: METHODOLOGY/REQUIREMENT ANALYSIS AND DESIGN																			
Define System Requirements	2	14/12/2024	16/12/2024																
Develop System Architecture	4	17/12/2024	21/12/2024																
Create Detailed Design Diagrams	5	22/12/2024	27/12/2024																
Outline Project Development Tools and Environment	2	28/12/2024	30/12/2024																
Develop Testing Strategy	1	31/12/2024	1/1/2025																
Submit to Supervisor for review and approval	2	2/1/2025	4/1/2025																
Submit to FYP System	0	5/1/2025	5/1/2025																
Final report & Paper for assessment																			
Document Results and Findings	1	6/1/2025	7/1/2025																
Outline Limitations and Future Work	1	6/1/2025	7/1/2025																
Finalize Conclusion and Recommendations	1	8/1/2025	9/1/2025																
Submit to Supervisor for review and approval	2	10/1/2025	12/1/2025																
Submit to FYP System	0	12/1/2025	12/1/2025																

Figure 1.1: Project Schedule FYP 1

The Project Schedule FYP 1 (Figure 1.1) focus is on structured content development, including key tasks such as proposal writing, literature review, and methodology. The detailed timeline for these activities is illustrated in Figure 1.1.

Project Activities (FYP2)	Duration	Start Date	End Date	Week																											
				Mar				Apr				May				Jun				July											
				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4								
Detailed Requirements Analysis																															
Confirm system requirements gathered in FYP 1	2 week	25-Mar	7-Apr																												
Analyze feedback and identify any gaps or refinements needed.																															
Document a detailed list of functional and non-functional requirements																															
System Design																															
Design and create a architecture of the system.	2 week	8-Apr	21-Apr																												
Design preliminary wireframes and mockups for the user interface.																															
Choose tools, sensor and development frameworks																															
CHAPTER 4: User Interface and User Experience Design																															
Design App UI/UX Flow	3 week	22-Apr	12-May																												
User-Centric Features																															
Enable basic user interactions																															
CHAPTER 5: Communication Framework																															
Select appropriate communication protocols	3 week	13-May	2-Jun																												
Define the data format for communication																															
Establish reliable and secure data transfer between sensors, the cloud, and the app																															
Configure IoT sensors for seamless connection with the cloud																															
Test real-time data synchronization between the app and sensors.																															
Debug potential issues in data transmission.																															
CHAPTER 6: Data Management and Security																															
Database Design	3 week	3-Jun	23-Jun																												
Security Implementation																															
Testing and Validation																															
Evaluation and User Feedback																															
Collect feedback on system usability, accuracy, and functionality.	1 week	24-Jun	30-Jun																												
Use feedback to identify and prioritize enhancements.																															
Final Adjustments & Optimization																															
Refine sensor performance and app functionality.	1 week	1-Jul	7-Jul																												
Optimize the system for scalability and robustness.																															
Documentation & Final Report																															
System design documentation	1 week	8-Jul	14-Jul																												
Development processes and challenges faced																															
Testing results and feedback analysis																															
Provide user manuals for the app and the system																															
Final Presentation Preparation																															
System features and functionality	1 week	8-Jul	14-Jul																												
Live demo of the IoT-based system and mobile app																															
Highlighted results from testing and evaluation																															

Figure 1.2: Project Schedule FYP 2

The Project Schedule FYP 2 (Figure 1.2) outlines the timeline for activities such as design, implementation, testing, and deployment of the IoT-Based Smart Garment Donation Station System. This schedule ensures a structured and efficient approach to completing the development phase during FYP 2.

1.8 Expected Outcomes

The Smart Garment Donation Station System aims to revolutionize the donation process by addressing key challenges through innovative technology. Real-time monitoring of bins and staff verification mechanisms will enhance security and efficiency, ensuring that donation stations operate smoothly and securely. Additionally, the system leverages push notifications to inform users about events and promotions, driving increased donations and fostering stronger community involvement. The inclusion of a user-friendly mobile app further enhances convenience by enabling donors to easily locate nearby donation stations, check bin status, and confidently donate garments, knowing their contributions are secure.

1.9 Chapter Outline

This report is organized into six chapters to thoroughly document the development of the IoT-Based Smart Garment Donation Station System. Chapter 1 introduces the project by presenting the background, problem statement, objectives, scope, significance, methodology, and expected outcomes. Chapter 2 provides a literature review, analyzing existing donation systems and related technologies to identify current challenges and justify the need for the proposed solution. Chapter 3 outlines the methodology used throughout the project, including requirement analysis, system architecture, hardware-software integration, and development tools. Chapter 4 focuses on the implementation phase, detailing the integration of IoT components, mobile application development using Flutter, and backend services powered by Firebase. Chapter 5 presents the testing strategies and results, covering unit testing, integration testing, and real-device testing on an Android smartphone. Finally, Chapter 6 concludes the report by summarizing key achievements, addressing identified limitations, and proposing future improvements to enhance the system's performance and scalability.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter explores existing systems and technologies relevant to IoT-based donation management, focusing on their architecture, operational efficiency, and user engagement features. The primary goal is to analyse how IoT integration, mobile application functionalities, and security mechanisms have been utilized to address challenges in donation systems, such as inefficiencies, lack of monitoring, and theft. By comparing these systems with the proposed IoT-Based Smart Garment Donation Station System, this review highlights gaps in existing solutions and discusses how the proposed system aims to overcome them.

IoT-based technologies have transformed the way donation systems operate by enabling real-time monitoring and automation. For instance, IoT sensors allow administrators to track bin status, optimize collection schedules, and enhance resource allocation. Additionally, mobile applications are increasingly used to engage donors through features like station locators, notifications, and impact tracking. Security mechanisms, such as tamper-proof designs and verification systems, further protect donations and build trust among donors.

Despite these advancements, existing systems often lack the specificity required for garment donation management. Many solutions focus on general waste or resource management, failing to address unique requirements like clothing categorization, scalable deployment, or enhanced donor feedback mechanisms. Additionally, security remains a challenge, with most systems relying on basic measures that do not comprehensively prevent unauthorized access or fraud.

The subsequent sections provide a detailed analysis of similar systems, examining their strengths and limitations. A comparison with the proposed system highlights its novel contributions, including IoT-enabled garment-specific features, robust security mechanisms, and scalable infrastructure. By addressing the gaps in existing solutions, the proposed system aims to set a benchmark for modern donation management.

2.2 IoT in Donation Systems

The incorporation of IoT technologies in donation systems has revolutionized how they are managed, offering significant benefits in terms of real-time monitoring, automation, and operational efficiency. By enabling continuous data collection, IoT sensors provide administrators with actionable insights, such as bin capacity, environmental conditions, and potential security breaches. For example, Sehrawat and Gill (2019) highlighted the utility of IoT sensors for detecting bin capacity, which can automatically notify administrators when bins are nearly full. This functionality prevents overflow, reduces manual checks, and ensures timely collection and sorting.

Krishna et al. (2017) explored the integration of Arduino microcontrollers within smart donation machines, automating the donation collection process. This integration allows for precise control over operations like locking mechanisms and weight-based sorting of donated items, reducing errors and enhancing reliability. Similarly, Zanella et al. (2014) emphasized the role of IoT technologies in smart cities, focusing on their applications in waste management systems, which share similarities with donation bin management. By leveraging IoT-enabled devices, these systems improve resource allocation, minimize human intervention, and enhance overall efficiency.

Despite these advancements, many traditional donation systems lack advanced IoT features, such as real-time tamper detection or automated alerts for unauthorized access. Howell (2013) and Civil Society (n.d.) reported instances where thieves exploited unmonitored donation bins by posing as legitimate staff members, highlighting a critical gap in current systems. Addressing these challenges through IoT solutions—such as motion detectors, security cameras, and tamper-proof locks—can significantly enhance donor trust and confidence in the system.

IoT technologies provide a solid foundation for transforming donation systems by enhancing efficiency, improving security, and fostering donor engagement. However, there remains considerable scope for innovation, particularly in integrating advanced tamper detection features and real-time analytics for predictive maintenance and fraud prevention. Bridging these gaps can lead to smarter, more secure, and sustainable donation systems.

2.3 Practical Applications in Donation Systems

IoT-based solutions have proven transformative in various practical applications, including the optimization of donation systems. For instance, Setiawan and Hariono (2024) showcased how IoT-enabled donation boxes streamline processes through real-time monitoring and secure access features like QR code authentication. Such systems reduce operational inefficiencies by automating tasks like bin capacity checks and tamper detection. Similarly, Krishna et al. (2017) demonstrated the use of Arduino microcontrollers for smart donation machines, highlighting their ability to enhance security and ensure timely collection through automated notifications.

Mobile applications also play a pivotal role in practical implementations. Pandey and Kumar (2022) and Satam et al. (2021) explored Android applications designed to simplify the donation process, focusing on donor convenience. Features like location tracking and event notifications ensure donors stay engaged, while administrative tools improve operational management at donation centres. However, the lack of integration between IoT-enabled devices and mobile applications in these solutions limits their potential for a comprehensive donation management system.

While these applications have made significant strides in modernizing donation systems, they often fail to address specific requirements for garment donations. The proposed system aims to bridge these gaps by combining IoT technologies, advanced security, and garment-specific features, ensuring both practicality and scalability in real-world applications.

2.4 Analysis of Comparable Systems

This section reviews existing systems related to IoT-enabled donation management and mobile application integration. These systems highlight various technological approaches to enhancing donation processes and provide insights into their strengths, limitations, and areas for improvement.

2.4.1 Smart Donation Machine Using Arduino

Krishna et al. (2017) introduced an innovative automated donation system leveraging Arduino microcontrollers and sensors to enhance the efficiency of donation management. The system utilizes an ultrasonic sensor to monitor bin capacity, ensuring timely notifications are sent to administrators when the bins are full. This feature prevents overflow and ensures the smooth operation of the donation process. Additionally, the inclusion of a GSM module facilitates real-time communication by sending SMS alerts, while the LCD display provides users with relevant information and status updates. The system also incorporates a Bluetooth module to enable local interactions, such as configuration or user communication.

The tamper-proof design of the donation bins enhances security, protecting donated items from theft or unauthorized access. The interaction between the system's subsystems—comprising ultrasonic sensors, GSM modules, Bluetooth modules, and LCD displays—is illustrated in Figure 2.1, demonstrating how the system operates seamlessly to manage donations.

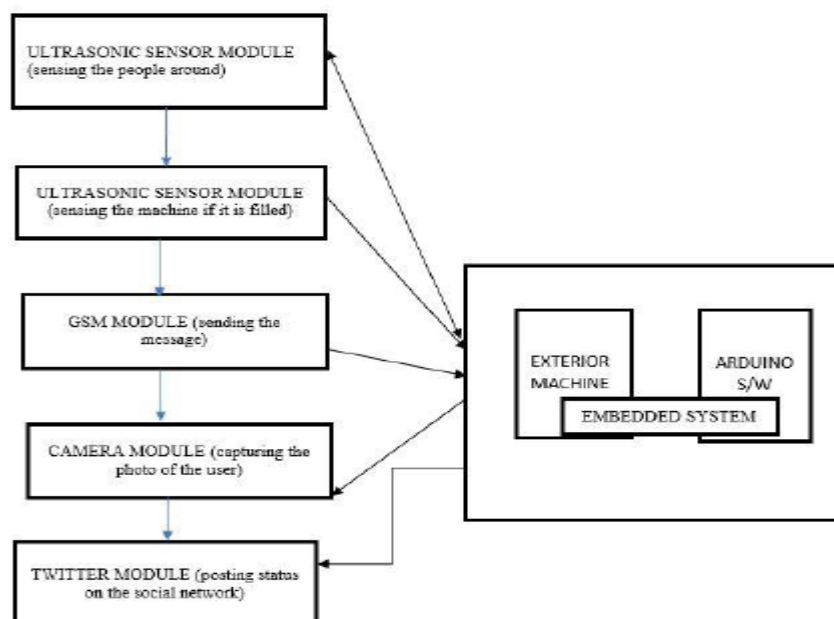


Figure 2.1: Hardware Integration of the Smart Donation Machine (Krishna et al., 2017)

While the system effectively addresses operational efficiency for general donations, it lacks features aimed at engaging donors, such as real-time updates or notifications via a mobile application. Moreover, the absence of advanced security measures, such as multi-factor authentication or remote tamper alerts, limits its application in broader and more complex donation scenarios, such as garment-specific systems. Despite these limitations, the system offers a strong foundation for small-scale implementations and provides valuable insights for enhancing donation management technologies.

2.4.2 Android-Based Application for Food, Clothes, and Books Donation

Pandey and Kumar (2022) developed an Android-based application designed to streamline the management of donations for food, clothes, and books. The app emphasizes donor engagement by integrating features such as station locators, real-time updates on donation campaigns, and notification systems to encourage active participation. Donors can enter their personal information, log into the system, and perform tasks such as checking the details of ongoing charitable events.

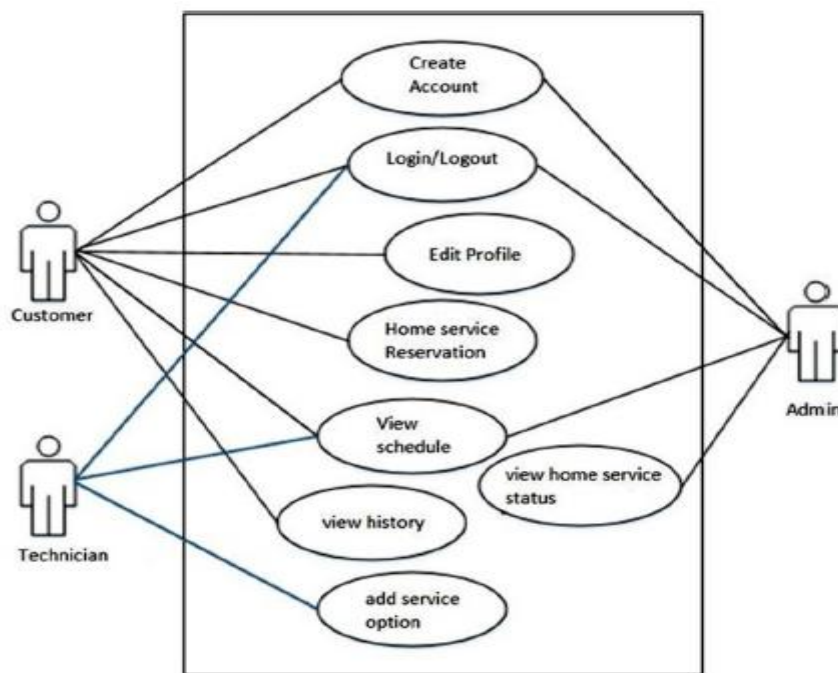


Figure 2.2: Use Case Diagram of Communication Between Customer, Admin, Technician and System (Pandey & Kumar, 2022)

As illustrated in Figure 2.2, the use case diagram highlights the interaction between users and the system, showcasing distinct functionalities for customers, administrators, and

technicians. This design, according to Pandey and Kumar (2022), ensures streamlined operations and an intuitive user experience for managing accounts, services, and schedules.



Figure 2.3: HCI-Based Design Phases of the SeVa App (Pandey & Kumar, 2022)

The app's design follows Human-Computer Interaction (HCI) principles, prioritizing usability and accessibility. As shown in Figure 2.3, the design process incorporates user feedback and requirements to ensure a seamless experience. This iterative approach to app development aims to simplify navigation and enhance functionality, making it easier for donors to interact with the system.

While the app excels in providing user-friendly interfaces and promoting donor engagement, it lacks integration with IoT technologies, which limits its ability to monitor physical donation bins or provide real-time updates on their status. Additionally, the absence of advanced security features, such as tamper detection or multi-factor authentication, reduces its capability to secure physical donation bins against unauthorized access.

Despite these limitations, the app provides a solid foundation for engaging donors and managing donations through mobile platforms. Enhancing the app with IoT features and improved security mechanisms would significantly expand its scope and effectiveness in donation management systems.

2.4.3 Application for Donation Centers

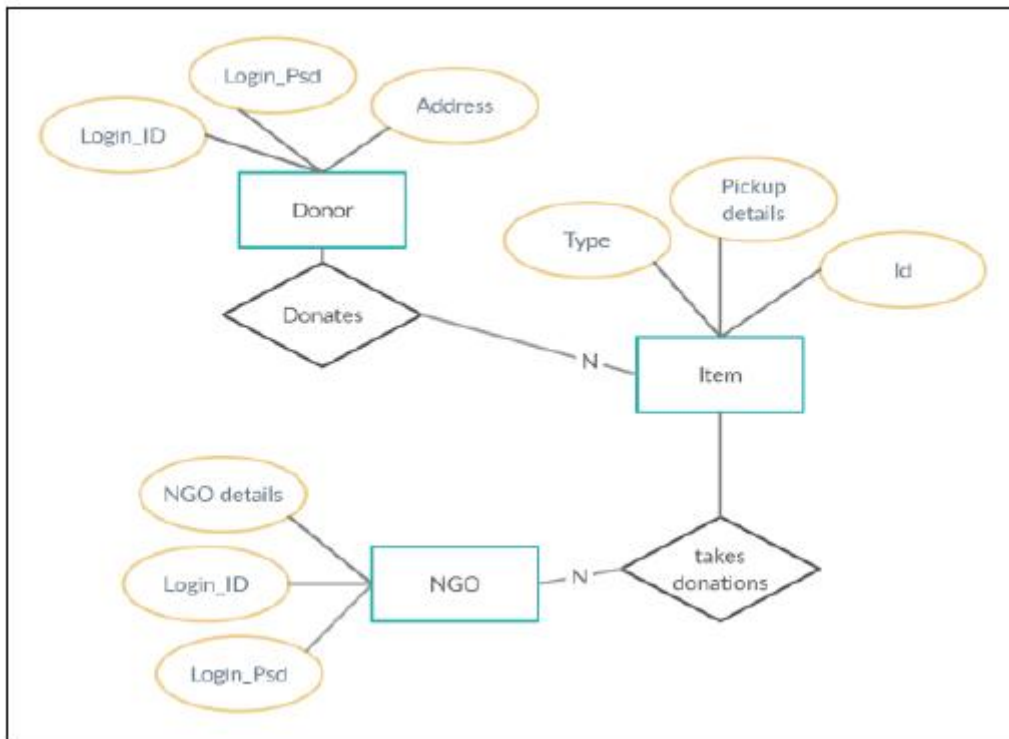


Figure 2.4: Entity Relationship Diagram (Satam et al., 2021)

The interaction between donors and NGOs is depicted in Figure 2.4. Satam et al. (2021) proposed a cross-platform mobile application aimed at optimizing donation center operations while providing a convenient user experience for both donors and NGOs. The system emphasizes flexibility and ease of access, especially useful during pandemics or for individuals with busy schedules. The application enables donors to create posts describing the commodities they wish to donate, which NGOs can view and accept through their dedicated login module.

The application leverages modern technologies like Flutter and Dart for development and integrates Firebase for cloud storage and live updates. Flutter's cross-platform capabilities ensure the app functions seamlessly across Android and iOS devices, while Firebase enhances real-time data synchronization. The system is divided into two primary modules:

1. NGO Login Module

- NGOs can log in to view and manage donation posts made by donors.
- After accepting a donation, staff members can initiate a chat with the donor for further details.

- The donation details are recorded in a history tab, and a receipt is generated for the donor upon completion.

2. Donor Login Module

- Donors can log in or register with their details to access the home page.
- They can create posts describing the commodities they wish to donate, along with accompanying photographs.
- Accepted posts enable chat functionality for seamless communication between donors and NGOs.

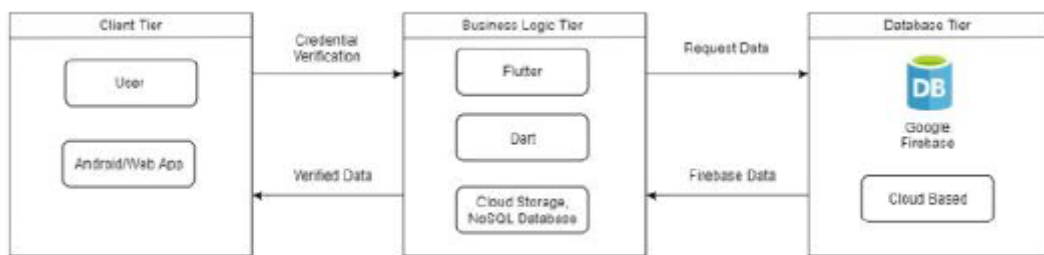


Figure 2.5: System Design (Satam et al., 2021)

The system design, shown in Figure 2.5, outlines the interaction between the client tier (user interface for donors and NGOs), the application tier (built using Flutter and Dart), and the Google Firebase cloud storage. This architecture ensures real-time data management and a smooth user experience.

While the application excels in software functionalities, such as user-friendly interfaces and communication features, it lacks IoT integration for real-time bin monitoring or tamper-proof security. Furthermore, the system does not include garment-specific features, such as clothing categorization or tracking donor contributions. These limitations make it less applicable to scenarios requiring physical donation bin management or specialized garment donation systems.

Nevertheless, the application's use of cross-platform technologies, live data updates, and streamlined workflows demonstrates its potential for enhancing donation centre operations and improving donor engagement. Integrating IoT features and advanced security mechanisms could further extend its functionality and scope.

2.4.4 IoT-Based Smart Donation Box

Setiawan and Hariono (2024) developed an advanced IoT-enabled smart donation box designed to enhance donation management through automation and interactivity. The system features multiple sensors and modules that work in tandem to provide real-time monitoring, tamper detection, and an engaging donor experience. This system integrates seamlessly with a mobile application and a web-based platform, allowing donors to check bin status, track their contributions, and securely access donation bins via QR code authentication.

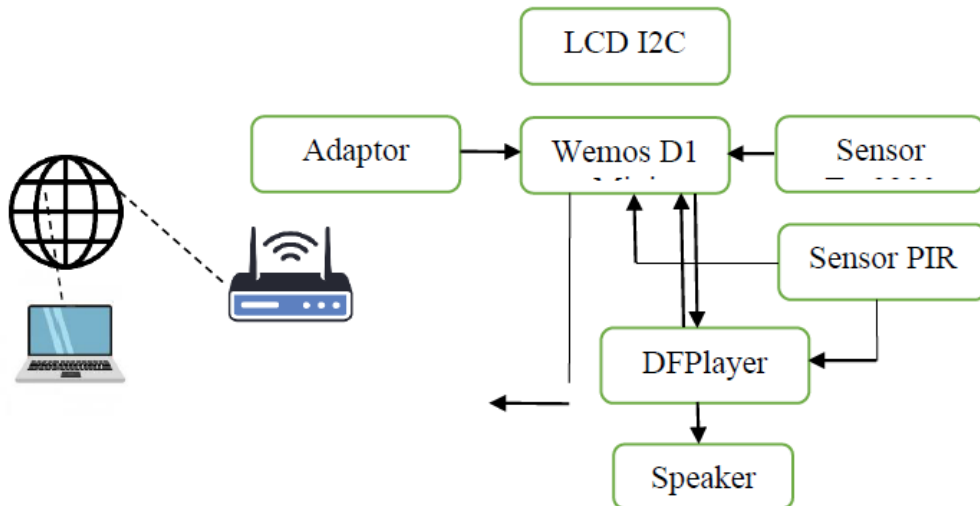


Figure 2.6: Block Diagram of Smart Donation Box (Setiawan & Hariono, 2024)

The block diagram of the system, shown in Figure 2.6, highlights the integration of several components, including sensors, a microcontroller, an LCD screen, and a website interface. Each component contributes to the system's functionality:

- PIR Sensor HC-SR501: Detects the presence of individuals near the donation box, activating the system and playing audio messages to engage donors.
- TCS3200 Colour Sensor: Identifies and records the colour of inserted items (e.g., money), enabling the system to log donation details accurately.
- DFPlayer Mini with Sound Speaker: Delivers audio messages, such as thank-you notes or updates about donation utilization, creating an interactive experience for donors.
- I2C LCD Screen: Displays relevant information, such as the total amount collected and updates about the bin's capacity, ensuring transparency for donors.
- Website Interface: Provides administrators with real-time monitoring, financial reports, and control over donation box settings, enhancing operational efficiency.

This comprehensive system combines IoT and mobile technologies to streamline donation management, offering both interactive elements for donors and robust monitoring capabilities for administrators. However, while the system is versatile and innovative, it does not specifically address garment donation needs, such as sorting or categorization. Additionally, it lacks advanced donor engagement features, such as tracking the impact of contributions or personalized notifications.

By incorporating garment-specific functionalities and enhancing donor interaction tools, the proposed IoT-Based Smart Garment Donation Station System can build upon the strengths of this smart donation box to address gaps in managing non-monetary donations effectively.

2.5 Comparison of Existing Systems

The existing systems reviewed offer various functionalities, from IoT-based monitoring to mobile app-based donor engagement and blockchain-secured donation tracking. However, these systems fall short in addressing the specific needs of garment donation management, which include categorization, advanced security features, and personalized donor engagement.

