



Module Code & Title

CS6P05 Final Year Project MAD
Food Share – Android App

Assessment Weightage & Type

30% FYP Interim Report

Supervisor By

Mr. Shekhar Timsina
Mr. Bishal Gharti Chhetri (GC)

Student Details

Name: Sita Ram Thing
London Met Id: 22015892
College Id: NP01MA4S220003
Islington College, Kathmandu
03 Jan 2024

Git Hub: [FYP-23-24/22015892-SitaRamThing \(github.com\)](https://github.com/FYP-23-24/22015892-SitaRamThing)

Abstract

This report illustrates the mid-term development of the Weaste Food Management system with food donations and the distribution system based on the Android Mobile Application. Many hotel businesses, party palaces and farmers still rely on food dumps, but the homeless or poor people face many problems because of it, this project aims to solve those problems and make the food donation system even more efficient and helpful. After the completion of this project, there would be a system that food donations in real life and through the donation system food are picked up and distributed.

Contents

1.	Introduction.....	1
1.1.	Project Introduction	1
1.2.	Problem as Scenario.....	2
1.3.	Project as Solution	3
1.4.	Aims and Objectives	4
1.4.1.	Aim	4
1.4.2.	Objectives.....	4
2.	Background	5
2.1.	Technology and Tools Used	5
2.1.1.	Technology	5
2.1.2.	Tools.....	7
2.2.	Methodology	9
2.2.1.	Methodology Consideration	9
2.2.2.	Methodology Selection	15
2.2.3.	Methodology Section Justification.....	16
2.3.	Similar Systems.....	23
2.3.1.	Similar System Consideration.....	23
2.3.2.	Similar System Comparison.....	27
3.	Work to Date	28
3.1.	Requirement Gathering	29
3.1.2.	Feature List (As per systems and roles)	34
3.2.	Use Case.....	36
3.2.1.	High-Level Use Case	37
3.3.	Sequence Diagram	41
3.3.1.	View profile sequence diagram.....	41
3.3.4.	Donation rating sequence diagram	44

3.3.5. Complaint with admin	45
3.4. Context Diagram	46
3.5. Data Flow Diagram Level -1.....	47
3.6. Activity Diagram	48
3.6.1. Take Membership Activity.....	48
3.6.2. Donation Activity	49
3.6.3. View History Activity	50
3.6.4. Donation Rating Activity.....	50
3.6.5. Complaint with Admin Activity.....	51
3.7. Wireframes	52
3.7.1 Mobile UI.....	52
3.7.2. Web UI for Admin	66
3.8. ER-Diagram.....	68
3.9. System Architecture Diagram	69
3.10. Class Diagram.....	70
3.11. DFD's level 2	71
3.12. Milestone Chart	76
3.13. WBS	77
3.14. Gantt Chart.....	78
3.15. Analysis of Progress	81
3.15.1. Progress Table	81
3.15.2. Progress Review.....	82
4. Future Work.....	87
5. Conclusion.....	90
6. References	91
7. Appendix.....	92
7.1. Project as a Solution	92

7.2. Technology Use	93
7.2.1. Django Rest Framework	93
7.2.1. Kotlin jetpack compose.....	93
7.3.1. Draw.io.....	94
1.3.2. Luci Chart	95
1.3.3. Figma.....	96
7.4 Expanded Use Case	97
7.5 DFD Level 2	111
7.5.1. Register Details	111
7.5.2. Login System.....	112
7.6 Activity Diagram	113
7.6.1. Login Activity.....	113
7.6.2. Register Activity	114
7.6.3. Logout Activity	115
7.7 Wireframe UI	116
7.7.1 Splash screen	116
7.8 Sequence Diagram.....	123
7.9 Survey Form.....	125
7.10 Work to date	132
7.10.1. Backend.....	132
7.10.2. Frontend	136

List of tables

Table 1: Justification scenario 1 for waterfall methodology.....	16
Table 2: Justification scenario 2 for waterfall methodology.....	16
Table 3: Justification scenario 3 for waterfall methodology.....	17
Table 4: Justification scenario 1 for prototype methodology	18
Table 5: Justification scenario 2 for Prototype methodology	18
Table 6Justification scenario 3 for Prototype Methodology.....	19
Table 7: Justification scenario 1 for scrum methodology.....	20
Table 8Justification scenario 2 for scrum methodology	20
Table 9: Justification scenario 3 for scrum methodology.....	21
Table 10: Justification scenario 4 for scrum methodology	21
Table 11: Justification scenario 5 for scrum methodology	21
Table 12: Justification scenario 6 for scrum methodology	22
Table 13: Comparison with similar applications	27
Table 14: The client approaches the latter.....	31
Table 15: Register user high-level use case.....	37
Table 16: Take membership user high-level use case	38
Table 17: Login user high level use case	38
Table 18: Food donation high-level use case.....	39
Table 19: View donation info high-level use case.....	39
Table 20: Donation rating high-level use case	39
Table 21: View history high-level use case.....	40

Table 22: View profile high-level use case.....	40
Table 23: Complaint with high-level use case	40
Table 24: Progress Table.....	81
Table 25: Progress timeline in Table	84
Table 26: Plan due worktable.....	89
Table 27: Register user expanded use case table	98
Table 28: Table membership Expanded use case	99
Table 29: Take membership expanded use case table.....	100
Table 30: Login user expanded use case table	102
Table 31: Food donation expended use case diagram.....	103
Table 32: Food donation user expanded use case table.....	104
Table 33: View Donation history expended use case diagram.....	105
Table 34: View history user expanded use case table.....	106
Table 35: Donation rating expanded use case diagram.....	107
Table 36: Donating rating expanded use case table	108
Table 37: Complaint with admin expanded use case diagram	109
Table 38: Complaint with admin expanded use case table	110

List of Figures

Figure 1: MS-word tools	7
Figure 2: Balsamiq tool.....	8
Figure 3: Waterfall Methodology	9
Figure 4: Prototype Methodology	11
Figure 5: Scrum Methodology	13
Figure 6: Selected Scrum Methodology	15
Figure 7: FlashFood similar system.....	24
Figure 8: Food Rescue similar app.....	25
Figure 9: No Food Waste Similar System	26
Figure 10: Work to date durations	28
Figure 11: Survey form response list of name	32
Figure 12 Survey form response list of email	33
Figure 13: Use Case Diagram	36
Figure 14: View profile sequence diagram	41
Figure 15: View history sequence diagram	42
Figure 16: Food Post for donation sequence diagram.....	43
Figure 17: Donation rating sequence diagram	44
Figure 18: Complaint with admin sequence diagram	45
Figure 19: Context diagram.....	46
Figure 20: DFD Level-1	47
Figure 21: Take membership details DFD Level-2	71

Figure 22: Food donation DFD Level-2	72
Figure 23: View Donation History DFD Level-2.....	73
Figure 24: Donation rating DFD Level-2.....	74
Figure 25: Complaint with admin DFD Level-2.....	75
Figure 26: Take the membership activity diagram.....	48
Figure 27: Food donation system activity diagram	49
Figure 28: View the history system activity diagram.....	50
Figure 29: Rating donation system activity diagram	50
Figure 30: Complaint with admin system activity diagram.....	51
Figure 31: Walkthrough screen mobile UI	52
Figure 32: Welcome screen mobile UI	53
Figure 33: Login screen mobile UI	54
Figure 34: Register screen mobile UI	55
Figure 35: Forgot password screen mobile UI.....	56
Figure 36: Donor Home screen mobile UI	57
Figure 37: Donor History screen mobile UI	58
Figure 38: Donor Post/Donation screen mobile UI	59
Figure 39: Volunteer Home screen UI	60
Figure 40: Volunteer Pending screen UI.	61
Figure 41: Volunteer History Screen	62
Figure 42: NGO Home screen UI	63
Figure 43: NGO History screen UI	64
Figure 44: NGO Setting Screen UI.....	65

Figure 45: Admin login page web UI	66
Figure 46: Admin forgot password UI.....	66
Figure 47: Admin dashboard UI	67
Figure 48: Class Diagram.....	68
Figure 49: System Architecture Diagram.....	69
Figure 50: Class diagram of food donation application.....	70
Figure 51: Milestone Chart	76
Figure 52: Milestone Sheet	76
Figure 53: Work Breakdown Structure	77
Figure 54: Gantt chart	80
Figure 55: System development progress data Trello	82
Figure 56: Progress timeline Figure	85
Figure 57: Draw.io tools	94
Figure 58: Luci chart tools	95
Figure 59: Figma Tools	96
Figure 60: Register expanded use case.....	97
Figure 62: Register details DFD Level-2	111
Figure 63: Login details DFD Level-2.....	112
Figure 64: Login system activity diagram	113
Figure 65: Register system activity diagram.....	114
Figure 66: Logout system activity diagram	115
Figure 67: Splash screen mobile UI	116
Figure 68: Register response message mobile UI	117

Figure 69: Donor Profile screen mobile UI	118
Figure 70: Donor Setting screen mobile UI	119
Figure 71: Volunteer Setting screen UI	120
Figure 72: NGO Users screen UI	121
Figure 73: NGO Profile screen UI	122
Figure 74: Survey other response email.....	125
Figure 75: User age status in Pai chart	126
Figure 76: Select your occupation. response in chart	126
Figure 77: find the excess food that could be donated. response in a pie chart.....	127
Figure 78: participated in food donation? response in the pai chart	127
Figure 79: Not donation reason and why waste it in a dump? response in a pie chart	128
Figure 80: Donation options individual or through the organization? Response pie chart	128
Figure 81: Donation application rating? response in a pie chart.....	129
Figure 82: Ration for online food donation app? response in a pie chart	129
Figure 83: How to track the donation food where distributed? response in pain chart	130
Figure 84: Donation received update form app? response in a pie chart.	130
Figure 85: Suggestion for a food donation app? response in a pie chart	131
Figure 86: Database model development.....	132
Figure 87: Database table in MySQL	132
Figure 88: Rest API Url	133
Figure 89: Rest API login authentication to get access token	134
Figure 90: Login authentication API test.....	134

Figure 91: Access token to get the user profile	135
Figure 92: Login page web frontend development	136
Figure 93: Dashboard web frontend development	136
Figure 94: Reset password web frontend development	137
Figure 95: Mobile Walkthrough Screen 1	138
Figure 96: Mobile Walkthrough Screen 2	139
Figure 97: Mobile Walkthrough Screen 3	140
Figure 98: Mobile welcome screen.....	141
Figure 99: Mobile login screen	142
Figure 100: Mobile register screen.....	143
Figure 101: Mobile reset password screen.....	144
Figure 102: Mobile dashboard screen	145
Figure 103: Mobile Profile screen.....	146
Figure 104: Mobile setting screen	147

1. Introduction

Many countries are in the development phase, and in these countries, numerous people live without their basic needs being fulfilled due to a lack of jobs or income sources. Some individuals are homeless or impoverished, struggling to afford everyday meals. Nepal faces similar issues, especially in urban areas where unemployed individuals often go hungry. One potential solution to this problem involves collecting food waste to distribute among the homeless or impoverished, ensuring they do not remain hungry. This project can surplus food collected to distribute to poor people or homeless people. Many organizations and NGOs help waste food management by distributing programs for free.

1.1. Project Introduction

In a world where technology continues to transform various facets of our lives, it's not surprising that the realm of charitable food donation is also benefiting from these advancements. The Weaste Food Donation App, known as "Food Share," represents a significant step forward in leveraging technology to address the critical issue of food scarcity and waste. The Food Share app seeks to revolutionize the way we approach food donations and redistribution.

The client's name is Yuvraj Tamang, and he worked at ING Food Company. He provides daily lunch food supplies for workers in different companies like Innovate Tech, Virav, and more. He needs an online food donation application that can easily collect surplus food to donate to homeless or poor people. The system should solve the issue of managing waste food and help in preventing food wastage by managing food dumping. Recognizing these dual issues, the Food Share app aims to bridge the gap between surplus food sources and those in need, while also reducing food wastage. This innovative digital solution serves as a bridge between surplus food providers and those in need, leveraging technology to streamline the process of redistributing excess food to marginalized communities.

1.2. Problem as Scenario

In a global context, surplus food waste has become a glaring issue, with developed nations discarding substantial amounts of perfectly edible food daily. Supermarkets, restaurants, and households often dispose of significant quantities of food due to overproduction, expiration dates, or aesthetic imperfections. This surplus contributes to environmental degradation through methane emissions in landfills and squanders resources like water and land used for food production. Meanwhile, in many developing nations like Nepal, despite grappling with food insecurity among significant segments of their population, food waste is a prevalent issue as well. Factors such as inadequate infrastructure for storage and transportation, poor market linkages, and limited awareness contribute to substantial food wastage even as a considerable portion of the population faces hunger and malnutrition.

In Nepal, the proportion of the employed population below \$1.90 purchasing power parity/day in 2022 was 5.1%. Nepal, like many other nations, encounters this paradoxical scenario where surplus food, often deemed unfit for sale due to cosmetic imperfections or nearing expiration dates, ends up wasted while a significant number of people struggle to access nutritious meals. This surplus, if effectively managed, could alleviate hunger and malnutrition in communities where access to food is a daily challenge. Addressing this issue requires concerted efforts to implement efficient distribution channels, educate communities about food preservation and utilization, and create systems that redirect surplus food to those in need, ensuring that valuable resources are not needlessly discarded while many go without adequate nutrition.

1.3. Project as Solution

The project's vision to address surplus food distribution through a specialized application is incredibly promising. By serving as a comprehensive platform, the application streamlines the process of collecting surplus food from diverse sources like supermarkets, restaurants, and households. This streamlined system allows for easy input of surplus food details, including type, quantity, and expiry dates, ensuring a seamless donation process. The waste food management system can have different user roles the donor can post the food with details then the system notifies all volunteers and users where new food is available for donation and volunteers to contact to pick up food and go distribute it somewhere, but case volunteers cannot accept the food and the data is expired. The food is not suitable for eating the farmer can go to receive the used for farming.

The application acts as a vital link between surplus food providers and organizations or communities in need. Its functionality enables precise matching of surplus food to the specific requirements of shelters and communities, guaranteeing that the surplus reaches those areas where it can make the most significant impact. Additionally, the project's commitment to incorporating educational resources and awareness campaigns is commendable. Educating both donors and recipients about reducing waste, proper food handling, and innovative utilization of surplus food is key to fostering a sustainable solution.

[\(project solicitation for appendix\)](#)

1.4. Aims and Objectives

1.4.1. Aim

The food donation system aims to minimise food waste by efficiently redistributing surplus food to needy communities, fostering a more equitable and sustainable food distribution network.

1.4.2. Objectives

The major objectives of the project are given below:

- To develop a user-friendly interface facilitating surplus food input, donation matching, and recipient organization interaction.
- To establish a secure database management system to efficiently store and retrieve surplus food data.
- To integrate an effective communication system between food providers and recipient organizations through API programming.
- To implement innovative design elements, incorporating UI/UX principles and material design for an engaging user experience.
- To conduct comprehensive research on existing technologies and tools, identifying and integrating the most suitable ones for optimal application functionality.
- To ensure scalability and adaptability, considering the potential expansion and growth of the food donation platform.
- To promote community engagement and awareness through features that educate users about food waste reduction and proper food utilization.

2. Background

2.1. Technology and Tools Used

2.1.1. Technology

The technology can be used for system development where the waste food management system has Django, DRF, Kotlin, jetpack compose, and MySQL databases.

2.1.1.1 Django and Rest Framework:

Django is a robust web framework primarily utilized for constructing the backend infrastructure of websites and web applications. Its functionalities encompass managing databases, handling user authentication, and structuring the core architecture of the site. Most of the REST API's descriptive work should go into specifying the media type or media types that are utilized to drive the application state and represent resources. Django is used for the backend development where Rest APIs development and admin panel web. (Ganth, 2023)

[\(Django DRF for appendix\)](#)

2.1.1.2. MySQL database:

MySQL is a popular relational database management system it is used to store and manage data across food donation systems. It's utilized as the primary data store, housing and organizing all the information required for the system's functionalities and operations. The system interacts with MySQL as it accesses, retrieves, updates, and manages the stored data within the system.

2.1.1.3. Kotlin and Jetpack Compose:

The base of Jetpack Compose is Kotlin. Kotlin has certain patterns in certain situations that facilitate writing quality Compose code. You will probably lose out on some of Composer's power if you mentally convert your thoughts in another programming language to Kotlin. (Developers, 2023)

Jetpack Compose, a specialized toolkit designed within the Android framework exclusively for Kotlin, plays a pivotal role in UI development. This toolkit streamlines the creation of UI elements, allowing developers to craft visually striking interfaces using concise code. Through Jetpack Compose, your project gains efficiency and flexibility in UI development, fostering the creation of dynamic and engaging user experiences for the mobile app. The Kotlin jetpack compose is used for front-end mobile application development.

[\(compose for appendix\)](#)

2.1.2. Tools

The tools are used for system architecture development and documentation. There are some tools are used like MS Word, balsamic, Figma, and lichi-chart, which are given below and are used in the project development and design phase.

1.1.2.1. MS-Word

MS Word allows users to create, edit, format, and print text documents. With a wide range of features, it enables tasks like spell-checking, formatting paragraphs, inserting images, creating tables, and more. Its user-friendly interface and extensive functionalities make it a popular tool for various writing and documentation needs. It is used for documentation on this coursework.

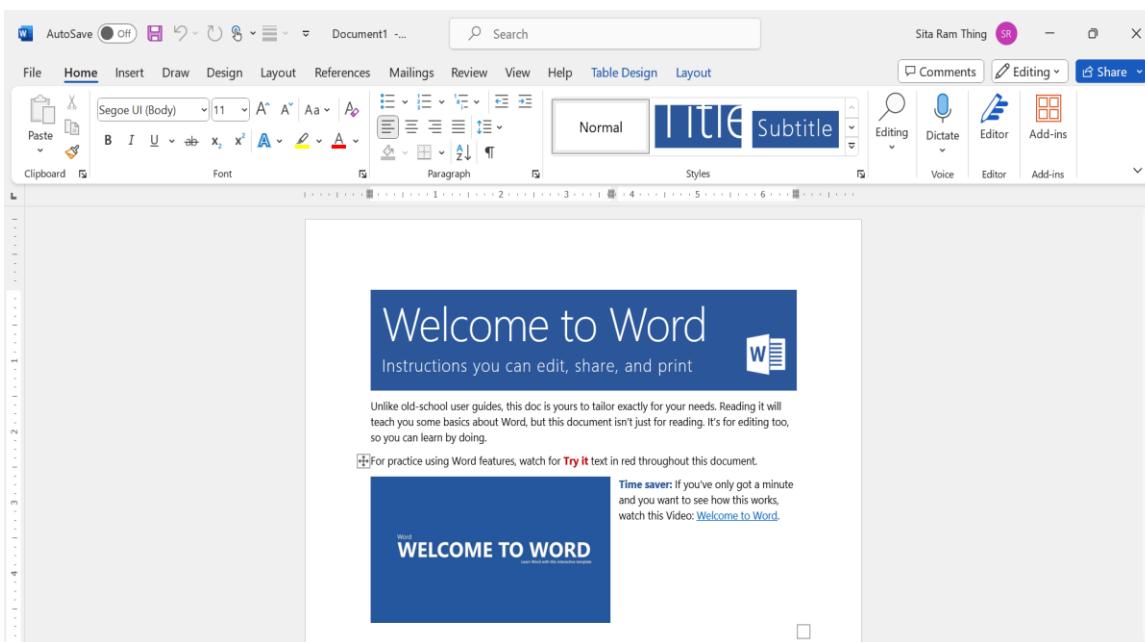


Figure 1: MS-word tools

1.1.2.2. Balsamiq

Balsamiq is a tool for simple wireframing. It produces wireframes and mockups for desktop software, web apps, and websites. Balsamiq Wireframes is the most user-friendly, quickest, most enjoyable, and well-supported wireframing application available on the Internet. The balsamic is used to create the wireframe of the system.

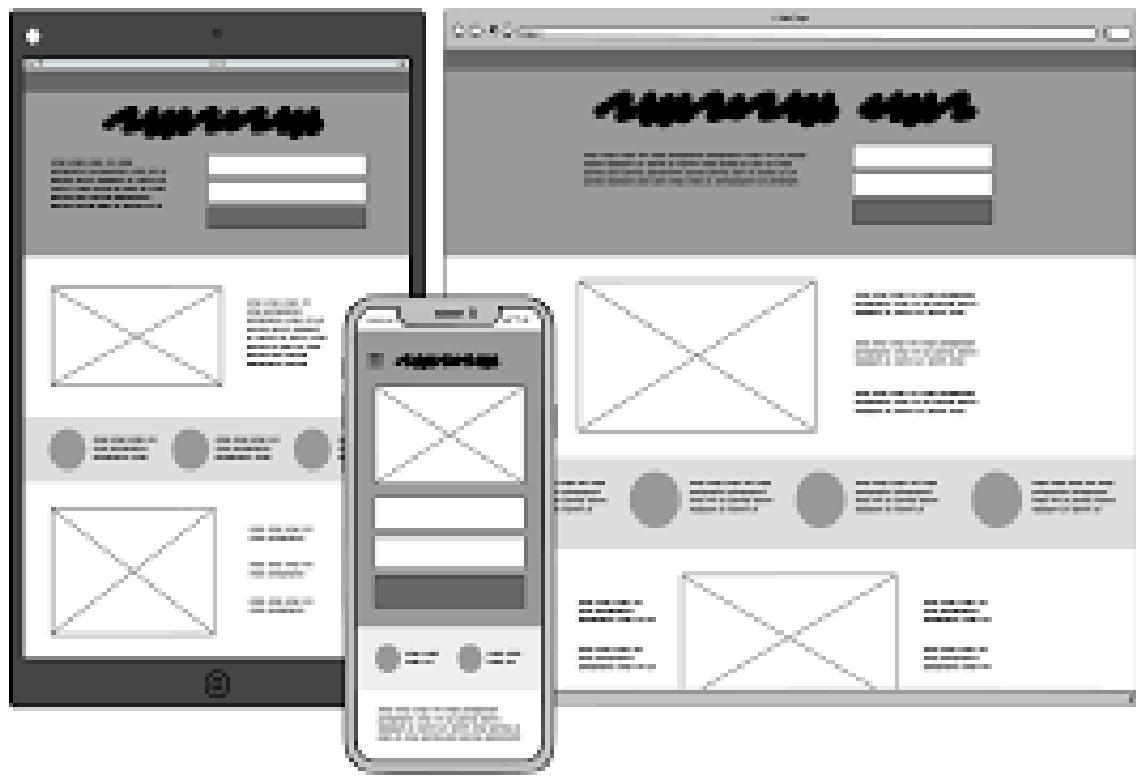


Figure 2: Balsamiq tool

[\(Other tools for appendix\)](#)

2.2. Methodology

2.2.1. Methodology Consideration

Methodology Considerations in a project, it's essential to provide a clear and concise overview of each methodology under consideration. The Final project development has three considerations of the methodology. Here's a structured way to approach it:

2.2.1.1. Waterfall Methodology:

The Waterfall methodology is a linear and sequential approach to project management. It follows a structured process where each phase—such as planning, design, development, testing, and deployment—flows in a top-down, cascading manner. One phase begins only after the previous one is complete. It's known for its detailed upfront planning and documentation, aiming for a clear and predefined project scope. Different execution teams are not required to be in continual touch under this strategy and are normally self-contained until integrations are necessary. (Adobe, 2023)

The Waterfall Method

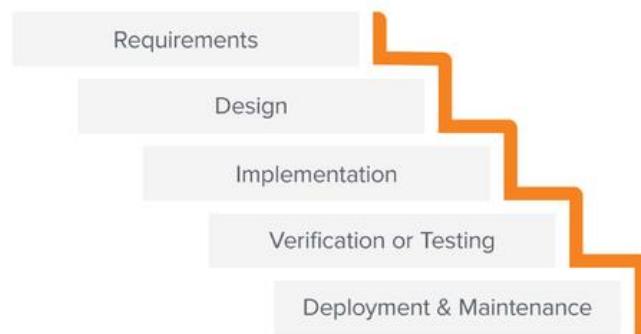


Figure 3: Waterfall methodology

The advantages and disadvantages of the Waterfall Methodology in the format you've provided:

Advantages of Waterfall Methodology:

- Provides clear project organization and control structure.
- Records errors and design details early, saving time in development.
- Efficiently estimates time and cost by strictly following the methodology.
- Helps recognize defects and faults before proceeding to the next phase.