Table 1.1: Comparison of Existing Systems with proposed system

Feature	Smart Donation Machine	Android-Based Application	Donation Centers App	IoT-Based Smart Donation Box	Proposed System
IoT Integration	Yes (Arduino for monitoring)	No	No	Yes (IoT sensors for real-time monitoring)	Yes (Ultrasonic sensor, real-time monitoring)
Mobile App Features	Limited (Admin notifications)	Yes (Locators, updates)	Yes (Registration, notifications)	Yes (Tracking, QR-based access)	Yes (Real-time notifications, station locators)
Cloud Integration	No	Limited (Firebase for data sync)	Yes (Firebase for live updates)	Yes (Web interface for real-time monitoring)	Yes (Real-time updates and synchronization)
Bin Monitoring (Capacity Status)	Yes (Real-time, via ultrasonic sensor)	No	No	Yes (Displayed via LCD screen)	Yes (Real-time status updates via IoT sensors)
Security Mechanisms	Basic (Tamper-proof bin design)	No	No	QR code-based access	Yes (secure access)
User Engagement	Low	High (Notifications, campaign updates)	Medium (Chat functionality)	Medium (Audio messages, donor transparency)	High (Event updates, impact tracking)
Garment-Specific Features	No	No	No	No	Yes (Categorization, tailored functionalities)

Donation Categories	General	Food, clothes, books	General	General	Garment-specific donation system
Focus Area	General donation automation	Food, clothes, and books donations	Cross-platform donor-NGO interaction	Monetary donations	Garment-specific donation management

As presented in Table 1.1, the comparison underscores the limitations of existing systems in addressing garment-specific donation needs. While systems like the Smart Donation Machine, Application for Donation Centers, Android-Based Application for Food, Clothes, and Books Donation, Smart Donation Machine Using Arduino and IoT-Based Smart Donation Box provide many key features, none fully integrate the features required for efficient and secure garment donation management. The proposed system fills this gap by combining IoT-enabled monitoring, comprehensive mobile app functionalities, and tailored solutions for garment donations.

2.6 Current Gaps in Research

While the reviewed systems highlight the advantages of IoT integration, mobile application features, and security mechanisms in donation management, significant gaps remain unaddressed. These limitations present opportunities for the proposed IoT-Based Smart Garment Donation Station System to offer a more comprehensive and innovative solution tailored to garment donations.

2.6.1 Garment-Specific Features

None of the existing systems reviewed are specifically designed to cater to the unique requirements of garment donations. For example, general donation systems focus on items like food or monetary contributions, overlooking essential functionalities like clothing categorization, size tracking, or weight-based sorting. The proposed system addresses this by incorporating garment-specific features that streamline donation processing, enabling more efficient sorting and allocation of donated clothes. This focus not only improves operational efficiency but also enhances the experience for both donors and administrators, creating a system specifically optimized for the complexities of garment donations.

2.6.2 Advanced Security

Although some systems incorporate basic security features such as QR code-based authentication or tamper-proof mechanisms (Setiawan & Hariono, 2024; Krishna et al., 2017), they lack advanced tools for addressing sophisticated security threats. The proposed system introduces multi-factor authentication, real-time tamper alerts, and logging of all bin interactions through secure digital records. These advanced features aim to deter theft and unauthorized access, providing a level of security unmatched by current solutions. This comprehensive approach strengthens trust among donors, ensuring that their contributions are protected and used appropriately.

2.6.3 Enhanced User Engagement

Donor engagement in existing systems is often limited to basic features like notifications or station locators (Pandey & Kumar, 2022; Satam et al., 2021). Few systems offer interactive features that track a donor's contributions, provide real-time feedback on their impact, or personalize their user experience. The proposed system integrates these functionalities, allowing donors to view the status of their donations, receive updates on how

their contributions are utilized, and gain insights into the overall impact of their involvement. By fostering a sense of transparency and connection, the system aims to encourage repeat contributions and greater community participation.

2.6.4 Scalability

Many existing solutions, including IoT-based systems and blockchain-enabled platforms, are designed for small-scale implementations or specific geographic areas (Li et al., 2022). These systems face challenges in scaling across multiple locations due to infrastructure limitations, cost inefficiencies, or lack of interoperability. The proposed system addresses scalability by employing modular IoT devices, cloud-based data storage, and mobile app integration. These features ensure that the system can expand to serve multiple locations without compromising performance or functionality, making it adaptable to diverse operational environments.

The proposed IoT-Based Smart Garment Donation Station System offers innovative solutions that fill critical gaps in existing donation systems. By focusing on garment-specific needs, incorporating advanced security measures, enhancing user engagement, and enabling scalable deployment, this system is positioned to redefine how garment donations are managed, ultimately benefiting donors, administrators, and recipients alike.

2.7 Future Trends and Research Directions

Emerging technologies, such as artificial intelligence (AI) and blockchain, are likely to shape the future of donation systems. AI algorithms could enhance donor engagement by personalizing notifications and analysing donor behaviour to optimize campaign strategies. Machine learning models can also be used to predict donation patterns and manage bin logistics, ensuring efficient allocation of resources. For example, Zanella et al. (2014) emphasized the potential of AI in optimizing resource management in smart cities, an approach that can be adapted for garment donation systems.

Blockchain technology, as discussed by Li et al. (2022), offers unparalleled transparency and security in donation tracking. By recording every transaction in an immutable ledger, blockchain ensures accountability and trust between donors and organizations. While current systems leverage blockchain for general material donations, future research could explore its application in garment-specific systems to track and categorize clothing donations securely.

Another promising trend is the integration of Internet of Behaviour (IoB), which combines IoT data with behavioural insights. IoB could enable donation systems to better understand user preferences and improve donor retention strategies. Additionally, advancements in low-power wide-area networks (LPWAN) can facilitate scalable IoT deployments, reducing costs and expanding the reach of donation systems.

Future research should focus on combining these technologies to create comprehensive, user-centric solutions. Exploring interdisciplinary approaches that integrate IoT, AI, and blockchain into a unified platform could revolutionize how garment donation systems operate, ensuring sustainability, security, and scalability.

2.8 Summary

The review highlights the transformative potential of IoT, mobile applications, and advanced technologies in modernizing donation systems. Existing systems have demonstrated features such as real-time bin monitoring, automated notifications, and mobile app-based donor engagement. However, most solutions remain generic, with limited focus on garment-specific requirements or robust security mechanisms. These gaps include the lack of clothing categorization, advanced user engagement tools like impact tracking, and scalable implementations for multi-location operations.

Emerging trends like AI IoT integration present exciting opportunities to address these challenges. Future systems could leverage AI for personalized donor engagement, blockchain for enhanced transparency and security, and IoT for automated monitoring and tamper-proof mechanisms. Innovations such as low-power networks and behaviour-driven analytics could further enhance scalability and operational efficiency.

The proposed IoT-Based Smart Garment Donation Station System seeks to address these limitations by combining IoT sensors, advanced security features, and mobile applications tailored for garment donations. It aims to fill critical gaps in existing systems, providing an innovative, scalable, and user-friendly solution for modern donation management. By integrating these advancements, the system holds the potential to set a new standard for donation platforms, ensuring efficiency, security, and donor satisfaction.

CHAPTER 3

Methodology

3.1 Introduction

This chapter outlines the methodology used to design and develop the IoT-Based Smart Garment Donation Station System. The system integrates IoT-enabled hardware, a mobile application, and backend support for real-time monitoring and secure management of garment donations. The methodology follows a systematic approach that includes requirement planning, system design, hardware and software development, and testing phases to ensure the system meets its objectives.

3.2 System Requirements

To ensure the development of an efficient and user-friendly IoT-Based Smart Garment Donation Station System, a detailed analysis of system requirements was conducted. The requirements were identified through stakeholder input, focusing on both functional and non-functional needs to address the operational challenges of garment donation systems.

The functional requirements include core features that the system must provide. IoT sensors will be deployed to monitor the capacity of donation bins, offering real-time status updates such as "full," "half-full," or "empty." This feature ensures that bins are emptied promptly, preventing overflow and maintaining operational efficiency. A mobile application will be developed to send real-time notifications about bin status, donation events, and alerts. The app will also include an intuitive interface to facilitate seamless interaction for both donors and administrators, enabling features like event updates, and real-time monitoring. Additionally, the app will allow users to locate nearby donation stations and bins using integrated mapping features. It will also provide details on the number and roles of staff available at each station, helping to optimize operational management and ensure timely assistance when needed.

The non-functional requirements focus on system performance and scalability. To support the expansion of the donation network, the system must be scalable, capable of handling multiple donation stations across different locations without compromising performance. High system availability will be crucial to provide uninterrupted real-time data synchronization between the IoT devices, the mobile app, and the cloud infrastructure. Additionally, secure data transmission protocols will be implemented to protect sensitive

information shared between IoT devices, the mobile application, and the backend system, ensuring the integrity and confidentiality of user data.

This requirement analysis provides the foundation for designing a robust and scalable system tailored to the unique challenges of managing garment donations while addressing both operational and user-centric needs.

3.3 Tools and Technologies

The development of the IoT-Based Smart Garment Donation Station System requires a combination of hardware components and software platforms to ensure seamless integration, functionality, and scalability. Below is a detailed description of the tools and technologies utilized for the system's implementation.

3.3.1 Hardware Tools

The hardware plays a crucial role in enabling IoT capabilities and interaction with the physical environment.

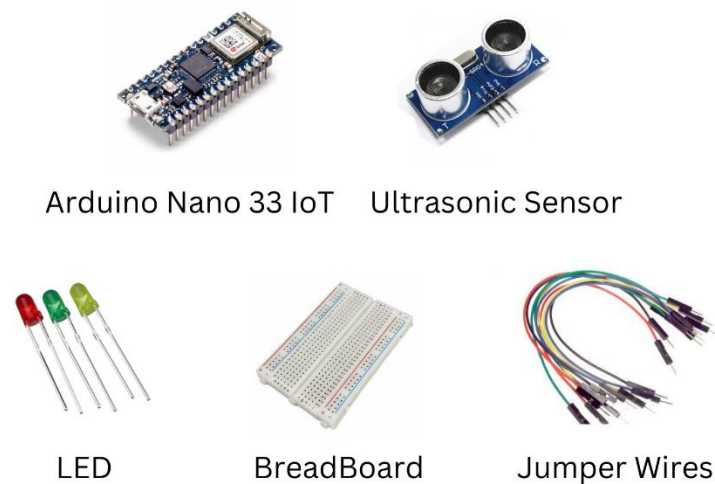


Figure 3.1: IoT Device for Hardware Tools

The hardware setup, as shown in Figure 3.1, integrates the Arduino Nano 33 IoT with an ultrasonic sensor and LED indicators to monitor and display bin statuses. Prototyping was performed using a breadboard and jumper wires to ensure reliable connections and functionality.

- **Arduino Nano 33 IoT:** This microcontroller acts as the central processing unit, managing data from sensors and handling communication with the cloud. Its built-in Wi-Fi capability simplifies cloud connectivity while its compact size ensures easy integration within the donation bin system.
- **Ultrasonic Sensor:** This sensor measures the bin's capacity by detecting the distance from the sensor to the topmost garment inside the bin. It provides real-time data about the bin's status: "empty," "half-full," or "full."
- **LED Indicators:**
 - **Red LED:** Lights up when the bin is "full."
 - **Yellow LED:** Indicates the bin is "half-full."
 - **Green LED:** Shows the bin is "empty."
- **Breadboard:** Used to assemble and test connections between the Arduino Nano, sensors, and LEDs. It enables a non-permanent setup for prototyping and testing the system.
- **Jumper Wires:** Used to connect components on the breadboard to the Arduino Nano. These ensure reliable communication and power transmission between all parts of the system.

3.3.2 Software Tools

The software development process involves designing a Flutter-based mobile application and backend systems to support hardware functionality and ensure seamless user interaction.

- **Flutter Framework:** Flutter, a cross-platform UI toolkit developed by Google, is used to build a mobile application compatible with both Android and iOS platforms. This ensures a responsive, visually consistent, and user-friendly interface for both donors and administrators.
- **Programming Language:** Dart, Flutter's native programming language, is employed to construct the application's interface, core features, and backend integrations, enabling smooth user interaction and efficient data processing.
- **Firebase Platform:** Firebase is utilized as the cloud backend for real-time database operations, user authentication, and data storage. It supports live updates, multi-device synchronization, and secure handling of donation-related data, such as bin status and event submissions.
- **PlatformIO Environment:** PlatformIO, an advanced development environment for embedded systems, is used for programming the Arduino Nano 33 IoT. It manages sensor inputs, controls LED indicators, and facilitates secure communication with Firebase, ensuring efficient integration between hardware and software components.

The combination of Flutter, Firebase, and PlatformIO provides a robust and scalable foundation for the system, enabling real-time monitoring, secure data handling, and an intuitive user experience. By leveraging these technologies, the application addresses key challenges in garment donation logistics while improving operational efficiency and user engagement.

3.4 System Design

System design serves as a critical phase in the development process, bridging the gap between system requirements and implementation. It involves creating a detailed blueprint that outlines the architecture, components, and interactions of the system. The primary objective is to ensure that the system is designed to meet functional and non-functional requirements, optimize performance, and maintain scalability and reliability.

In the context of the IoT-Based Smart Garment Donation Station, system design includes hardware and software architecture. This encompasses selecting appropriate IoT devices, database design for cloud integration, user interface layouts, and defining interactions between components like mobile apps and IoT devices. Effective system design ensures the system operates cohesively, supports future enhancements, and provides a seamless user experience.

3.4.1 Use Case Diagram

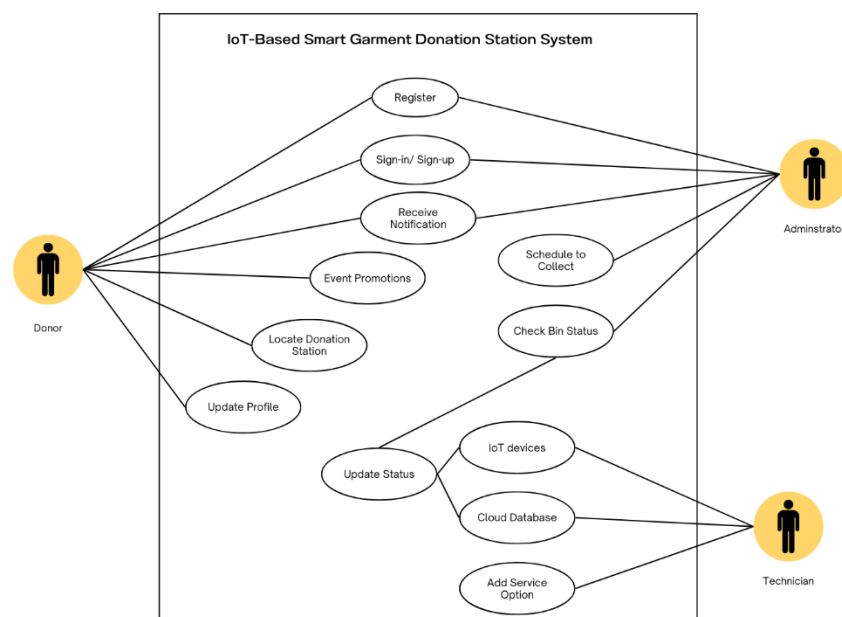


Figure 3.2: Use Case Diagram

The use case diagram (Figure 3.2) highlights the interactions between the main actors—donors, administrators, technician and the system. Donors can register log in or log out, locate donation stations, receive notification, view the event promotion and update profile. Administrators can register log in or log out, check bin status and schedule for collection. Technician can add service option into the system, set up the IoT devices and Cloud database

for update the bin status then send to the mobile application to let administrators to check bin status.

3.4.2 Flow Chart of the Proposed System

Introduce the flow chart by explaining that it outlines the step-by-step process of the proposed donation station system, from initializing hardware to updating bin status and interacting with users.

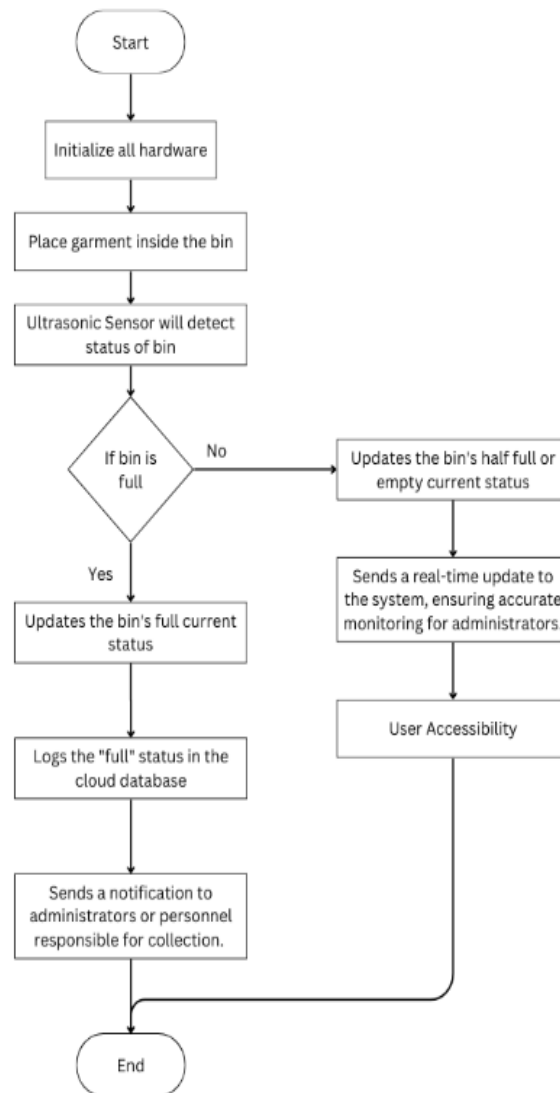


Figure 3.3: Flow Chart of the Proposed IoT-Based Smart Donation Bin

As depicted in Figure 3.3, the IoT-Based Smart Donation Bin system begins with hardware initialization, followed by real-time status detection using an ultrasonic sensor. The system differentiates between empty, half-full, and full statuses, with corresponding actions such as logging the status in the cloud database and notifying administrators for timely intervention. These steps ensure efficient monitoring and operation of the donation bin.

Step 1 Start: The system begins when powered on.

Step 2 Initialize All Hardware: Arduino Nano initializes connected components: ultrasonic sensor

Step 3 Place Garment Inside the Bin: The donor places the garment into the bin.

Step 4 Ultrasonic Sensor Detects Bin Status: The ultrasonic sensor measures the distance to determine the bin's fill level ("empty," "half-full," or "full").

Step 5 Is the Bin Full:

Yes:

- Updates the bin's full current status
- Logs the "full" status in the cloud database
- Sends a notification to administrators or personnel responsible for collection.

No:

- Updates the bin's half full current status
- Updates the bin's empty current status
- Sends a real-time update to the system, ensuring accurate monitoring for administrators.
- User Accessibility - users can continue depositing items as the bin is still operational.

Step 6 End: Wait for the next interaction.

3.4.3 Hardware System Architecture Design

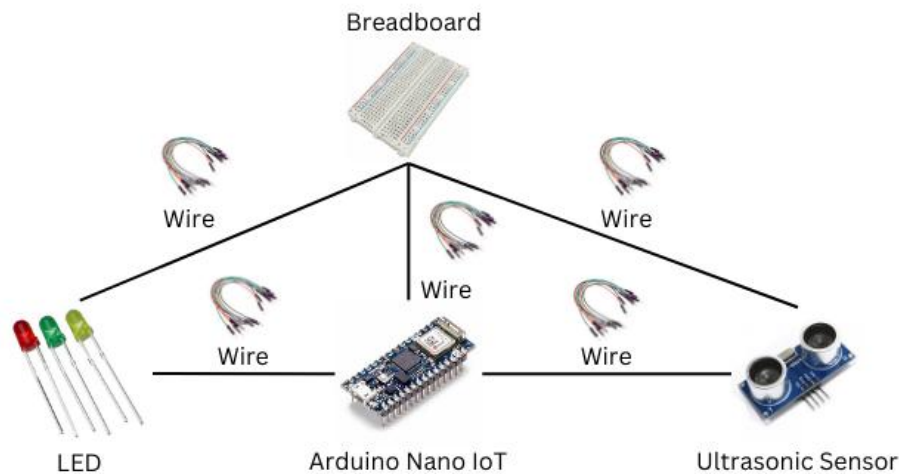


Figure 3.4: System Architecture Design showing the connection of all the hardware

As depicted in Figure 3.4, the system architecture integrates the Arduino Nano 33 IoT with ultrasonic sensors and LED indicators to monitor and display bin status. Connections are established using a breadboard for organized wiring, with the Arduino acting as the central hub for data processing and cloud communication. This streamlined design ensures efficient hardware operation and real-time monitoring capabilities.

Hardware Components and Connections

Arduino Nano 33 IoT: Acts as the central processing unit for the system.

Connections:

- Powers the ultrasonic sensor, LEDs, IR sensor, and servo motor via its 3.3V or 5V pins.
- Communicates with the cloud using its built-in Wi-Fi module.

Ultrasonic Sensor: Measures the bin's capacity by detecting the distance to the topmost garment in the bin.

Connections:

- Trigger Pin: Connected to a digital output pin on the Arduino Nano.
- Echo Pin: Connected to a digital input pin on the Arduino Nano.
- Power: Connected to the Arduino Nano's 5V pin.
- Ground: Connected to the Arduino Nano's GND pin.

LED Indicators (Red, Yellow, Green): Provides a visual indication of the bin's fill level.

- Green LED: Bin is empty.
- Yellow LED: Bin is half-full.
- Red LED: Bin is full.

Connections:

- Each LED is connected to a separate digital output pin on the Arduino Nano through a 220-ohm resistor.
- Power: Connected to the Arduino Nano's 5V pin.
- Ground: Connected to the Arduino Nano's GND pin.

Breadboard:

Used to connect and organize components such as the LEDs and sensors. Ensures proper wiring for testing and debugging before deployment.

Power Supply:

- The Arduino Nano 33 IoT can be powered via USB or a 5V DC adapter.
- Components requiring higher current, such as the servo motor, should use a dedicated external power supply.

System Workflow

- The ultrasonic sensor detects the current bin level and sends data to the Arduino Nano 33 IoT.
- The Arduino processes the sensor data and determines the bin's status (e.g., empty, half-full, or full)
- LED Indicators for red LED it will lights up when the bin is "full", yellow LED indicates the bin is "half-full" and green LED indicates the bin is "empty"
- The processed data can also be sent to the cloud for real-time monitoring through a mobile application.

3.4.4 Architecture Design

The architecture of the IoT-Based Smart Garment Donation Station System is designed to seamlessly integrate hardware, software, and cloud technologies to provide a robust, scalable, and user-friendly solution. The system is divided into three main layers, each serving a specific purpose and ensuring the efficient operation of the donation management process.

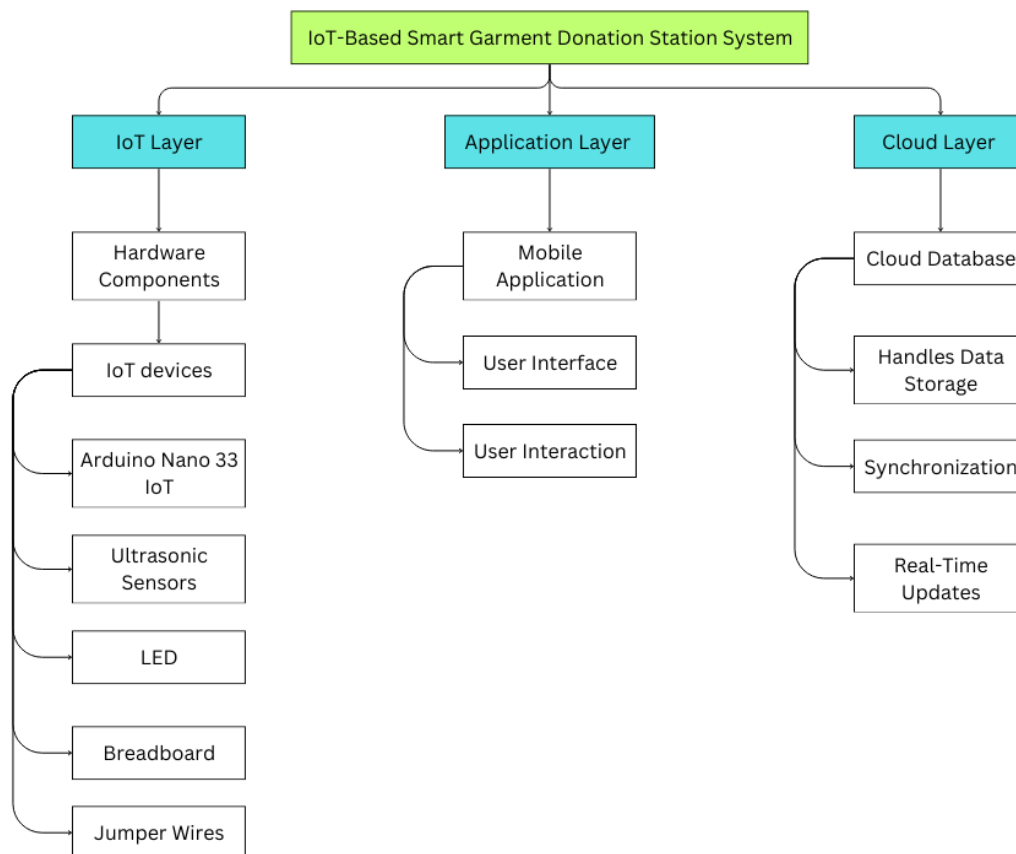


Figure 3.5: Architecture of the IoT-Based Smart Garment Donation Station System

As illustrated in Figure 3.5, the system's architecture integrates the IoT layer, application layer, and cloud layer to deliver a comprehensive solution. The IoT layer facilitates real-time monitoring and local feedback through sensors and LEDs, while the application layer provides user-facing tools for interaction. The cloud layer ensures reliable storage and secure data transmission, supporting scalability and efficient operation across multiple stations.

3.4.4.1 IoT Layer

The IoT layer forms the foundation of the system, facilitating interaction with the physical environment through integrated hardware components. At its core is the Arduino Nano 33 IoT, which acts as the central processing unit. This microcontroller collects and processes data from sensors and manages communication with the cloud using its built-in Wi-Fi module.

The ultrasonic sensor plays a crucial role in monitoring the bin's capacity. It measures the distance from the sensor to the topmost item in the bin, providing real-time data to determine whether the bin is "empty," "half-full," or "full." This information is processed by the Arduino Nano and transmitted to a cloud platform for storage and further analysis.

To enhance user interaction, LED indicators are integrated into the system. These LEDs provide immediate visual feedback about the bin's status like green LED for "empty," yellow LED for "half-full," and red LED for "full." This layer's functionality ensures that the system operates efficiently, offering real-time monitoring, local feedback, and remote data accessibility.

3.4.4.2 Application Layer

The application layer serves as the primary interface for user interaction with the system. It facilitates features tailored for both donors and administrators. Donors can utilize the mobile application to locate nearby donation bins, check their status (e.g., full or empty), and receive updates about donation events. Administrators, on the other hand, can monitor bin conditions in real-time, manage donation records, and receive alerts or notifications about system status. This layer focuses on providing an intuitive user experience with responsive design and seamless functionality, ensuring efficient communication between users and the underlying system components.

3.4.4.3 Cloud Layer

The cloud layer handles data storage, synchronization, and real-time updates across connected devices. It manages donation records, monitors bin statuses, and facilitates seamless communication between IoT hardware and the mobile application. This layer ensures scalability and reliability, allowing the system to efficiently support multiple donation stations without performance degradation. It also emphasizes data security, providing encryption and secure data access protocols to maintain the integrity and confidentiality of donation-related information.

The architecture enables seamless integration of IoT hardware, mobile applications, and cloud-based services, ensuring a scalable and adaptable platform capable of addressing operational challenges in garment donation systems. By combining these technologies, the system delivers real-time monitoring, efficient data management, and enhanced user experience for both donors and administrators.

3.4.5 Prototype (UI Design)

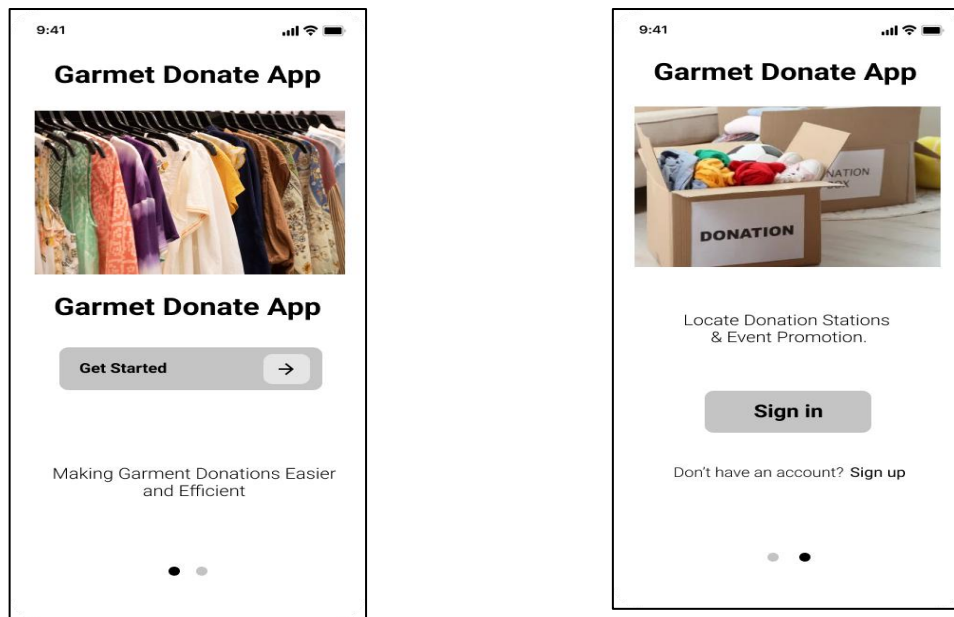


Figure 3.6: UI Design for Splash Screen 1 and 2

The Figure 3.6 for screen 1 introduces the app's purpose, emphasizing its role in simplifying the donation process and enhancing efficiency. It creates a welcoming impression, highlighting the app's commitment to user convenience. The figure 3.6 for screen 2 highlights the app's key features, such as locating donation stations and promoting events. It quickly communicates the app's value in connecting users with donation opportunities and keeping them informed.

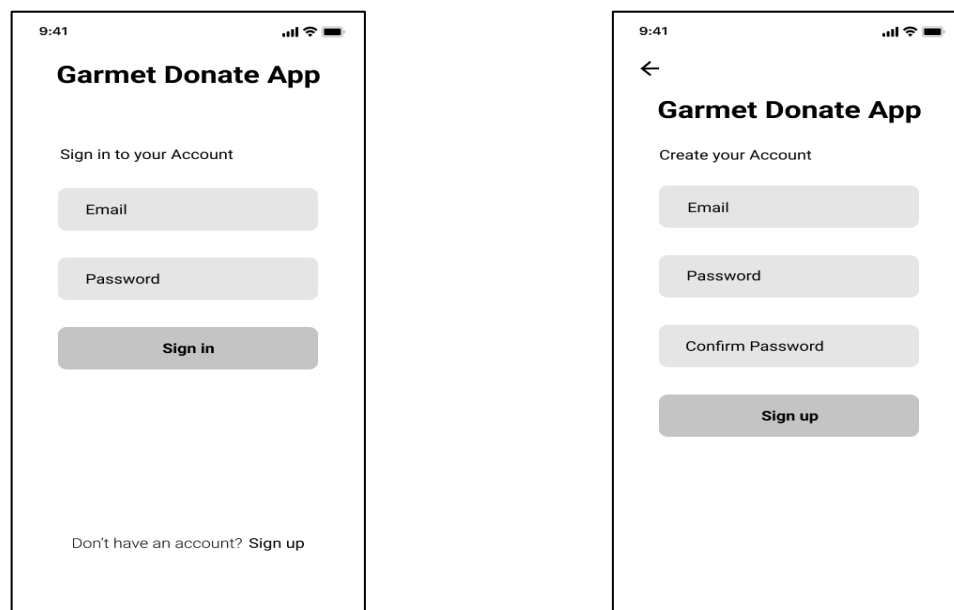


Figure 3.7: UI Design for Sign-in and Sign-up Page

The Figure 3.7 for sign-in page allows returning users access their accounts. It includes fields for entering credentials (e.g., email and password) and a "Sign In" button. If the user doesn't have an account, they can navigate to the sign-up page from the sign-in screen. The figure 3.7 for sign-up page collects essential details like name, email, and password to create a new account. It ensures new users can quickly register and access the app's features.

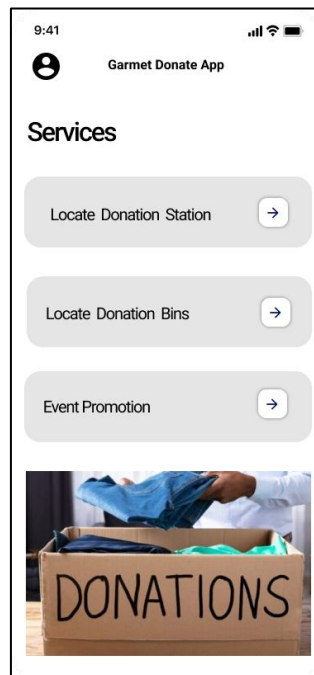


Figure 3.8: UI Design for Home Page

The Figure 3.8 for home page serves as the central navigation hub for the app, providing users with quick access to key features:

Profile User Icon (Top-Left Corner):

- Allows users to view and update their profile information, such as name, email, and phone number.

Main Features (Centre Buttons):

- Locate Donation Stations: Helps users find nearby donation stations on a map.
- Locate Donation Bins: Allows users to locate specific donation bins.
- Event Promotion: Provides details about ongoing or upcoming donation events, enabling users to participate effectively.

This intuitive design ensures that users can navigate seamlessly and access core functionalities with minimal effort.

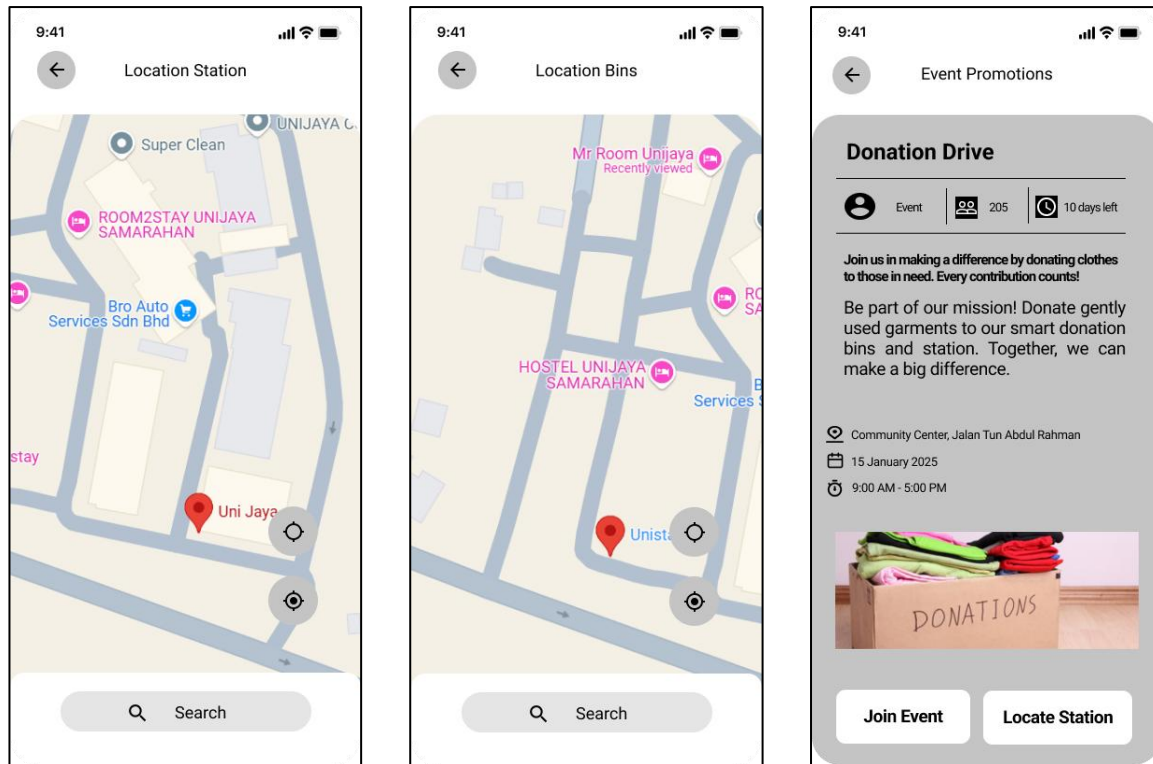


Figure 3.9: UI Design for Locate Station Page, Locate Bins Page and Event Promotion Page

The Figure 3.9 for locate station page features an interactive map that helps users locate nearby donation stations. Markers on the map indicate the exact locations of stations, with additional details available when users click on a marker, such as station names and operating hours. Similar to the Locate Station Page, the locate bins page displays an interactive map but focuses on the locations of specific donation bins.

The Figure 3.9 for Event Promotion Page highlights ongoing or upcoming donation events, such as "Donation Drive." It includes a brief description, event date and time, and location details. Users can also access event updates or notifications to stay informed.

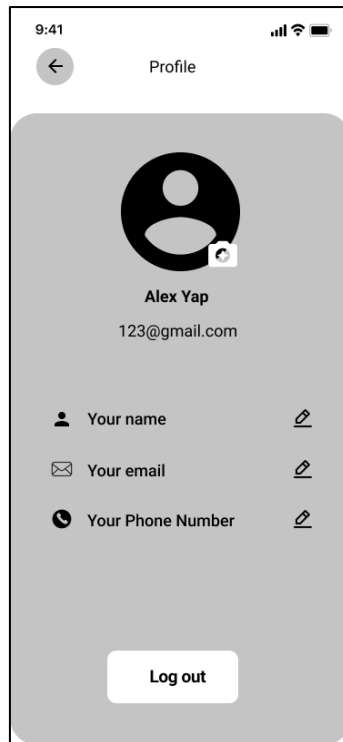


Figure 3.10: UI Design for Profile Page

The Figure 3.10 for profile page allows users to view and manage their account details.

- **Profile Information:** Displays user details such as name, email, and donation history.
- **Edit Profile Button:** Allows users to update their personal information, such as changing their name or email.
- **Logout Button:** Provides an option for users to log out of the application.

3.4.6 Database Design

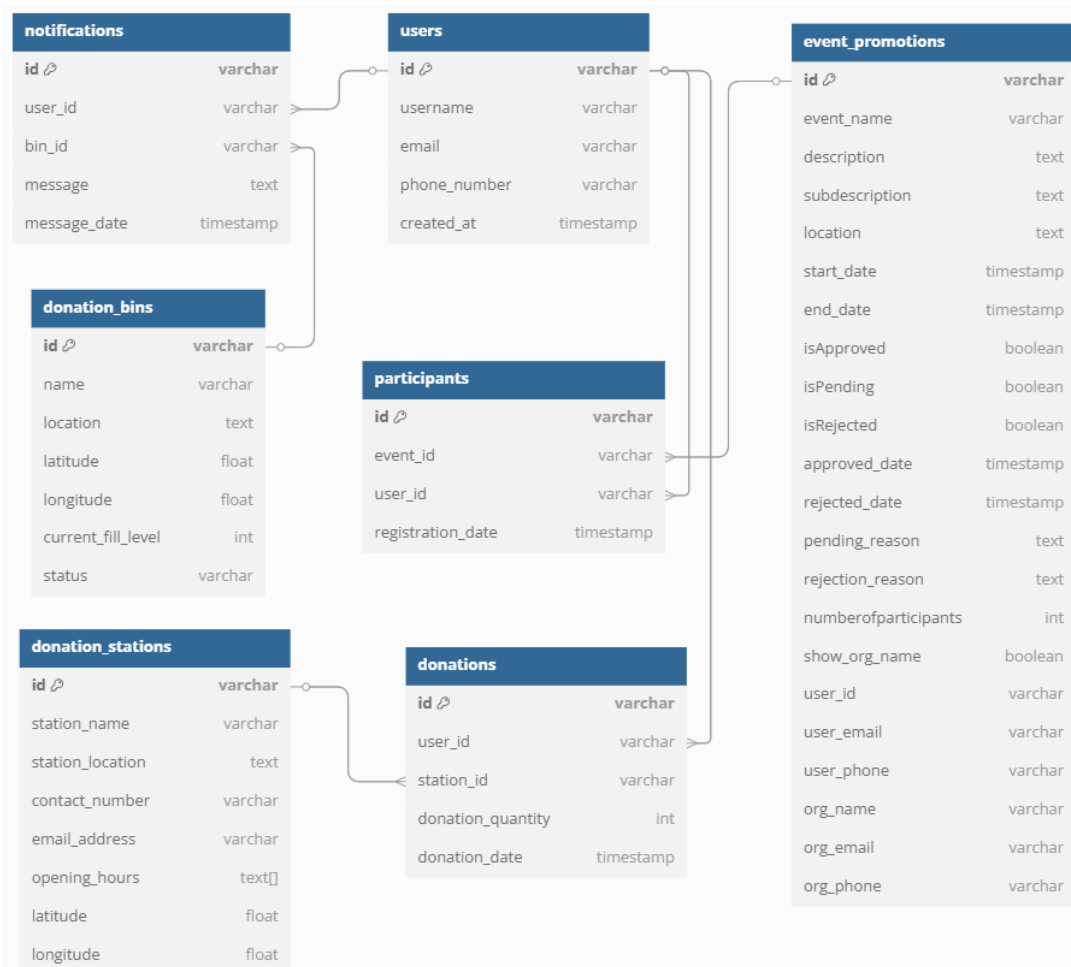


Figure 3.11: Database Design

As illustrated in Figure 3.11, the firestore database design for the Garment Donation System consists of multiple collections that reflect the functional components of the application. Each collection stores documents with relevant fields, and relationships between data are maintained through document ID references rather than traditional foreign key constraints.

The `donation_bins` collection stores details of all donation bins, including fields such as `id` (document ID), `name`, `location`, `latitude`, `longitude`, `current_fill_level` (as a percentage), and an optional `status` field to indicate whether the bin is active, full, or offline. This collection supports real-time bin monitoring through integration with the Firebase Realtime Database.

The `users` collection contains user account information, with each document identified by the user's Firebase Authentication UID. Fields include `username`, `email`, `phone_number`, and `created_at` timestamp. This structure supports role-based access and personalized user interactions within the app.

The `donation_stations` collection represents fixed donation drop-off points. Each station document includes `station_name`, `station_location`, `contact_number`, `email_address`, an array of `opening_hours`, and geographic coordinates (latitude, longitude). These details are used for geolocation, navigation, and UI presentation.

The `event_promotions` collection manages all event-related data. Each document stores `event_name`, `description`, `subdescription`, `location`, `start_date`, and `end_date`, along with status fields such as `isApproved`, `isPending`, and `isRejected`. It also includes `approved_date`, `rejected_date`, `pending_reason`, and `rejection_reason` for admin moderation. Organizer and submitter details are captured through fields like `org_name`, `org_email`, `org_phone`, `user_id`, `user_email`, and `user_phone`. The `numberofparticipants` field tracks engagement, while the `show_org_name` boolean determines public visibility of the organizer.

The `participants` collection logs event participation. Each document contains a `user_id`, `event_id`, and `registration_date`, enabling tracking of which users joined which events. This collection can be used for analytics and historical participation records.

The `donations` collection records individual donation activities, including `user_id`, `station_id`, `donation_quantity`, and `donation_date`. This supports data-driven reporting and tracking of community contributions.

The `notifications` collection is used to deliver system alerts to users. Each document contains the `user_id`, the related `bin_id`, a message, and the `message_date`. Notifications are triggered by events such as bin overflow, event status changes, or new updates.

Although Firestore does not enforce traditional relational constraints, logical relationships are maintained through document references. For example, `donations.user_id` links to `users.id`, and `donations.station_id` links to `donation_stations.id`. Similarly, `notifications.user_id` and `bin_id` reference their respective documents in the `users` and `donation_bins` collections. The `participants` collection references both `users` and `event_promotions`, enabling a many-to-one relationship for both user-event mappings.

This document-based design ensures scalability, real-time data synchronization, and flexible schema evolution, making it highly suitable for the needs of a mobile-first, cloud-integrated donation platform.

3.5 Development Phase

The development phase for the IoT-Based Smart Garment Donation Station System involves two key aspects: hardware development and software development. These processes work in tandem to ensure seamless integration between the physical and digital components of the system.

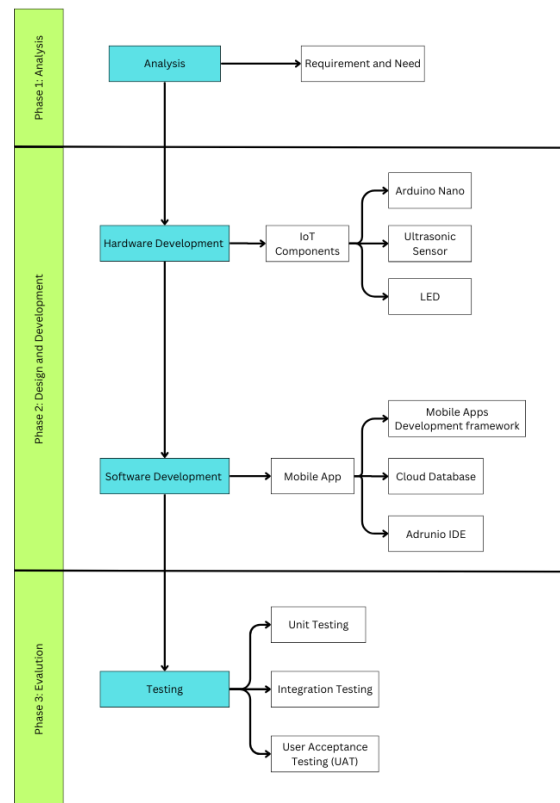


Figure 3.12: Phased Development Workflow for an IoT-Based Smart Garment Donation Station

As illustrated in Figure 3.12, the development workflow for the IoT-Based Smart Garment Donation Station System is structured into phases encompassing analysis, hardware development, software development, testing, and deployment. Each phase ensures that the system meets its functional and non-functional requirements while providing a user-friendly and efficient donation management solution.

3.5.1 Hardware Development

The hardware development process focuses on assembling and configuring IoT devices to enable real-time monitoring and interaction. The Arduino Nano is programmed to act as the central processing unit, handling data from sensors and controlling other connected components. Ultrasonic sensors are installed to monitor bin capacity by measuring the distance from the sensor to the collected items, identifying statuses such as "empty," "half-full," or

"full." LED Indicators for red LED it will lights up when the bin is "full", yellow LED indicates the bin is "half-full" and green LED indicates the bin is "empty". The processed data can also be sent to the cloud for real-time monitoring through a mobile application.

3.5.2 Software Development

The software development process focuses on creating a mobile application and backend system to efficiently manage and monitor donation activities. The application is designed to ensure compatibility across multiple platforms, providing users with a seamless experience on various devices. It features an intuitive user interface and core functionalities, including locating nearby donation stations, tracking the status of donation bins in real-time, and receiving notifications about important updates or events. The backend system supports data processing, secure communication, and integration with IoT devices to ensure smooth operation and reliable performance.

To facilitate real-time updates and data synchronization, cloud database is a key important part. Cloud database manages donation records, bin statuses, and system notifications, ensuring seamless communication between the hardware and the mobile application. These features are critical in ensuring the system's reliability and user trust.

By combining these hardware and software development efforts, the system achieves its goal of providing an efficient, scalable, and user-friendly platform for managing garment donations. This phase ensures that both physical components and digital interfaces work harmoniously to deliver a seamless experience for donors and administrators alike.

3.5.3 Testing

The testing phase ensures the functionality, integration, and usability of the IoT-Based Smart Garment Donation Station System. A structured approach is followed to verify the system's hardware components, software features, and overall user experience. This phase is critical for identifying and addressing any issues, ensuring the system performs reliably and meets the intended requirements.

3.5.3.1 Unit Testing

Unit testing focuses on validating individual components of the system to ensure they function as expected. Each hardware component, such as the ultrasonic sensors display is tested independently to verify its accuracy and responsiveness. Similarly, the software elements, including the mobile app's features like bin location tracking, notifications, and user authentication, are evaluated to ensure proper functionality. This step is crucial for identifying and resolving issues at the component level before integration.

3.5.3.2 Integration Testing

Integration testing examines the seamless communication and interaction between the system's hardware and software layers. The Arduino Nano is tested to ensure it processes data from the ultrasonic sensors accurately. The integration between the mobile application and the cloud database is also validated to confirm real-time synchronization of bin statuses and notifications. This phase ensures that the system components work harmoniously, providing a unified and efficient platform for users.

3.5.3.3 Real-Device Testing

Real-device testing was conducted using a physical Android smartphone to evaluate the application in an environment that closely mirrors actual user conditions. The app was installed and run on an OPPO Reno5 F device to test performance, interface responsiveness, and compatibility with hardware features such as GPS and Firebase integration. Functionalities such as station locating, event browsing, form submissions, and live bin status updates were verified under real network conditions. This testing helped ensure the system performs reliably on actual devices, identifying potential issues related to layout rendering, input handling, or API communication.

By employing unit testing, integration testing, and Real-device testing, the testing phase ensures that the system is robust, reliable, and user-friendly. This comprehensive approach

identifies potential issues and refines the system, laying the foundation for a successful deployment.

3.6 Summary

This chapter presents the comprehensive methodology used in developing the IoT-Based Smart Garment Donation Station System, focusing on delivering a scalable, secure, and user-centric solution. It begins with a detailed requirement analysis, outlining both functional needs—such as real-time bin monitoring, secure authentication, and mobile app features—and non-functional requirements like scalability and secure data transmission. The selected tools and technologies highlight the seamless integration of hardware components (e.g., Arduino Nano 33 IoT, ultrasonic sensors, and LED indicators) with software platforms like Flutter, Firebase, and PlatformIO. These tools enable efficient communication between hardware and cloud services, real-time data updates, and intuitive mobile interfaces, supporting smooth interactions for both users and administrators.

The system architecture is structured into three layers: IoT, application, and cloud. This layered approach ensures reliable data flow from the sensors to the cloud and onward to the mobile app, balancing real-time performance with long-term scalability. The UI prototype emphasizes usability through features like donation bin locators, event promotion pages, and live status tracking. Complementing this, the database design defines clear relationships between entities such as users, donation bins, and events to support smooth system operations. The development phase includes hardware assembly, software integration, and cloud setup, while the testing phase—conducted through unit, integration, and real-device testing—validates system functionality and performance. This structured methodology ensures the solution effectively addresses existing gaps in donation management, setting a foundation for real-world deployment and future user-driven refinements.

CHAPTER 4

Implementation

4.1 Introduction

This chapter discusses the complete implementation of the smart bin level monitoring system developed for this project. The implementation combines both hardware and software components to create an IoT-enabled solution capable of detecting bin fill levels and transmitting real-time data to a centralized database. The system leverages an ultrasonic distance sensor to monitor waste accumulation in bins, visual indicators for physical feedback, and wireless communication for data logging. The data collected by the hardware is sent to a Firebase Realtime Database, which is then accessed and displayed through a mobile application built using Flutter. All code, both embedded and application-level, is developed in Visual Studio Code using PlatformIO and the Flutter development environment.

4.2 Hardware Components

4.2.1 Arduino Nano 33 IoT board

The core hardware of the system is centered around the Arduino Nano 33 IoT board. This microcontroller was chosen for its compact form factor and built-in WiFi capabilities, which eliminate the need for external communication modules. The Arduino Nano 33 IoT serves as the central control unit, managing sensor inputs, output indications, and communication with the Firebase Realtime Database over a secure HTTPS connection.

4.2.2 HC-SR04 ultrasonic sensor

To detect the level of waste in a bin, two HC-SR04 ultrasonic sensor is utilized. This sensor measures the distance between the sensor and the surface of the waste inside the bin by emitting ultrasonic waves and calculating the time taken for the echo to return (Pratama & Harijono, 2024). These measurements are then converted into percentage values that represent how full the bin is. The sensor is mounted at the top of the bin, and its data is sampled periodically by the Arduino.