Disadvantages of Waterfall Methodology:

- Requires clear upfront requirements, challenging as changes often occur during development.
- Restricts changes due to its linear structure.
- Lacks management of risks and security issues.
- Not suitable for large projects as real products aren't completed sequentially.

2.2.1.2. Prototype Methodology:

Prototyping involves creating quick, simplified models or versions of the final product or features to gather feedback early in the development process. It's highly iterative, focusing on creating mock-ups or prototypes that allow stakeholders and users to interact with the concept and provide feedback. Prototyping helps in refining requirements, understanding user needs, and validating design concepts before investing heavily in full-scale development. However, without careful control, it can lead to scope creep—where the project scope expands without proper management. For the new system, a basic design is built. The users extensively analyze the initial prototype, noting its virtues and faults, as well as what should be added and removed. (Lewis, 2023)

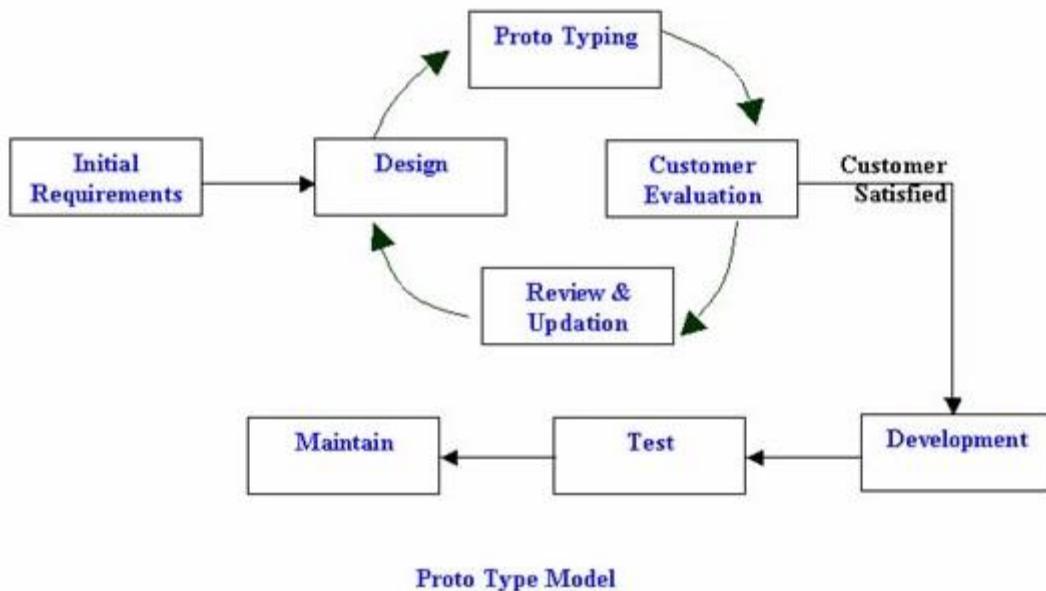


Figure 4: Prototype methodology

The advantages and disadvantages of the Waterfall Methodology in the format you've provided:

Advantages of Prototype Methodology:

- This methodology offers design flexibility, allowing for adaptability in the design process.
- It facilitates easy error detection during the development phase.
- Scope for refinement enables seamless accommodation of new requirements.
- Ensures higher customer satisfaction and comfort through active user involvement.
- Actively engages users in the development phase, fostering a collaborative approach.

Disadvantages of Prototype Methodology:

- Implementation of this model can incur high costs during the development process.
- Variations in requirements might lead to inconsistencies and frequent changes.
- Rapid development of prototypes might result in suboptimal solutions due to haste.
- Complexity within the system could increase as prototypes evolve.
- Insufficient or incomplete problem analysis may lead to inadequacies within the model.

2.2.1.3. Scrum Methodology:

Scrum is an agile framework that emphasizes flexibility, collaboration, and iterative development. It divides the project into smaller, manageable iterations called sprints, usually lasting a few weeks. During each sprint, a small part of the project is completed, allowing for frequent reviews, adjustments, and adaptability to changing requirements. Scrum fosters a highly collaborative environment, encouraging teamwork and continuous communication. While it offers adaptability and responsiveness, it requires consistent involvement and management to ensure the project progresses smoothly within the iterative cycles. Scrum is like the conductor of an agile symphony in the fast-paced world of software development. (S, 2023)

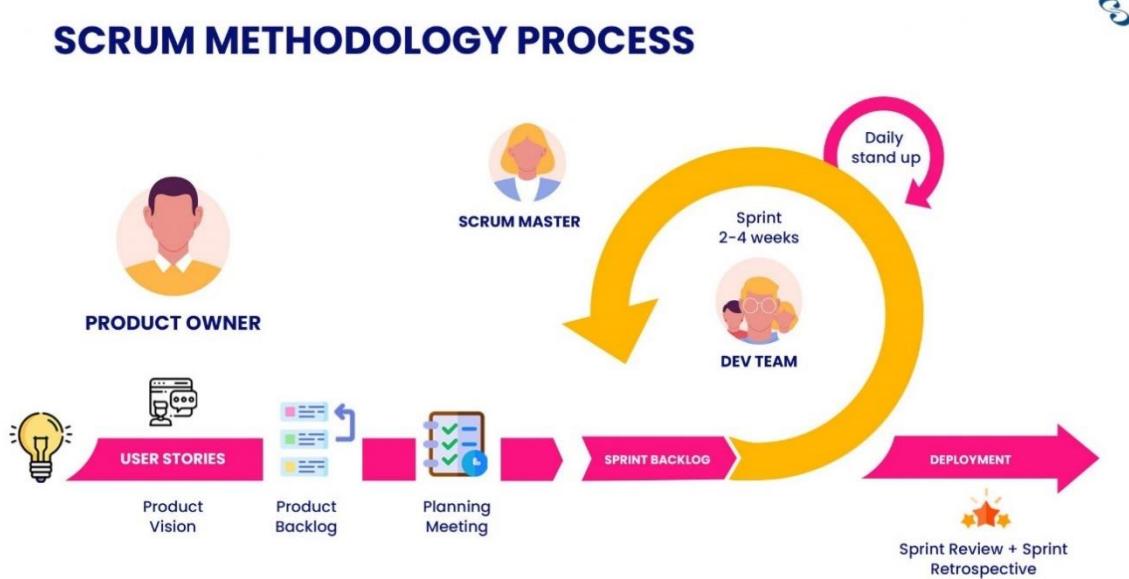


Figure 5: Scrum Methodology

The advantages and disadvantages of the Scrum Methodology in the arrangement you've given:

Advantages of Scrum Methodology:

- Scrum guarantees efficient use of both time and resources.
- Divides large projects into smaller sprints.
- Works well for quick development projects.
- Developments are coded and tested during the sprint review.
- Scrum is agile and welcomes feedback from stakeholders and customers.

Disadvantage of Scrum Methodology:

- Because Scrum lacks a set completion date, it frequently causes scope creep.
- Project failure is most likely to occur if participants lack commitment or cooperation.
- It can be difficult to implement the Scrum framework in large teams.
- Projects might suffer greatly from the departure of any team members in the middle of them.

2.2.2. Methodology Selection

Scrum proves invaluable in crafting an optimized Food Donation System by fostering step-by-step improvements and swift adaptability to changes. It ensures that new ideas or needs can be seamlessly integrated or removed within a short timeframe, facilitating agile adjustments to my project's features. Scrum's iterative approach allows early detection and resolution of issues, preventing potential disruptions, as each sprint involves thorough testing and review. This methodology's structured progression, combining development, testing, and problem-solving, offers the ideal framework for my system's development, offering flexibility and ease in accommodating changes within the development process.

1. Vision
2. User stories
3. Scrum planning
4. Product backlog
5. Sprints
 - a) Sprint planning
 - b) Sprint backlog
 - c) Daily Scrum Meeting
 - d) Development
 - e) Testing / QA
 - f) Sprint Review
 - g) Sprint Retrospective
6. Release Product



Figure 6: Selected Scrum Methodology

Benefits of having a scrum methodology for project management:

The project greatly follows the Agile Methodology. It promotes adaptability to changing requirements, ensuring that the project remains aligned with evolving needs. Through iterative cycles, it enables regular feedback, enhancing stakeholder engagement and ensuring the final product meets their expectations. Scrum's structured yet flexible approach fosters improved collaboration within teams, leading to increased productivity and innovation.

2.2.3. Methodology Section Justification

2.2.3.1. Reason for not choosing Waterfall methodology.

Scenario	The initial understanding of food donation project requirements is limited, and there's a high likelihood of requirements evolving as the project progresses.
Methodology	Waterfall Methodology
Justification	In the Waterfall model, the project progresses through sequential stages, where the initial requirements play a significant role. However, when the requirements are uncertain or likely to change, this methodology's inflexible nature might lead to challenges in accommodating evolving needs. This could result in a gap between the initial project plan and the actual requirements, impacting the project's success.

Table 1: Justification scenario 1 for waterfall methodology

Scenario	The food donation project requires frequent adjustments and learning from ongoing experiences during its implementation.
Methodology	Waterfall Methodology
Justification	Waterfall's linear structure doesn't easily accommodate changes or incorporate learnings during project execution. In a project like food donation, where continuous improvement is vital, a more flexible methodology might be better suited. Students might face difficulties when they encounter the need for ongoing adaptations in a Waterfall framework, potentially affecting their ability to respond to changing circumstances effectively.

Table 2: Justification scenario 2 for waterfall methodology

Scenario	Client involvement and feedback are crucial for the success of the food donation project, but the Waterfall methodology restricts Client involvement until the project's final stages.
Methodology	Waterfall Methodology
Justification	In the Waterfall model, clients usually provide feedback mainly during the requirements phase and might not be actively engaged until the project's completion. For a project like food donation, where continuous client input is vital, this limited involvement might result in missed opportunities for critical feedback and adjustments throughout the project lifecycle, potentially leading to a mismatch between the project's outcome and client expectations.

Table 3: Justification scenario 3 for waterfall methodology

The Waterfall methodology, known for its linear progression through predefined stages, may not be the most suitable choice for a Food Donation Final Year Project (FYP). Its rigid structure poses challenges in accommodating evolving project requirements and lacks flexibility for continuous adaptation. In a dynamic project like food donation, where needs might change, Waterfall's fixed approach could hinder the project's responsiveness to these changes. Therefore, opting for a more adaptable and flexible methodology, such as Agile or iterative approaches, could better cater to the project's evolving needs, fostering continuous improvement and responsiveness throughout the project's lifecycle. So, this methodology is not suitable for the project and is not chosen for system development.

2.2.3.2. Reason for not choosing Prototype Methodology.

Scenario	The FYP project encounters ongoing changes in project scope due to evolving community needs or unforeseen challenges.
Methodology	Prototype Methodology
Justification	The Prototype methodology emphasizes building and refining a working model based on clear initial requirements. However, in a project with continually shifting needs, a prototype might struggle to keep up with rapid changes, leading to frequent modifications in the prototype itself. This iterative nature could create inefficiencies and make it challenging to maintain a stable prototype in line with evolving project needs.

Table 4: Justification scenario 1 for prototype methodology

Scenario	The FYP project has strict deadlines and limited resources for development and testing.
Methodology	Prototype Methodology
Justification	Prototyping involves iterative cycles of development, testing, and refinement. In a constrained environment where time and resources are limited, the iterative nature of prototypes might lead to delays or inefficiencies. This methodology might not be suitable when there's a need to meet fixed deadlines or optimize resource utilization within a confined timeframe.

Table 5: Justification scenario 2 for Prototype methodology

Scenario	The FYP project lacks initial clarity regarding client requirements and expectations of the project.
Methodology	Prototype Methodology
Justification	Prototyping relies on a clear understanding of requirements and continuous client involvement for iterative improvements. If the client's needs or project expectations are unclear initially, this methodology might struggle to deliver a refined prototype that aligns with clients' evolving expectations. The lack of early-stage clarity could impede the iterative development process.

Table 6 Justification scenario 3 for Prototype methodology

In these scenarios, the Prototype methodology might face challenges in accommodating evolving scopes, managing limited resources and time constraints, and aligning with unclear stakeholder requirements. These limitations could impact its effectiveness for an FYP project focused on food donation. so will not select the prototype methodology.

2.2.3.3. Reason for Choosing Scrum Methodology

Scenario	The project encounters frequent changes in community needs or regulatory requirements.
Methodology	Scrum Methodology
Justification	Scrum allows seamless adaptation to changes. Its iterative nature accommodates evolving requirements, ensuring that the project remains aligned with dynamic community needs throughout development.

Table 7: Justification scenario 1 for scrum methodology

Scenario	New features or functionalities need to be incorporated swiftly into the project.
Methodology	Scrum Methodology
Justification	Scrum enables the quick integration of new features. Its iterative cycles prioritize and incorporate new functionalities into subsequent sprints, ensuring timely integration without disrupting the development flow.

Table 8: Justification scenario 2 for scrum methodology

Scenario	Continuous client review and feedback are crucial for project success.
Methodology	Scrum Methodology
Justification	Scrum allows seamless adaptation to changes. With its iterative nature, Scrum promotes regular review and feedback loops. With sprint reviews and retrospectives, stakeholders can provide continuous input, ensuring alignment with their expectations throughout the project lifecycle.

Table 9: Justification scenario 3 for scrum methodology

Scenario	We are ensuring thorough testing for each feature or component completed in the project.
Methodology	Scrum Methodology
Justification	Scrum mandates testing for every increment. By completing and testing features within sprints, the methodology ensures each component is thoroughly examined, reducing the risk of undetected errors or bugs.

Table 10: Justification scenario 4 for scrum methodology

Scenario	In the project continuous improvement is essential to enhance project outcomes where scrum helps to development phase.
Methodology	Scrum Methodology
Justification	Scrum fosters iterative improvement. Each sprint's retrospective allows the team to reflect on what went well and what needs improvement, leading to incremental enhancements throughout the project.

Table 11: Justification scenario 5 for scrum methodology

Scenario	The scrum methodology can project encounter sudden priority shifts due to unforeseen emergencies.
Methodology	Scrum Methodology
Justification	Scrum's flexibility in prioritizing tasks helps handle emergencies. The method allows reprioritization within sprints, ensuring that urgent tasks can be addressed without disrupting the project's overall progress.

Table 12: Justification scenario 6 for scrum methodology

Scrum's adaptability to change, rapid feature integration, regular review cycles, comprehensive testing, task prioritization flexibility, and iterative improvement capabilities make it a highly suitable methodology for the structured and efficient development of a Food Donation system development FYP project most helpful and suitable, so I chose the scrum methodology.

2.3. Similar Systems

2.3.1. Similar System Consideration

This project has been proposed for development by keeping about the several others. previously developed waste or surplus food managed to distribute and provide alerts about them. This section covers a number of these systems, from which the project's concepts were developed. Discussions are also held regarding the systems' creation, operation, and critical analysis.

2.3.1.1. Flashfoods

Author: local government bodies, NGOs,

It involves early warning systems to anticipate flash floods, activating a network of collection points, and swiftly mobilizing volunteers to gather non-perishable food items like canned goods, grains, and bottled water. Key features of this system include its rapid response strategy triggered by early warnings, the mobilization of volunteers and local organizations, partnerships with businesses and farmers for resources, transparent communication about relief efforts, and adaptability to different disaster scales. Through these features, the Flashflood Food Donation System strives to provide timely and essential food aid to those impacted by sudden disasters, ensuring a more efficient and organized response to mitigate hunger during crises. (Josh Domingues, 2023)

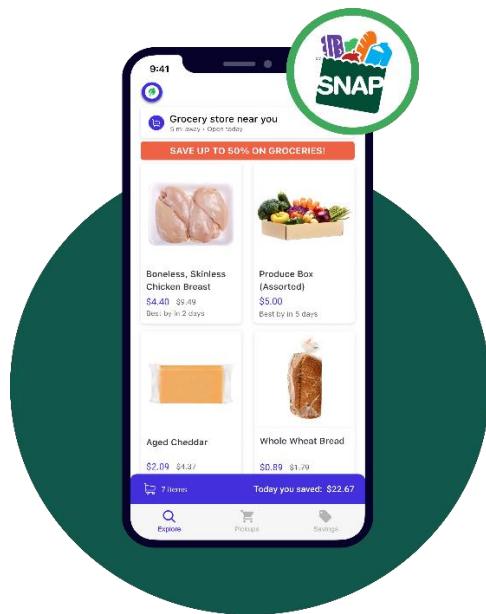


Figure 7: FlashFood similar system

2.3.1.2. Food Rescue

Author: Dave Lampert, Melissa Spiesman, Jeff Schacher and Kevin Mullins

Food Rescue, originating as Community Plates in Fairfield County in 2011, was conceived by Jeff Schacher and Kevin Mullins in response to the pressing issues of food insecurity and food waste. The primary goals of the system were to reduce food waste while simultaneously feeding those in need. This unique model garnered substantial support from volunteers, food donors, and social service agencies. Over the years, they furthered their expansion, eventually renaming the organization to Food Rescue US in 2017, reflective of their nationwide presence. With an upgraded app and a growing network, the initiative has provided more than 117 million meals and prevented over 147 million pounds of excess food from reaching landfills. (Mullins, 2024)



Figure 8: Food Rescue similar app

2.3.1.3. No Food Waste

Author: Anne Frank

No Food Waste is a food donation system controlled by an Indian organization which aims to redistribute excess food from weddings, parties, and events to those who are hungry. The system started with 2 Shopper Bags and a volunteer to collect the surplus food and deliver it to the homeless through the Public Transportation System from October 16th, 2014. Surplus food is distributed or served to the needy before the expiry of surplus food. The food is collected to distribute every day, but the system cannot show all the history of the donation details. The system has shown donation location and distributed location display in the system. (Frank, 2024)



Figure 9: No Food Waste Similar System

2.3.2. Similar System Comparison

Features	Flashfood	Food Rescue	No Waste	Food	Food Share
Food Post	Yes	Yes	Yes	Yes	Yes
View Location on Google Map	Yes	Yes	No	Yes	
Push Notification	Yes	Yes	Yes	Yes	
Local Notification	No	No	No	Yes	
Email Notification for Distributed Message	Yes	Yes	Yes	Yes	
Donation complaint for administration	No	Yes	Yes	Yes	
Time-expired foods are recommended for farmer	Yes	No	No	Yes	
User View History	No	No	No	Yes	

Table 13: Comparison with similar application

The comparison involves three similar systems alongside mine, each with distinct features. I have incorporated a new feature, local notification. Additionally, features such as location and email notification are borrowed from the system. The 'Food Share' application was developed primarily to distribute surplus food to homeless or impoverished individuals. The system also allows farmers to participate by donating surplus food, while other expired food, can be used for farming purposes.

3. Work to Date

Research to topic	Durations	Stat Date	End Date	Status	
Phase 1: Planning and Requirement Analysis	3 weeks	10/09/2023	30/09/2023	Completed	
Task 1: Research and project finalize	7 days	10/09/2023	16/09/2023		
Task 2: Client meeting	3 days	17/09/2023	19/09/2023		
Task 3: Requirement collection	4 days	20/09/2023	23/09/2023		
Task 4: Proposal writing	7 days	24/09/2023	30/09/2023		
Phase 2: Defining Requirements	2 weeks	1/10/2023	21/10/2023	100%	
Task 1: User stories	7 days	1/10/2023	7/10/2023	Completed	
Task 2: Scrum planning	7 days	8/10/2023	14/10/2023		
Task 3: Product Backlog	7 days	15/10/2023	21/10/2023		
Phase 3: Designing the Project Architecture	4 weeks	22/10/2023	11/11/2023	100%	
Task 1: Research about UML Diagram and backend development	14 days	22/10/2023	4/11/2023	Completed	
Task 2: Usecase, Sequence, and Context diagram develop	7 days	5/11/2023	11/11/2023		
Task 3: DFD, Class diagram, Activity diagram develop	7 days	12/11/2023	18/11/2023		
Task 4: ERD, System architecture, Wireframe develop	7 days	19/11/2023	25/11/2023		
Phase 4: Building or Developing the Project	14 weeks	26/11/2023	16/3/2024	Pending	
Sprint 1: DBMS creation	2 weeks	26/11/2023	9/12/2023	100%	
Sprint Planning	1 day	26/11/2023	26/11/2023	Completed	
Sprint Backlog	1 day	27/11/2023	27/11/2023		
Development	7 days	28/11/2023	4/12/2023		
Testing	2 days	5/12/2023	6/12/2023		
Sprint Review	1 day	7/12/2023	7/12/2023		
Retrospective and Release	1 day	8/12/2023	9/12/2023		
Sprint 2: Rest API development	Same process of Sprint 1	2 weeks	10/12/2023	23/12/2023	50% (Incomplete)
Sprint 3: Web admin panel UI development	Same process of Sprint 1	2 weeks	24/12/2023	6/1/2024	50% (Incomplete)
Sprint 4: Local database with CRUD and Retrofit to API implement	Same process of Sprint 1	2 weeks	7/1/2024	20/1/2024	Due
Sprint 5: Mobile UI development	Same process of Sprint 1	2 weeks	21/1/2024	03/02/2024	50% (Incomplete)
Sprint 6: Authentication development	Same process of Sprint 1	2 weeks	04/02/2024	17/02/2024	100% (Completed)
Sprint 7: Frontend and backend connection	Same process of Sprint 1	2 weeks	18/02/2024	2/3/2024	Due
Sprint 8: Google services and Firebase implement	Same process of Sprint 1	2 weeks	3/3/2024	16/03/2024	Due
Phase 5: Testing	1 week	17/3/2024	23/03/2024		
Task 1: Final Testing	7 days	17/3/2024	23/03/2024	Due	
Phase 6: Project Deployment	2 weeks	24/03/2024	06/04/2024		
Task 1: Summarize the deployment project	7 days	24/03/2024	30/03/2024	Due	
Task 2: Document and mitigate risks	7 days	31/03/2024	6/4/2024		
Task 2: Document and mitigate risks	7 days	31/03/2024	6/4/2024		
Phase 7: Maintenance and Release	1 week	7/4/2024	13/4/2024		
Task 1: Bugs and problem maintain	3 days	7/4/2024	9/4/2024	Due	
Task 2: Project review and submit	4 days	10/4/2024	13/4/2024		

Figure 10: Work to date durations

All the development progress evidence is given in the appendix side.

([Work to date for appendix](#))

3.1. Requirement Gathering

Purpose:

The purpose of the waste or surplus food collection to donation application is to minimize food wastage by connecting surplus food providers with local charities, shelters, and individuals in need. It aims to streamline the process of food donation, ensuring that excess food is efficiently redirected to those who can benefit from it. By leveraging technology, this application intends to bridge the gap between food surplus and scarcity, contributing to a more sustainable and equitable society.

Intended Audience:

- Food Providers: Restaurants, hotels, event organisers, grocery stores, etc., seeking to donate surplus food.
- Organizations or People: Charities, shelters, poor or homeless people, food banks, community centres, and NGOs are involved in distributing food to those in need.
- Individuals: People interested in volunteering or donating directly to help redistribute surplus food.

Project Scope:

- User-Friendly Interface: Develop an intuitive mobile and web application accessible to food providers and receiving organizations but web to admin control the system. This interface should allow easy registration, food listing, scheduling, and tracking of donations.
- Registration and Verification: Implement a secure registration system for both food providers and receiving organizations, ensuring credibility and safety in the donation process.
- Real-Time Donation Listings: Enable food providers to list surplus food items with details like quantity, expiration date, and pick-up/delivery options.
- Matching Algorithm: Develop an algorithm that matches surplus food listings with nearby receiving organizations based on location, quantity, and their specific needs.
- Communication Platform: Include a messaging system to facilitate communication between food providers and receiving organizations for coordination and logistics.
- Volunteer and Donation Support: Integrate features for individuals to volunteer time or donate resources directly through the app to support the cause.
- Analytics and History: Incorporate analytics tools to track the amount of food donated, number of meals provided, and overall impact, providing history details to showcase the application's effectiveness.