4.2.3 LED Indicators

Visual feedback is provided using five LEDs: two green, two yellow, and two red. These LEDs indicate the fill percentage of the bin in stages. The green LEDs light up when the bin is less than 40% full, the yellow LEDs represent medium fill levels from 40% to 80%, and the red LED signals that the bin is almost full or completely full, indicating a percentage above 80%. This immediate feedback helps nearby users or staff assess the bin status at a glance without relying solely on the app.

4.2.4 Breadboard and Jumper Wires

The entire circuit is built on a breadboard using jumper wires, which provide a temporary and flexible wiring setup ideal for prototyping. This setup allows for quick modifications during testing and development before transitioning to a more permanent configuration if needed.

4.3 Hardware Components Connection

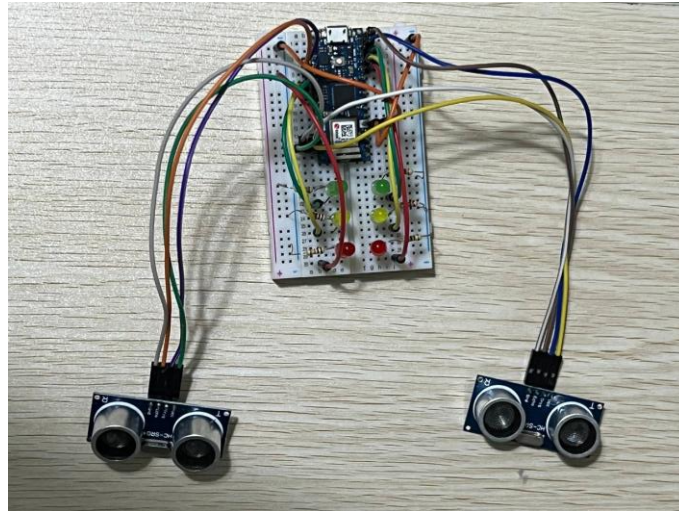


Figure 4.1: Connection of All of Hardware Components

4.3.1 First Ultrasonic Sensor (HC-SR04) and LED Indicators

This setup involves first HC-SR04 ultrasonic sensor and three LED indicators connected to an Arduino Nano 33 IoT. The purpose of this hardware configuration is to measure the fill level of a donation bin using ultrasonic distance sensing and indicate the bin's status using colored LEDs.

Pin on HC-SR04 and Led	Color Wire	Arduino Pin	Purpose
VCC	Yellow	VIN	Power
GND	White	GND	Ground
TRIG	Blue	D11	Trigger signal
ECHO	Brown	D12	Receive echo signal
Green Led	Green	D10	LOW fill
Yellow Led	Yellow	D9	Moderate fill
Red Led	Red	D8	High fill

Table 2.1: Connection of Arduino Nano 33 IoT with First Ultrasonic Sensor (HC-SR04) and Led

Ultrasonic Sensor (HC-SR04) Connections (Table 2.1):

- VCC (Yellow wire) → VIN: Supplies power to the ultrasonic sensor from the Arduino's VIN pin.
- GND (White wire) → GND: Connects the ground of the sensor to the Arduino's ground to complete the circuit.

- TRIG (Blue wire) → D11: The trigger pin is used to send out an ultrasonic pulse. It is connected to digital pin D11 of the Arduino.
- ECHO (Brown wire) → D12: The echo pin receives the reflected ultrasonic pulse and is connected to digital pin D12 to measure the time taken for the echo to return.

LED Indicators Connections, these LEDs provide a visual indication of the bin's fill level based on the distance measured by the HC-SR04 sensor.

- Green LED (Green wire) → D10: Indicates LOW fill level – the bin is mostly empty.
- Yellow LED (Yellow wire) → D9: Indicates MODERATE fill level – the bin is partially filled.
- Red LED (Red wire) → D8: Indicates HIGH fill level – the bin is nearly full and needs attention.

This configuration allows real-time monitoring of the donation bin's status using both sensor data and color-coded LEDs, enabling immediate visual feedback and helping with effective bin management.

4.3.2 Second Ultrasonic Sensor (HC-SR04) and LED Indicators

This configuration consists of a second HC-SR04 ultrasonic sensor and three additional LED indicators, connected to the Arduino Nano 33 IoT. It is designed to measure and indicate the fill level of another donation bin, using ultrasonic distance sensing and colored LEDs to visually represent the bin status.

Pin on HC-SR04 and Led	Color Wire	Arduino Pin	Purpose
VCC	Green	VIN	Power
GND	White	GND	Ground
TRIG	Purple	D14	Trigger signal
ECHO	Orange	D15	Receive echo signal
Green Led	Green	D16	LOW fill
Yellow Led	Yellow	D17	Moderate fill
Red Led	Red	D118	High fill

Table 2.2: Connection of Arduino Nano 33 IoT with Second Ultrasonic Sensor (HC-SR04) and Led

Ultrasonic Sensor (HC-SR04) Connections (Table 2.2):

- VCC (Green wire) → VIN: Supplies power to the sensor from the Arduino's VIN pin.
- GND (White wire) → GND: Provides the ground connection to complete the circuit.
- TRIG (Purple wire) → D14: Sends the ultrasonic pulse. Connected to digital pin D14 on the Arduino.
- ECHO (Orange wire) → D15: Receives the echo signal and calculates the distance. Connected to digital pin D15.

LED Indicators Connections:

- The LEDs are used to indicate the fill status of the bin based on distance measurements:
- Green LED (Green wire) → D16: Lights up when the bin is at low fill level, indicating it is mostly empty.
- Yellow LED (Yellow wire) → D17: Indicates a moderate fill level, meaning the bin is partially filled.
- Red LED (Red wire) → D18: Signals a high fill level, showing the bin is nearly full and may need to be emptied soon.

This second sensor and LED setup functions independently but identically to the first, allowing the system to monitor multiple donation bins with real-time visual feedback. This design supports scalable bin management, where each set of sensors and LEDs corresponds to a specific bin for efficient monitoring.

4.4 Software Components

4.4.1 Embedded System Integration

All embedded code for this project was developed using Visual Studio Code with the PlatformIO extension. PlatformIO provides a modern, flexible, and efficient development environment compared to the standard Arduino IDE. It offers several advantages, including advanced dependency management, integrated debugging tools, and highly customizable build configurations. Unlike the Arduino Cloud platform—which limits certain features behind a subscription—PlatformIO allows unrestricted access to essential APIs and tools, making it an ideal choice for cost-effective and scalable embedded development.

4.4.1.1 PlatformIO with Arduino Configuration and Firebase Realtime Database Integration

The Arduino Nano 33 IoT board acts as the main microcontroller for the donation bin system. All real-time interactions and hardware-level processes are handled by the embedded code written in PlatformIO. Key responsibilities of this embedded system include:

- **Ultrasonic Sensor Data Reading:** Continuously reads distance values from HC-SR04 ultrasonic sensors.
- **Fill Level Calculation:** Converts the measured distance into a percentage to indicate how full each donation bin is.
- **LED Indicators Control:** Activates green, yellow, or red LEDs based on the calculated fill level, providing a quick visual reference.
- **Firebase Communication:** Utilizes the WiFiNINA and ArduinoHttpClient libraries to securely send data (distance and fill percentage) to the Firebase Realtime Database via HTTPS POST requests.

This embedded code runs continuously on the hardware, ensuring accurate and up-to-date monitoring of bin statuses, which are then reflected in the mobile application for both users and administrators. The seamless integration between PlatformIO, Arduino hardware, and Firebase services is central to the system's reliability and responsiveness.

4.4.2 Flutter

Flutter is used to develop the mobile application. It is a modern, cross-platform framework by Google that allows for the creation of responsive and visually appealing apps (Bhagat, 2022). One of the key reasons for choosing Flutter is its seamless integration with Firebase, which significantly simplifies backend connectivity (Payne, 2019). With Flutter, developers can directly access Firebase services using official packages, making it easy to implement real-time updates, user authentication, and cloud data management. In this project, Flutter is used to build an app that enables users to monitor donation bin levels and view related information in real time through a clean and intuitive interface.

4.4.2.1 UI of Flutter

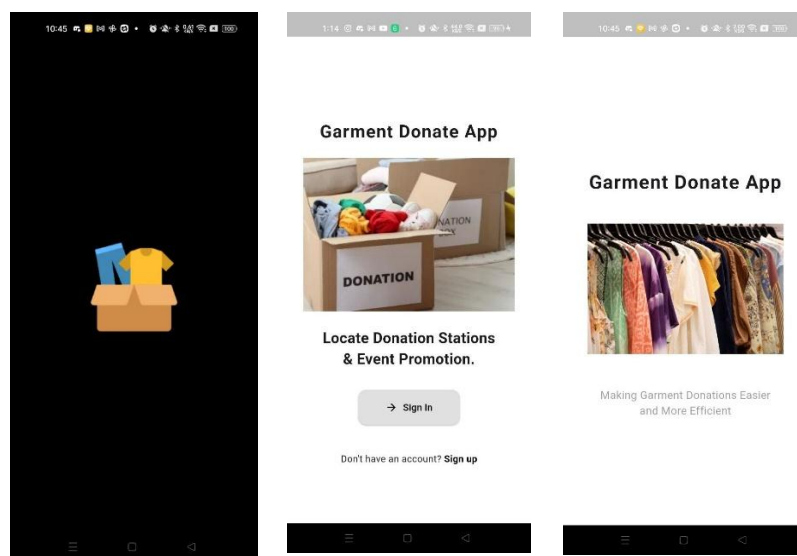


Figure 4.2: UI for While User Loading the Apps

Figure 4.2 showcases the user interface flow during the initial loading and first launch of the mobile application. It includes three key stages illustrated through three images:

First Image – App Loading Screen: When the user opens the app, a loading screen appears with a black background and the app icon centered on the screen. This minimal splash screen indicates that the application is initializing and loading necessary resources in the background.

Second Image – Introduction and Sign In Screen: Upon launching the app for the first time, the user is presented with an introductory screen that outlines the app's core features and purpose. This screen includes a prominent Sign In button that navigates users to the sign-in page. Below that, a text prompt "Don't have an account? Sign up" is displayed, allowing users to access the Sign-Up page for new account registration.

Third Image – Splash Screen After Launch: Once the user has interacted with the app previously, subsequent launches display a branded splash screen. This screen briefly appears while the app prepares the home interface, offering a visually consistent transition into the main experience.

This multi-stage flow ensures a user-friendly onboarding process, guiding both new and returning users seamlessly through login and access to core features.

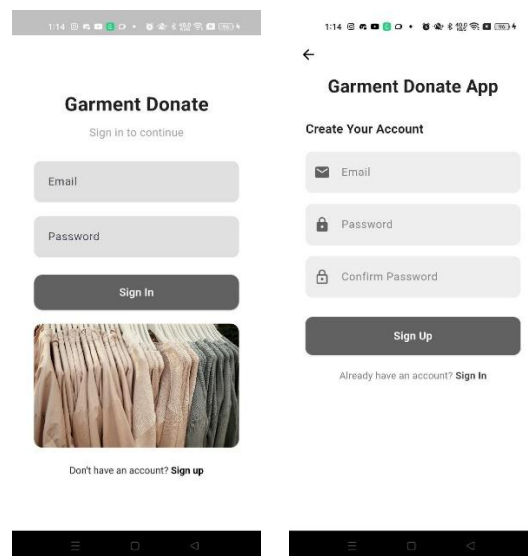


Figure 4.3: UI for User Sign In and Sign UP

This interface (Figure 4.3) allows users to either log into an existing account or create a new one. The Sign In page provides fields for email and password, enabling registered users to securely access their profiles and personalized content. The Sign-Up page offers a registration form that collects essential user details such as email, and password to create a new account.

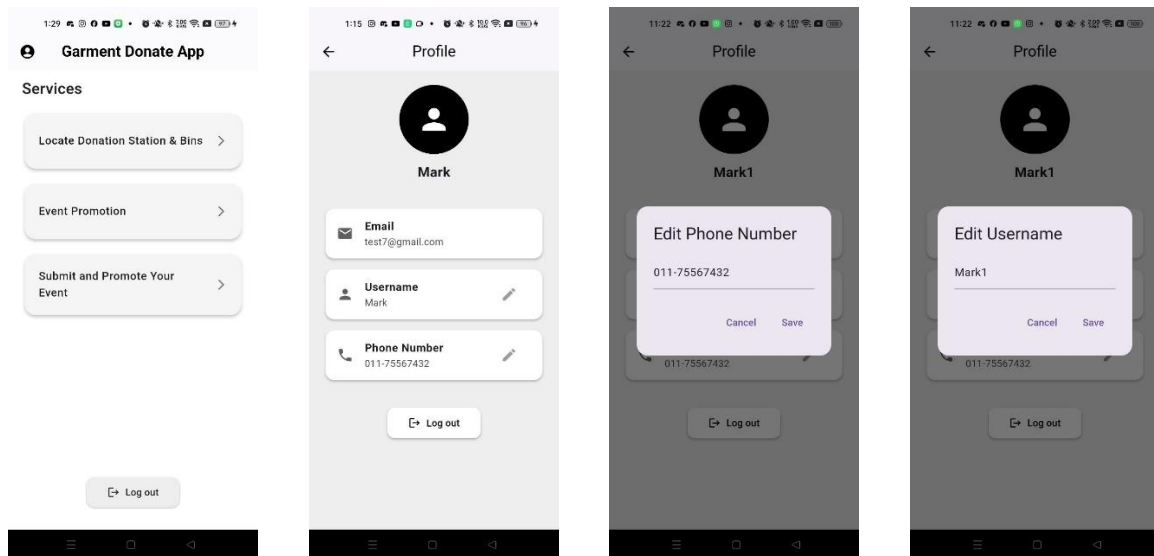


Figure 4.4: UI for User Dashboard and Profile

Figure 4.4 represent User Dashboard serves as the main navigation hub once the user is successfully signed in. It provides quick access to four core services:

- **Locate Donation Station and bins:** Allows users to view nearby donation stations and bins based on their current location, for the donation bins can Displays real-time data on donation bin availability and fill levels.
- **Event Promotion:** Showcases upcoming donation-related events and promotions to encourage participation.
- **Submit and Promote Your Event:** Enables users or organizations to collaborate with the platform by submitting details of corporate social responsibility (CSR) events. These submissions are reviewed by the admin before being published, allowing companies to promote their donation drives and align with community impact goals through the app.
- A Logout button is also available on this page to securely end the session.

The Profile Page allows users to view and manage their personal information. Users can edit their username and phone number through a clean and simple interface. The Logout button is also provided here for user convenience and session security.

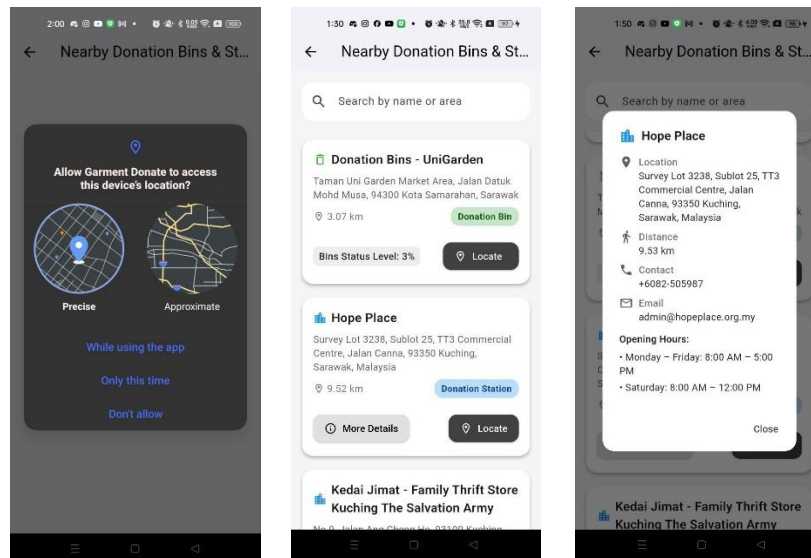


Figure 4.5: UI for Locate Donation Stations and Bins

Figure 4.5 illustrates when users open the unified “Locate Donation Stations & Bins” page, the app first request’s location permission. Granting this permission allows the system to calculate the user’s exact position and sort every station or bin by real-world proximity. Without it the screen cannot display meaningful results, so the permission prompt is essential to the feature’s operation.

Once permission is approved, two kinds of entries appear in a single, continuous list:

Entry type	Key information shown	Visual cues
Donation Station	• Station name • Address • Straight-line distance in km	A building-style icon (blue) distinguishes stations at a glance.
Donation Bin	• Bin name • Address • Straight-line distance in km • Bin-level status – the current fill level expressed as a percentage	A green bin icon marks ordinary bins. If the fill level exceeds 80 %, the entry is flagged with a red warning icon and the percentage badge turns red, signalling that the bin is nearly full and should be collected soon.

Table 4.3: Visual and Informational Differences Between Donation Stations and Donation

Bins

For every entry the user is offered two clear actions:

- More Details – opens a pop-up card with the full profile of the chosen station, for stations this includes location, distance, contact number, email address, and opening hours.
- For bins it shows the same location and distance data plus the live fill-level percentage so managers can plan collections efficiently.

The interface lets users find the nearest, most suitable drop-off location—and see at a glance which bins need attention.

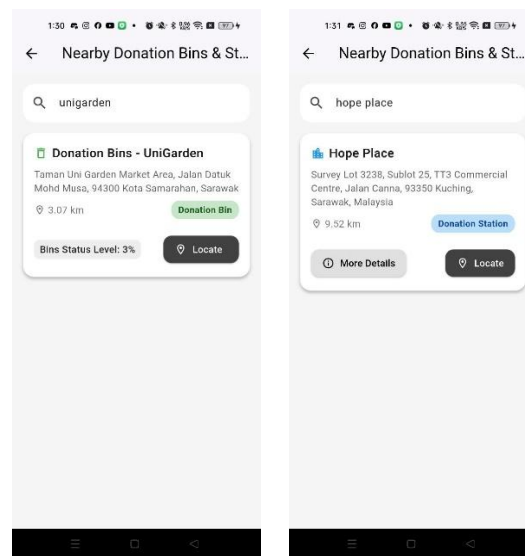


Figure 4.6: UI for Search Feature of Locate Donation Station and Bins

Figure 4.6 illustrates the search functionality within the combined Locate Donation Stations and Bins interface. This feature enables users to quickly find specific donation points—whether bins or stations—by typing relevant keywords such as names or area locations.

The search bar is positioned at the top of the page, allowing users to initiate queries without scrolling. As the user types, the app filters the combined list in real-time, dynamically displaying results that match the entered text. The results may include either donation bins or stations depending on the search term and are clearly distinguished by icons and labels.

The first image in the figure shows a sample search result for donation bins, displaying relevant entries that include the bin name, location, distance from the user, and bin fill level status.

The second image presents a search result for donation stations, where each entry displays the station name, address, and distance from the user.

This unified search feature improves usability by allowing users to locate specific donation points instantly, regardless of their type or distance, ensuring a smooth and efficient navigation experience.

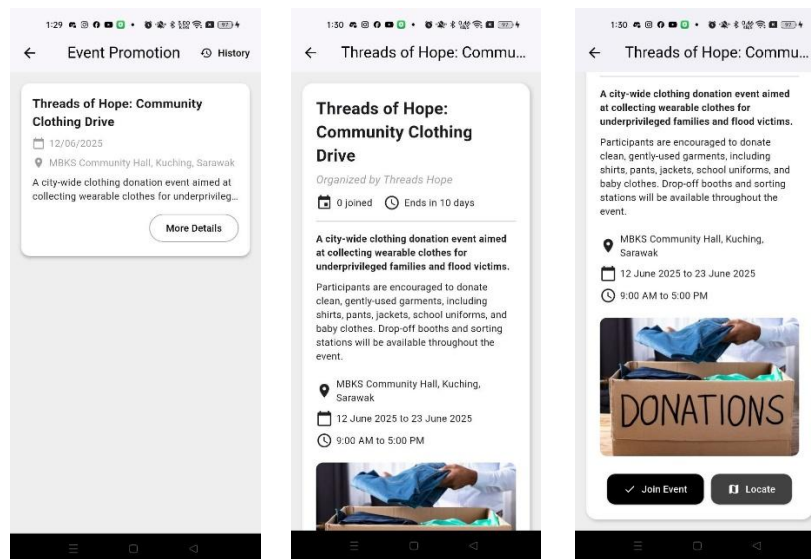


Figure 4.7: UI for Event Promotion

Figure 4.7 showcase Event Promotion page displays a scrollable list of upcoming donation-related events and campaigns. Each event card presents key event information, including:

- Event Name
- Start Date
- Location
- Brief Description

At the top right corner of the page, there is a History button with a history icon. Tapping this button allows users to view past events that have already ended, separating current promotions from historical ones.

Each event card includes a "More Details" button. When tapped, it navigates users to a dedicated Event Detail page that provides comprehensive event information such as:

- Event Name
- Organization Name (Displayed if the organizer chose to reveal it)
- Location
- Full Description and Sub-description
- Start Date and End Date

The Event Detail page also features two action buttons:

- Join Event – Lets users express interest or confirm participation.
- Locate – Opens Google Maps with the exact location of the event for navigation.

This feature improves user engagement by making it easy to explore and participate in current events, while offering a clear and organized path to access past event history when needed.

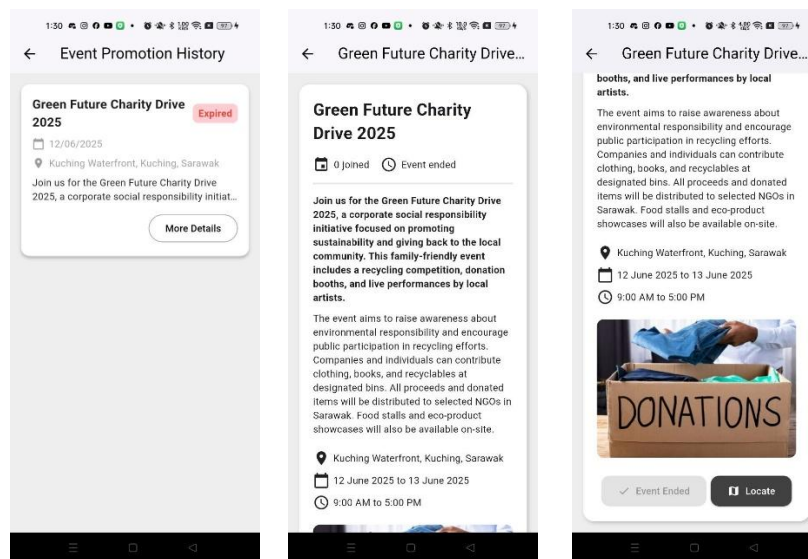


Figure 4.8: UI for Event Promotion History

Figure 4.8 represent Event Promotion History page allows users to view past donation-related events that have already ended. This feature ensures that users can still refer to or reflect on expired events even after their conclusion.

The first image displays the main Event Promotion History interface. It presents a scrollable list of all expired events. Each event card includes the following information:

- Event Name
- Start Date
- Location
- Brief Description

Each card includes a “More Details” button, which allows users to explore the full event information.

The second and third images show the Event Details page for a selected expired event. This page displays comprehensive event information:

- Event Name
- Organization Name (if applicable)
- Location
- Full Description and Sub-description
- Start and End Dates

For expired events:

- The Join Event button is replaced with a non-clickable “Event Ended” indicator, clearly showing that the event has concluded and can no longer be joined.
- The Locate button remains active, enabling users to view the event’s location in Google Maps, which can be useful for future reference or directions.

This history feature provides users with an organized way to review previous donation campaigns, access their details, and retain awareness of past community efforts.

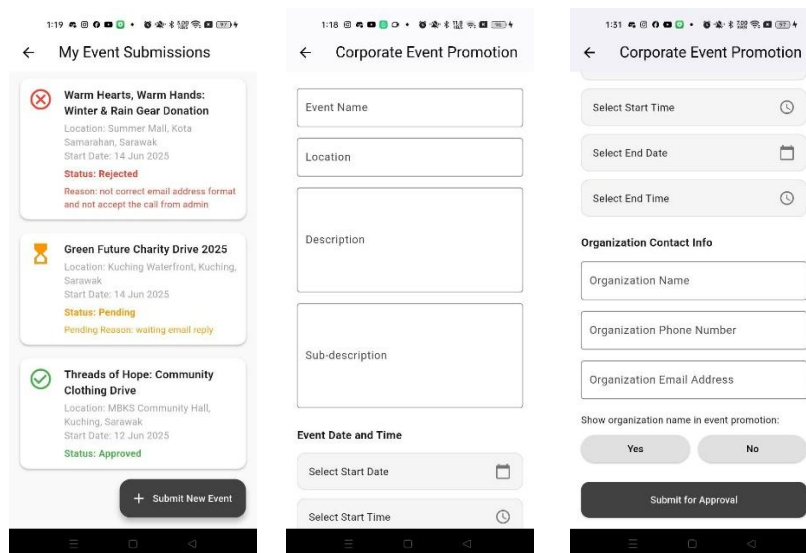


Figure 4.9: UI for Corporate Event Promotion

Figure 4.9 highlights the Submit and Promote Your Event feature, showcasing two main stages: the main interface and the event submission form. This feature empowers users—particularly organizations or community partners—to propose and submit their own donation-related events for promotion within the app. In the first image, the main interface displays a list

of previously submitted events by the user. Each listed event shows its name, location, and start date, along with a status badge indicating whether the event is Pending, Approved, or Rejected. For events still under review, a pending reason is also displayed—such as "Waiting for email reply"—to inform the user why the event has not yet been approved. In the case of rejected events, a rejection reason is clearly shown to ensure transparency.

A floating “Submit New Event” button is positioned at the bottom right corner, making it easy for users to initiate a new submission. When tapped, the app navigates to the event submission form as shown in the second and third images. The form collects comprehensive event details including event name, location, description, sub-description, start date and time, and end date and time. In addition, the form requires users to enter their organization’s name, phone number, and email address, ensuring that event organizers can be contacted by the admin team if needed.