3.1.1. Survey Conducted (if a client with the client, if not with users)

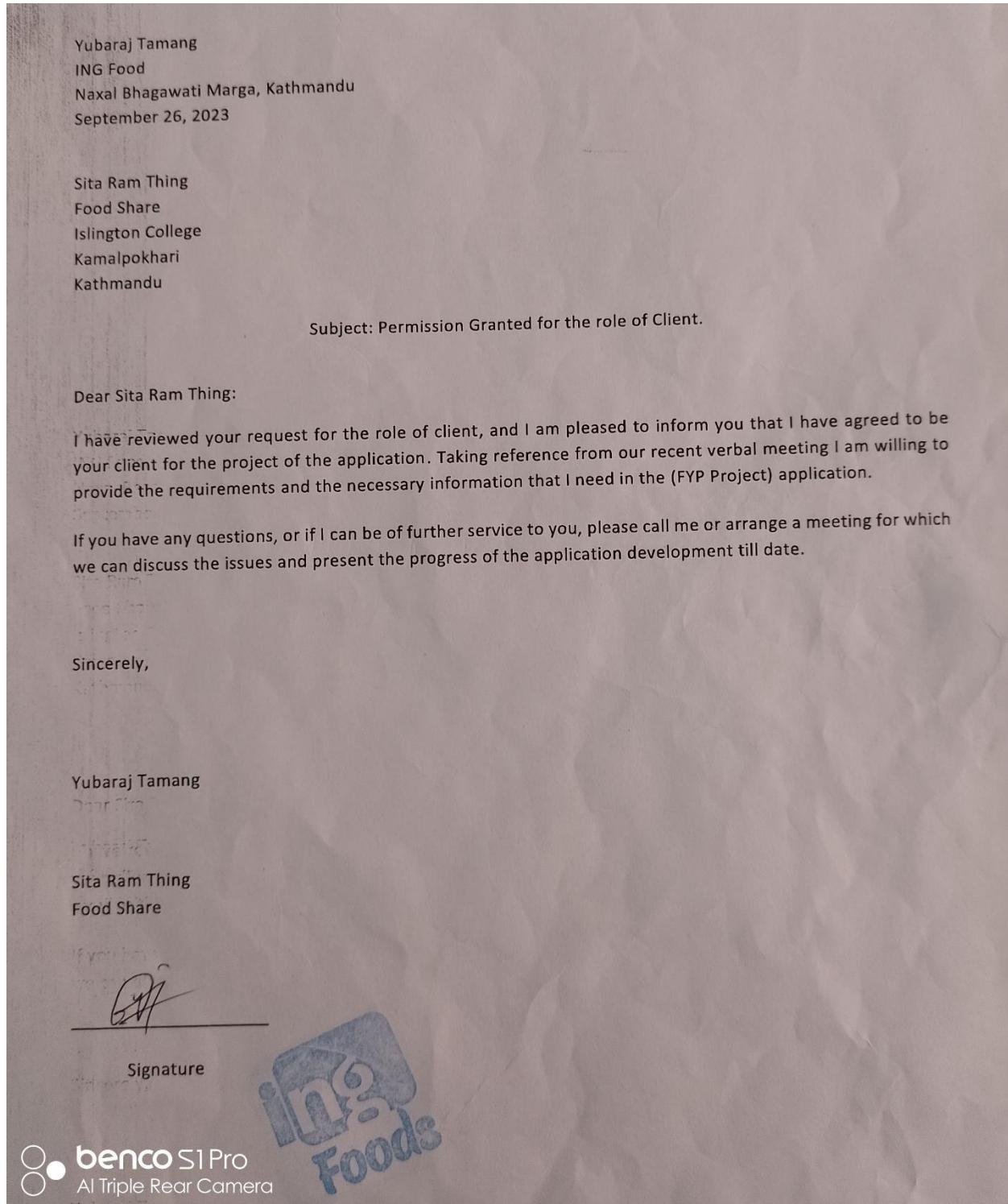


Table 14: The client approaches the latter.

The system is developed for client requirements where meet the client and some features are developed. Also, in the survey for food donations, many users can suggest different requirements. The client and survey form to collect the requirement and develop the system where any features are moved on development processes like UML diagram, database development, login authentication, web and mobile UI development, and some features carer continues to develop where pending, but some features are upcoming sprint to develop.

The client-approved letter is given below:

The survey response is given below:

A screenshot of a survey response interface. At the top, there are tabs: 'Questions', 'Responses' (which is underlined and shows the number '30'), and 'Settings'. Below this, the title '30 responses' is displayed, along with a 'View in Sheets' button and a three-dot menu icon. A toggle switch labeled 'Accepting responses' is turned on. There are three navigation buttons: 'Summary' (underlined), 'Question', and 'Individual'. The main area shows a list of names under the question 'Write your full name?'. The names listed are: Ram Pariyar, Utsarga KC, Kushal Poudel, Abinash Bhattarai, and Manjil.

Figure 11: Survey form response list of names

Write your email address?

30 responses

pariyaram2023@gmail.com

kcutsarga77@gmail.com

np01nt4a210078@islingtoncollege.edu.np

abinashbhattarai30@gmail.com

NP01MA4A210003@islingtoncollege.edu.np

bimesh.shrestha@innovatetech.co

mynameispratikpoudel32@gmail.com

niken1727@gmail.com

reeyashrestha00@gmail.com

Figure 12 Survey form response list of email

The user survey response link is: https://docs.google.com/forms/d/1nvGA9vr6h--5PxgHTYFd_Ycq6iJtVw_YeNFvg3xbwVQ/edit#responses

[\(Survey form for appendix\)](#)

3.1.2. Feature List (As per systems and roles)

3.1.2.1. Web

Admin:

- User management: Create, modify, and delete user accounts.
- Dashboard: Overview of system activities, user statistics, and reports.
- Content management: Update, edit, or remove website content.
- Analytics: Access to data and analytics for user behaviour and system performance insights.
- Settings: Manage system configurations and preferences.
- Data operations: Add, update, and delete data within the system for accuracy.

3.1.2.2. Mobile:

Donor:

- Facilitate food and resource donations for the poor people of surplus food.
- the donation track and monitor donated items for transparency. It can access donation history and receipts.
- Receive updates on campaigns and activities.

Volunteer:

- Browse and sign up for volunteering events or activities.
- View, manage, and communicate about assigned tasks or shifts.
- Connect with admins, donors, and other volunteers for coordination.

Farmer:

- Access agricultural tools and resources.
- Receive guidance on crops, weather, and farming techniques.
- Obtain market information for trends and pricing.
- Handle expired food for animals and post surplus food for donation.

NGO(Admin):

- Create, manage, and update projects or initiatives.
- Utilise donation management tools for fundraising.
- Connect and coordinate with volunteers, donors, and other users.
- Manage and oversee roles within the platform.

Each role on the mobile platform serves specific functions catering to their responsibilities within the system, whether it's facilitating donations, managing agricultural activities, coordinating volunteers, overseeing projects, or administering the platform.

3.2. Use Case

A use case diagram is a visual representation that illustrates the interactions between actors and a system. It showcases various use cases of a system and how different actors interact with those use cases. The Use Case Diagram which is given below:



Figure 13: Use Case Diagram

3.2.1. High-Level Use Case

The system includes an extensive collection of use cases, many of which are high-level use cases that facilitate system design and development. Each high-level use case focuses on providing a brief explanation of each procedure as it develops and as a result, would be recognized by the customer.

The high-level use case is given below:

3.2.1.1. Register user.

Use Case:	Register Details
Actors:	Donor, Volunteer, NGO, Farmer
Descriptions:	All users can input their respective details into the system. Upon submission, the system automatically registers the provided information, ensuring seamless integration and efficient data management. If the register details are correct show the success message otherwise display the error message.

Table 15: Register user high-level use case

3.2.1.2. Take Membership

Use Case:	Take Membership
Actors:	Volunteer, NGO
Descriptions:	A new volunteer provides the personal details, and his/her details are registered with the system. The NGO provide the membership, and then volunteers take the new membership.

Table 16: Take membership user high-level use case

Use Case:	Privilege
Actors:	Volunteer
Descriptions:	After taking a new member, a new volunteer gets the privilege of the food donation system.

3.2.1.3. Login System

Use Case:	Login
Actors:	Donor, Volunteer, Farmer, NGO
Descriptions:	After registering details in the system, all the users can provide valid details and log in to the system. Then successfully log in to the system. If the login details are valid display the success message otherwise error message.

Table 17: Login user high-level use case

3.2.1.4. Food Donate

Use Case:	Food Donate
Actors:	Donor, Farmer, Volunteer
Descriptions:	The Donor or Farmer can donate the proper food information and details with location. The system can show the donation food details in the history after posting the donated food. All the volunteers can get the donation information (Notification).

Table 18: Food donation high-level use case

3.2.1.5. View Donation info

Use Case:	Donation Info
Actors:	Volunteer
Descriptions:	After receiving the donation info, the volunteer can view the donation details if it is possible or not possible to distribute.

Table 19: View donation info high-level use case

3.2.1.6. Donation Rating

Use Case:	Donation Rating
Actors:	Volunteer, Donor
Descriptions:	After the food is completely donated to some people the volunteer can give the donation rating to the donor with food distributed information.

Table 20: Donation rating high-level use case

3.2.1.7. View History

Use Case:	View History
Actors:	Donor, Volunteer, NGO, Farmer
Descriptions:	All the users can view the history of food donation rate, where to donate or more details.

Table 21: View the history of high-level use case

3.2.1.8. View NGO profile

Use Case:	View NGO profile
Actors:	Volunteer, NGO
Descriptions:	The volunteer can view the NGO profile details where some information gets more details.

Table 22: View profile high-level use case

3.2.1.9. Complain with Administration

Use Case:	Complain with Admin
Actors:	Donor, Volunteer, Admin
Descriptions:	After contact with donors and volunteers the donors cannot be provided or donate food and the receiver cannot come to receive the donation food, they can complain to the admin

Table 23: Complaint with high-level use case

[\(Expanded use case for appendix\)](#)

3.3. Sequence Diagram

3.3.1. View profile sequence diagram

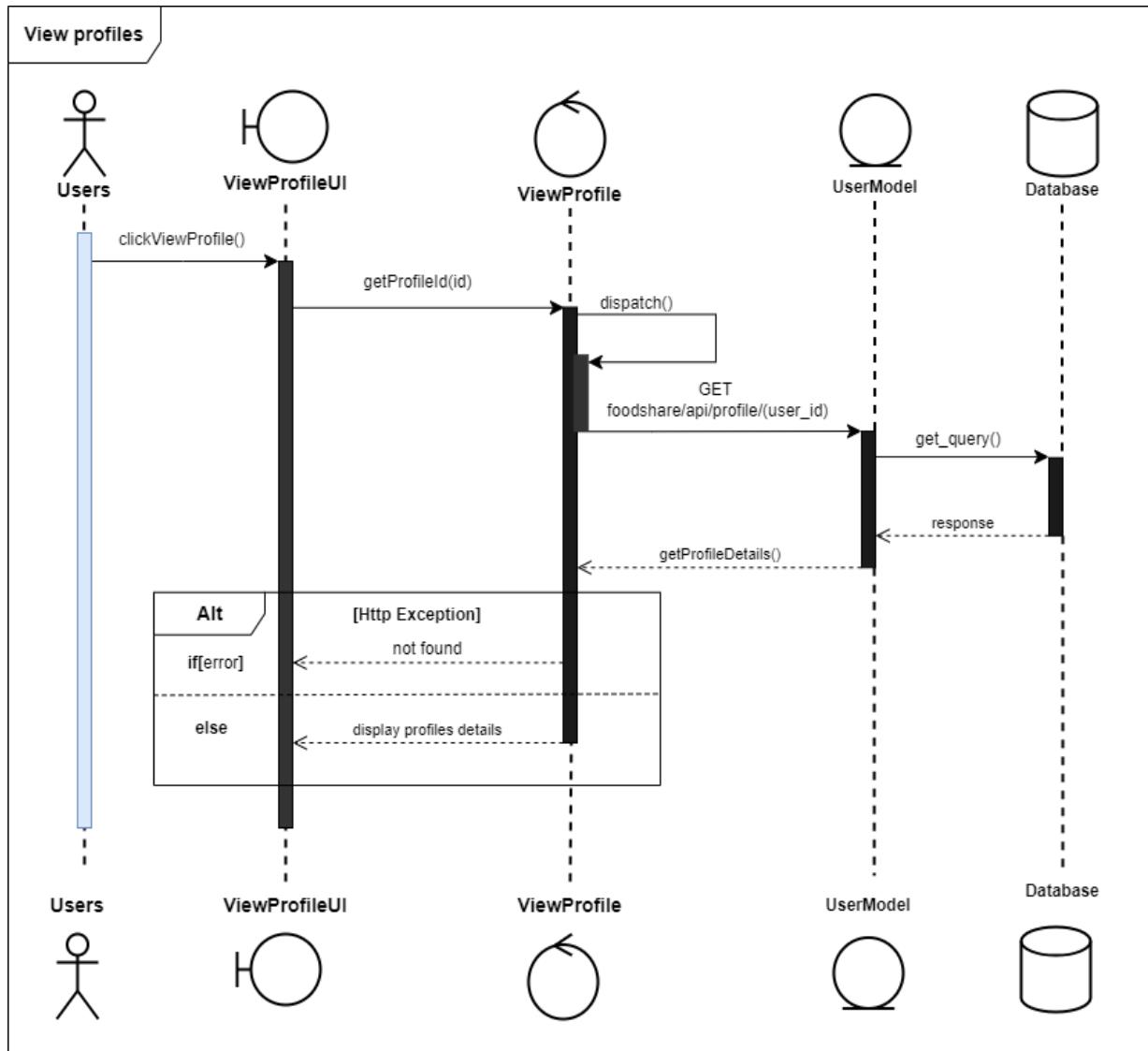


Figure 14: View profile sequence diagram

3.3.2. View history sequence diagram

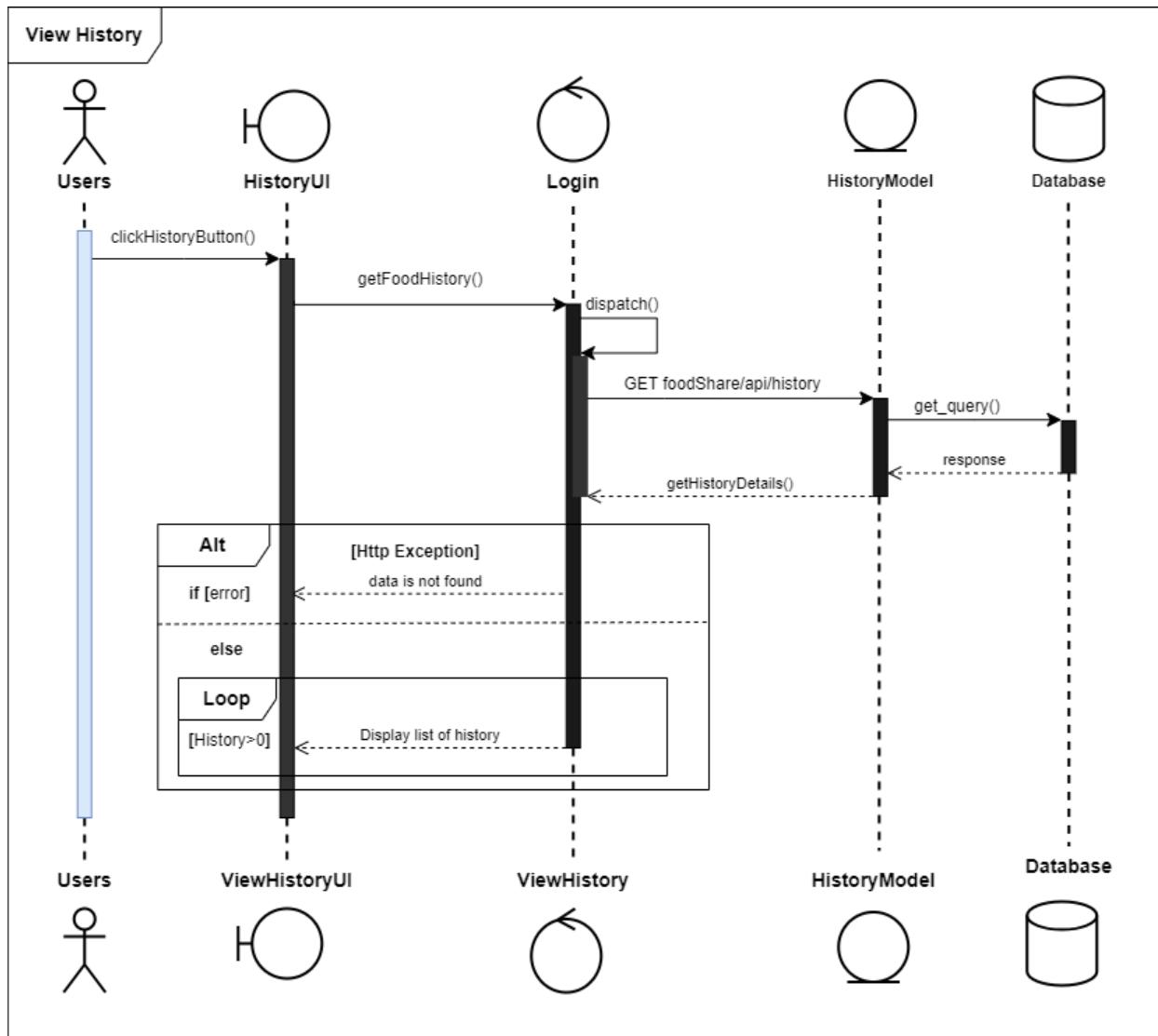


Figure 15: View the history sequence diagram

3.3.3. Post food donation sequence diagram

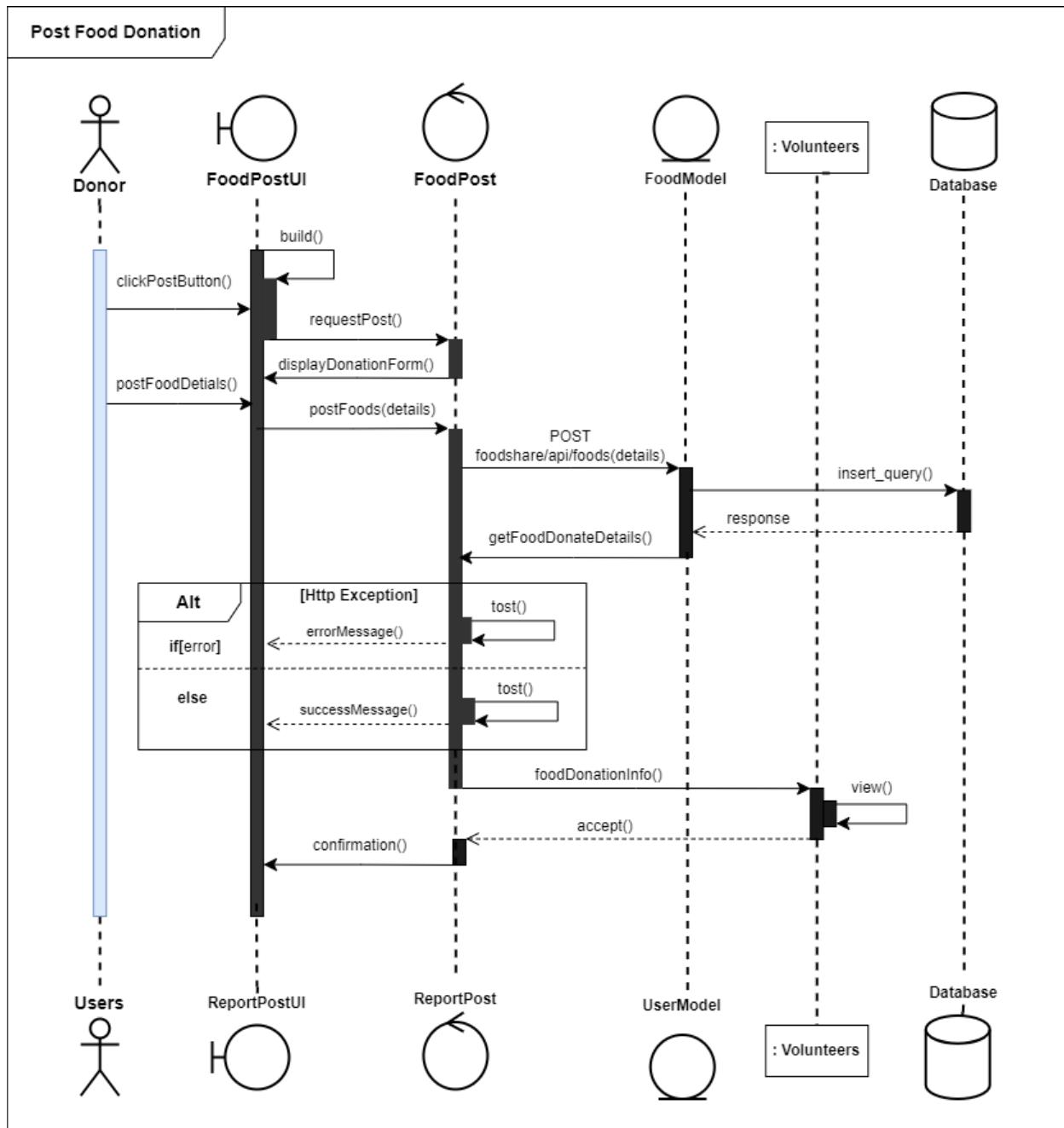


Figure 16: Food Post for donation sequence diagram

3.3.4. Donation rating sequence diagram

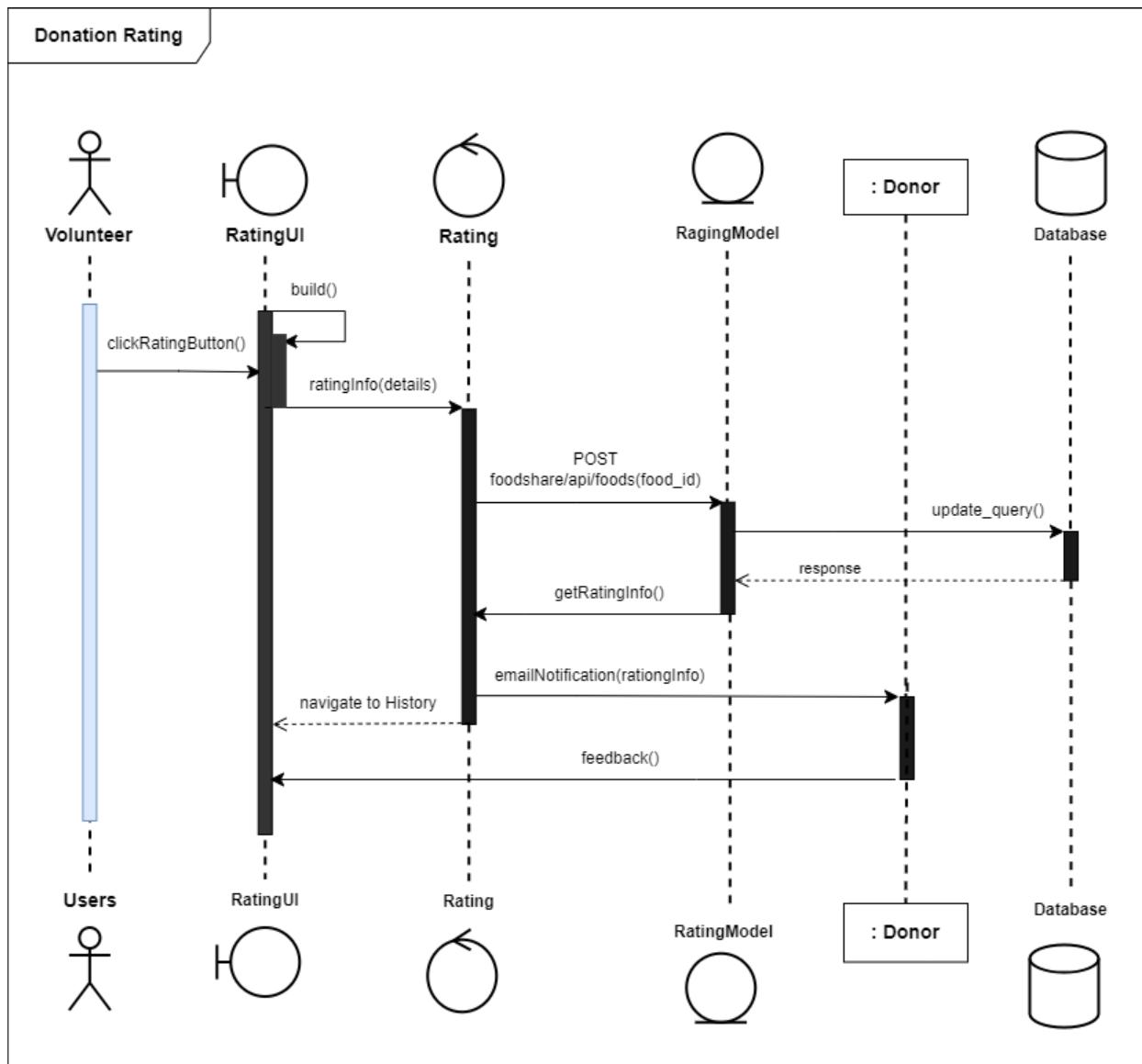


Figure 17: Donation rating sequence diagram

3.3.5. Complaint with admin

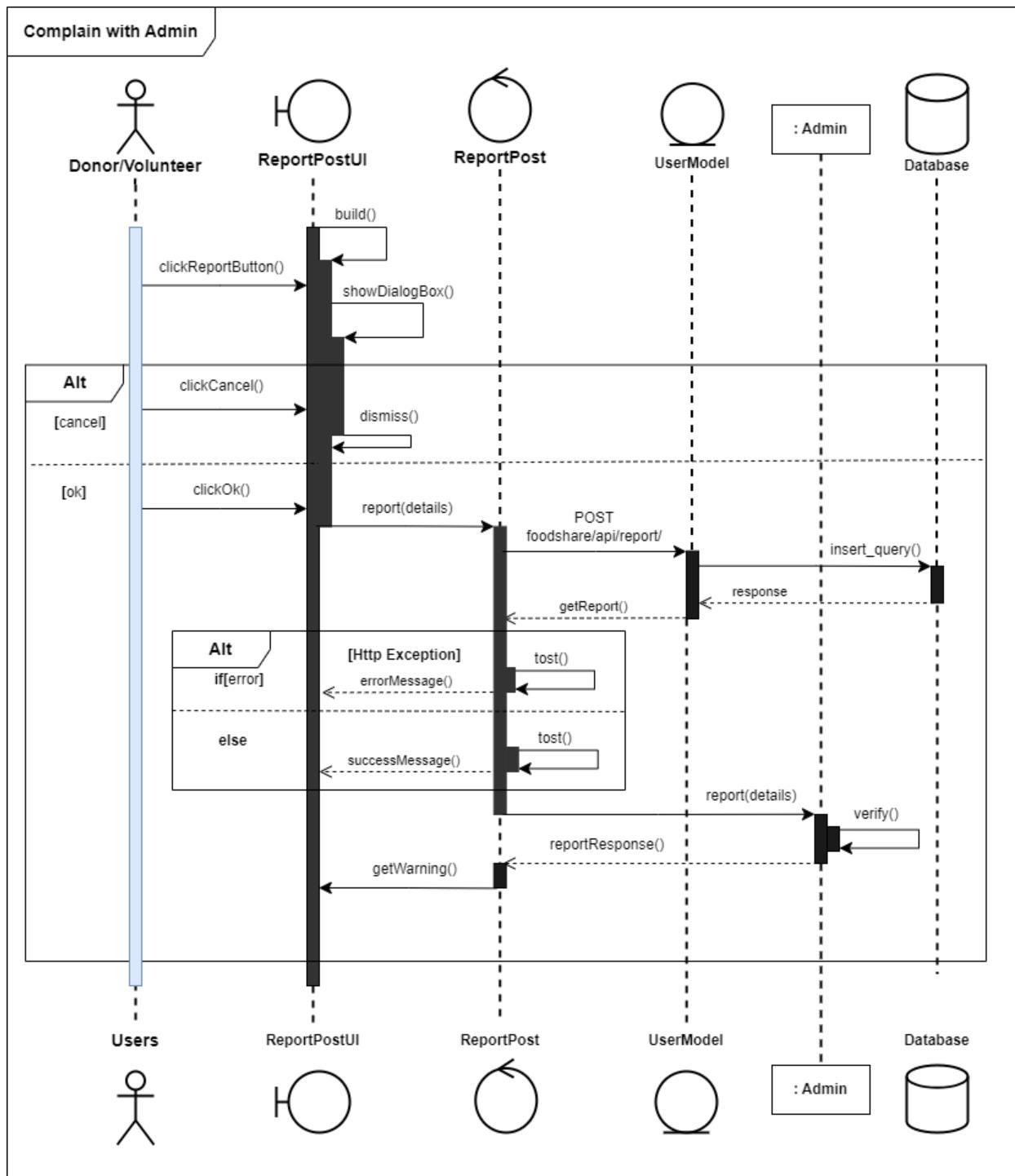


Figure 18: Complaint with admin sequence diagram

([Sequence Diagram for appendix](#))

3.4. Context Diagram

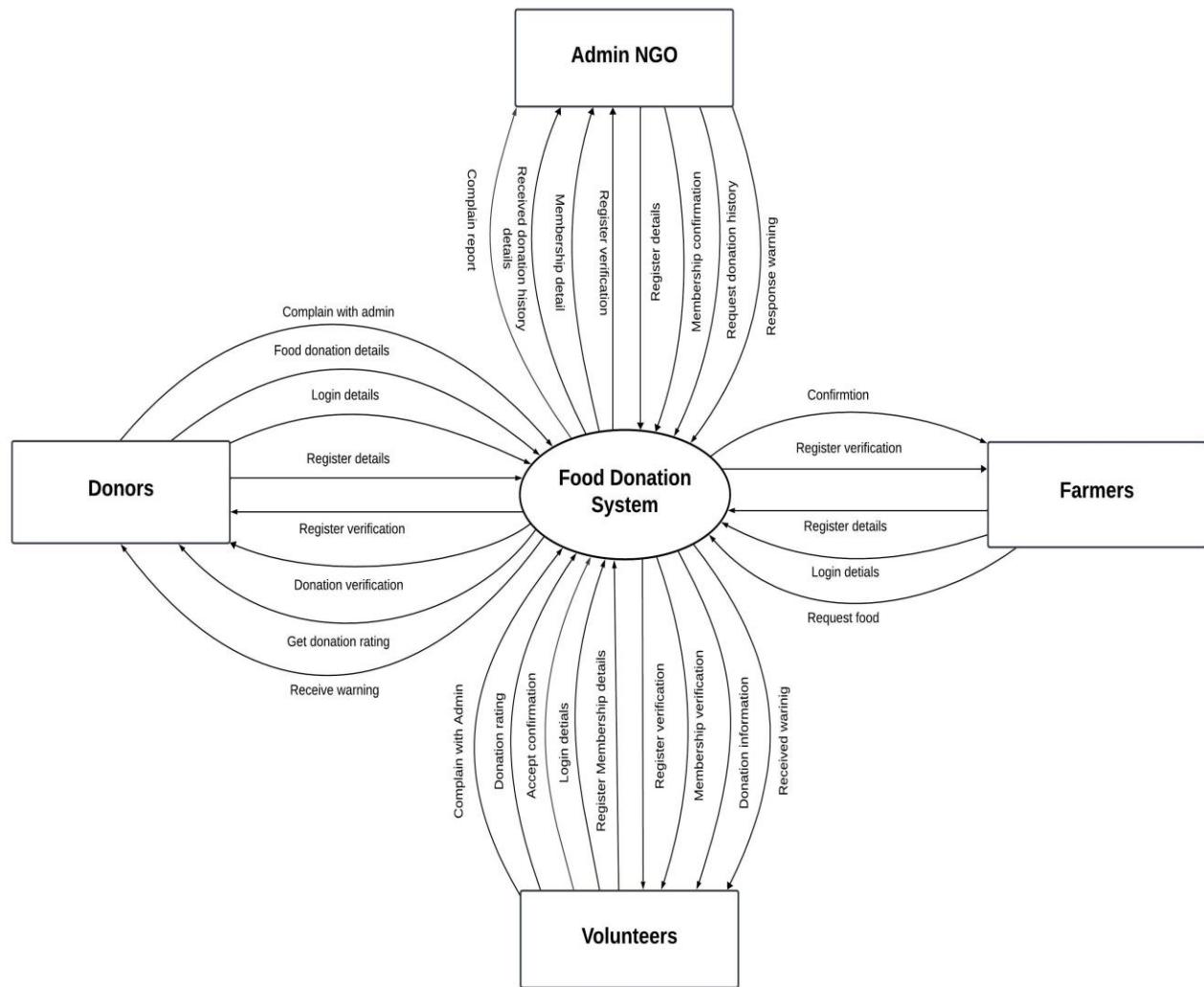


Figure 19: Context diagram

3.5. Data Flow Diagram Level -1

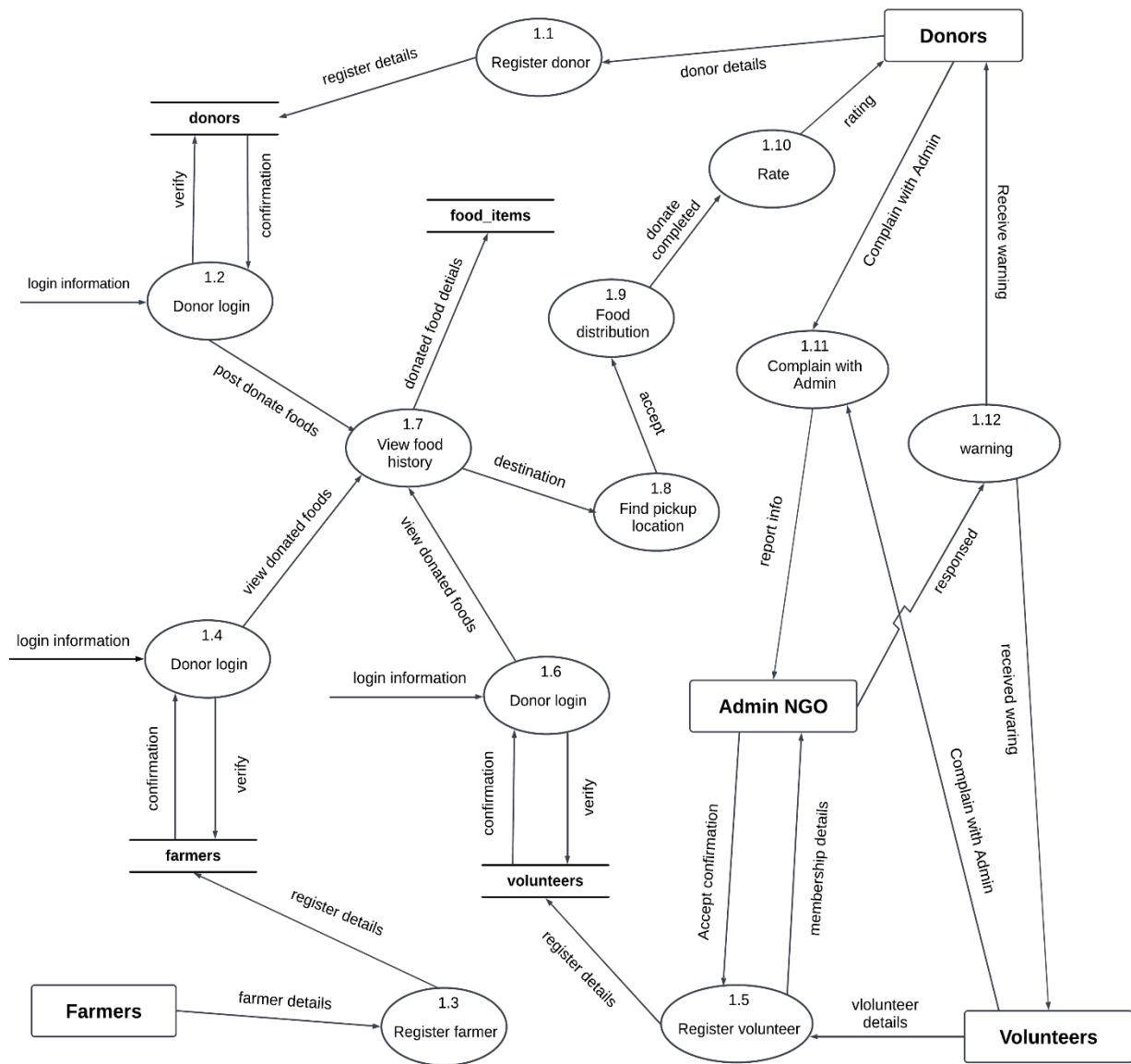


Figure 20: DFD Level-1

3.6. Activity Diagram

3.6.1. Take Membership Activity

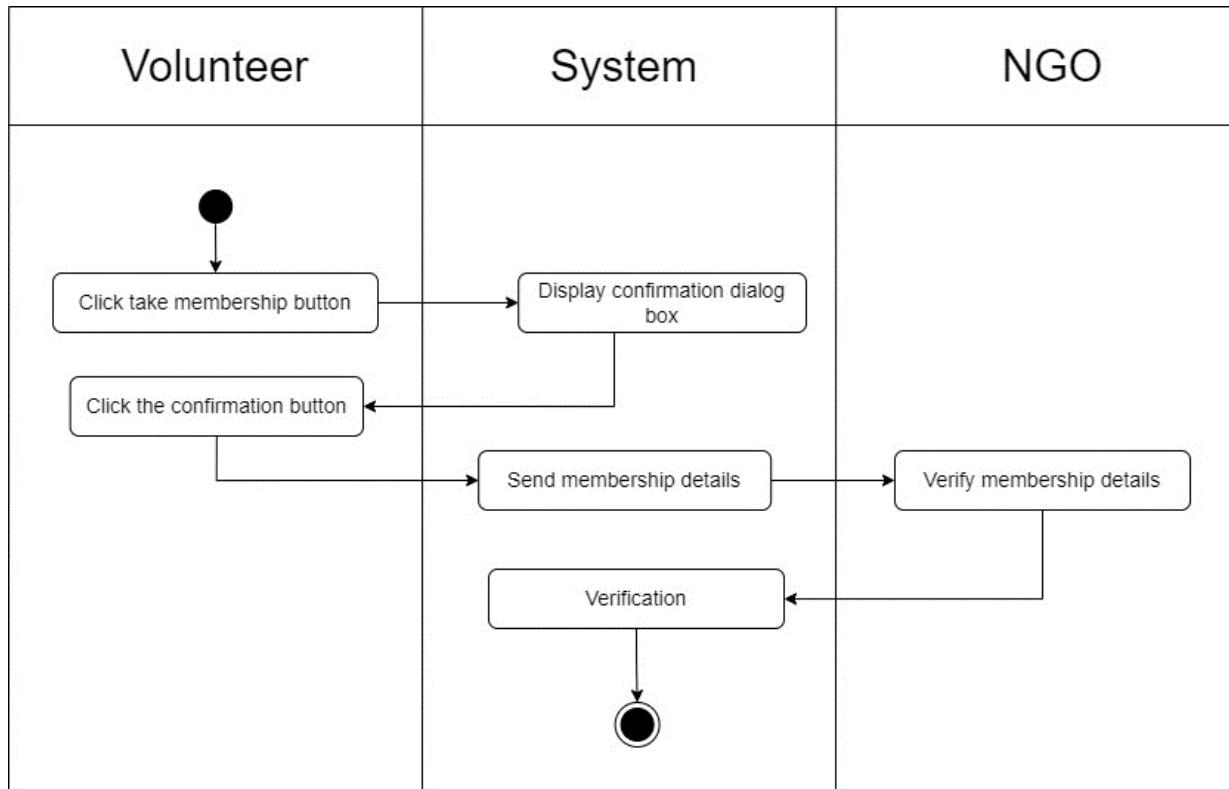


Figure 21: Take the membership activity diagram

3.6.2. Donation Activity

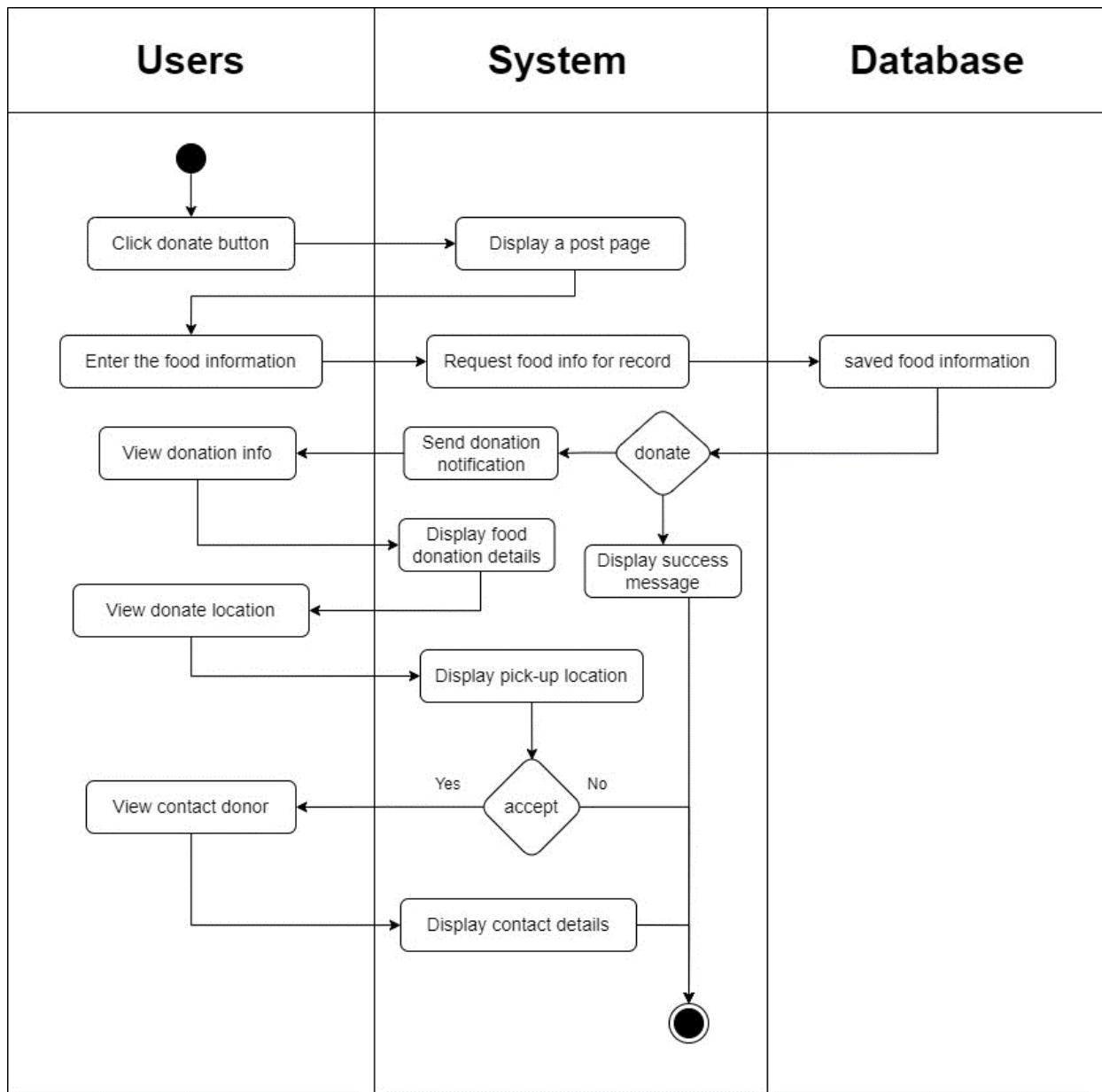


Figure 22: Food donation system activity diagram

3.6.3. View History Activity

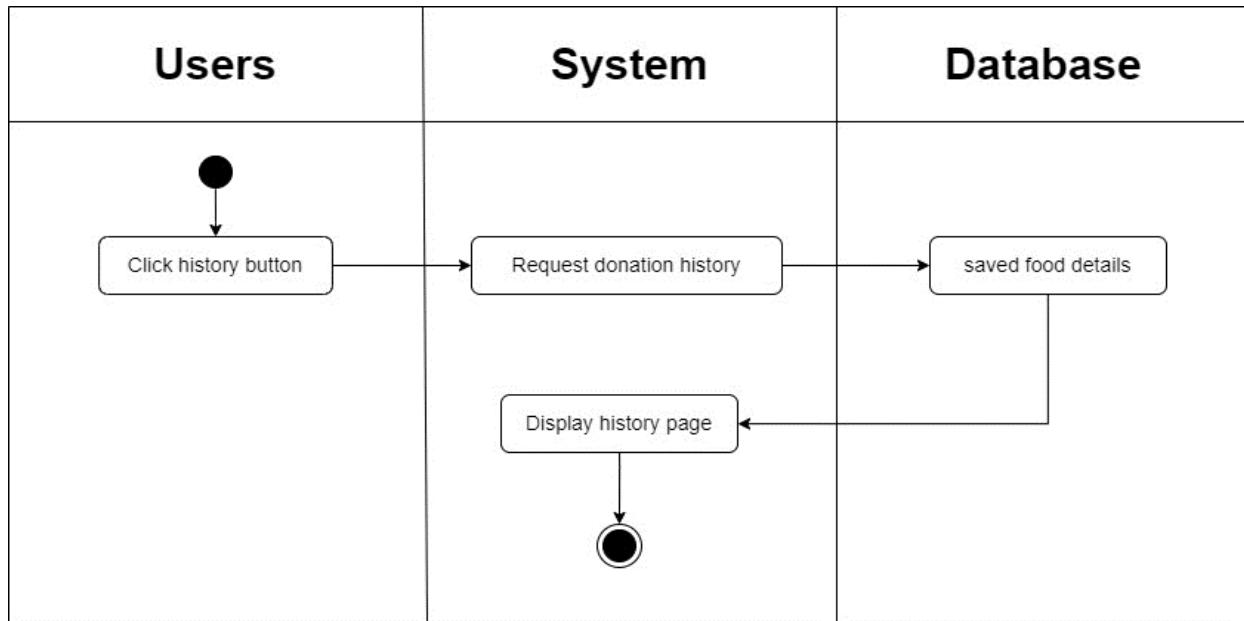


Figure 23: View the history system activity diagram.