An important option included in the form is the ability for the organizer to choose whether or not to display the organization name on the publicly promoted event. This is facilitated by a simple "Yes" or "No" selection. Once the form is completed and submitted, the event request is forwarded to the admin panel for manual review and approval.

This feature supports structured and transparent collaboration between external organizations and the platform, encouraging community engagement and enhancing the app’s involvement in corporate social responsibility initiatives while ensuring admin oversight remains in place.

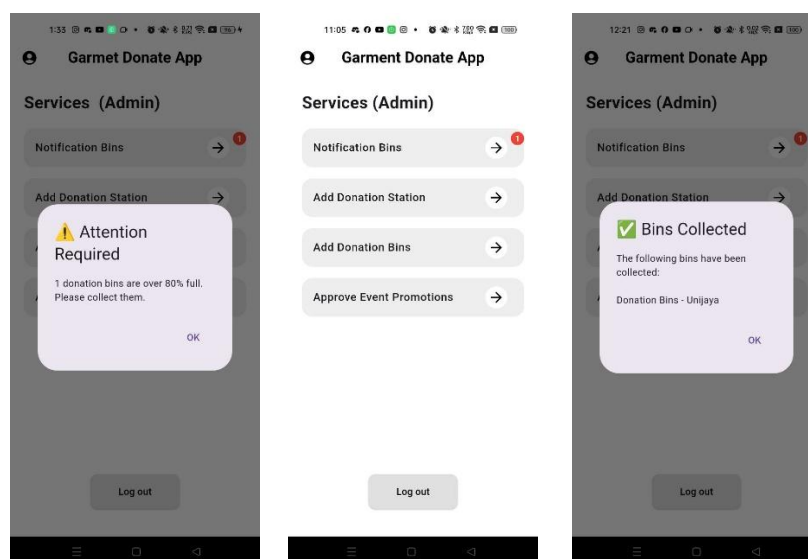


Figure 4.10: UI for Admin Dashboard

Figure 4.10 illustrates the Admin Dashboard, which equips administrators with tools to manage and monitor donation operations efficiently. The figure includes three images that showcase key functionalities designed to maintain smooth platform operations.

The first image shows an alert interface where bins that exceed 80% fill level are highlighted. When a bin reaches this threshold, a pop-up message immediately notifies the admin, identifying which specific bins require attention. This feature ensures prompt collection actions to prevent overflow and maintain public cleanliness.

The second image displays the main Admin Dashboard interface, presenting several core features:

- Notification Bins – Shows a red badge (e.g., "1") on the icon to indicate how many bins are currently overfilled and need admin attention.
- Add Donation Station – Allows the admin to add and register new donation stations by entering the station's name, address, contact number, email, coordinates, and operating hours.
- Add Donation Bins – Enables the addition of new bins by providing necessary details such as bin name, location, and GPS coordinates.
- Approve Event Promotion – This new feature allows the admin to review corporate event submissions sent by users. Each request includes event details such as name, location, description, schedule, and optional contact information. Admins can choose to approve or reject submissions. If rejected, a reason must be provided, which will be visible to the user.

The third image highlights a pop-up message that appears when bins previously over 80% are collected and now report a fill level near 0%. This collection alert helps the admin track bin maintenance activity and confirms successful pickups, maintaining efficient waste management.

Together, these features provide the admin with a comprehensive control panel to oversee bin operations, manage user-submitted events, and ensure the platform functions effectively in both service and community engagement.

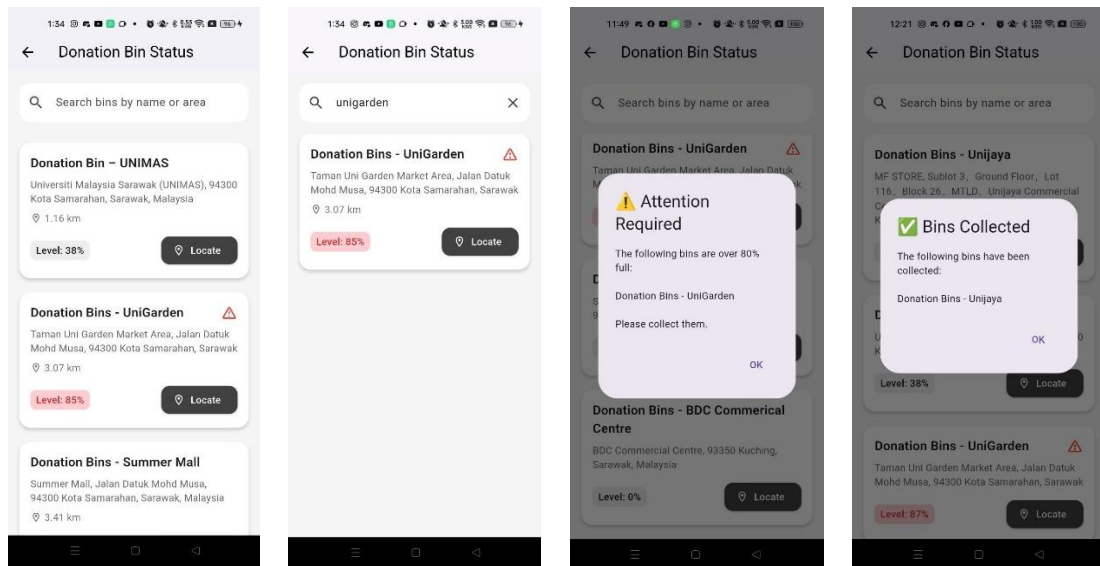


Figure 4.11: UI for Notification Bins

Figure 4.11 showcases the Notification Bins interface, which provides the admin with real-time insights and alerts related to the current status of all donation bins in the system. This interface is essential for monitoring and responding to bin usage efficiently.

The first image displays the main Notification Bins page, listing all registered bins. Each bin entry includes important details such as:

- Bin Name
- Location
- Distance from the admin's current location
- Bin Level Status – Displayed as a percentage (0%–100%). If a bin reaches over 80%, the status bar turns red to visually indicate urgency.
- A Locate button is available for each bin, which opens Google Maps and directs the admin to the specific bin's location for collection.

The second image highlights the search functionality, which allows the admin to search bins by name or area. This feature improves usability, especially when managing a large number of bins across various locations.

The third image illustrates a pop-up alert that appears when one or more bins exceed 80% capacity. This "Attention" notification prompts the admin to take immediate action, helping to prevent overflow or service delays.

The fourth image presents another pop-up message that appears when bins previously marked as overfilled are now collected and their levels drop to 0%–20%. This "Bin Collected"

notification confirms that the bin has been serviced and allows the admin to verify collection status without manually checking each bin.

This Notification Bins feature ensures responsive bin management, enabling admins to act quickly on fill levels, streamline maintenance tasks, and maintain operational efficiency across donation sites.

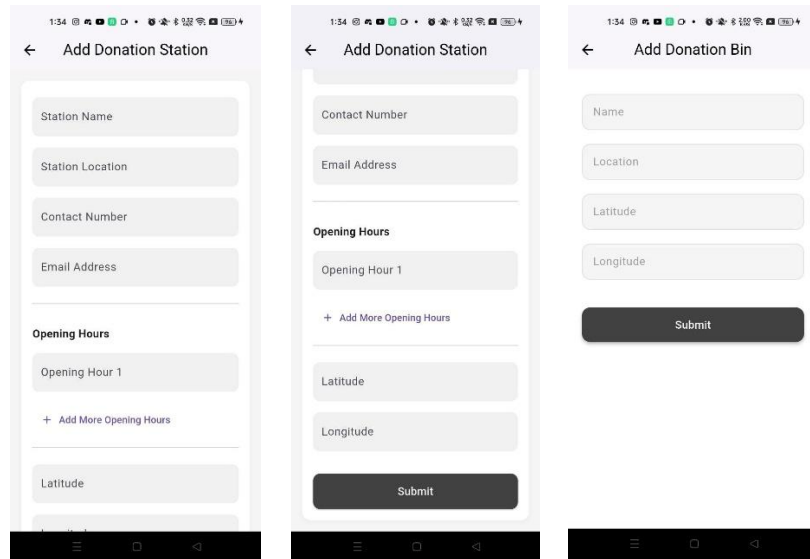
The image displays three mobile application screens side-by-side. The first screen, titled 'Add Donation Station', contains input fields for 'Station Name', 'Station Location', 'Contact Number', 'Email Address', a section for 'Opening Hours' with 'Opening Hour 1' and an 'Add More Opening Hours' button, and a 'Latitude' field. The second screen, also titled 'Add Donation Station', shows 'Contact Number', 'Email Address', the 'Opening Hours' section, and 'Latitude' and 'Longitude' fields. The third screen, titled 'Add Donation Bin', includes 'Name', 'Location', 'Latitude', 'Longitude', and a 'Submit' button. All screens feature a back arrow in the top left corner and a standard Android navigation bar at the bottom.

Figure 4.12: UI for Add Donations Station and Bins

Figure 4.12 represent admin interface allows administrators to add new donation stations and bins into the system.

The Add Donation Station page enables the admin to input:

- Station Name
- Address Location
- Contact Number
- Email Address
- Opening Hours
- Coordinates (Latitude and Longitude)

The Add Donation Bins page allows the admin to input:

- Bin Name
- Location Address
- Coordinates (Latitude and Longitude)

This functionality ensures that new stations and bins can be registered and made available to users in real-time, supporting expansion and accurate geolocation for user features like locating the nearest donation points.

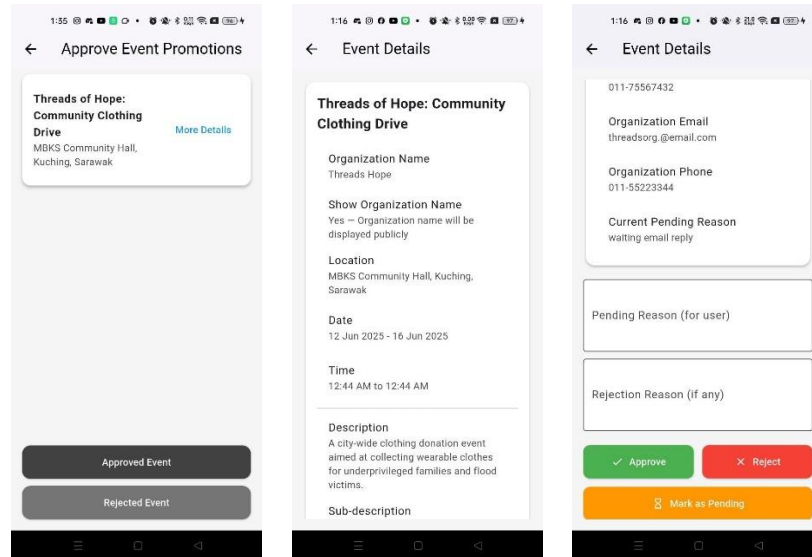


Figure 4.13: UI for Approve Event Promotions

Figure 4.13 depicts the workflow an administrator follows when moderating Corporate Event Promotion requests.

In the main list view (first image) every submitted event appears in a tidy table that shows the Event Name and Location plus a More Details button. An extra column displays the submitting organization’s name (if the organizer chose “Show organization name: Yes — Organization name will be displayed publicly” when completing the form); if they selected No, that field is show No — Organization name will not be shown to the public. Two quick filters— Approved Events and Rejected Events—sit at the bottom of the screen so the admin can instantly narrow the list by status.

Tapping More Details opens the Event Details panel (second image). Here the admin can review the complete brief:

- Event Name
- Organized by <Org Name> (shown only when the Show Org Name flag = Yes)
- Location
- Description & Sub-description
- Start Date & Time – End Date & Time

A Pending Reason field is also present. If the admin needs additional information—e.g., “Waiting email reply”—they enter it here and hit the orange Pending button at the bottom. Doing so stores the reason in the pending reason field and keeps the submission in a Pending state.

Scrolling reveals the Submitter Contact section (third image) with:

- User Email
- User Phone
- Organization Email
- Organization Phone

All organization contact fields are mandatory in the submission form, guaranteeing that the admin can always reach the organizer.

At the bottom of the detail view the admin has three decisive actions:

- Approve – marks the event Approved and publishes it.
- Reject – requires a typed rejection reason, copies it to rejection_reason, and notifies the submitter.
- Pending (orange) – saves any text in pending_reason and leaves the request awaiting further response from the organizer.

This moderation pipeline gives the platform full quality control while keeping organizers informed—with visible Pending, Approved, or Rejected statuses and any admin-supplied reasons—before an event ever reaches the public Event Promotion list.

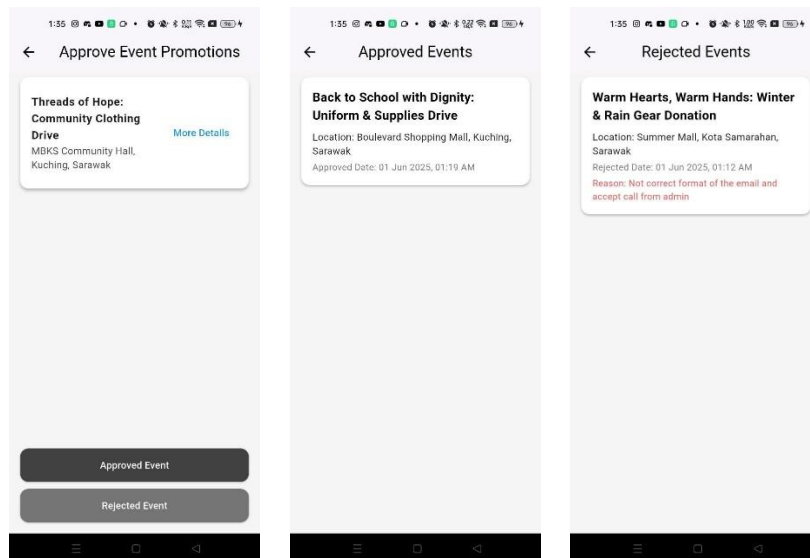


Figure 4.14: UI for Approve Event Promotions of Approved Events and Rejected Events

Figure 4.14 showcases the extended functionality of the Approve Event Promotions module, enabling the admin to efficiently manage and review both approved and rejected corporate event submissions. This structured interface helps in maintaining transparency and provides an organized record of decisions made on each event.

The first image presents the main Approve Event Promotions page, listing all event promotion requests submitted by users. Each row includes:

- Event Name
- Location
- A More Details button for in-depth information
- At the bottom of the page, two navigation buttons—Approved Event and Rejected Event—allow the admin to switch views and access the corresponding categorized lists.

The second image shows the Approved Events page, which appears after the admin presses the Approved Event button. This page displays all the events that have been approved by the admin. Each row in the list includes:

- Event Name
- Location
- Approved Date

This view provides a quick reference to all successfully reviewed and accepted events, allowing for tracking and validation purposes.

The third image shows the Rejected Events page, accessed by pressing the Rejected Event button. It lists all the events that were reviewed and rejected by the admin. Each entry contains:

- Event Name
- Location
- Rejected Reason
- Rejected Date

This layout not only documents which events were denied but also provides a reason for the rejection, helping users understand the decision and allowing room for improvement if they wish to resubmit.

Overall, this interface supports better administrative control and event oversight while ensuring that both approved and rejected records are accessible and clearly organized.

4.4.3 Firebase

Firebase is used as the backend-as-a-service (BaaS) platform for the project (Kunitsyn & Fedorov, 2025). It offers multiple integrated services that support both the hardware and mobile application components. The current implementation uses:

- Firestore Database – to store structured data such as user profiles, donation station details, bin locations, and event promotions.
- Realtime Database – to handle dynamic sensor data (e.g., bin fill levels) sent by the Arduino Nano 33 IoT. This ensures that any change in bin levels is instantly reflected in the app.
- Firebase Authentication – to simplify the user sign-in process, currently supporting email/password login. Google Sign-In (Gmail login) is planned but not yet deployed.

The integration of Firebase with Flutter enables a smooth and efficient communication channel between the Arduino hardware, the cloud backend, and the mobile application (Kunitsyn & Fedorov, 2025)

4.2.3.1 Firestore Database

Firestore Database is a scalable, real-time NoSQL cloud database provided by Firebase that enables structured data storage using collections and documents (Kunitsyn & Fedorov, 2025). In this project, Firestore is used to store and manage structured data for various modules of the donation app, including donation bins, donation stations, event promotions, and user profiles. Below is a breakdown of the key collections and the type of information each stores.

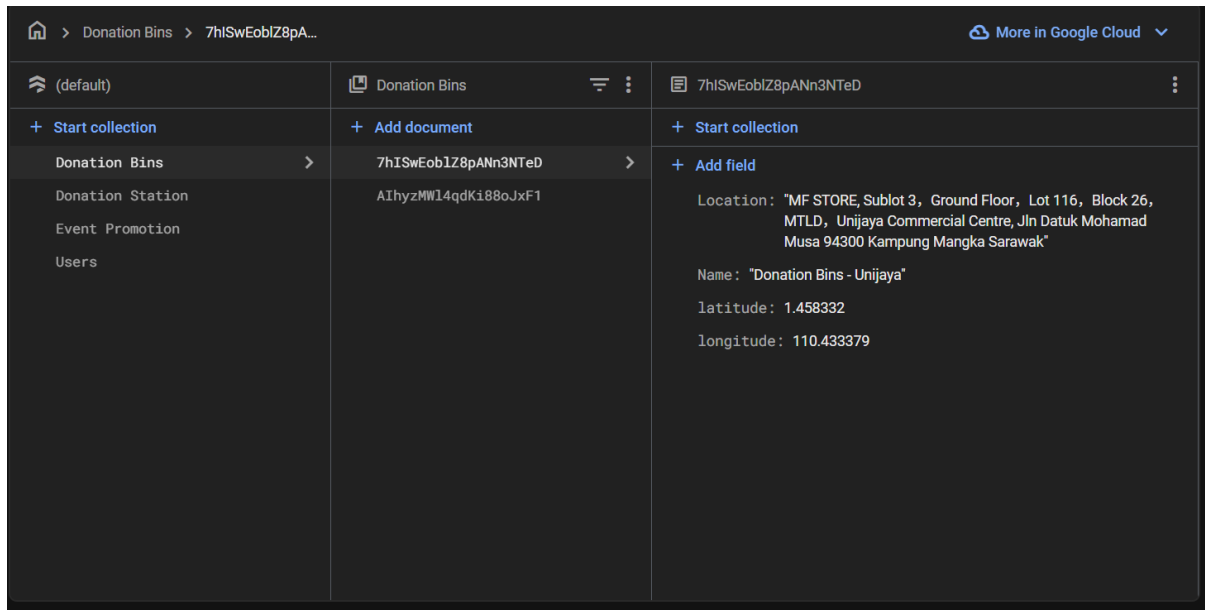


Figure 4.15: Firestore Database of Collections for Users Collection

The figure 4.15 show the users collection is designed to store profile information for every individual who registers or logs into the app. Each document in this collection is uniquely identified by the user's Firebase Authentication UID, ensuring that user data is securely linked to their authenticated account.

- **created_at (Timestamp):** Records the exact date and time the user account was created in the system. Example: 31 May 2025 at 00:28:24 UTC+8
- **email (String):** Stores the registered email address of the user. This is typically the same email used for authentication and communication. Example: "test0@gmail.com"
- **phone_number (String):** Captures the user's mobile or contact number. This field may initially be blank and is usually editable via the user's profile settings. It is especially important for features that require user verification or contact, such as event submission or support.
- **username (String):** Represents the user's chosen display name. Like the phone number, this can also be updated by the user to personalize their profile.

The Users collection acts as the foundational data source for managing individual user identities within the app. It supports both personalization (via username) and communication (via email and phone number). The inclusion of the created at timestamp helps with user account tracking and analytics. As the app grows, this structure supports scalability and secure access control based on each user's unique document ID.

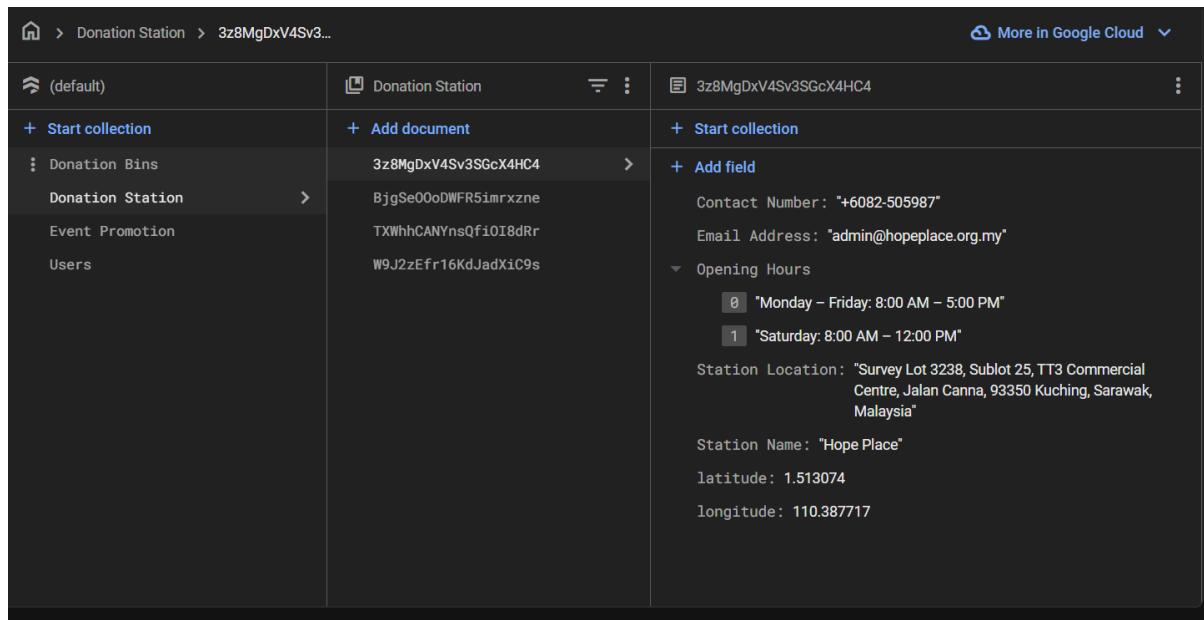


Figure 4.16: Firestore Database of Collections for Donation Station Collection

The Figure 4.16 show Donation Station collection holds documents representing registered donation stations—these are larger, often permanent donation drop-off points. Each station document contains both contact and location information:

- Station Name (String): Name of the donation station. *Example:* "Hope Place"
- Station Location (String): Full physical address. *Example:* "Survey Lot 3238, Sublot 25, TT3 Commercial Centre..."
- Contact Number (String): Telephone number for inquiries. *Example:* "+6082-505987"
- Email Address (String): Official email for communication. *Example:* "admin@hopeplace.org.my"
- Opening Hours (Array): Operational time slots listed by day. *Example:* "Monday – Friday: 8:00 AM – 5:00 PM"
- Latitude and Longitude (Number): Used for map positioning and navigation features.

The Donation Station collection holds detailed records of official donation centers or stations. Each document includes contact information (phone number and email), station name, precise address, operating hours (as an array of strings), and latitude and longitude for location services. This collection enables the app to provide comprehensive details for users seeking nearby donation stations, including navigation support via Google Maps and access to station-specific information. The structure is essential for enhancing user trust, facilitating contact, and ensuring accessibility to physical donation points.

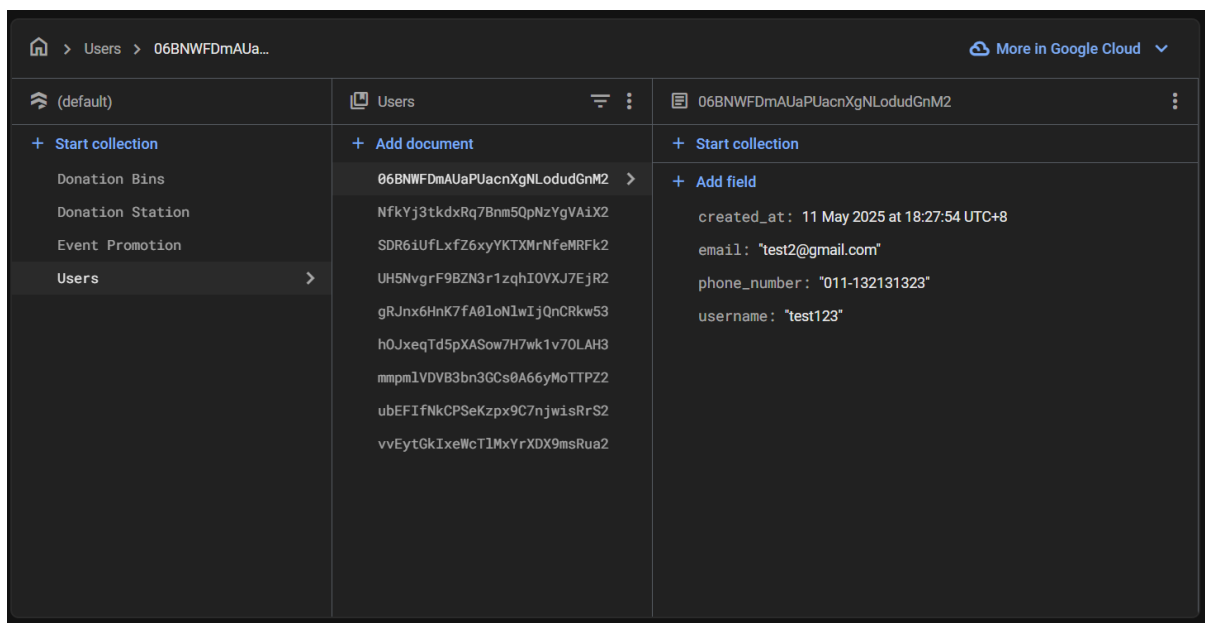


Figure 4.17: Firestore Database of Collections for Donation Bins Collection

The Figure 4.17 show Donation Bins collection represents a physical donation bin. The documents include location metadata that helps users locate nearby bins through geolocation-based services in the app. Example fields include:

- Location (String): The full address of the donation bin. *Example:* "MF STORE, Sublot 3, Ground Floor, Lot 116, Block 26, MTLA..."
- Name (String): The label or name assigned to the bin. *Example:* "Donation Bins - Unijaya"
- Latitude and Longitude (Number): Geographic coordinates used for distance calculation and Google Maps integration.