3.6.4. Donation Rating Activity

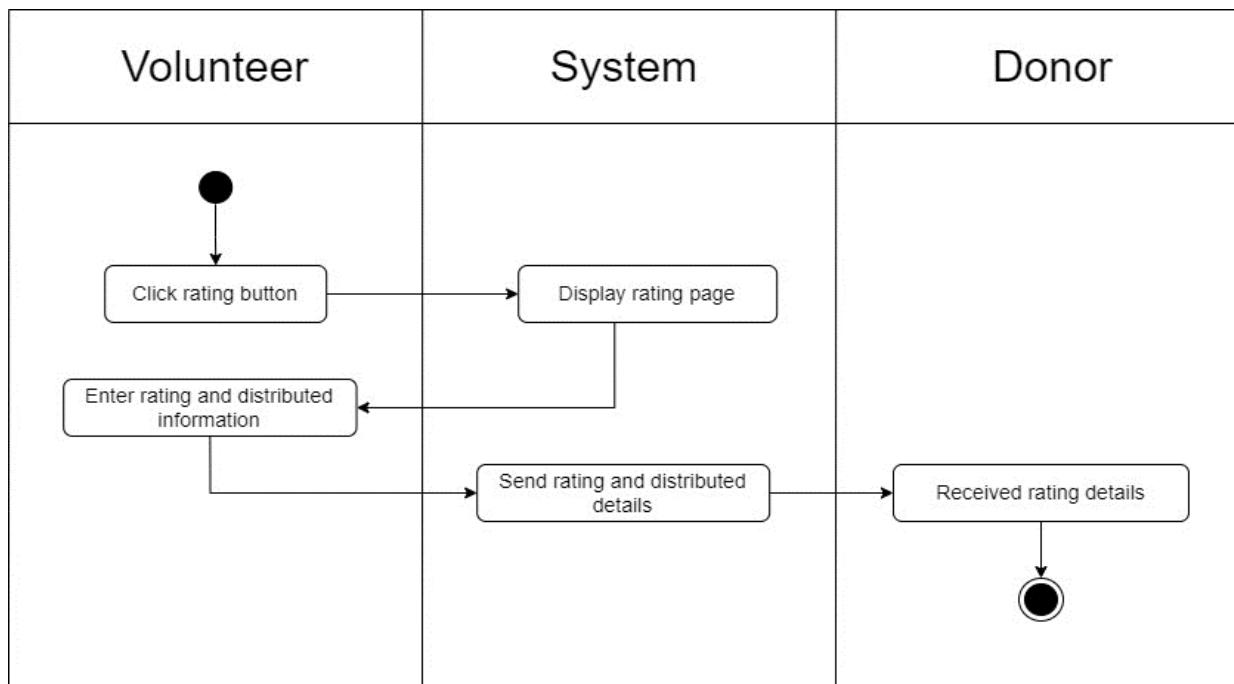


Figure 24: Rating donation system activity diagram

3.6.5. Complaint with Admin Activity

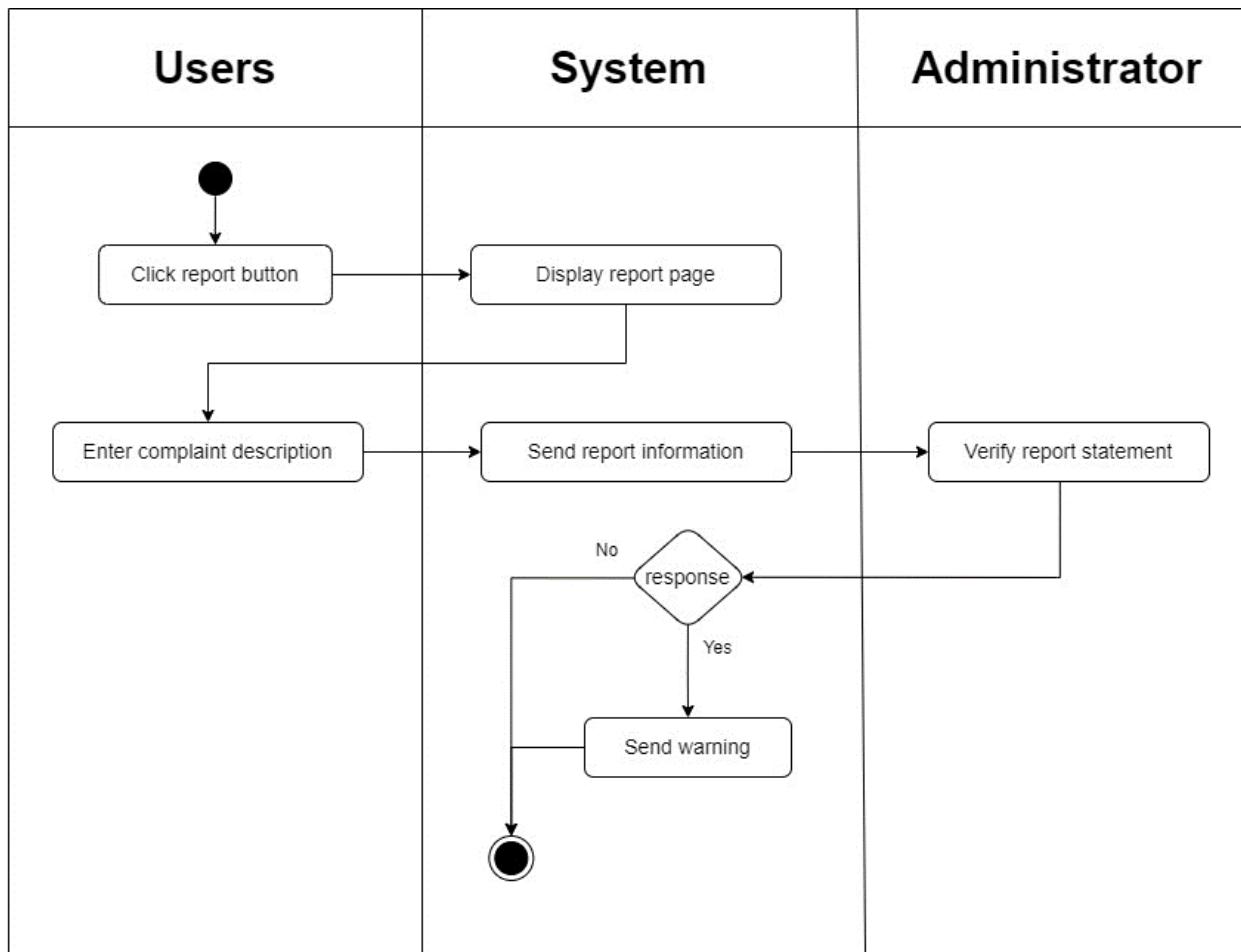


Figure 25: Complaint with admin system activity diagram

([Activity diagram for appendix](#))

3.7. Wireframes

3.7.1 Mobile UI

3.7.1.1 Walkthrough Screen

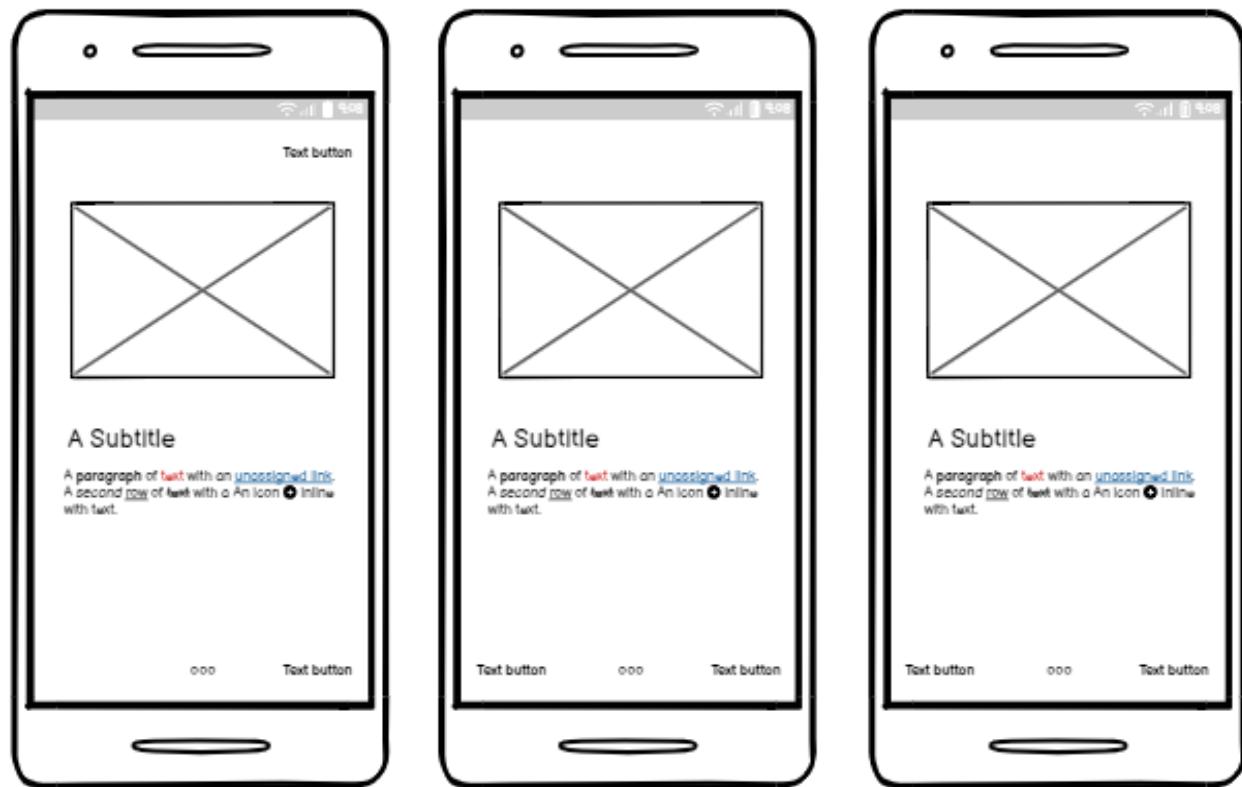


Figure 26: Walkthrough screen mobile UI

3.7.1.2 Welcome Screen

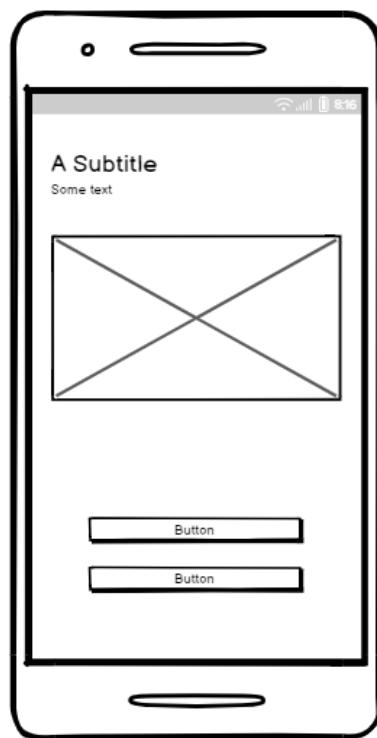


Figure 27: Welcome screen mobile UI

3.7.1.3 Login Screen

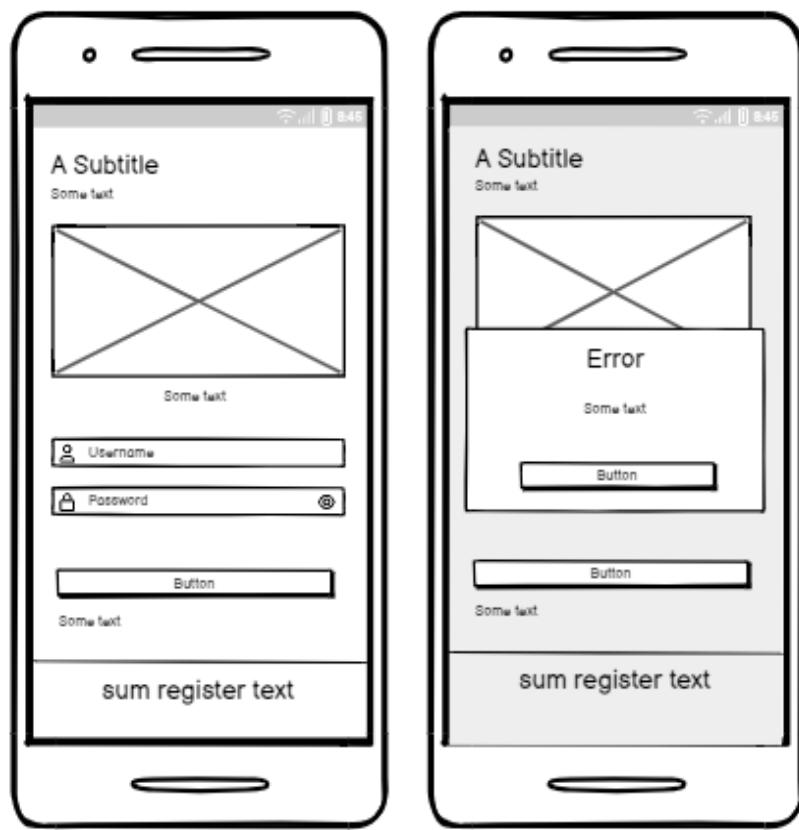


Figure 28: Login screen mobile UI

3.7.1.4 Register Screen

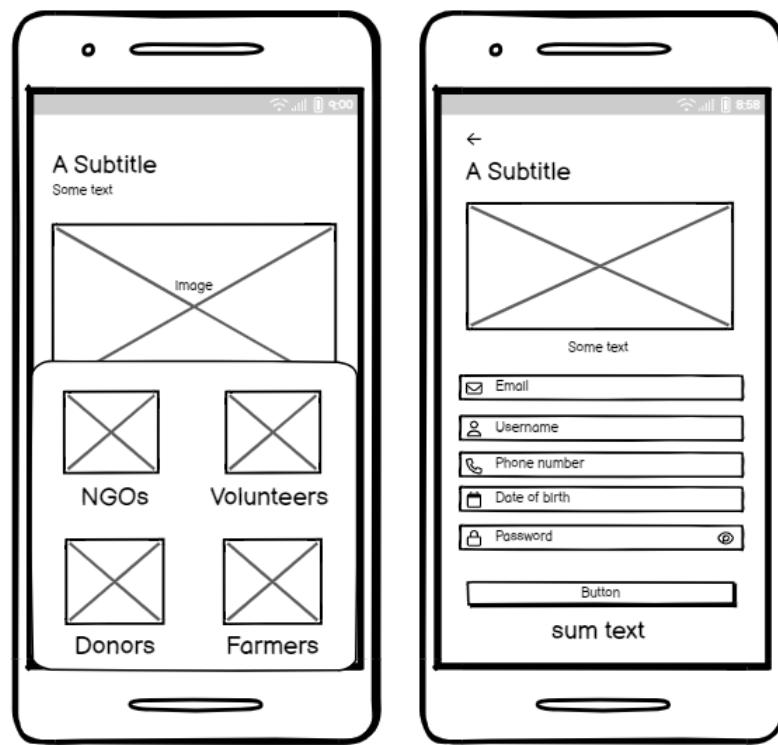


Figure 29: Register screen mobile UI

3.7.1.5 Forgot Password Screen

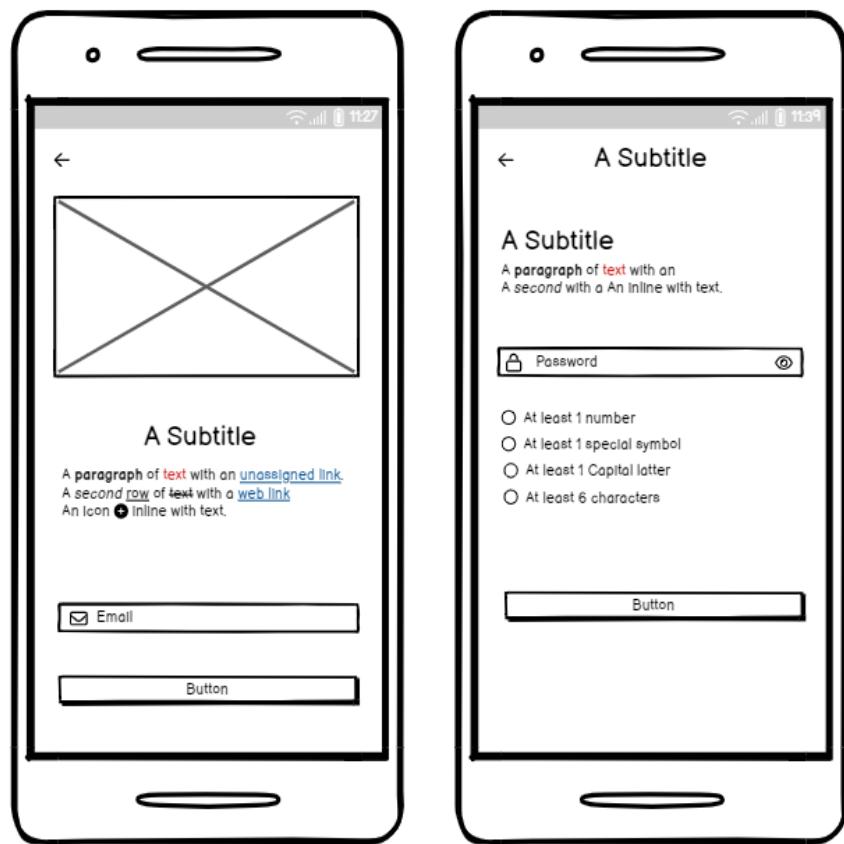


Figure 30: Forgot password screen mobile UI

3.7.1.6. Dashboard for Donors

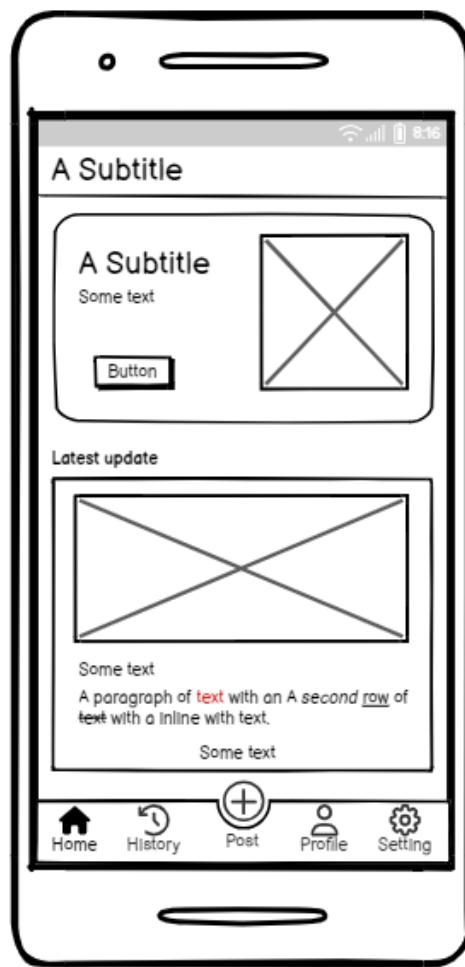


Figure 31: Donor Home screen mobile UI

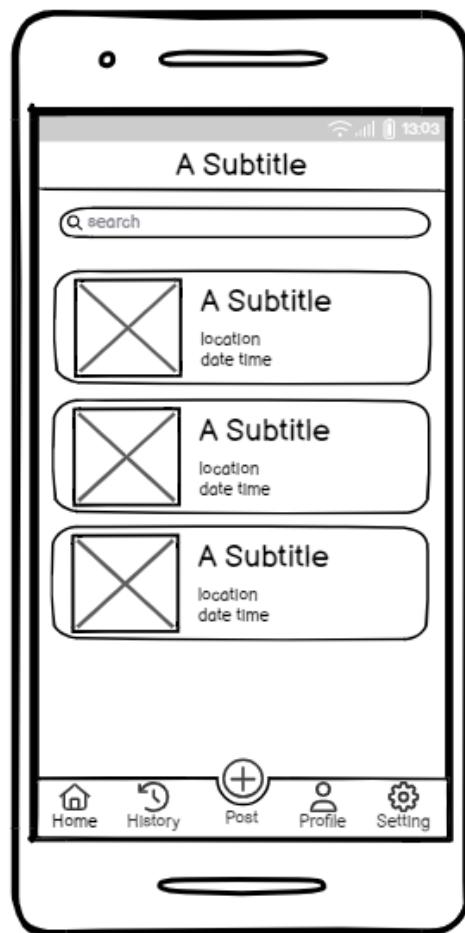


Figure 32: Donor History screen mobile UI

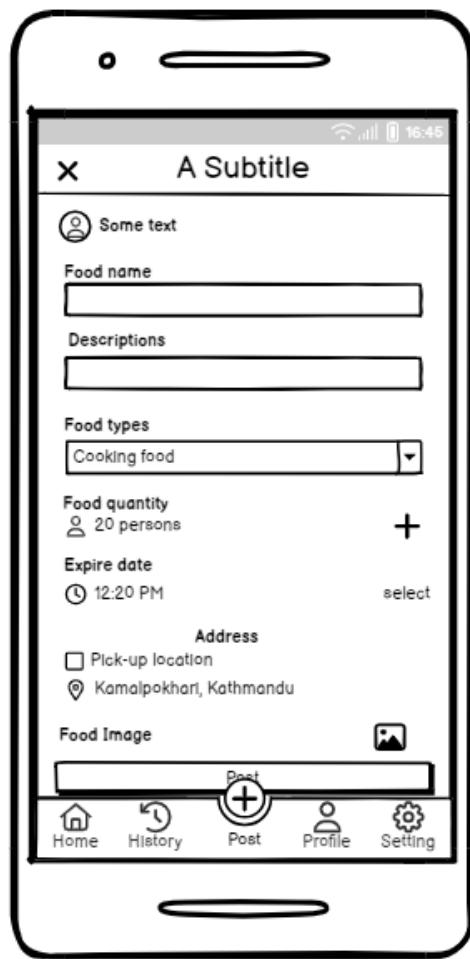


Figure 33: Donor Post/Donation screen mobile UI

3.7.1.7. Dashboard for Volunteer and Farmer

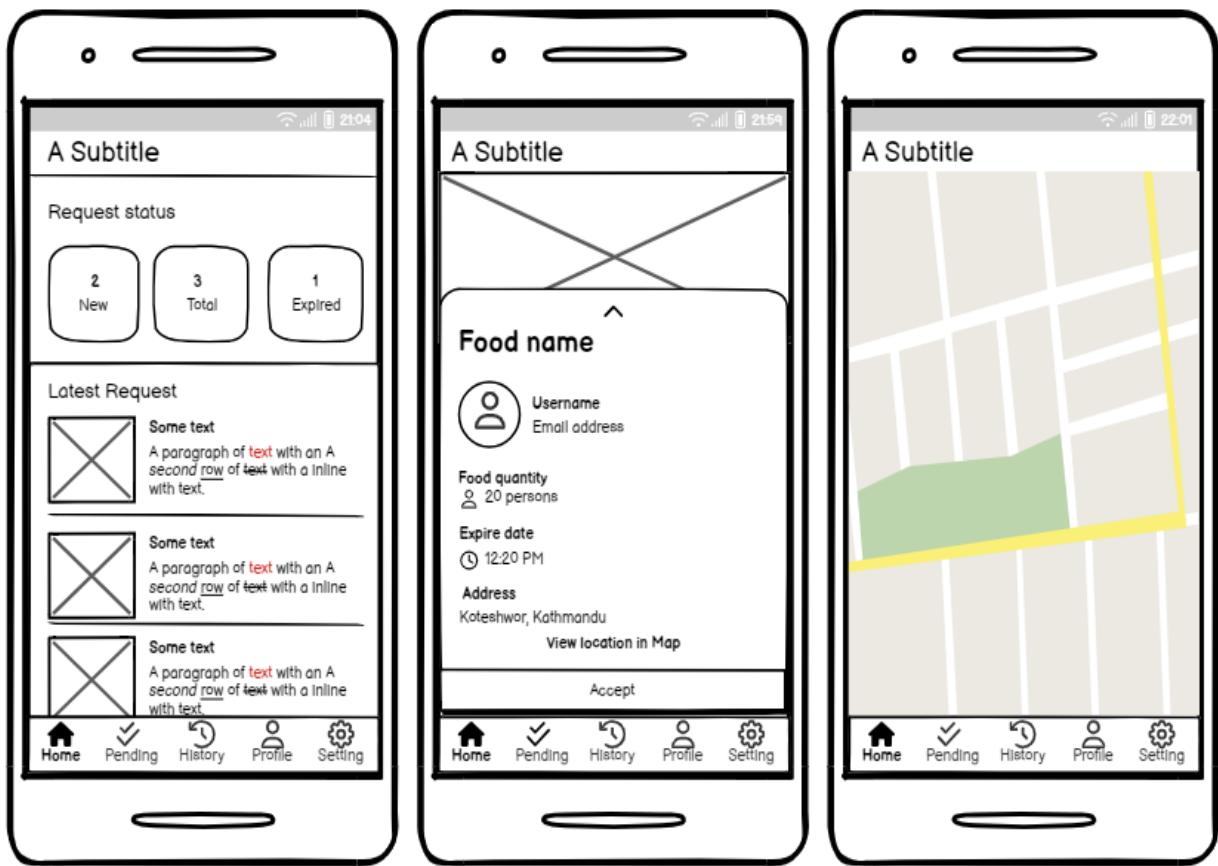


Figure 34: Volunteer Home screen UI

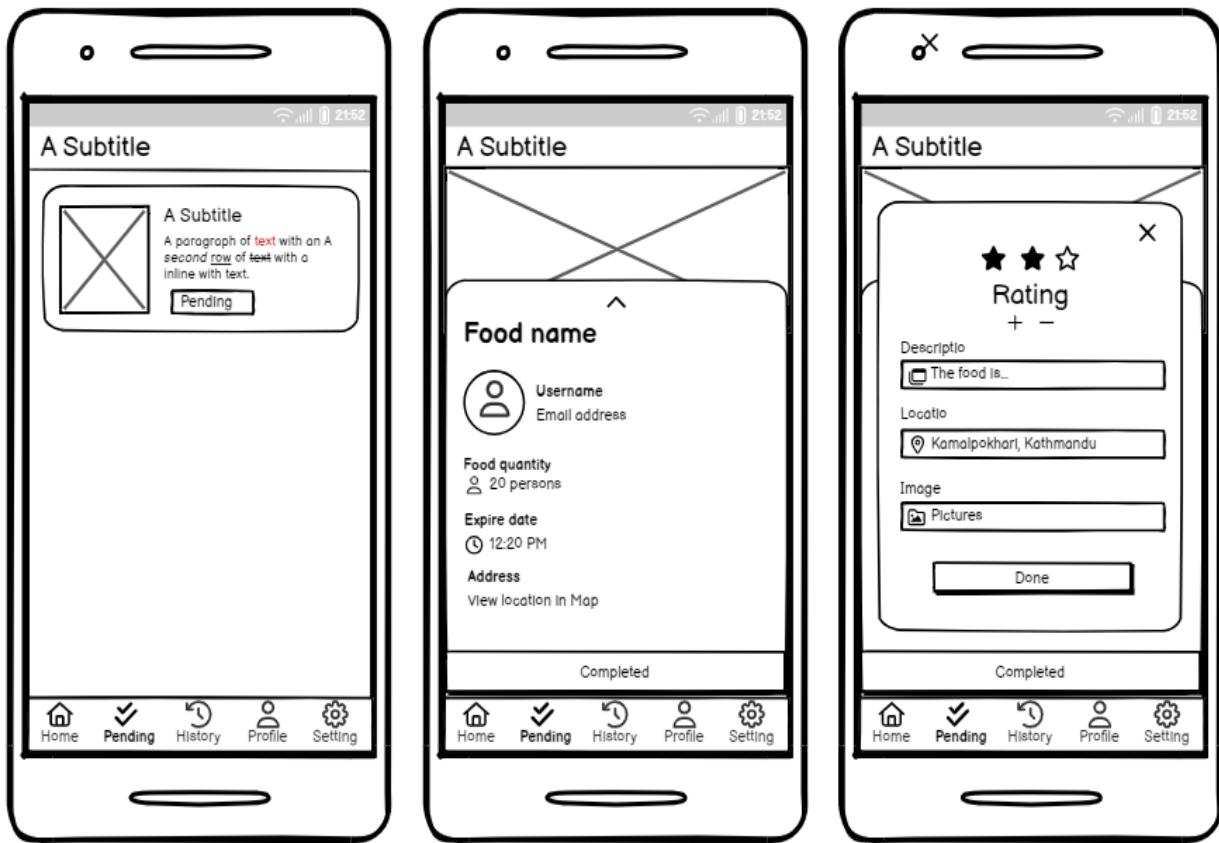


Figure 35: Volunteer Pending screen UI.

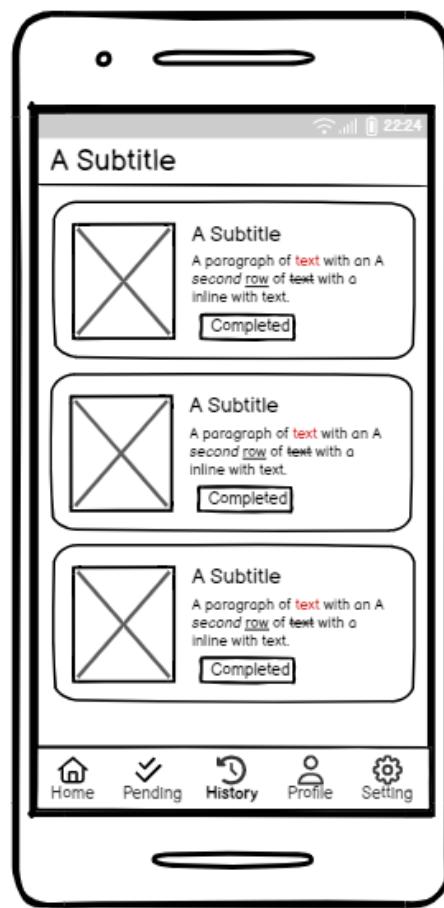


Figure 36: Volunteer History screen

3.7.1.8. NOG Dashboard

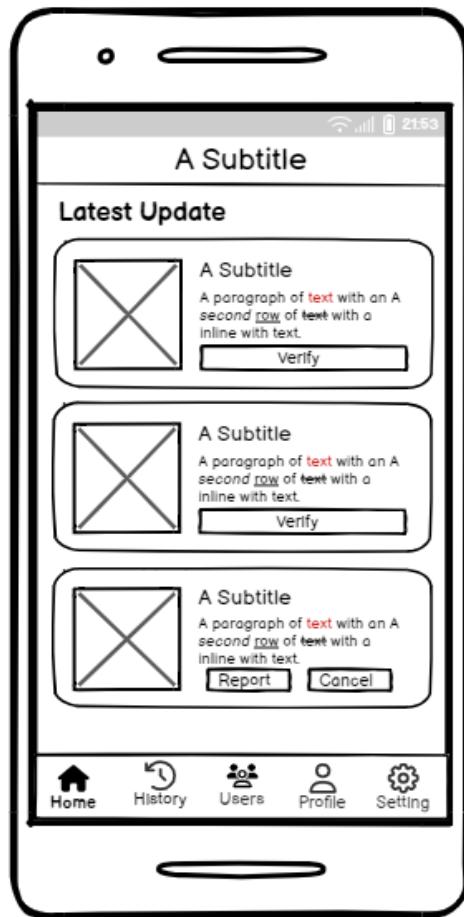


Figure 37: NGO Home screen UI

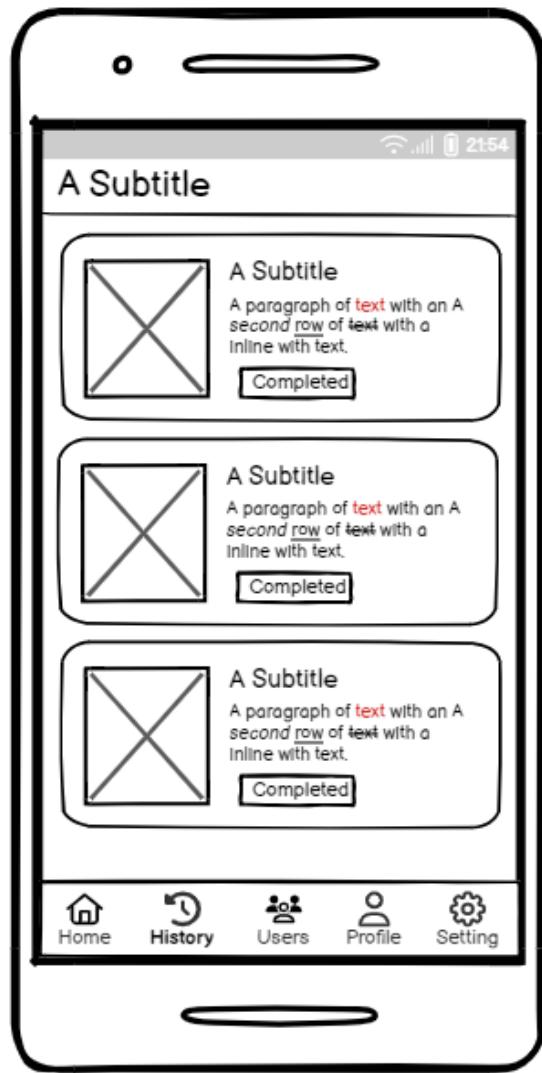


Figure 38: NGO History screen UI



Figure 39: NGO Setting Screen UI

3.7.2. Web UI for Admin

The image shows a wireframe of an admin login page. On the left, there is a 'Food Share' logo with the text 'Please login your account'. Below it are two input fields: 'Email' (placeholder 'Enter mail address') and 'Passw' (placeholder 'Enter a password'). A 'Sign Up for Balsamic Cloud' button is located below the password field. On the right, there is a large placeholder area with a large 'X' drawn through it, and a 'Waste Food Management System' logo with a long URL below it.

Figure 40: Admin login page web UI

The image shows a wireframe of an 'Forgot Password' page. It features a title 'Rest Password' at the top. Below it is a form with three input fields: 'Email', 'New Password', and 'Confirm Password'. At the bottom of the form are two buttons: 'Cancel' and 'Confirm'.

Figure 41: Admin forgot password UI.

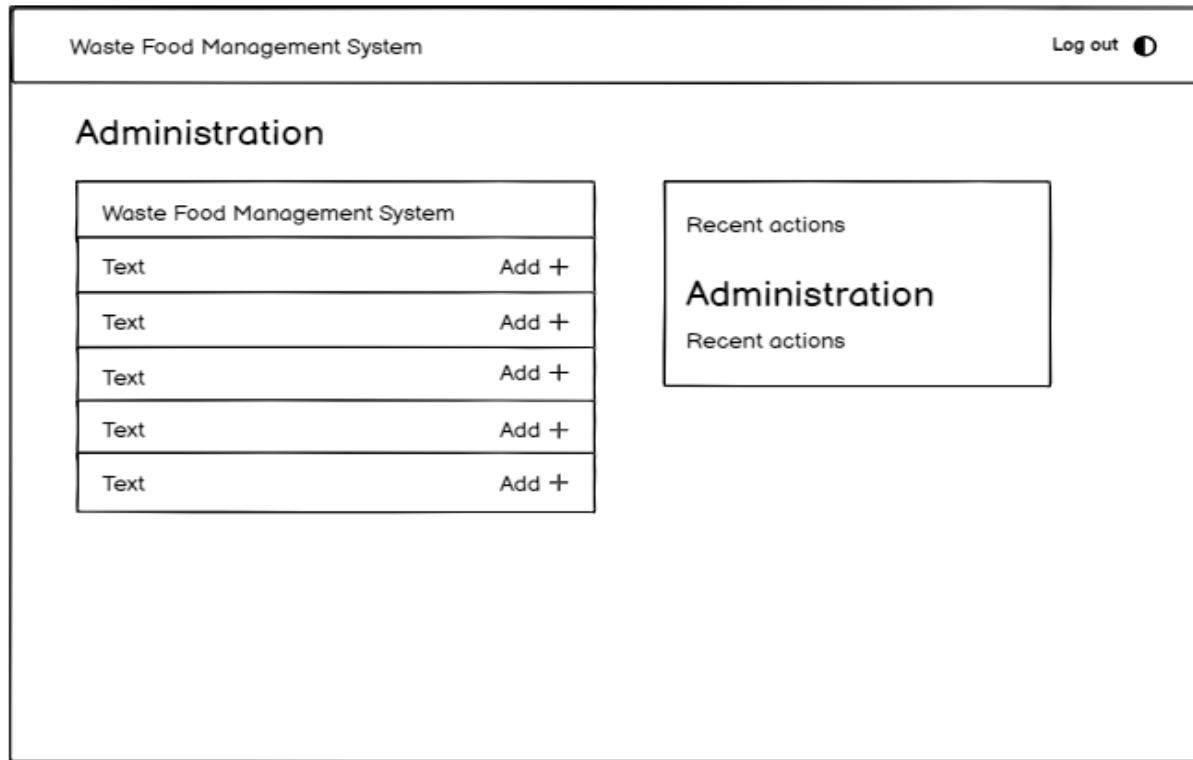


Figure 42: Admin dashboard UI

[\(Wireframe UI for appendix\)](#)

3.8. ER-Diagram

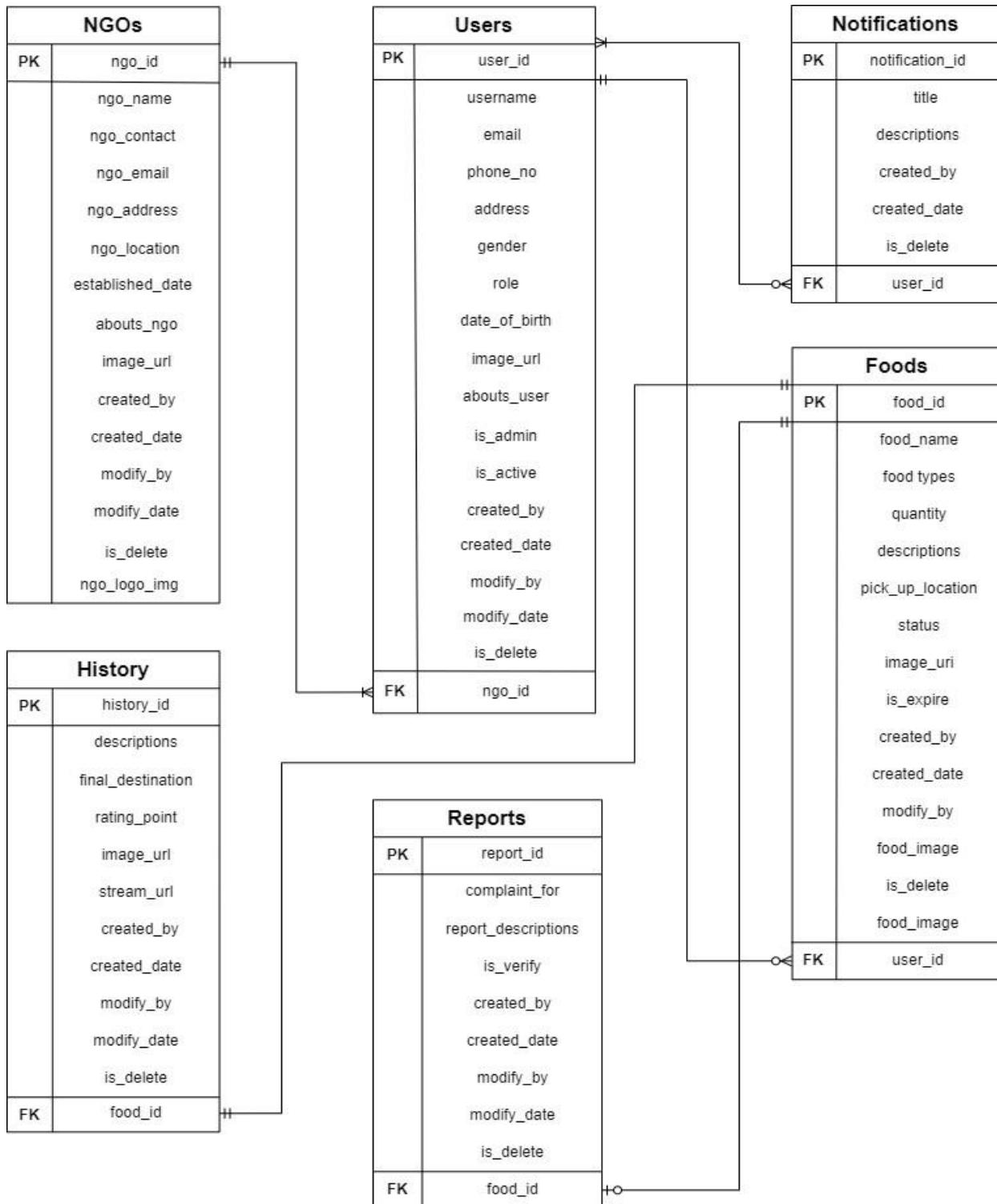


Figure 43: Class Diagram

3.9. System Architecture Diagram

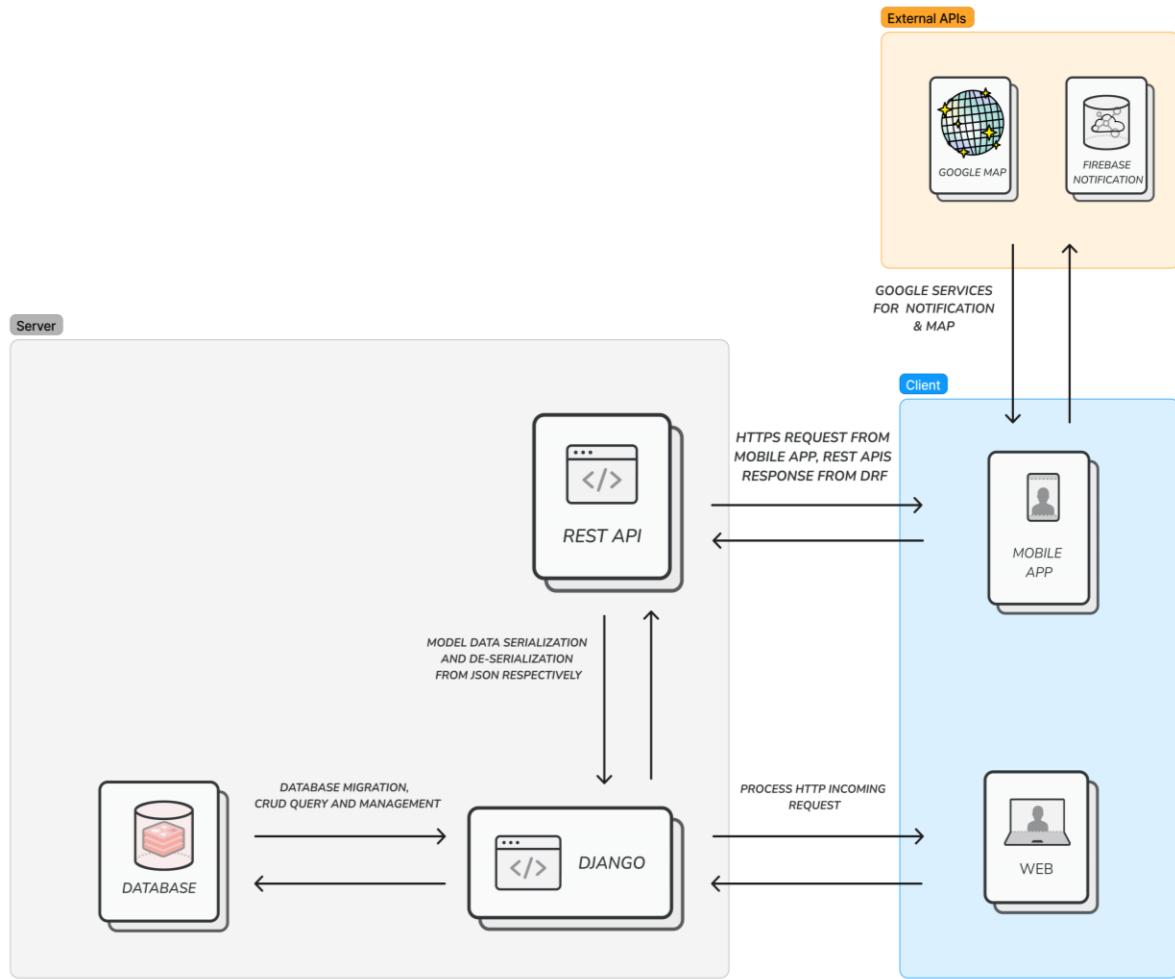


Figure 44: System Architecture Diagram

3.10. Class Diagram

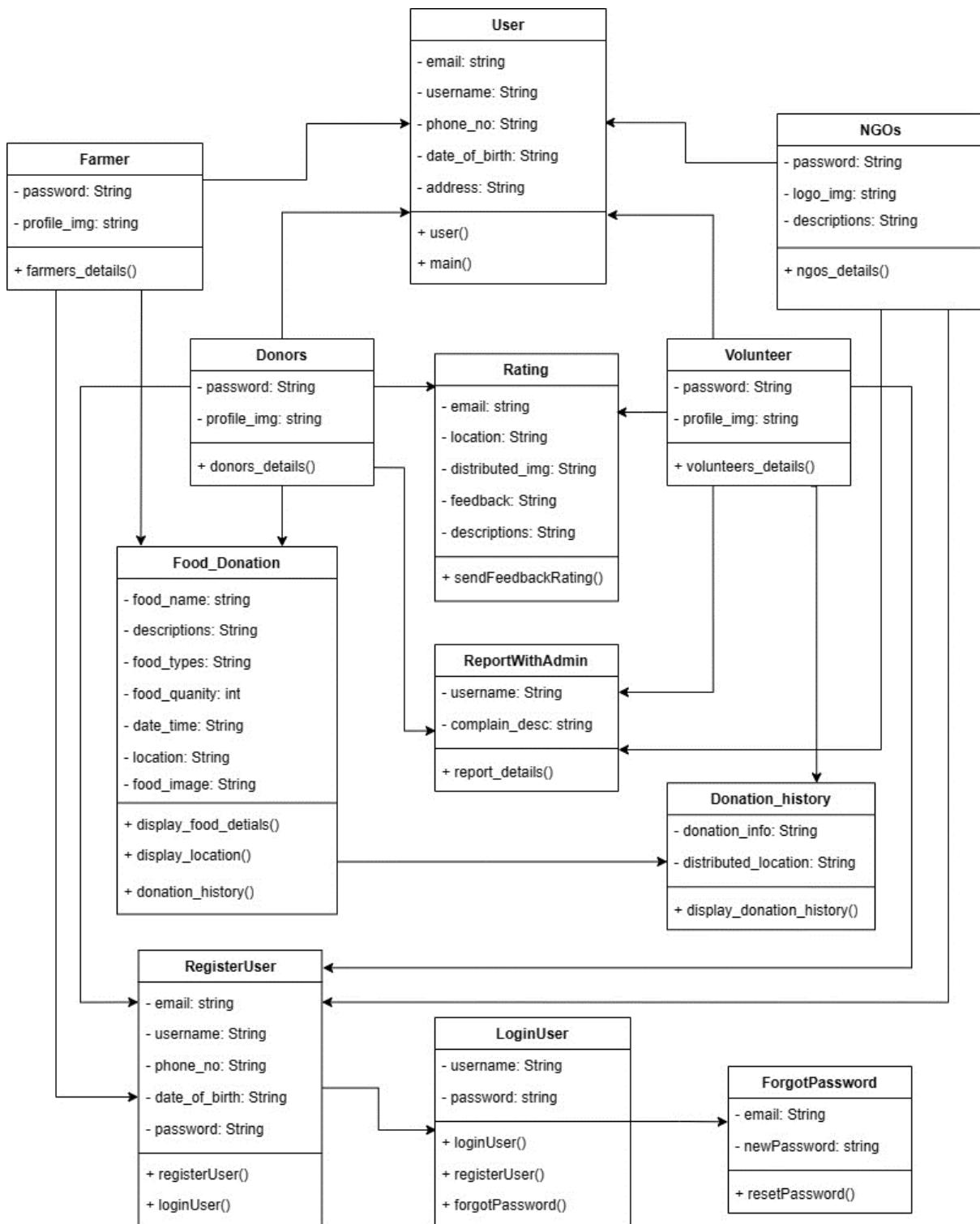


Figure 45: Class diagram of food donation application

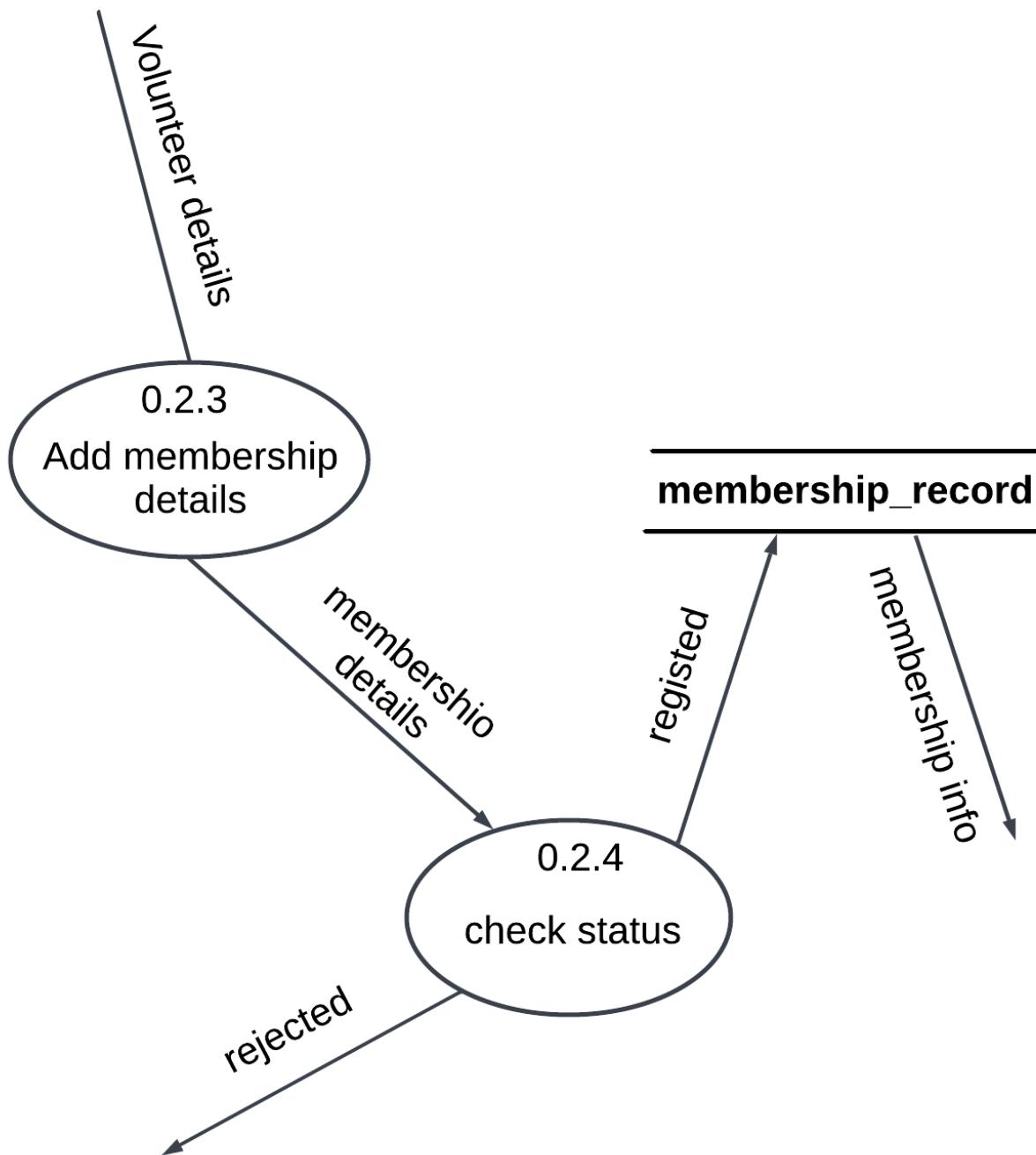
3.11. DFD's level 2**3.11.1. Take Membership**