The Donation Bins collection stores information about all physical donation bin locations managed within the app. Each document represents a single donation bin and includes details such as the bin's name, exact location, and geographical coordinates (latitude and longitude). This data supports features like bin location display on a map, distance calculation from the user, and linking to real-time bin fill levels stored in the Firebase Realtime Database. The structure enables users and admins to efficiently find, monitor, and manage bin usage based on location-specific data.

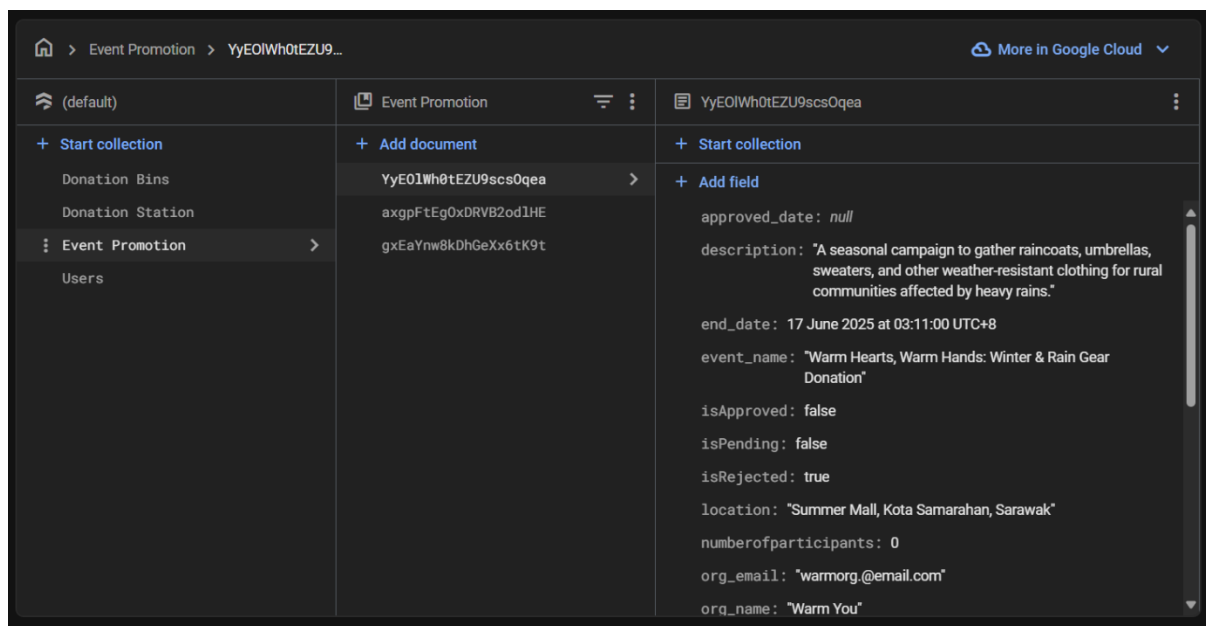


Figure 4.18: Firestore Database of Collections for Event Promotion Collection

The Figure 4.18 show the Event Promotion collection stores all data related to donation or community events, particularly those submitted by organizations or public users. Each document corresponds to one submitted event and includes comprehensive event metadata:

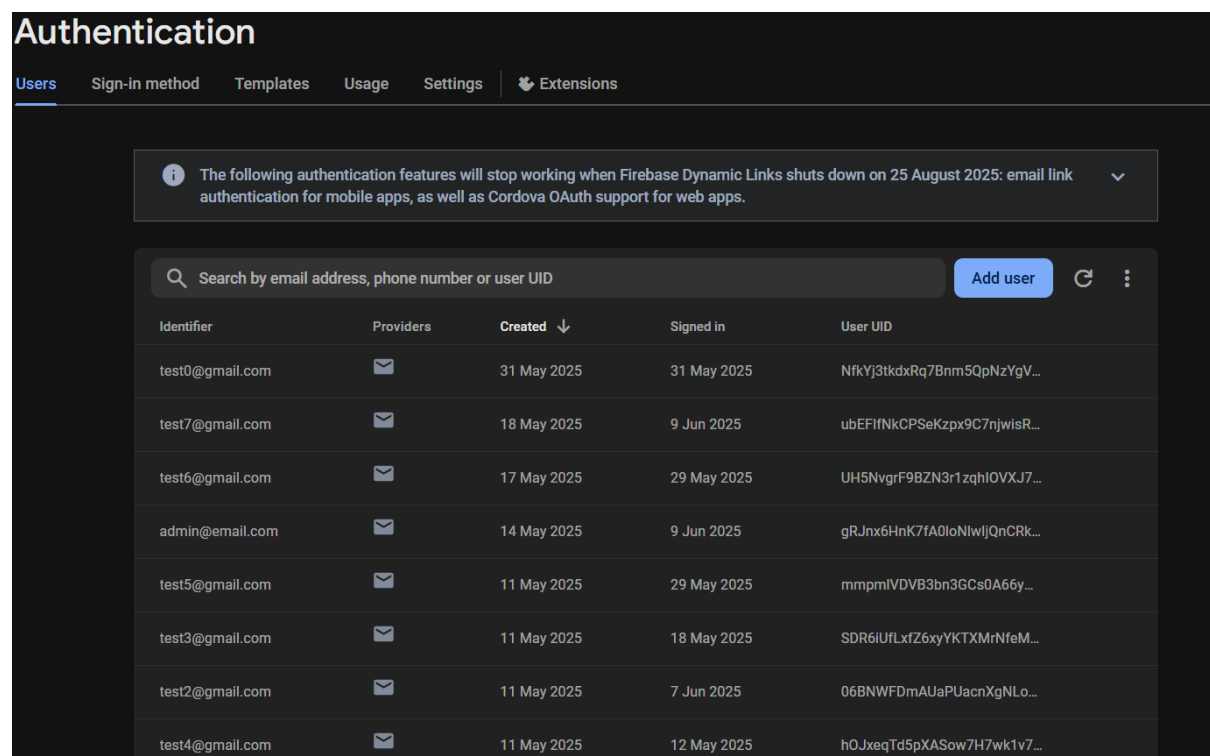
- `event_name` (String): Title of the event. *Example*: "Green Future Charity Drive 2025"
- `description` / `subdescription` (String): Detailed and brief summaries of the event purpose and activities.
- `location` (String): Event venue. *Example*: "Kuching Waterfront, Kuching, Sarawak"
- `start_date` / `end_date` (Timestamp): Date and time boundaries for the event.
- `isApproved`, `isPending`, `isRejected` (Boolean): Approval status indicators managed by admin.

- `approved_date`, `rejected_date` (Timestamp or null): Stores time when admin action was taken.
- `rejection_reason`, `pending_reason` (String): Text feedback from admin (e.g., “waiting email reply”).
- `user_id`, `user_email`, `user_phone` (String): Identifies the submitting user for communication and tracking.
- `org_name`, `org_email`, `org_phone` (String): Organization info; mandatory upon submission.
- `show_org_name` (Boolean): Determines if the organization name will be shown publicly.
- `numberofparticipants` (Number): Tracks user engagement or event sign-ups.

The Event Promotion collection is designed to manage community and corporate event campaigns submitted by users. Each document stores extensive event details such as event name, description, sub-description, location, date and time range, number of participants, and organizer contact info. It also includes administrative fields for managing approval status (`isPending`, `isApproved`, `isRejected`) and related metadata like rejection reasons, approval date, and visibility of the organizer’s name. This collection powers the event discovery and participation features in the app, supporting admin moderation, transparent status tracking, and enhanced user engagement in donation-related initiatives.

Firestore Database plays a crucial role in supporting real-time data access and modular storage for this app. By organizing information into dedicated collections (Donation Bins, Donation Station, Event Promotion, and Users), the app can efficiently perform location-based services, event management, and user role differentiation. Each collection is well-structured, allowing both admin and user functionalities to operate smoothly within the app environment.

4.2.3.2 Firebase Authentication



The screenshot shows the Firebase Authentication console interface. At the top, there's a navigation bar with tabs: Users, Sign-in method, Templates, Usage, Settings, and Extensions. Below this, a warning message states: "The following authentication features will stop working when Firebase Dynamic Links shuts down on 25 August 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps." Below the warning, there's a search bar with the placeholder text "Search by email address, phone number or user UID" and an "Add user" button. The main content is a table listing registered users.

Identifier	Providers	Created ↓	Signed in	User UID
test0@gmail.com	✉	31 May 2025	31 May 2025	NfkYj3tkdxRq7Bnm5QpNzYgV...
test7@gmail.com	✉	18 May 2025	9 Jun 2025	ubEFiFNkCPSeKzpx9C7njwisR...
test6@gmail.com	✉	17 May 2025	29 May 2025	UH5NvgrF9BZN3r1zqhiOVXJ7...
admin@email.com	✉	14 May 2025	9 Jun 2025	gRJnx6HnK7fA0loNIwIjQnCRk...
test5@gmail.com	✉	11 May 2025	29 May 2025	mmpmIVDVB3bn3GCs0A66y...
test3@gmail.com	✉	11 May 2025	18 May 2025	SDR6iUfLxZ6xyYKTXMrNfeM...
test2@gmail.com	✉	11 May 2025	7 Jun 2025	06BNWFDmAUaPUacnXgNLo...
test4@gmail.com	✉	11 May 2025	12 May 2025	hOJxeqTd5pXASow7H7wk1v7...

Figure 4.19: Firebase Authentication

Figure 4.19 illustrates the Firebase Authentication interface, specifically showcasing the Email/Password sign-in method. Firebase Authentication is a core security feature that manages user identity and access in a secure and scalable way. In this app, it allows users to create an account and sign in using their email address and password combination.

Within the Firebase Console's Authentication tab, the interface displays key user information in a tabular format:

- Identifier – Shows the email address used as the login credential for each registered user.
- Providers – Indicates the sign-in method, in this case, Email/Password.
- Created – Displays the date and time when the user account was created.
- Signed In – Shows the most recent login timestamp, helping administrators monitor user activity.
- User UID – The unique identifier automatically assigned to each user by Firebase. This UID is used throughout other Firebase services (e.g., Firestore) to securely link user-specific data like profiles, submissions, and preferences.

This authentication setup supports secure access control, user management, and streamlined login processes across the app. By using Firebase Authentication, the app benefits from built-in security features such as password hashing, session handling, and real-time authentication state monitoring.

4.2.3.3 Firebase Realtime Database

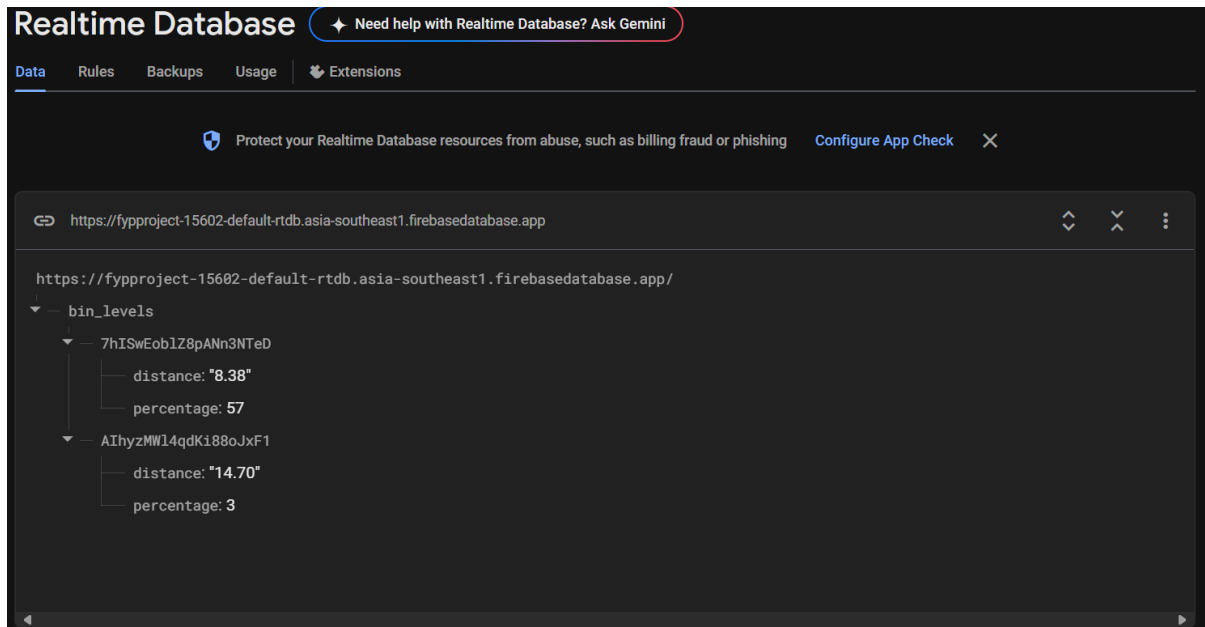


Figure 4.20: Firebase Realtime Database

Figure 4.20 displays the structure and data stored within the Firebase Realtime Database, a cloud-hosted NoSQL database that stores and syncs data between users in real time (Madaminov & Allaberganova, 2023). In this app, the Realtime Database is primarily used to monitor and display live updates of donation bin statuses, such as their fill level percentages and distance from the user.

The database structure includes a root node called `bin_levels`, under which each donation bin is represented by a unique document ID (e.g., `7hISwEoblZ8pANn3NTeD`). Each document stores the following key data:

- **percentage** – A numeric value representing the current fill level of the bin (0% to 100%). This value is used to visually indicate whether the bin is low, moderately full, or needs urgent collection (e.g., >80%).
- **distance** – A string representing the calculated distance (in kilometers) between the user's current location and the bin. This helps users identify the nearest available bins.

This live data is typically updated by IoT hardware (e.g., Arduino with ultrasonic sensors) and allows the app to reflect changes instantly without requiring page refreshes or manual updates. This real-time capability ensures efficient waste management operations, enabling admins and users to make informed decisions about when and where to donate.

The Firebase Realtime Database complements other Firebase services by offering synchronized, scalable, and fast data operations—especially crucial for features that depend on up-to-date sensor information (Madaminov & Allaberganova, 2023)

4.5 Key Features

The garment-donation monitoring system now uses two HC-SR04 ultrasonic sensors, each wired to an Arduino Nano 33 IoT and its own tri-colour LED stack. The board streams data directly to Firebase, which is consumed by the Flutter mobile application.

4.5.1 Dual-Bin Monitoring with Ultrasonic Sensors

To simulate real-time monitoring of multiple donation bins, the system employs two HC-SR04 ultrasonic sensors connected to a single Arduino Nano 33 IoT board. Each sensor is responsible for monitoring one bin, making it possible to track the fill levels of two separate bins simultaneously. This setup reflects a realistic deployment scenario where each bin is equipped with a dedicated sensor for accurate and independent monitoring.

During testing, hand placement was used to simulate objects inside the bins. For this reason, a shorter distance range of 0cm to 15cm was applied to represent different fill levels. However, in actual deployment, the real distance range inside a bin is typically larger. A practical scale used for live bins would be:

- Above 20 cm → High fill (Red)
- 20 cm – 50 cm → Moderate fill (Yellow)
- Below 50 cm → Low fill (Green)

This practical threshold ensures a more accurate representation of how full the bin is based on its physical depth.

4.5.2 Real-Time, Per-Bin Data Streams to Firebase

Every three seconds the microcontroller converts the measured distance into a fill percentage and pushes a JSON object

```
{ distance: "<cm>", percentage: <int> }
```

to `/bin_levels/{binId}` in Firebase Realtime Database. Separate document IDs keep the two data streams isolated, so the cloud layer treats them exactly as if they originated from two independent devices.

4.5.3 On-Site LED Indicators

A quick glance at the bins tells staff whether they need servicing:

- **Green** = plenty of capacity
- **Yellow** = nearly half-full
- **Red** = over 80 % (collection required)

This visual cue is useful even when the mobile app or internet connection is unavailable.

4.5.4 Secure & Scalable Cloud Connectivity

The firmware relies on Wi-FiNINA and ArduinoHttpClient to perform HTTPS PUT requests, ensuring data integrity and confidentiality.

- Firebase Realtime Database holds live sensor readings.
- Cloud Firestore stores static metadata (bin names, station details, event information).

This split lets the solution scale smoothly as more bins or stations are added without restructuring existing data.

4.5.5 Instant Feedback in the Flutter Application

The Flutter app subscribes to the `/bin_levels` node; changes arrive via Firebase's real-time listeners and are rendered almost instantly.

Users can therefore:

- Locate nearby bins and stations on an integrated map.
- Check live fill percentages (bins > 80 % are highlighted in red).
- Receive in-app alerts when a bin transitions from full to empty or vice versa.

Flutter's cross-platform UI toolkit, combined with Firebase's SDKs, gives a polished, low-latency user experience on both Android and iOS devices.

4.5.6 Flutter Application Features

The Flutter mobile application serves as the user-facing interface that connects donors, organizations, and system administrators through a seamless and responsive experience. Built using Google’s Flutter framework and integrated with Firebase services, the app provides a centralized platform for real-time interaction, navigation, event participation, and community engagement. The core features are outlined below:

4.5.6.1 Locate Donation Bins and Stations

Users can access an interactive map and list view showing the real-time locations of:

- Donation Bins – Each bin displays its name, location, and current fill level (retrieved from Firebase Realtime Database). Bins above 80% capacity are flagged in red to indicate that collection is required.
- Donation Stations – Fixed donation drop-off points (e.g., charity offices) are shown using data from Cloud Firestore. Users can view station details like opening hours, address, contact number, and email.

The app uses the Geolocator package to show the user’s current position and calculate distances to each station or bin, making it easy to find the nearest drop-off point.

4.5.6.2 View and Participate in Event Promotions

The app includes an Event Promotion section where users can explore upcoming donation-related events submitted by other users or organizations. Each event card provides:

- Event Name and Location
- Start Date
- Short Description
- “More Details” button for complete event information, including start/end time, full description, and organizer details.

If the event is still active, users can choose to:

- Join the Event – to express interest or RSVP
- Locate the Event – to open directions in Google Maps

Expired events are clearly labeled as “Event Ended”, and historical events can be accessed from a History button at the top right corner of the Event Promotion page.

4.5.6.3 Submit and Promote Your Event

Users, especially organizations or partners, can use the Corporate Event Promotion feature to submit their own community events. The event submission form includes required fields such as:

- Event Name, Description, Sub-description
- Start and End Dates
- Location
- Organizer Contact Info: Organization Name, Phone Number, and Email (mandatory)
- A toggle option to “Show Organization Name” publicly on the event listing

Once submitted, the event enters a Pending state and awaits admin review. If any pending reason (e.g., “Waiting for Email Reply”) is provided by the admin, it will be shown along with a Pending badge in orange for clarity.

Users can track the status of their submissions (Pending, Approved, Rejected) along with corresponding feedback and edit options if needed.

Together, these Flutter app features enhance user engagement and accessibility, while fostering community collaboration and efficient donation flow through mobile technology and real-time cloud connectivity.

4.6 Summary

Chapter 4 provides a comprehensive overview of the implementation process for the smart garment donation bin monitoring system. It begins with an introduction to the hardware components, including the Arduino Nano 33 IoT board, HC-SR04 ultrasonic sensors, LED indicators, and essential accessories such as a breadboard and jumper wires. The chapter details the wiring connections for two ultrasonic sensors and their corresponding LED indicators, each sensor dedicated to monitoring a separate donation bin. These components work together to detect fill levels and provide on-site visual feedback, simulating a dual-bin monitoring setup within a compact hardware environment.

The software components are also extensively covered, showcasing how PlatformIO was used to program the Arduino for real-time data processing and secure communication with Firebase. The system integrates Cloud Firestore for static data, Firebase Realtime Database for dynamic bin updates, and Firebase Authentication for user access control. Additionally, a Flutter-based mobile application displays bin fill levels in real time and supports key features such as locating donation bins and stations, viewing event promotions, and submitting corporate events. This seamless integration between hardware, cloud services, and mobile application highlights the project's scalability, efficiency, and real-world applicability in promoting community-based donation efforts.

CHAPTER 5

Testing

5.1 Introduction

Testing plays a critical role in ensuring the reliability and success of any software application. It verifies that the system functions as intended, meets user expectations, and operates efficiently under various conditions. In this project, extensive testing was performed on both user-facing and admin functionalities of the Garment Donation App. The primary objective was to detect and resolve any bugs, performance issues, or usability concerns prior to deployment. Both functional and non-functional testing approaches were employed to evaluate the application's accuracy, responsiveness, and overall user experience.

5.2 Functional Testing

Functional testing focuses on verifying that the application performs its intended functions correctly. Each module and feature of the app was tested against the defined requirements and expected behavior. This included testing user authentication (sign-up and sign-in), dashboard navigation, event promotion submissions, donation station and bin locating features, and admin approval functionalities. The tests ensured that users could register, log in, submit events, view donation stations, and interact with bins as intended. Admins were also able to review, approve, or reject event submissions. These tests confirmed that each function delivered the correct output given a valid input and handled invalid scenarios appropriately.

The functional test plan began with authentication flows. Using Firebase's authentication emulator, valid and invalid sign-up / sign-in attempts were executed on a physical Android handset. Assertions verified correct redirection to the dashboard, storage of user UIDs, and user-friendly error banners for bad credentials or disabled accounts.

Next, core user journeys were scripted with the Flutter integration-test package. Scenarios covered, locating the nearest station, reading its detail sheet, and opening Google Maps, viewing live bin-level percentages, including the red "> 80 %" alert, joining an event, then seeing the participant counter increment and submitting a corporate event with mandatory organisation data and the "show org name" toggle. Each script compared on-screen values against mock documents in the Firebase emulator to ensure perfect data binding.

Admin-side tests focused on moderation workflows. With an admin account, we loaded the "Approve Event Promotions" list, filtered by Pending, tapped More Details, typed a pending_reason, and pressed Pending, Approve, and Reject in turn. After every action, a Firestore query confirmed that the booleans (isApproved, isPending, isRejected) and timestamps (approved_date, rejected_date) were updated atomically, and that the organiser's history screen reflected the change.

Embedded-code functional tests used PlatformIO's Unity unit-test framework. Stubbed echo-pulse durations were fed into the getDistance() routine to guarantee centimetre accuracy, after which the calculatePercentage() and updateLEDsByDistance() functions were validated against a truth-table (0-5 cm → Red ON, 6-10 cm → Yellow ON, 11-15 cm → Green ON). Finally, HTTPS requests were replayed through the Postman mock server to prove that each sensor wrote to the correct /bin_levels/{binId} path.

5.3 Non-Functional Testing

Non-functional testing focused on assessing the application's performance, usability, and stability. The app was tested across multiple device screen sizes to ensure a responsive layout. Load times, Firebase data retrieval, and real-time updates were evaluated to confirm that data transmission between the embedded system and the Flutter application remained seamless. The app's user interface, based on a clean white and grey theme, was reviewed for consistency, clarity, and ease of navigation. Stability testing involved monitoring the app for crashes or glitches, particularly after multiple hot reloads during development. One recurring issue identified was the failure of certain asset images to load properly after repeated reloads, underscoring the importance of robust asset management. Overall, non-functional testing validated that the app delivered a smooth and user-friendly experience under practical usage conditions.

5.3.1 Using Real Phone Device – OPPO Reno5 F

Non-functional testing was conducted to assess the overall usability, responsiveness, stability, and compatibility of the Garment Donation App under real-world conditions. A key part of this process involved testing the application directly on a physical mobile device — specifically, an OPPO Reno5 F (Model CPH2217), which runs Android 13, has an octa-core processor, 128GB storage, 16GB RAM, and a 6.43-inch screen. By enabling Developer Mode and connecting the phone via USB, the flutter run command was used to install and execute the app for hands-on evaluation. This approach allowed the system to be tested under realistic usage conditions that represent typical end-user behavior.

Testing on a real device offered insights that emulator or simulation environments might miss. The app's UI rendering, image loading, screen responsiveness, and navigation transitions were observed for fluidity and correctness. This setup helped verify that Firebase operations (such as authentication and data fetching) worked smoothly on a real Android environment, while also assessing device compatibility and resource handling across different screen sizes and hardware configurations. Furthermore, it highlighted any crashes, layout issues, or slow performance that might occur during continuous usage. This hands-on non-functional testing played a vital role in ensuring the application delivers a stable, user-friendly, and responsive experience across various Android devices, particularly for users with mid- to high-end smartphones like the OPPO Reno5 F.

5.3.2 Firebase Test Lab (Cloud-Based Non-Functional Testing)

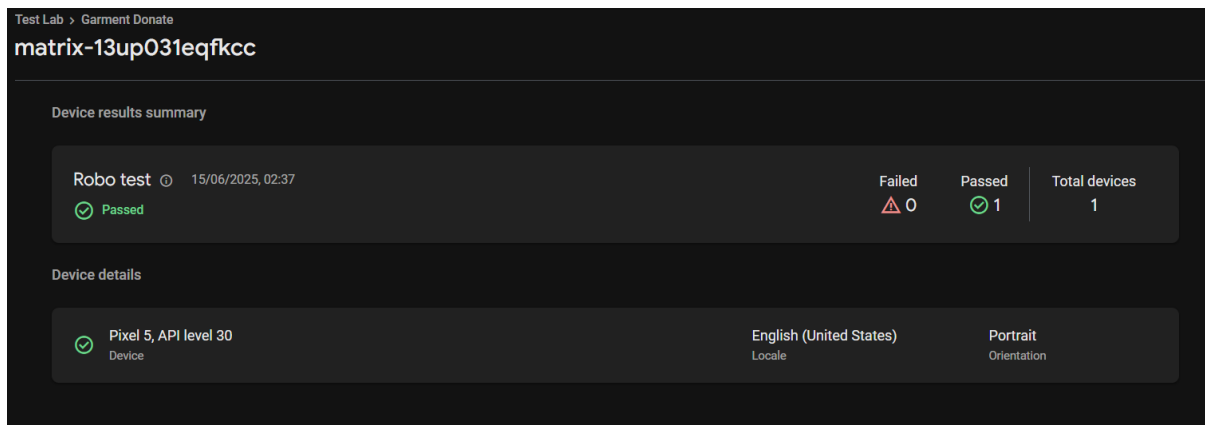


Figure 5.1: Firebase Test Lab about Robo Test of Pixel 5, API level 30

In addition to real-device testing, non-functional testing was also conducted using Firebase Test Lab (Figure 5.1), a cloud-based tool that allows testing on a wide range of physical and virtual devices hosted by Google. Two key screenshots were captured to document this testing process: one showing the Device Results Summary, and the other displaying Device Details of Test Issues. Firebase Test Lab automatically launched a Robo test, which simulated user actions such as tapping buttons, entering text, and navigating between pages.