Figure 46: Take membership details DFD Level-2

3.11.2. Food Donate

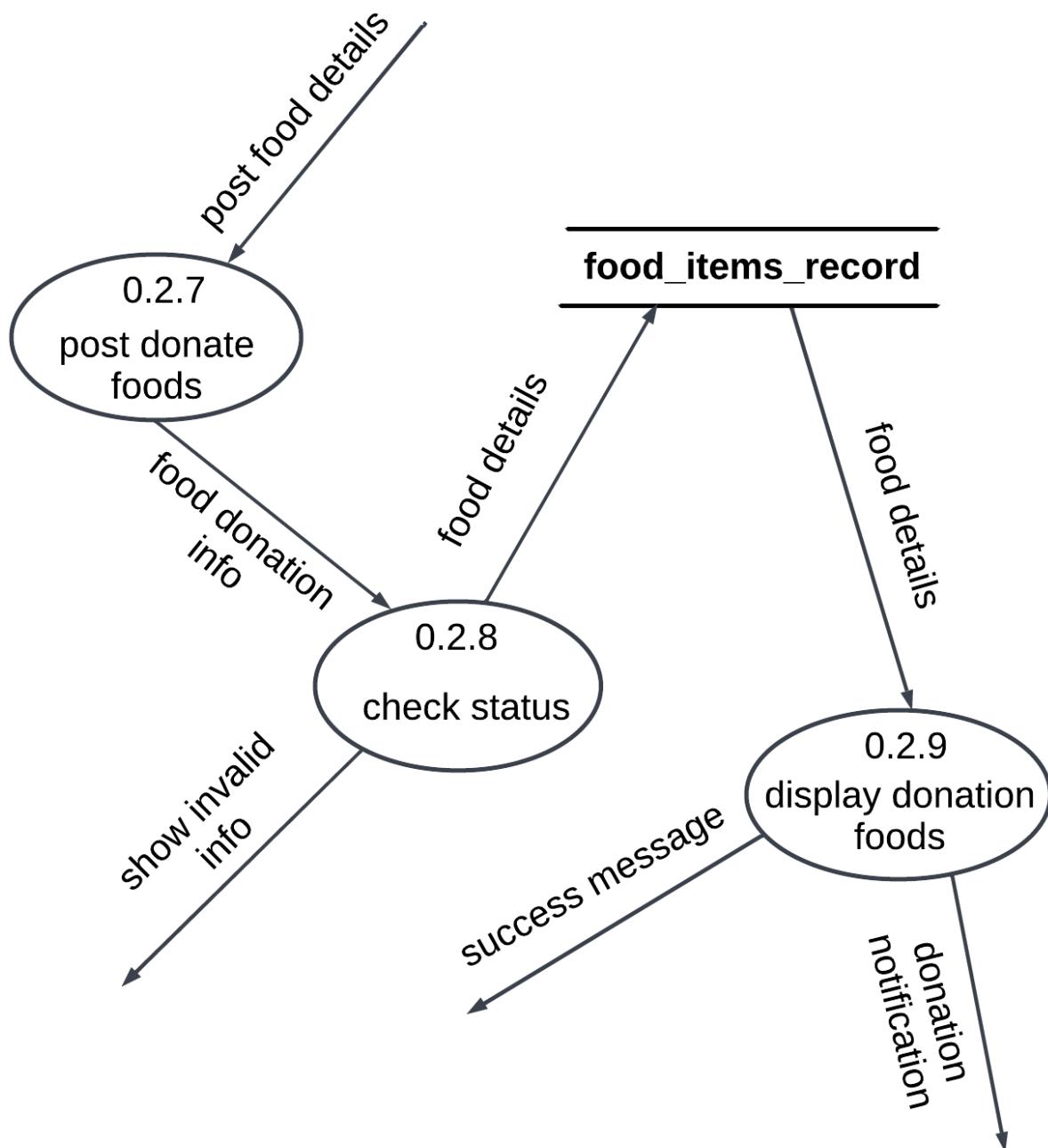


Figure 47: Food donation DFD Level-2

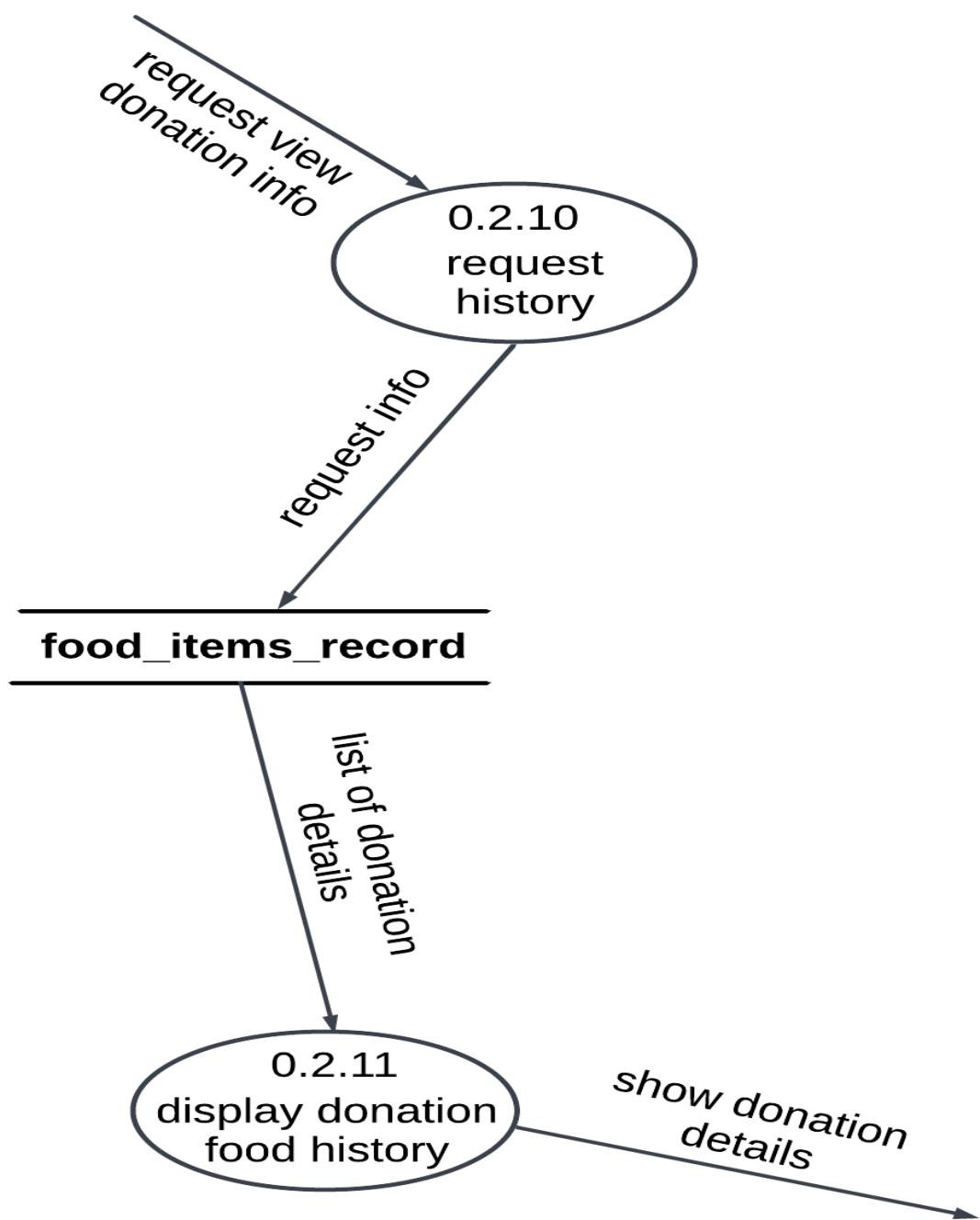
3.11.3. View Donation History

Figure 48: View Donation History DFD Level-2

3.11.4. Donation Rating

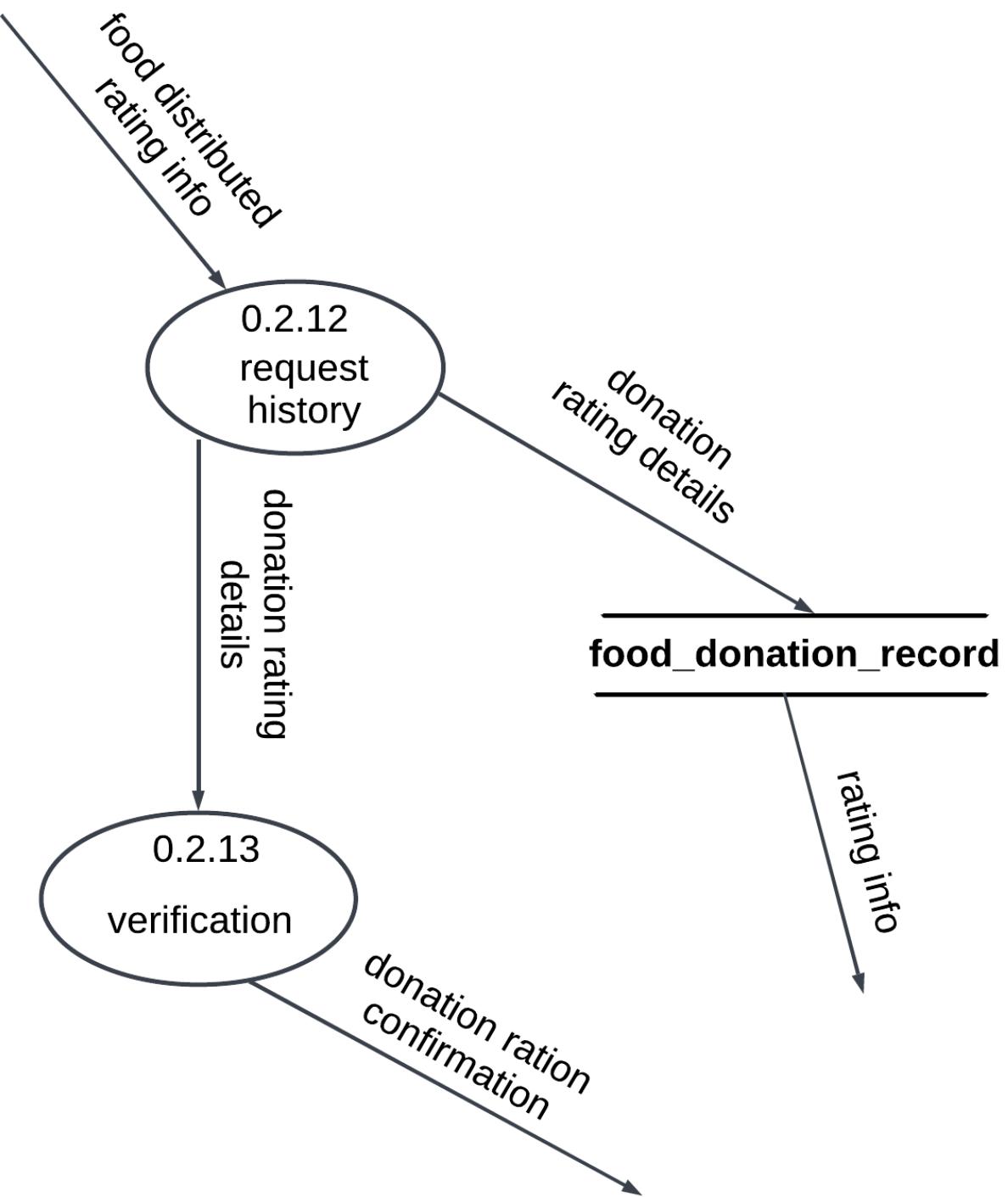


Figure 49: Donation rating DFD Level-2

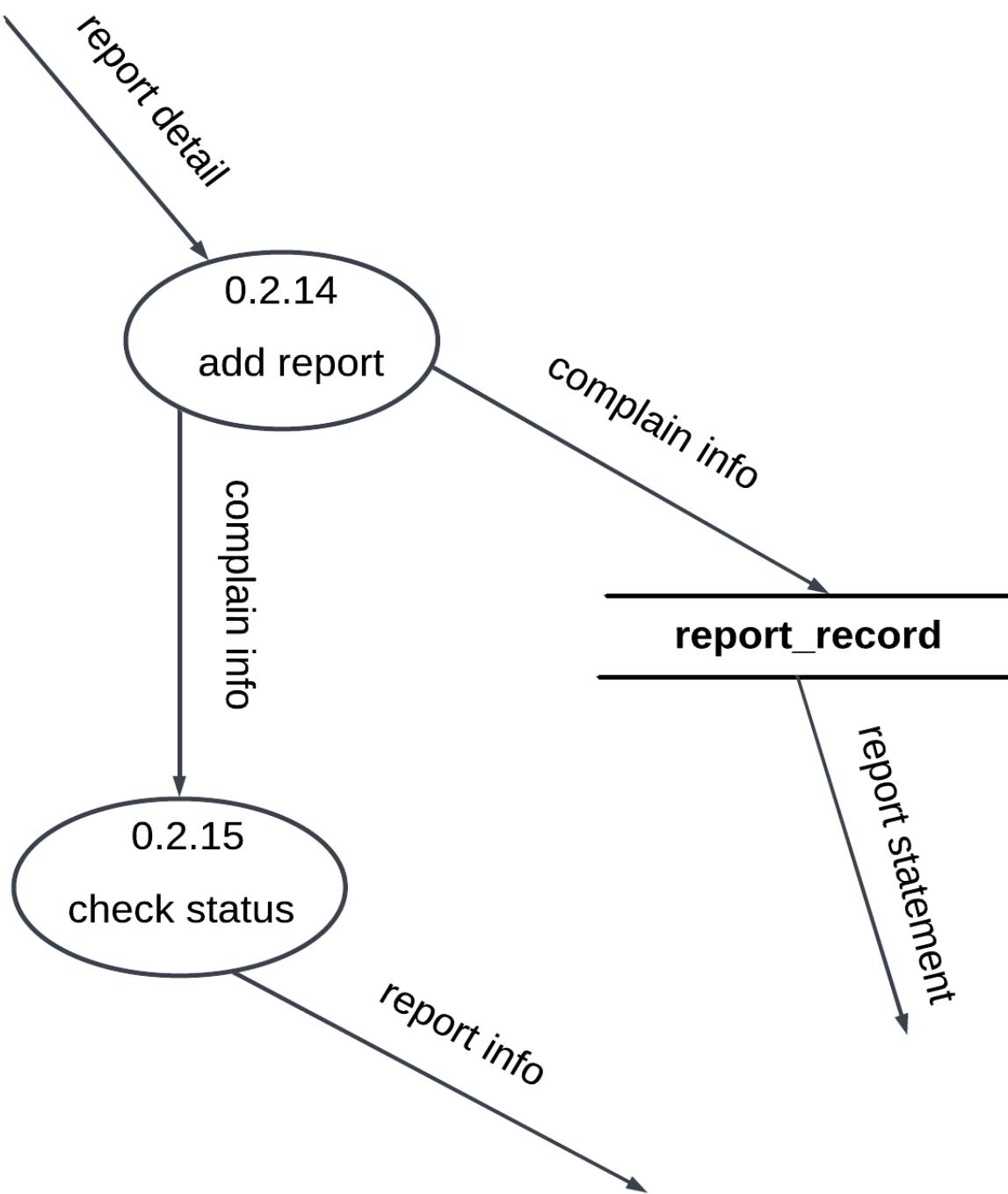
3.11.5. Complain with Admin

Figure 50: Complaint with admin DFD Level-2

[\(DFD Level-2 for appendix\)](#)

3.12. Milestone Chart

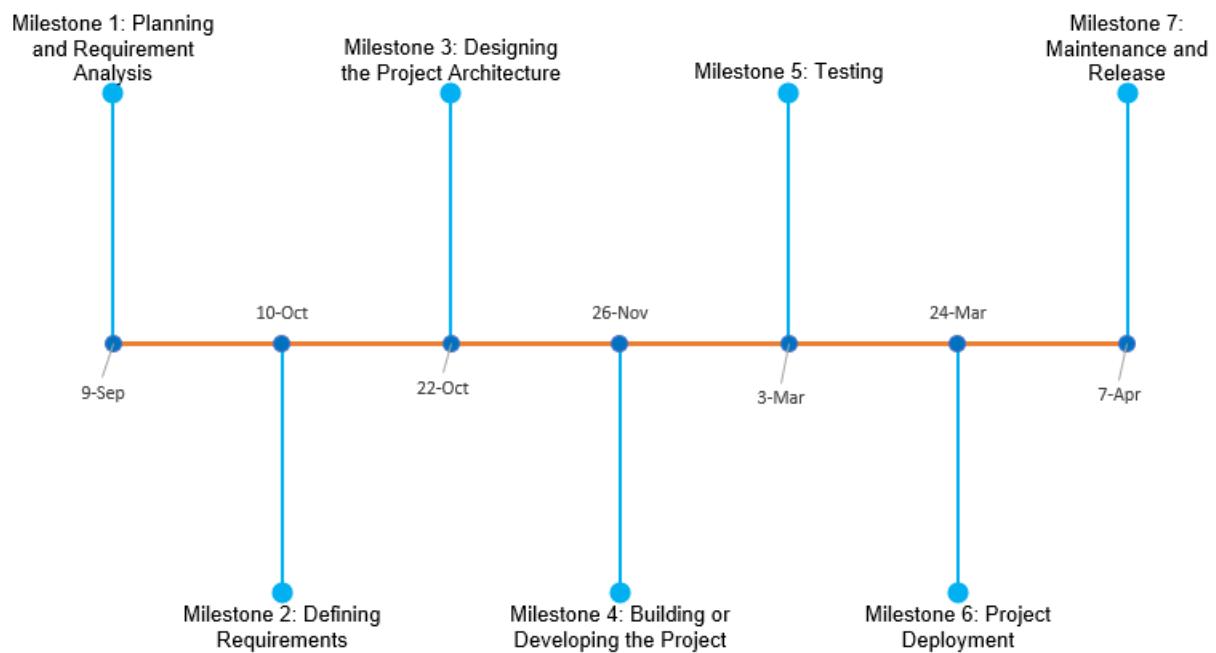


Figure 51: Milestone Chart

A	B	C	D
Date	Milestone	Height	Line Chart
9-Sep	Milestone 1: Planning and Requirement Analysis	1	0
10-Oct	Milestone 2: Defining Requirements	-1	0
22-Oct	Milestone 3: Designing the Project Architecture	1	0
26-Nov	Milestone 4: Building or Developing the Project	-1	0
3-Mar	Milestone 5: Testing	1	0
24-Mar	Milestone 6: Project Deployment	-1	0
7-Apr	Milestone 7: Maintenance and Release	1	0

Figure 52: Milestone Sheet

3.13. WBS

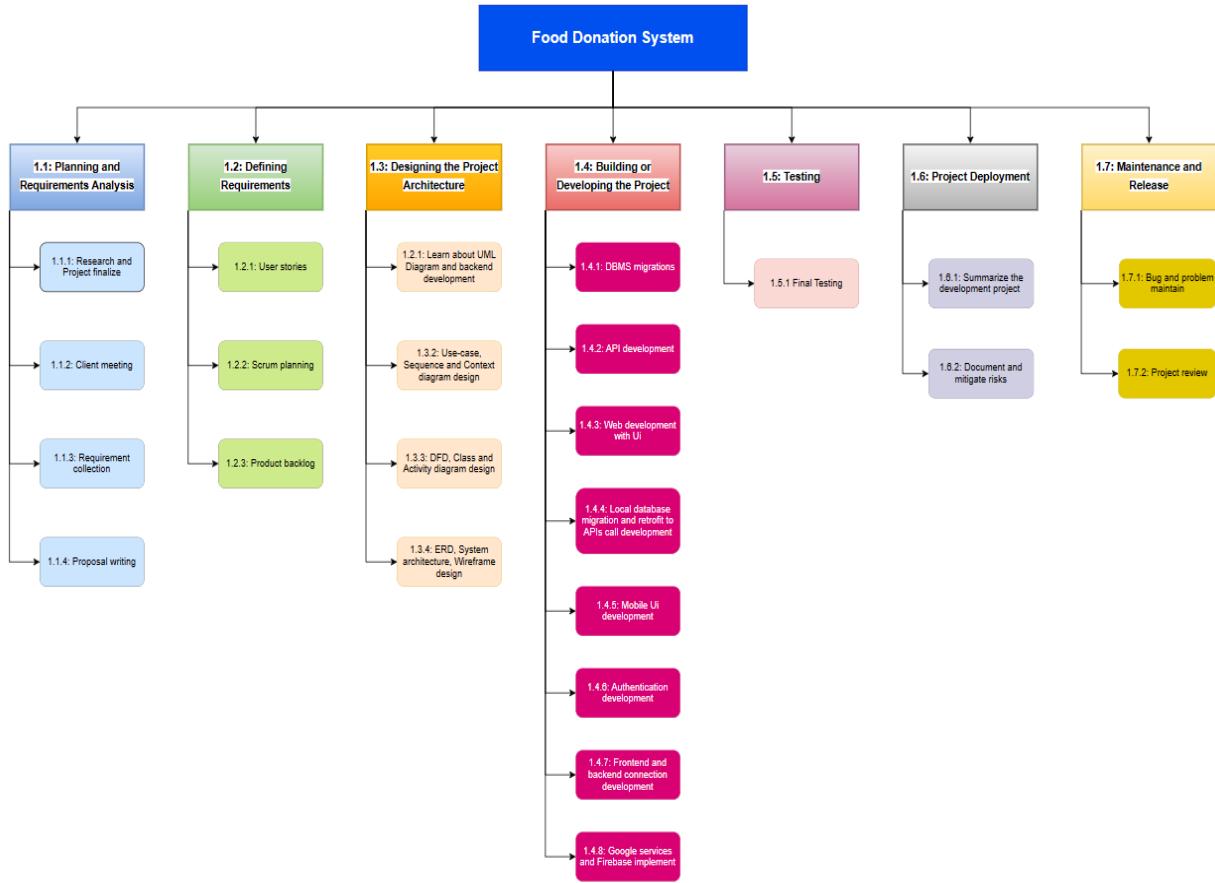
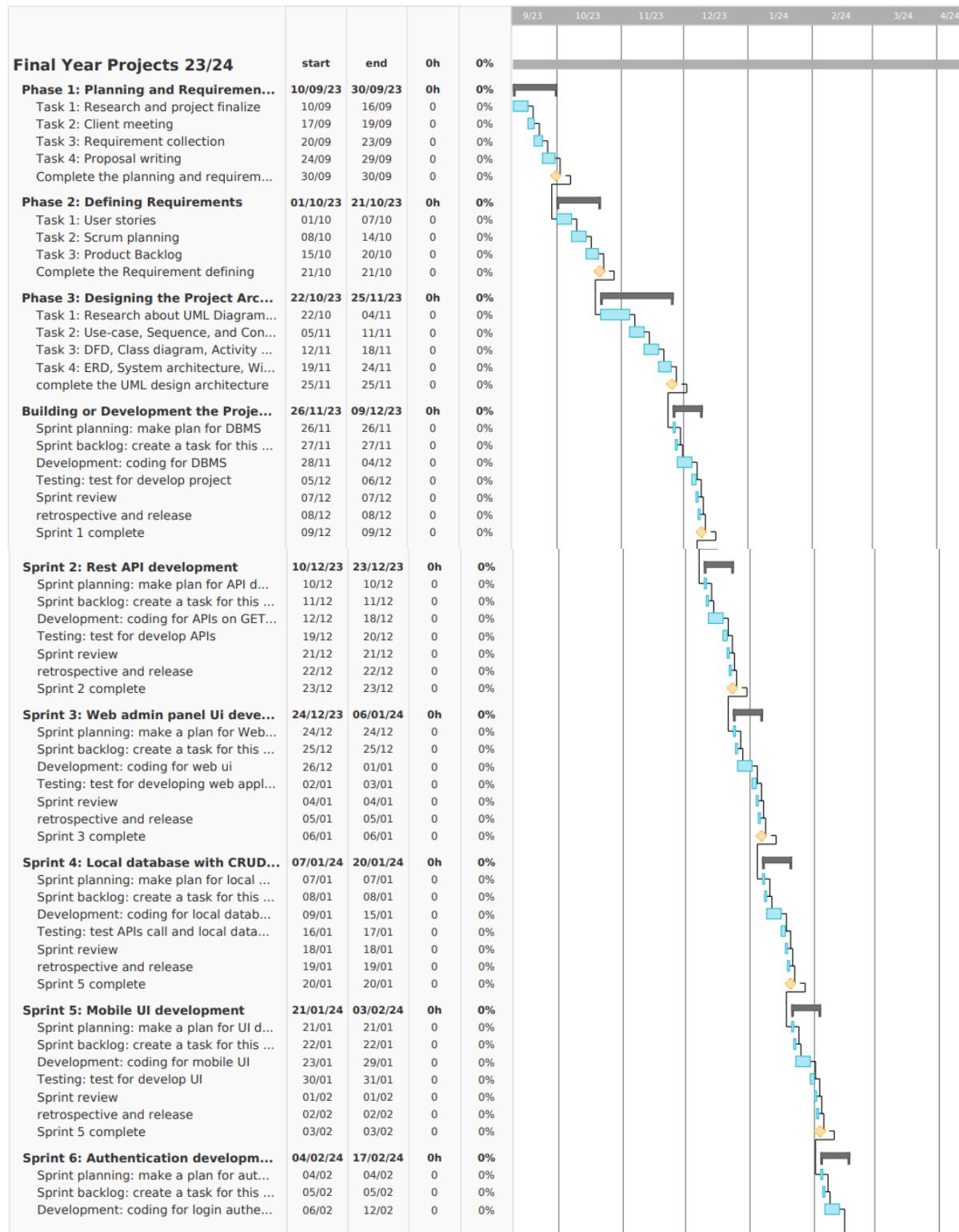


Figure 53: Work Breakdown Structure

3.14. Gantt Chart

A Gantt chart is a visual tool used in project management to display tasks or activities against time. The final year project can show the project duration and delivery time in the Gantt chart. It has seven phases of dividing the project work duration. There are some different phases like planning and requirement gathering, defining requirements, designing, development, Testing, project development, maintenance, and release etc. The development phase can be divided into the seven sprints. Sprint 1 Database development, sprint 2 API development, sprint 3 Web admin panel development, sprint 4 Mobile local development, sprint 5 Mobile app UI development, sprint 6 Authentication development, sprint 7 Frontend and backend development, and sprint 8 Google map and Firebase push notifications development.