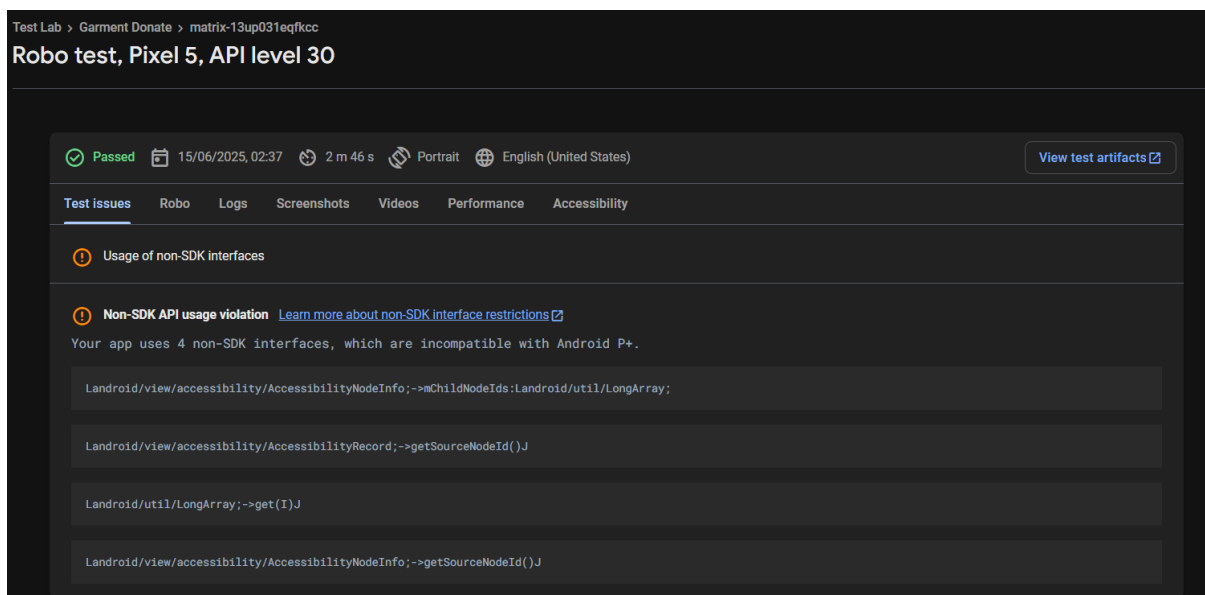


Figure 5.2: Firebase Test Lab about Robo Test of Test Issues

The test environment identified four non-SDK API usage issues (Figure 5.2), which are compatibility warnings related to newer Android versions (Android P and above). These issues involve restricted Android internal APIs, such as accessibility node information. However, after careful review, it was determined that these issues do not affect the functionality, security, or

usability of the application. Since the app operates as intended on both Firebase Test Lab devices and a real device without crashes or errors, no changes were made to the codebase regarding these warnings.

This test confirmed the stability and performance of the app across different Android environments, helping to ensure broader device compatibility. While the Test Lab flagged minor warnings, the app was deemed safe and ready for deployment, provided that its core functionality and user experience remained unaffected. Firebase Test Lab thus complemented real-device testing by providing an automated, large-scale validation of the application's behavior on diverse hardware and software configurations.

5.3.3 Firebase Security Rules Simulator (Security Testing)

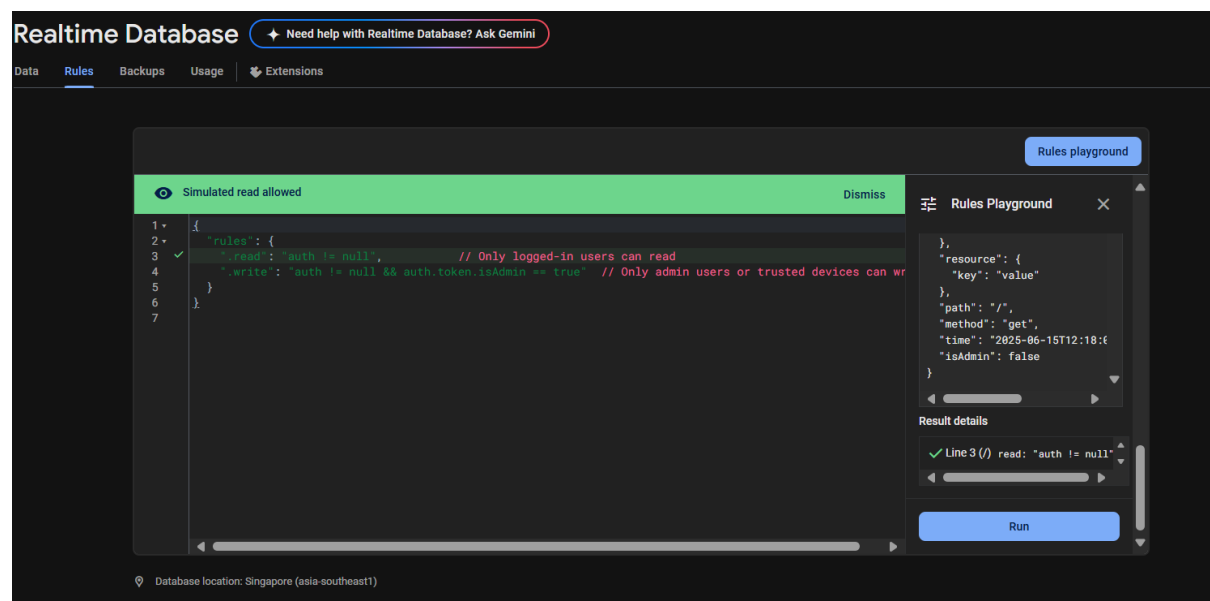


Figure 5.3: Results of Test Run Rules Playground in Rules of Real-Time Database

As part of the non-functional testing phase, security testing was conducted using the Firebase Security Rules Simulator (Madaminov & Allaberganova, 2023). Figure 5.3 show the tool is designed to simulate access attempts to the Firebase Realtime Database in order to verify that security rules are properly enforced and unauthorized access is prevented. The simulator allows developers to test different authentication scenarios without affecting the live database.

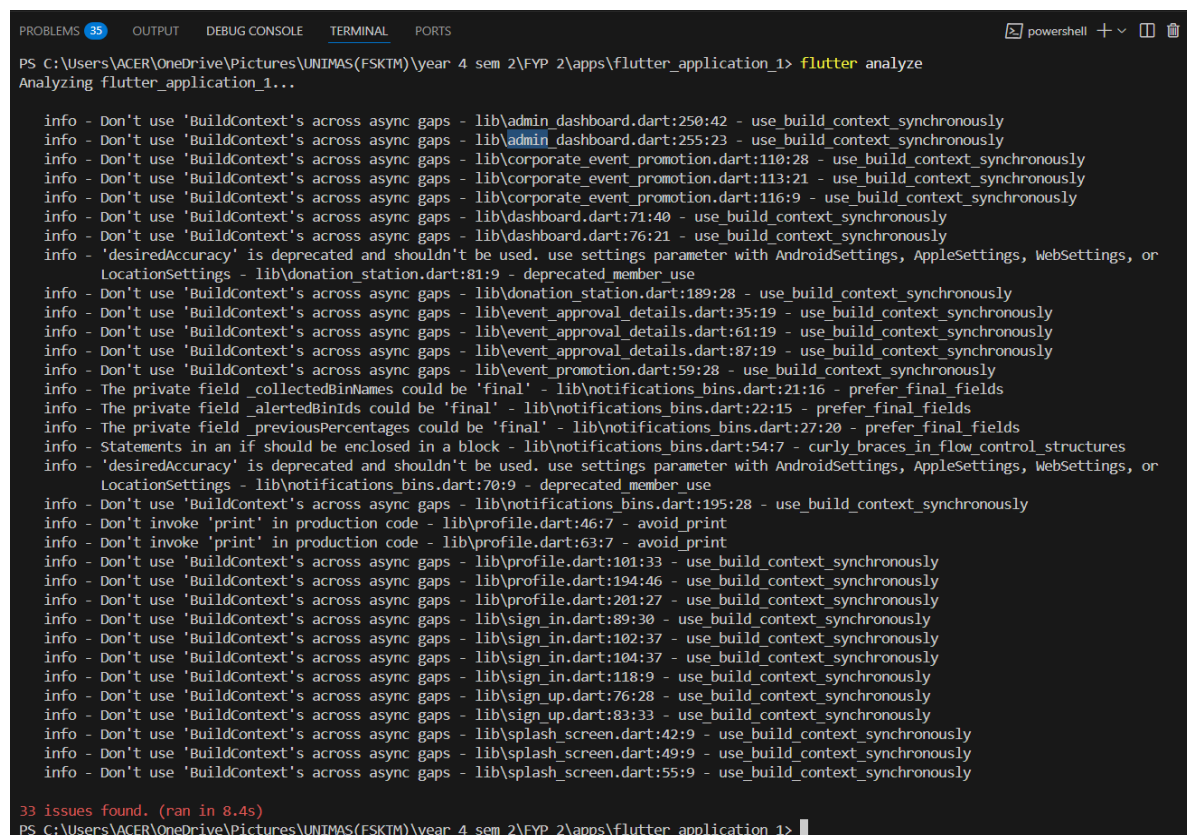
In the test case, a simulated user with Firebase UID 5212cdaf-7707-42d5-9d56-b7d8b5109506 attempted to read data from the root path (/) of the Realtime Database using a GET request. The simulated authentication payload indicated that the user had signed in via

Google but did not have a verified email, phone number, or additional profile details. According to the Firebase security rules, access was granted due to the condition on line 3: `read: "auth != null"`, which allows any authenticated user to read from the database.

This simulation validated that the database was appropriately restricting access to only authenticated users, which is a foundational rule for maintaining data privacy and security. While the current rule permits broad read access, this test highlights the importance of reviewing and refining rules to apply role-based access control—such as distinguishing between regular users and admins—especially if sensitive or role-specific data exists.

The test confirmed that basic authentication checks are in place and functional, ensuring that unauthenticated users cannot access protected resources. Future enhancements could involve implementing more granular conditions such as `auth.token.email_verified == true` or checking for custom claims (e.g., `isAdmin`) to further improve the system's security posture.

5.3.4 Dart Analyze



```
PS C:\Users\ACER\OneDrive\Pictures\UNIMAS(FSKTM)\year 4 sem 2\FYP 2\apps\flutter_application_1> flutter analyze
Analyzing flutter_application_1...

info - Don't use 'BuildContext's across async gaps - lib\admin_dashboard.dart:250:42 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\admin_dashboard.dart:255:23 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\corporate_event_promotion.dart:110:28 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\corporate_event_promotion.dart:113:21 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\corporate_event_promotion.dart:116:9 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\dashboard.dart:71:40 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\dashboard.dart:76:21 - use_build_context_synchronously
info - 'desiredAccuracy' is deprecated and shouldn't be used. use settings parameter with AndroidSettings, AppleSettings, WebSettings, or LocationSettings - lib\donation_station.dart:81:9 - deprecated_member_use
info - Don't use 'BuildContext's across async gaps - lib\donation_station.dart:189:28 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\event_approval_details.dart:35:19 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\event_approval_details.dart:61:19 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\event_approval_details.dart:87:19 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\event_promotion.dart:59:28 - use_build_context_synchronously
info - The private field _collectedBinNames could be 'final' - lib\notifications_bins.dart:21:16 - prefer_final_fields
info - The private field _alertedBinids could be 'final' - lib\notifications_bins.dart:22:15 - prefer_final_fields
info - The private field _previousPercentages could be 'final' - lib\notifications_bins.dart:27:20 - prefer_final_fields
info - Statements in an if should be enclosed in a block - lib\notifications_bins.dart:54:7 - curly_braces_in_flow_control_structures
info - 'desiredAccuracy' is deprecated and shouldn't be used. use settings parameter with AndroidSettings, AppleSettings, WebSettings, or LocationSettings - lib\notifications_bins.dart:70:9 - deprecated_member_use
info - Don't use 'BuildContext's across async gaps - lib\notifications_bins.dart:195:28 - use_build_context_synchronously
info - Don't invoke 'print' in production code - lib\profile.dart:46:7 - avoid_print
info - Don't invoke 'print' in production code - lib\profile.dart:63:7 - avoid_print
info - Don't use 'BuildContext's across async gaps - lib\profile.dart:101:33 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\profile.dart:194:46 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\profile.dart:201:27 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\sign_in.dart:89:30 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\sign_in.dart:102:37 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\sign_in.dart:104:37 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\sign_in.dart:118:9 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\sign_up.dart:76:28 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\sign_up.dart:83:33 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\splash_screen.dart:42:9 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\splash_screen.dart:49:9 - use_build_context_synchronously
info - Don't use 'BuildContext's across async gaps - lib\splash_screen.dart:55:9 - use_build_context_synchronously

33 issues found. (ran in 8.4s)
PS C:\Users\ACER\OneDrive\Pictures\UNIMAS(FSKTM)\year 4 sem 2\FYP 2\apps\flutter_application_1>
```

Figure 5.4: Results of Dart Analyze

In this section, Figure 5.4 show using the flutter analyze tool to examine the code quality and identify any potential issues in the Flutter application. The analysis process helps to enforce good coding practices and ensure future maintainability. Upon running the analysis, a total of 33 info-level issues were identified. These issues include common best practice violations such as using BuildContext across asynchronous gaps, usage of deprecated properties like desiredAccuracy, suggestions to declare fields as final, and minor formatting improvements like adding curly braces in control structures.

Despite the identified issues, the application runs successfully on a real device without any crashes or malfunctions. Therefore, no immediate code changes were made. However, these warnings have been noted for future improvement, especially the repeated recommendation to avoid using BuildContext after await, which may lead to runtime errors if not handled carefully. Addressing these issues in future development phases can improve code stability, safety, and alignment with modern Flutter development standards.

The non-functional testing phase focused on evaluating the application's performance, security, and code quality across various environments. Firstly, the app was tested on a real device, the OPPO Reno5 F, to assess its behaviour, responsiveness, and user experience under real-world conditions. Next, Firebase Test Lab was used for cloud-based device testing, which highlighted a few non-critical issues that did not affect core functionality. Security testing was conducted using the Firebase Security Rules Simulator to ensure that data access rules were enforced correctly, confirming that only authenticated users could access the database. Lastly, a code quality check was performed using flutter analyze, revealing 33 minor issues, such as context usage across async gaps and deprecated features, which were acknowledged for future optimization. Overall, the app functioned as expected, and the non-functional testing validated its readiness for real usage while identifying areas for future improvement.

5.4 Test Case

This section outlines the test cases developed to verify the functionality, reliability, and usability of the donation platform's key features. Each test case is designed to ensure that the application behaves as expected under both normal and edge-case conditions.

5.4.1 Test Cases for SplashScreen and IntroScreen

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: SplashScreen and IntroScreen Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
SplashScreen 01: Display Splash Screen on Launch	Launch the app	Open app from device	App icon is shown centered on a black background (splash screen displayed)	Pass
SplashScreen 02: Duration of Splash Screen	Launch the app	Wait on splash screen	Splash screen remains visible for a few seconds before navigating to the next screen	Pass
IntroScreen 01: First-Time User Intro Display	First-time launch	Open app	Intro screen with app introduction and Sign In button is displayed	Pass
IntroScreen 02: Navigate to Sign In Page	Tap "Sign In"	Tap button	User is navigated to the Sign In screen	Pass
IntroScreen 03: Navigate to Sign Up Page	Tap "Sign Up" text	Tap "Don't have account? Sign up"	User is navigated to the Sign Up screen	Pass

Table 3.1: Test Cases for SplashScreen and IntroScreen

5.4.2 Test Cases for Sign-In

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: Sign-In Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
SignIn 01: Successful Login	Valid email and correct password	Tap "Sign In"	User is logged in and redirected to the home page or dashboard	Pass
SignIn 02: Invalid Email Format	Invalid email (e.g., user.com)	Tap "Sign In"	Error message: "Please enter a valid email address"	Pass
SignIn 03: Incorrect Password	Valid email, wrong password	Tap "Sign In"	Error message: "Incorrect email or password"	Pass
SignIn 04: Empty Fields Submission	Leave email and password blank	Tap "Sign In"	Error message: "Email and password are required"	Pass
SignIn 05: Navigate to Sign Up	Tap "Don't have an account? Sign up"	Tap link	User is redirected to the Sign-Up page	Pass

Table 3.2: Test Cases for Sign-In

5.4.3 Test Cases for Sign-Up

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: Sign-Up Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
SignUp 01: Successful Registration	Valid email and strong password	Tap "Sign Up"	Account is created, user redirected to Sign In page	Pass
SignUp_02: Existing Email	Email already registered	Tap "Sign Up" button	Show "Email already in use" error	Pass
SignUp 04: Invalid Email Format	Invalid email	Tap "Sign Up" button	Error message: "Enter a valid email address"	Pass
SignUp_03: Weak Password	Password less than 6 characters	Tap "Sign Up" button	Show "Password too weak" message	Pass
SignUp_04: Empty Fields	Leave some fields empty	Tap "Sign Up" button	Show "Required field" validation messages	Pass
SignUp 06: Navigate to Sign In	Tap "Already have an account? Sign in"	Tap link	User is redirected to the Sign In page	Pass

Table 3.3: Test Cases for Sign-Up

5.4.4 Test Cases for User dashboard

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: Sign-Up Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
Dashboard_ 01: Load UI	User logged in	Open User Dashboard	Show dashboard with all feature buttons	Pass
Dashboard_ 02: Navigate to Stations	Tap "Locate Donation Station and Bins"	Tap button	Navigate to Donation Station and Bins page	Pass
Dashboard_ 03: Open Event Promotions	Tap "Event Promotion"	Tap button	Navigate to Event Promotion page	Pass
Dashboard_ 05: View Corporate Events	Tap "Submit and Promote Your Event"	Tap button	Navigate to Corporate Event Promotion page	Pass
Dashboard_ 05: Profile	Tap Profile Icon on Top-left side	Tap Profile Icon	Navigate to User Profile page	Pass
Dashboard_ 07: Logout	Tap logout button/icon on bottom-middle side	Tap button	Navigate to Sign-In screen	Pass

Table 3.4: Test Cases for User dashboard

5.4.5 Test Cases for User Profile

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: User Profile Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
Profile_01: Load Profile	User logged in	Open Profile Page	Show current user details (email, phone, etc.)	Pass
Profile_02: Edit Phone	New valid phone number	Tap "Save/Update"	Phone number updated	Pass
Profile_03: Username	New valid username	Tap "Save/Update"	Username updated	Pass
Profile_04: Logout	Tap logout button/icon	Tap button	Navigate to Sign-In screen	Pass

Table 3.5: Test Cases for User Profile

5.4.6 Test Cases for Locate Donation Station and Bins Page

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: Locate Donation Station and Bins Page Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
Station and Bins 01: Load Page	Open page	Load page	Show list of donation stations and bins	Pass
Station and Bins 02: Search Station	Station or Bins name/area	Enter in search bar	Filter results based on search	Pass
Station and Bins 03: locate button	Tap locate button	Tap on locate button	Show station on Google Map	Pass
Station and Bins 04: More details	Tap More details button	Tap on more details button	Show the details of donation station	Pass
Station and Bins 04: Bins level status	Bin data with different fill percentages	Load page and view bin level status	Show bin fill level in percentage; bins over 80% are highlighted in red	Pass

Table 3.6: Test Cases for Locate Donation Station and Bins Page

5.4.7 Test Cases for Event Promotion Page

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: Event Promotion Page Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
Event Promo 01: Load Event Promotion Page	Open page	Navigate to Event Promotion page	Display list of upcoming events with event name, start date, location, and description	Pass
Event Promo 02: Scroll Event List	Scroll input (touch)	Scroll down event list	List scrolls smoothly and dynamically loads more events if available	Pass
Event Promo 03: View Event Details	Tap "More Details" on any event card	Tap More Details button	Navigate to Event Detail page showing full event info, including sub-description, start/end date, organization name (if applicable)	Pass
Event Promo 04: Join Event Button	Tap Join Event	Tap Join Event on Event Detail page	User joined the event successfully, confirmation shown	Pass
Event Promo 05: Locate Button	Tap Locate	Tap Locate on Event Detail page	Open Google Maps with event location	Pass

Event Promo 06: View Event History	Tap History button (top-right corner)	Navigate to Event Promotion History page	Show list of expired events with event name, date, location, and brief description	Pass
Event Promo 07: View	Tap "More Details" on expired event	Tap to open Event Details for past event	Show full details with "Event Ended" label	Pass
Event Promo 08: Locate Expired Event	Tap Locate on expired event	Tap Locate button	Open Google Maps with past event location	Pass
Event Promo 09: Organization Visibility	View event with hidden organizer name	Open details of event where organizer chose to hide name	Organization name not displayed (hidden if opted)	Pass
Event Promo 10: UI Button Visibility Logic	Event is expired	Open expired event detail page	"Join Event" replaced with "Event Ended" label; Locate button still active	Pass

Table 3.7: Test Cases for Event Promotion Page

5.4.8 Test Cases for Submit and Promote Your Event Page

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: Submit and Promote Your Event Page Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
SubmitEvent 01: Load Submit and Promote Event Page	Open page	Navigate to Submit & Promote Event section	Show list of previously submitted events with name, location, start date, and status badge	Pass
SubmitEvent 02: Status Badge - Pending	Event under review	View pending event in list	Show "Pending" badge with pending reason (e.g., "Waiting for email reply")	Pass
SubmitEvent 03: Status Badge - Approved	Approved event	View approved event in list	Show "Approved" badge without any pending/rejection reason	Pass
SubmitEvent 04: Status Badge - Rejected	Rejected event	View rejected event in list	Show "Rejected" badge with rejection reason (e.g., "Incomplete details")	Pass
SubmitEvent 05: Tap Submit New Event Button	Tap floating button	Tap "Submit New Event" button	Navigate to Event Submission Form page	Pass
SubmitEvent 06: Submit Form with Complete Details	Fill all required fields correctly	Tap Submit button	Show confirmation message; event forwarded for admin review	Pass

SubmitEvent 07: Submit Form with Missing Required Fields	Leave required fields blank (e.g., phone number)	Tap Submit button	Show validation error messages; form not submitted	Pass
SubmitEvent 08: Organization Visibility Option – Yes	Select “Yes” to show org name	Submit form	Organization name will be publicly visible on promoted event	Pass
SubmitEvent 09: Organization Visibility Option – No	Select “No” to hide org name	Submit form	Organization name will be hidden on promoted event	Pass
SubmitEvent 10: Data Format Validation	Enter invalid email or phone	Fill form with wrong format	Show validation error message (e.g., “Invalid email format”)	Pass
SubmitEvent 11: Back Navigation	Tap back or cancel while filling form	Navigate back to previous screen	Return to previous page without submitting form	Pass
SubmitEvent 12: Retain Previously Submitted Events	After submission or refresh	Re-open page	Previously submitted events still listed with updated statuses	Pass

Table 3.8: Test Cases for Submit and Promote Your Event Page

5.4.9 Test Cases for admin Dashboard

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: Corporate Event Promotions Page Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
AdminDashboard 01: Admin Login and Access Dashboard	Valid admin credentials	Login and navigate to Admin Dashboard	Dashboard is displayed with admin tools (bins, stations, events)	Pass
AdminDashboard 02: Alert for Bins Over 80%	Bin fill level > 80%	Open dashboard	Pop-up alert appears showing which bins are over 80%	Pass
AdminDashboard 03: Notification Bins Badge	1 or more bins > 80%	View Notification Bins icon	Red badge (e.g., "1") shows on bin icon	Pass
AdminDashboard 04: Pop-Up Alert for Bin Collection	Bin recently collected and level = 0%	Load dashboard	Pop-up confirms successful bin collection	Pass
AdminDashboard 05: View All Core Features	Navigate through tabs/buttons	Open each module	Able to access Notification Bins, Add Station, Add Bin, Approve Events	Pass

Table 3.9: Test Cases for admin Dashboard

5.4.10 Test Cases for Notifications bins

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: Notifications bins Page Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
NotifBins 01: Load Notification Bins Page	Open Notification Bins tab	Navigate to page	Display all registered bins with name, location, distance, and bin level status	Pass
NotifBins 02: Bin Level Over 80%	Bin level = 85%	View bin entry	Status bar turns red to indicate urgency	Pass
NotifBins 03: Locate Button Functionality	Tap Locate button	Tap on Locate next to any bin	Open Google Maps with bin's location	Pass
NotifBins 04: Search Bin by Name or Area	Enter bin name or area in search bar	Search	Filtered results show matching bin(s)	Pass
NotifBins 05: Pop-Up Alert – Bin Overfilled	Bin level exceeds 80%	Open page / refresh	Pop-up alert appears with “Attention” and list of affected bins	Pass
NotifBins 06: Pop-Up Alert – Bin Collected	Bin level drops from >80% to 0–20%	Load page	Pop-up message appears with “Bin Collected” notification	Pass
NotifBins 07: Distance Display	Admin has location services enabled	View bin entries	Each bin displays estimated distance from current admin location	Pass

NotifBins 08: Scroll Through Bin List	Scroll down on page	Scroll through bin entries	List scrolls smoothly and loads additional bin entries if available	Pass
NotifBins 09: No Matching Search Result	Enter invalid bin name	Search	Show "No bins found" message or empty list	Pass
NotifBins 10: Verify Real-Time Status Update	Bin fill level changes in backend	Refresh Notification Bins page	Updated level status is reflected on screen immediately	Pass
NotifBins 11: Multiple Overfilled Bins	2 bins exceed 80%	Open page	Pop-up lists all 2 bins needing attention; all marked in red	Pass

Table 3.10: Test Cases for Notifications bins

5.4.11 Test Cases for Add Donation Station Page

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: Add Donation Station Page Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
AddStation 01: Load Add Donation Station Page	Open page	Navigate to Add Donation Station	Display station input form with all fields	Pass
AddStation 02: Add Valid Station	Fill all required fields correctly	Submit form	Station added successfully, confirmation message shown	Pass
AddStation 03: Submit with Empty Fields	Leave one or more fields blank	Submit form	Validation error shown; form not submitted	Pass
AddStation 04: Invalid Email Format	Enter wrongly formatted email	Submit form	“Invalid email format” error shown	Pass
AddStation 05: Navigate Back	Click back or while filling	Leave the page	Return to previous screen without saving	Pass

Table 3.11: Test Cases for Add Donation Station Page

5.4.12 Test Cases for Add Donation Bins Page

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: Add Donation Bins Page Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
AddBin 01: Load Add Donation Bins Page	Open page	Navigate to Add Donation Bins	Display bin input form with required fields	Pass
AddBin 02: Add Valid Bin	Fill in bin name, address, and coordinates	Submit form	Bin added successfully, confirmation shown	Pass
AddBin 03: Submit with Missing Fields	Omit bin name or coordinates	Submit form	Show validation error, prevent submission	Pass
AddBin 04: Verify Real-Time Bin Update	Input bin with same name & location	Submit form	Bin is displayed instantly in list	Pass
AddBin05: Navigate Back	Click back or while filling	Leave the page	Return to previous screen without saving	Pass

Table 3.12: Test Cases for Add Donation Bins Page

5.4.13 Test Cases for Approve Event Promotion Page

Test Case	Input	Action	Expected output	Actual Test Result
Test Case Name: Approve Event Promotion Page Created by: YAP WING TOM (81529) Date created: 2025-06-15 Tested by: YAP WING TOM (81529)				
AEP 01: Load Approve Event Promotions Page	Open page	Navigate to Approve Event Promotion section	Display table with event name, location, org name (if applicable), and More Details button	Pass
AEP 02: Filter Approved Events	Tap "Approved Event" button	Switch view	Show only approved events with event name, location, and approval date	Pass
AEP 03: Filter Rejected Events	Tap "Rejected Event" button	Switch view	Show only rejected events with event name, location, rejected reason, and rejected date	Pass
AEP 06: Open Event Details Panel	Tap "More Details" button	View event detail	Display full event name, location, descriptions, date/time, and contact section	Pass
AEP 07: Org Name Flag = Yes	View details	If "Show org name" = Yes	Display "Organized by: [Organization Name]" under event name	Pass

AEP 08: Org Name Flag = No	View details	If “Show org name” = No	Do not show org name in the detail view	Pass
AEP 09: View Contact Information	View detail page	Scroll to Submitter Contact section	Display user and organization contact details (phone and email)	Pass
AEP 10: Approve Event	Click Approve button	Confirm action	Event marked as Rejected, reason saved, appears in Rejected list	Pass
AEP 11: Reject Event with Reason	Type reason + click Reject	Confirm action	Event marked as Rejected, reason saved, appears in Rejected list	Pass
AEP 12: Reject Event without Reason	Leave reason blank + click Reject	Submit	Show validation error: “Rejection reason required”	Pass
AEP 13: Mark as Pending with Reason	Type reason + click Pending	Save	Event remains in Pending state, reason saved in Pending Reason field	Pass
AEP 14: Mark as Pending without Reason	Leave reason blank + click Pending	Save	Event remains in Pending state, no reason shown	Pass
AEP 15: Check Approved Event Record	Event marked as Approved	View Approved list	Event shown with name, location, and approval date	Pass
AEP 16: Check Rejected Event Record	Event marked as Rejected	View Rejected list	Event shown with name, location, rejected reason and date	Pass

AEP 17: Navigate Between Lists	Tap buttons Approved Event or Rejected Event	Switch view	Switches between categorized event lists	Pass
--------------------------------	--	-------------	--	------

Table 3.13: Test Cases for Approve Event Promotion Page

Test Case comprehensively documents the test cases designed to validate the core functionalities of the donation platform. The test cases span across both user-facing and admin-facing features, ensuring the system performs reliably, securely, and as expected. The major components tested include:

- Splash and Intro Screens – Confirmed proper display and navigation from app launch to authentication.
- Sign In and Sign Up – Verified user authentication workflows, input validation, and navigation between login and registration pages.
- Locate Donation Station and Bins – Tested searching, locating, and viewing details of donation stations and bin fill levels, including alerts for bins exceeding 80%.
- Event Promotion – Ensured that upcoming and historical donation events are listed correctly, and users can view event details or join events.
- Submit and Promote Event – Tested the submission form for external organizations to propose events, including admin review status visibility.
- Admin Dashboard – Verified all core admin operations such as adding stations and bins, receiving bin alerts, and tracking maintenance.
- Notification Bins – Ensured real-time bin monitoring and notification handling for overfilled and emptied bins.
- Add Donation Station and Bins – Tested accurate registration of new locations and bin data into the system.
- Approve Event Promotions – Confirmed admin review processes including approval, rejection, and pending status with contact traceability.

These test cases play a critical role in maintaining high-quality standards for the application, facilitating both operational efficiency for administrators and an engaging experience for users. They help to ensure that all features function seamlessly and that the system supports both user-driven and admin-controlled workflows effectively.

5.5 Summary

Chapter 5 presents a comprehensive overview of the testing phase conducted to ensure the functionality, performance, and security of the donation management application. It begins with an introduction to the overall testing objectives, followed by detailed descriptions of functional and non-functional testing. Functional testing verifies that the app behaves correctly across various features, while non-functional testing explores the app's performance under real-world conditions. This includes testing on a physical device (OPPO Reno5 F), cloud-based testing through Firebase Test Lab, applying Firebase Security Rules to protect data access, and using Dart Analyze to detect code issues and enforce best practices.

The chapter then outlines a structured set of test cases that evaluate both user and admin functionalities. These cover user sign-in and sign-up processes, user dashboards, event promotions, bin monitoring, and location services. Admin-related test cases focus on critical operations like managing donation stations and bins, approving event promotions, and responding to bin status notifications. Each test case ensures proper validation, UI responsiveness, and feature accuracy, reinforcing the application's usability and reliability. Overall, Chapter 5 highlights the thorough testing process that ensures the system's quality, user satisfaction, and secure data management.

CHAPTER 6

Conclusion and Future Work

6.1 Introduction

The development of the Garment Donation App aimed to create a user-friendly platform that bridges the gap between individuals who want to donate garments and donation stations or organizers who manage donation events. With core functionalities such as user authentication, locating donation stations and bins, and promoting events—especially corporate ones—the app provides a centralized solution for managing and participating in donation-related activities. By integrating Firebase services such as Authentication, Firestore, and Realtime Database, the app ensures a seamless backend infrastructure, while Flutter offers a consistent and responsive frontend experience. The app also supports admin features for reviewing and approving event promotions, thereby maintaining content quality and control.

Throughout the development process, emphasis was placed on clean UI design using a white and grey theme, intuitive navigation, and basic error handling. Various modules were tested to confirm their correct behavior, and the app was refined through multiple iterations and user feedback. The result is a functional application that supports both donors and administrators in achieving efficient garment donation management. However, as with any software project, there are areas that remain to be enhanced, and a few limitations were identified during testing and usage.

6.2 Achievement of Objectives

Original Objective	Status & Evidence of Completion
Objective 1 – Implement IoT sensors that monitor donation-bin capacity in real time and notify administrators when bins are full or require attention.	Achieved. Dual HC-SR04 ultrasonic sensors mounted on each side of the bin stream percentage-full data to Firebase Realtime Database every 15 s. The Admin “Notification Bins” page highlights bins $\geq 80\%$ in red and raises a pop-up alert; a second pop-up announces when the same bin level drops to 0 %, confirming collection.
Objective 2 – Develop a mobile app that allows users to locate nearby donation stations, view operating hours, and receive staff verification details.	Achieved. The Locate Donation Station module uses geolocation, distance sorting, and a card-based UI (white/grey scheme) to list stations with name, address, distance, opening hours, contact phone and e-mail. A Locate button deep-links to Google Maps for real-time navigation.
Objective 3 – Enable the app to push notifications to users about upcoming donation events or promotional activities.	Achieved. Within the Event Promotion module, users can browse events; expired events are labelled “Event Ended” in Event Promotion History page.

Table 4.1: Achievement of Objectives

The system’s development journey successfully aligned with the outlined objectives, combining both hardware and software innovations to address real-world donation logistics. The first objective was met through the integration of IoT hardware—specifically, the Arduino Nano 33 IoT and dual HC-SR04 ultrasonic sensors—which enabled accurate, periodic bin fill level measurements (Pratama & Harijono, 2024). These readings were transmitted to Firebase Realtime Database and dynamically reflected in the admin panel. The Notification Bins interface not only displayed the bin status with colored indicators (e.g., red for $\geq 80\%$) but also provided instant visual alerts when bins were full or emptied, enhancing admin responsiveness.

The second objective centered on creating a practical and user-friendly mobile application using Flutter. This app successfully allows users to locate donation stations based on their geographic position, view important details such as contact information and operational hours, and navigate directly to the station via deep linking to Google Maps. The station and bin cards were developed with a clean and intuitive UI/UX design using a white

and grey theme, which supports clarity and ease of use. Additionally, basic user profile management and authentication were incorporated using Firebase Authentication and Firestore.

For the third objective, a foundational structure for event promotions and notifications was implemented. Users are able to browse upcoming donation events, access details, and view their status (e.g., "Event Ended" for past events). For the current release, in-app updates are achieved through Firebase real-time listeners and Flutter state management, providing near-instantaneous UI updates when events are added, approved, or modified. Together, these implementations demonstrate significant progress toward the project's vision of a connected, responsive, and community-oriented donation ecosystem.

6.3 Limitations

Despite the system meeting its core objectives and delivering the expected functionalities, several limitations were identified during development and testing phases that could impact scalability, user experience, and real-world deployment.

6.3.1 Internet Dependency for IoT and Mobile App

Both the Arduino Nano 33 IoT hardware and the mobile application rely heavily on a stable internet connection to function correctly. The Arduino Nano 33 IoT uses Wi-Fi connectivity (limited to 2.4GHz networks), which may not be compatible with some modern routers or environments that primarily use 5GHz. Additionally, when used in the field with mobile data hotspots, the board occasionally experiences unstable connections. This affects real-time data transmission to Firebase Realtime Database, potentially delaying bin status updates. Similarly, the mobile application depends on Firebase services—including Realtime Database, Authentication, and Firestore (see Sections 4.4.1.1 and 4.4.3)—for core features like bin monitoring, event promotion, and profile access. Without internet access, users are unable to locate donation stations, submit events, or receive notifications, which significantly limits functionality in low-connectivity areas.

6.3.2 Image Loading Issues

During development, especially while using hot reload features in Flutter, image assets occasionally fail to load properly from the local assets folder. Although this issue may be minimized in the final release build, it presents challenges during testing and UI validation (as observed in Section 5.3 testing results).

6.3.3 Lack of Offline Support

The current version of the app does not support offline caching or data access. All content, including donation station listings, bin levels, and event information, requires an active internet connection to be retrieved from Firebase. This limits accessibility and usability in rural or offline scenarios where connectivity is unreliable, as noted in both functional and non-functional testing phases (see Sections 5.2 and 5.3.2).

6.3.4 Limited Validation and User Feedback

While basic validation exists on forms such as sign-up, event submission, and profile updates, the system lacks more sophisticated error handling. Feedback for invalid or incomplete inputs is minimal, which may confuse users or lead to submission errors. Improved

real-time feedback and field-specific validation are needed for better user interaction, especially in the event submission module (refer to Section 5.4.8).

6.3.5 Simplified Admin Workflow

Although the admin dashboard (Section 5.4.9) includes basic management tools like adding stations, viewing bin statuses, and approving events, it lacks key productivity features. For example, there is no centralized dashboard overview, alert notification system, or summary cards to show pending tasks. These limitations may slow down decision-making, especially when managing multiple user submissions or bins in real time.

6.3.6 Limited User Profile Features

The user profile functionality (Section 5.4.5) currently captures only basic user data such as name, email, and phone number. There is no support for profile image uploads, donation activity logs, or customizable account settings, which could enhance user engagement and personalization. Expanding this feature would contribute to a more complete and engaging user experience.

6.3.7 Firebase Plan Limitations (FCM and Storage Not Implemented)

Features such as Firebase Cloud Messaging (FCM) for real-time notifications and Firebase Storage for uploading images were not implemented due to limitations of the free Firebase plan. These services require an upgraded (Blaze) plan for full access, including secure media hosting and background notification capabilities. As a result, the current system lacks real-time alerts for users and cannot support dynamic image uploads for events or profiles. These features are planned for future implementation once an upgraded plan becomes available.

6.3.8 Limited Adoption & Community Engagement

As a newly developed system, the platform is not widely recognized or adopted by the public or local organizations. This can limit its effectiveness in promoting donation events or encouraging regular community use. Since the system currently lacks widespread visibility or integration with local councils or NGOs, it may take time and effort to build trust and demonstrate value before it becomes a preferred tool for garment donation management or event promotion.

6.3.9 Justification of Hardware Use in Certain Areas

While the use of IoT sensors (ultrasonic) enables real-time bin level monitoring, in some low-traffic or rural areas, deploying such hardware may not be necessary or cost-effective. Certain locations may not generate enough donation activity to warrant sensor-based tracking.

Therefore, future deployments should consider scalability and location-specific needs before investing in sensor hardware.

Overall, the system faces several key limitations that impact its usability, scalability, and deployment in real-world environments. Both the Arduino Nano 33 IoT and the mobile application are highly dependent on stable internet connectivity, with the hardware limited to 2.4GHz Wi-Fi and the app reliant on Firebase services, making offline use currently unsupported. Development challenges such as occasional image loading failures, minimal input validation, and limited user feedback can affect the overall user experience. Additionally, user profile features are basic, lacking customization options like profile pictures or donation history tracking. These limitations must be addressed to enhance performance, reliability, and long-term user engagement.

6.4 Future Improvements

To enhance the functionality, reliability, and scalability of the Garment Donation System, several future improvements are proposed. These enhancements aim to address current limitations, improve real-world performance, and offer a more robust, user-friendly, and impactful solution for both users and administrators.

6.4.1 Enhanced Arduino-Based Hardware Setup

Currently, the system uses a single Arduino Nano 33 IoT board, two HC-SR04 ultrasonic sensors, LED indicators, a breadboard, and jumper wires. While functional for testing, future versions should consider expanding the setup by integrating multiple Arduino boards for multi-location deployments, waterproof ultrasonic sensors for outdoor use, and enclosures for environmental protection. Additionally, replacing the breadboard setup with a custom PCB (Printed Circuit Board) will improve durability and reduce loose connections in long-term installations. Power reliability can be enhanced using battery-backed modules or solar-powered solutions, ensuring the system remains operational in remote or unpowered locations.

6.4.2 Robust Image and Asset Handling

Currently, image assets are loaded locally due to cost constraints and the limitations of the Firebase free tier. Firebase Storage was not used because it requires upgrading to a paid plan to enable secure and scalable image hosting and file management. In future iterations, Firebase Storage can be integrated to allow users to upload profile pictures, event images, and promotional content. This transition will resolve image loading issues and improve the flexibility of media handling.

6.4.3 Offline Mode and Local Caching

Currently, the application relies heavily on internet connectivity to fetch and update data from Firebase. Implementing offline capabilities using local storage (e.g., SQLite or shared preferences) and real-time synchronization upon reconnection would improve accessibility, especially in areas with unstable or limited internet coverage. This will allow users to view previously loaded content such as donation stations or event details even when offline.

6.4.4 Improved Admin Dashboard with Analytics

The existing admin panel (as detailed in Section 5.4.9) provides only basic moderation features. Future improvements should include dashboard widgets, summary analytics, bin fill level charts, and event submission alerts to enhance real-time decision-making. This will streamline admin workflows and provide a clearer overview of system status and user activity.

6.4.5 Push Notification Integration (FCM)

Firestore Cloud Messaging (FCM) was not implemented in the current version due to the free tier's limitations, which restrict background notifications and advanced targeting features. In future versions, once the plan is upgraded, FCM will be integrated to notify users about event approvals, donation reminders, bin overflow alerts, and more. This will significantly increase user engagement and system interactivity.

6.4.6 Advanced User Profile Features

The user profile feature (Section 5.4.5) could be extended to include profile picture uploads, donation history tracking, contribution badges, and account settings for better personalization. These enhancements can promote user retention and increase their sense of contribution within the app ecosystem.

6.4.7 Event Categorization and Scheduling Tools

Event organizers would benefit from additional tools such as tag-based categorization (e.g., charity, corporate, community), event calendar integration, and RSVP functionality. This would streamline event planning, increase participation accuracy, and improve coordination.

6.4.8 Enhanced Map and Location Features

The current map features could be expanded by including real-time bin fill levels, color-coded markers, routing directions, and distance-based filtering. Additionally, clustering nearby bins can declutter the interface and enhance usability in dense urban areas.

6.4.9 Accessibility and Multilingual Support

To cater to a wider audience, the application should implement accessibility options such as voice support, larger text options, and high-contrast themes. Furthermore, introducing multilingual interfaces will allow non-English-speaking users to interact comfortably with the system.

6.4.10 Security Enhancements and Role-Based Access

To strengthen system security, especially in multi-role environments, future updates should implement role-based access control (RBAC), allowing different access levels for users, event organizers, station managers, and admins. This would prevent unauthorized access and streamline permission management.

6.4.11 Data Analytics and Reporting Tools

A dedicated analytics module will be developed to generate reports on event participation, bin usage trends, and donation statistics. This data will be valuable for NGOs, corporate partners, and government stakeholders looking to evaluate community engagement and optimize donation operations.

6.4.12 Adoption Campaigns and Outreach

To increase platform visibility and encourage widespread usage, future work will focus on organizing adoption campaigns in collaboration with local communities, non-governmental organizations (NGOs), and universities. These partnerships aim to promote the system through awareness initiatives, donation drives, and pilot programs within real-world environments. By engaging directly with potential users and stakeholders, the system can gather valuable feedback to improve its features, usability, and relevance. Furthermore, such outreach efforts foster greater public trust and support, especially among socially responsible organizations and institutions. This approach will not only enhance the impact of the system but also support long-term sustainability by integrating it into community-based environmental and charitable efforts.

To further enhance the functionality, scalability, and user experience of the IoT-Based Smart Garment Donation Station System, several key improvements are proposed. These include upgrading the hardware with multiple Arduino boards, waterproof sensors, and custom PCBs for durability; integrating Firebase Storage for image hosting; and adding offline capabilities for better accessibility. The admin dashboard can be expanded with analytics tools, bin tracking charts, and real-time alerts. Future updates will also include advanced user profile features, multilingual support, map enhancements, and push notifications through Firebase Cloud Messaging. Role-based access control and data analytics modules will strengthen security and improve insight into donation trends. Additionally, community adoption campaigns with NGOs and universities will help promote system usage, gather feedback, and ensure sustainability across real-world environments.

6.5 Summary

Chapter 6 provides a comprehensive reflection on the outcomes, challenges, and future directions of the Garment Donation System project. The chapter begins by reaffirming the core goal of developing a user-friendly platform that integrates IoT-based bin monitoring with a mobile app for donation station location and event engagement. It highlights the successful achievement of all main objectives, including real-time bin data transmission via Arduino Nano 33 IoT, user interaction through a Flutter-based mobile interface, and event promotion tools supported by Firebase backend services (Bhagat, 2022). These elements collectively demonstrate a well-integrated system that meets functional expectations for both users and administrators.

Despite these achievements, several limitations were identified, including the system's dependency on stable internet connections, image loading issues, and a lack of offline support. Limitations in form validation, simplified admin tools, and minimal user profile features also highlight areas for refinement. Additionally, Firebase Cloud Messaging and Firebase Storage were excluded due to the constraints of the free Firebase plan, though they are proposed for future versions. The chapter concludes with detailed future improvement plans, such as upgrading the hardware setup, implementing local caching, enhancing the admin dashboard, integrating FCM, and introducing role-based access control and multilingual support. These enhancements aim to make the system more scalable, accessible, and effective for real-world deployment.

References

- Smart textile recycling and efficient clothing banks*. (2023, February 10). Greener and Smarter Waste - Nordsense. <https://nordsense.com/clothing-banks/>
- Most charity fraud perpetrated by staff or volunteers, report suggests*. (n.d.). <https://www.civilsociety.co.uk/news/most-charity-fraud-perpetrated-by-staff-or-volunteers-report-suggests.html>
- H. W. Prajitno, M. B. Kristanda and A. Kusnadi, "Study of Push and Pull Techniques in Delivering Mobile Application Notifications," 2019 5th International Conference on New Media Studies (CONMEDIA), Bali, Indonesia, 2019, pp. 20-25, doi: 10.1109/CONMEDIA46929.2019.8981856.
- K. V. S. Krishna, P. V. I. Bhanu and B. Valarmathi, "Smart donation machine — Automation of donation process using arduino," 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2017, pp. 1222-1226, doi: 10.1109/ICCSP.2017.8286574.
- D. Sehrawat and N. S. Gill, "Smart Sensors: Analysis of Different Types of IoT Sensors," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 523-528, doi: 10.1109/ICOEI.2019.8862778.
- R. Deekshath, P. Dharanya, K. R. D. Kabadia, G. D. Dinakaran and S. Shanthini, "IoT Based Environmental Monitoring System using Arduino UNO and Thingspeak", *IJSTE - International Journal of Science Technology Engineering*, vol. 4, no. 9, 2018.
- A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, "Internet of Things for Smart Cities," in *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22-32, Feb. 2014, doi: 10.1109/JIOT.2014.2306328.
- S. Zafar, G. Miraj, R. Baloch, D. Murtaza and K. Arshad, "An IoT Based Real-Time Environmental Monitoring System Using Arduino and Cloud Service", *Engineering Technology Applied Science Research*, vol. 8, no. 4, pp. 3238-3242, 2018.
- G. Pandey and A. Kumar, "An Examination on Food, Clothes and Books Donation Based Android Application," 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), Mandya, India, 2022, pp. 1-6, doi: 10.1109/ICERECT56837.2022.10059757.

- Satam, A., Gavli, N., & Arora, B. (2021). Application for donation centers. *International Journal of Advance Research, Ideas and Innovations in Technology*, 7(4), 741–744.
<https://www.ijariit.com/manuscripts/v7i4/V7I4-1454.pdf>
- Setiawan, D., & Hariono, T. (2024). Design of a Smart Donation Box Based on IoT. *NEWTON: Networking and Information Technology*, 3(2), 43–53.
<https://doi.org/10.32764/newton.v3i2.4951>
- T. Li, D. Hu, M. Li, Y. Li and S. Zheng, "A Blockchain-based Material Donation Platform," 2022 International Conference on Blockchain Technology and Information Security (ICBCTIS), Huaihua City, China, 2022, pp. 246-254, doi: 10.1109/ICBCTIS55569.2022.00061.
- Dr. Sudha L K , Mohammed Tahir , Ruchitha E , Varshini R, Syed Saif Ali, 2023, Smart Dustbin Monitoring System using Arduino UNO, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 12, Issue 02 (February 2023)
- Howell, P. (2013, April 25). *Is your neighborhood donation bin laundering your clothing to for-profit retailers?* Business of Story | Storytelling Strategy, Workshops & Keynotes.
<https://businessofstory.com/is-your-neighborhood-donation-bin-laundering-your-clothing-to-for-profit-retailers/>
- Pratama, A. W., & Harijono, N. A. (2024). The analysis of ultrasonic sensors in smart trash bin technology based on Arduino nano microcontroller. *Deleted Journal*, 1(3).
<https://doi.org/10.47134/jme.v1i3.3206>
- Payne, R. (2019). *Beginning App Development with Flutter: Create Cross-Platform Mobile Apps*. <https://www.amazon.com/Beginning-App-Development-Flutter-Cross-Platform/dp/1484251806>
- Nanavati, J., Patel, S., Patel, U., & Patel, A. (2024). Critical Review and Fine-Tuning Performance of Flutter Applications. *2024 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, 838-841, 838–841.
<https://doi.org/10.1109/icmcsi61536.2024.00131>
- Bhagat, S. A. (2022). Review on mobile application development based on Flutter platform. *International Journal for Research in Applied Science and Engineering Technology*, 10(1), 803–809. <https://doi.org/10.22214/ijraset.2022.39920>

- Madaminov, U. A., & Allaberganova, M. R. (2023). Firebase Database Usage and Application Technology in Modern Mobile Applications. *2023 IEEE XVI International Scientific and Technical Conference Actual Problems of Electronic Instrument Engineering (APEIE), 1690-1694*, 1690–1694. <https://doi.org/10.1109/apeie59731.2023.10347828>
- Kunitsyn, V., & Fedorov, V. (2025). FIREBASE (REALTIME DATABASE) MOBILE AND WEB APPLICATION DEVELOPMENT PLATFORM. Materials of the All-Russian Scientific and Practical Conference «CHALLENGING ISSUES IN SYSTEMS MODELING AND PROCESSES», 349–352. https://doi.org/10.58168/cismp2024_349-352