The project duration of the Gantt Chart is given below:





Created with Free Edition

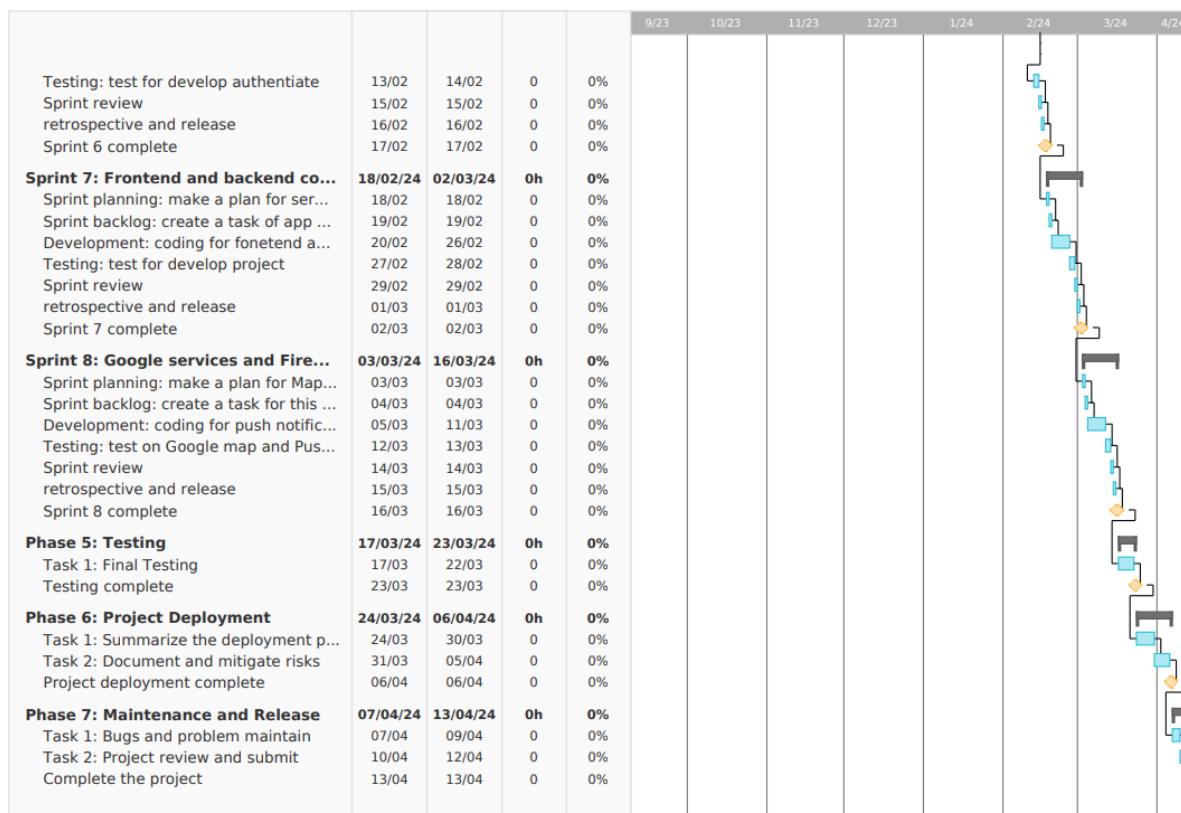


Figure 54: Gantt chart

3.15. Analysis of progress

3.15.1. Progress Table

S.N.	Task	Start date	End date	Status	Progress
1	Research the topic and client finalized	10/09/2023	20/09/2023	Completed	100%
2	Project requirements collection and project planning	21/10/2023	25/10/2023	Completed	100%
3	Proposal writing	26/10/2023	30/10/2023	Completed	100%
4	UML diagram	22/10/2023	11/11/2023	Completed	100%
5	Database development	26/11/2023	09/12/2023	Completed	100%
6	Web admin login UI, reset password UI, home page UI development	10/12/2023	16/12/2023	Completed	100%
8	Login Authentication	05/12/2023	09/12/2023	Completed	100%
7	Mobile walk-through screen UI, welcome screen UI, login, register UI, forget password screen UI and different user role dashboard UI development	27/11/2023	04/12/2023	Incomplete	50%
9	Rest API development	10/12/2023	24/12/2023	Incomplete	50%

Table 24: Progress table

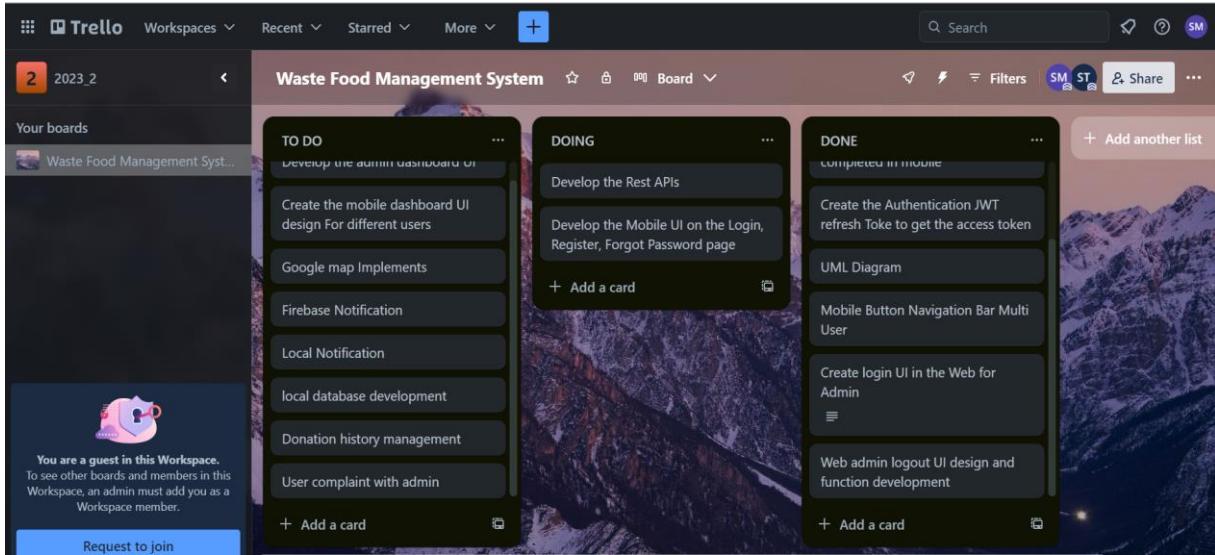


Figure 55: System development progress data Trello

3.15.2. Progress Review

The works of the project are complete according to the Gantt chart and some development is already 50% development which is up to upcoming tasks, but some tasks are not started. According to the project plan, the first is to research the project, make the plan and client finalize it in a timely completed. after meeting the client, the project requirement collection and getting the project requirement collection to move on the start the project. the first thing is database development. it is completed on project plain time then authentication, web application and mobile application UI development. The system development progress is better position because it is in a timely covered and development is fast moving on. up upcoming sprint is also better development and timely completion of the project. The client meeting and surveying to collect some features requirements and implement the system. Upcoming tasks or features are more default for development because are the major features of the project. where view location in Google map, push notifications, and data management which is a client requirement in the system. The major goal is to complete the project before the deadline.

3.15.3. Progress Timeline

S.N.	Task	Start date	End date	Status	Progress
1	Research the topic and client finalized	26/11/2023	09/12/2023	Completed	100%
2	Project requirements collection and project planning	21/10/2023	25/10/2023	Completed	100%
3	UML diagram			Completed	100%
4	Database development			Completed	100%
5	Web admin login UI, reset password UI, home page UI development			Completed	100%
6	Login Authentication	20/12/2023	27/12/2023	Completed	100%
7	Rest API development	10/12/2024	10/01/2024	Incomplete	90%
8	Mobile walk-through screen UI, welcome screen UI, login, register UI, forget password screen UI and different user role dashboard UI development	20/01/2024	31/01/2024	Incomplete	50%
9	Data management from the web admin side	02/02/2024	10/02/2024	Not Started	0%

	(add, update, delete)				
10	Mobile to the user, food details are added, updated, delete the rough of rest API with Food donation history management	11/02/2024	20/02/2024	Not Started	0%
11	Local and Firebase push notification	03/03/2024	08/03/2024	Not Started	0%
12	Google map	09/03/2024	16/03/2024	Not Started	0%

Table 25: Progress timeline in table

Research to topic	Durations	Stat Date	End Date	Status
Phase 1: Planning and Requirement Analysis				100%
Task 1: Research and project finalize	3 weeks	10/09/2023	30/09/2023	Completed
Task 2: Client meeting	7 days	10/09/2023	16/09/2023	
Task 3: Requirement collection	3 days	17/09/2023	19/09/2023	
Task 4: Proposal writing	4 days	20/09/2023	23/09/2023	
	7 days	24/09/2023	30/09/2023	
Phase 2: Defining Requirements	2 weeks	1/10/2023	21/10/2023	100%
Task 1: User stories	7 days	1/10/2023	7/10/2023	Completed
Task 2: Scrum planning	7 days	8/10/2023	14/10/2023	
Task 3: Product Backlog	7 days	15/10/2023	21/10/2023	
Phase 3: Designing the Project Architecture	4 weeks	22/10/2023	11/11/2023	100%
Task 1: Research about UML Diagram and backend development	14 days	22/10/2023	4/11/2023	Completed
Task 2: Usecase, Sequence, and Context diagram develop	7 days	5/11/2023	11/11/2023	
Task 3: DFD, Class diagram, Activity diagram develop	7 days	12/11/2023	18/11/2023	
Task 4: ERD, System architecture, Wireframe develop	7 days	19/11/2023	25/11/2023	
Phase 4: Building or Developing the Project	14 weeks	26/11/2023	16/3/2024	Pending
Sprint 1: DBMS creation	2 weeks	26/11/2023	9/12/2023	100%
	Sprint Planning	1 day	26/11/2023	26/11/2023
	Sprint Backlog	1 day	27/11/2023	27/11/2023
	Development	7 days	28/11/2023	4/12/2023
	Testing	2 days	5/12/2023	6/12/2023
	Sprint Review	1 day	7/12/2023	7/12/2023
	Retrospective and Release	1 day	8/12/2023	9/12/2023
Sprint 2: Rest API development	Same process of Sprint 1	2 weeks	10/12/2023	23/12/2023
Sprint 3: Web admin panel UI development	Same process of Sprint 1	2 weeks	24/12/2023	6/1/2024
Sprint 4: Local database with CRUD and Retrofit to API implement	Same process of Sprint 1	2 weeks	7/1/2024	20/1/2024
Sprint 5: Mobile UI development	Same process of Sprint 1	2 weeks	21/1/2024	03/02/2024
Sprint 6: Authentication development	Same process of Sprint 1	2 weeks	04/02/2024	17/02/2024
Sprint 7: Frontend and backend connection	Same process of Sprint 1	2 weeks	18/02/2024	2/3/2024
Sprint 8: Google services and Firebase implement	Same process of Sprint 1	2 weeks	3/3/2024	16/03/2024
Phase 5: Testing	1 week	17/3/2024	23/03/2024	
Task 1: Final Testing	7 days	17/3/2024	23/03/2024	Due
Phase 6: Project Deployment	2 weeks	24/03/2024	06/04/2024	
Task 1: Summarize the deployment project	7 days	24/03/2024	30/03/2024	Due
Task 2: Document and mitigate risks	7 days	31/03/2024	6/4/2024	
Phase 7: Maintenance and Release	1 week	7/4/2024	13/4/2024	
Task 1: Bugs and problem maintain	3 days	7/4/2024	9/4/2024	Due
Task 2: Project review and submit	4 days	10/4/2024	13/4/2024	

Figure 56: Progress timeline figure

3.15.4. Action Plan

The project began with intensive groundwork researching, identifying requirements, and creating UML diagrams. The initial step involved creating a strong MySQL database capable of accommodating various data changes. Subsequently, the focus shifted to constructing a Django Rest API to facilitate seamless communication between the front and back ends of the system. The web had completed the admin login page, logout, reset password, and home page design but the mobile has only a walkthrough screen, welcome screen, login, register and forgot password page's UI design.

After completing the database creation then develop the Django Rest API for data binding for frontend and backend connection. The next plan is to complete frontend UI development on Mobile and Web both systems. After completing the UI development then will start the mobile local database development for some static data saving it can help for not needed for API calls. The local database has used the room database for suitable mobile devices. The admin authentication has been already completed for web users but for the mobile users it is pending where I will do more research on error fixes from mobile to login and register. When the login and register phase is completed, they will develop the user dashboard where different users can navigate to actions. All the user's dashboards completed the design of the API call to the food donation part developed. Once the frontend and backend connection are completed, we will start the Firebase push notification and local notification development after that google map implemented to show to donation location. The final testing will start if any errors to solve and move to the next work. The project can be deployed and delivered the project.

The mobile stack is complete then check the web where mobile users can data in the backend server to the database it can be modified or updated. The system is fully developing the web system used by admin action and mobile for donors, farmers, volunteers, and Ngo admin can go to action for donation system.

4. Future Work

Moving forward, the focus will be on optimizing the user experience across mobile and web platforms. I'll enhance the mobile interface using Kotlin kit Jetpack Compose while employing HTML, CSS, and Bootstrap for the web. Additionally, plan to establish a local mobile database to expedite operations by storing certain data locally. Addressing the ongoing challenge with mobile user logins is a priority. Once resolved, we'll delve into creating intuitive user dashboards, ensuring seamless navigation and interaction. Integration between these dashboards and the crucial food donation aspect remains a key focus. As these components align, I'll introduce engaging features like notifications and Google Maps integration to highlight donation locations. Thorough testing will precede the project's completion to guarantee a flawless user experience upon launch. Upcoming tasks involve refining mobile-server synchronization and ensuring comprehensive coverage within my web system.

This inclusivity encompasses functionalities catering to administrators, donors, farmers, volunteers, and NGO admins, streamlining donation processes for all users. my goal is a strong and user-friendly platform accessible to diverse user bases, delivering an exceptional experience across all interactions. In the future, the plan is to expand this system into a fully developed multi-platform solution, catering to Android, iOS, and web users, providing access to a wide range of individuals. The goal is to make it easier for various types of social workers to assist impoverished and homeless individuals more effectively. Enhancements to the system will include additional messaging features to facilitate better communication and collaboration among users. The aim is continuous expansion and improvement to better serve the needs of those in need.

The future of due work is explained in table form and how to complete the project in the same duration.

All the future work is shown in the table as given below:

S.N.	Due future work	Descriptions
1.	Web dashboard for admin and mobile dashboard for different user	The web application for admin data management where due to work on the development admin dashboard. Where the admin views the user and more food-related data in tabular form. It can be completed in the upcoming next sprint. Where also mobile has many pages available for the user role dashboard. It also competes in the same sprint.
2.	Login, register and forget password authentication from mobile device	The login register, and forget password is due in mobile from mobile to backend connection. It is moving on development and the features are pending where this will be complete in a few days.
3.	Data add, update, and delete action for admin	In the admin panel data insert, update and deleted function work is due it can be completed in the coming sprint. Where admin action this feature.
4.	Local databases are developed for the temporary static data store	The local database is the due feature work of the system where static or temporary data is saved for the local database. The local database for using the room database.
5.	Frontend to backend data phase and post for data store	The frontend to backend data phase and post for data store means the mobile user can do some work and can also change the backend database. These features are most important for the system. Users can create, add, update, delete or modify some data that can change and back to give the information they used it

		can be changed.
6.	Local and Firebase push notification	The local and firebase push notification is the second main feature of the project where donors can post the food details and then notify the volunteers that new food is available for donation. So, it can be moved to the coming sprint using the Firebase.
7.	View map using Google API	The Google map feature is another major feature of the system, and it is also due to the work feature. It is the upcoming last sprint to develop the system, which can be developed using the Google Map API.
8.	Donation history management	The donation history management is the due work it is work on the coming sprint where make the plan the going the development.
9.	Complaint with admin	The users can complain to the admin through the mobile directly with complaint details. This task is also due to work, and it can be completed in the comment sprint where is it also covered.
10.	Testing	The final due work of the system testing the internal testing is every feature completed to test but the final testing whole system test to find any errors or bugs.

Table 26: Plan due worktable

5. Conclusion

To summarize what's been said earlier, this report demonstrates that the client was chosen, the project plan was created, and the development processes are detailed in the report. The development process shows the front-end mobile login, register, forgot password, walkthrough, and dashboard but it is not completed. Development, back-end Rest API, database, and login UI design are demonstrated in this report. Although the project appeared to be somewhat complex and messy at times, overcoming the milestones was an incentive to do more and more.

The final year project (FYP) is an ambitious aimed at creating a comprehensive Android platform solution facilitating streamlined interactions for various user groups involved in donation processes. The project's progression has been methodical, starting with foundational research, database design, and API development to ensure robust communication between front and back ends. The focus has been on enhancing user experiences across mobile and web platforms. Mobile development, utilizing Kotlin with Jetpack Compose, and web development employing HTML, CSS, and Bootstrap, have shaped the interface for both systems. Emphasis on mobile local database implementation and resolving user login challenges has been pivotal. As components align, intuitive user dashboards for diverse user categories have taken shape, intertwining seamlessly with crucial aspects like food donation. Integration of engaging features such as notifications and Google Maps has been planned to highlight donation locations, further enriching user engagement.

The project's inclusivity is evident in its coverage of functionalities catering to administrators, donors, farmers, volunteers, and NGO admins, ensuring a cohesive donation process for all involved parties. The goal is a user-friendly, robust platform accessible across Android, and the web application, aimed at aiding social workers in assisting the impoverished more effectively. Moving forward, plans include continuous expansion and improvement, extending the system's capabilities with additional messaging features to foster better communication and collaboration among users. The project's conclusion nears, with impending tasks focusing on final testing, ensuring a flawless user experience before deployment.

6. References

- Adobe. (2023, 12 25). *Waterfall Methodology: A Complete Guide*. Retrieved from Adobe Experience Cloud:
<https://business.adobe.com/blog/basics/waterfall#:~:text=The%20waterfall%20methodology%20is%20a,detailed%20documentation%2C%20and%20consecutive%20execution.>
- Developers, G. f. (2023, 12 24). *Developer*. Retrieved from Morden Android:
<https://developer.android.com/jetpack/compose/kotlin>
- Frank, A. (2024, 12 27). *Food donation*. Retrieved from No Food Waste Food:
<https://nofoodwaste.org/>
- Ganth, M. (2023, 12 24). *Django REST framework*. Retrieved from Django:
<https://www.django-rest-framework.org/topics/documenting-your-api/>
- Josh Domingues, N. B. (2023, 12 27). *Flash Food*. Retrieved from Save Money and fight against food waste: <https://www.flashfood.com/#How-it-works>
- Lewis, S. (2023, 12 25). *Prototyping model*. Retrieved from TeachTargate:
<https://www.techtarget.com/searchcio/definition/Prototyping-Model#:~:text=The%20prototyping%20model%20is%20a,or%20product%20can%20be%20developed.>
- Mullins, J. S. (2024, 12 26). *FOOD WASTE IN THE U.S. IS REAL*. Retrieved from Food Rescue Us: <https://foodrescue.us/food-waste/>
- S, S. (2023, 12 25). *What Is Scrum Methodology? & Scrum Project Management*. Retrieved from Nmble humanize work: <https://www.nimblework.com/agile/scrum-methodology/>

7. Appendix

7.1. Project as a solution

Furthermore, the project's emphasis on efficient distribution channels, including logistics planning and collaboration with local authorities, showcases a holistic approach. This approach not only ensures safe and timely transportation of surplus food but also focuses on building infrastructure for proper storage and preservation. Ultimately, the project's multifaceted strategy aims not only to curb food waste's environmental impact but also to combat food insecurity by empowering communities and fostering a more equitable food distribution system.

A list of solutions is given below:

- Utilize the waste food and waste food reduction,
- Utilize surplus food effectively,
- Raise social awareness about food waste,
- Facilitate food donation and distribution,
- Encourage responsible food handling and safety practices,
- Promote a sense of community and social responsibility,

[\(Go back to the same page\)](#)

7.2. Technology Use

7.2.1. Django Rest Framework

Django's admin panels empower site administrators to manage web application content without complex coding. When combined with Django REST Framework (DRF), Django facilitates the development of RESTful APIs, enabling seamless communication between different parts of a system and mobile app. In a food donation system, this allows users to input surplus food details via a mobile app or website and enables recipient organizations to request specific food types. Django simplifies backend development and pairs with DRF, creating robust APIs for effective communication across system components.

[\(Go back to the same page\)](#)

7.2.1. Kotlin jetpack compose.

Users interact indirectly with Kotlin and Jetpack Compose through the app's front end within the system. They encounter intuitive and responsive interfaces sculpted by these tools, effortlessly exploring various features and functionalities. Leveraging Kotlin and Jetpack Compose in frontend development ensures users encounter visually captivating and user-friendly interfaces, significantly enhancing their overall experience with your mobile app.

[\(Go back to the same page\)](#)

7.3 Tools use

7.3.1. Draw.io

Draw.io is an online tool for creating diagrams like flowcharts, org charts, and UML diagrams. It's user-friendly, offering shapes and templates for easy customization. Users drag and drop elements onto the canvas to visualize processes or systems. In UML diagram design, Draw.io has specific tools for depicting complex software systems, making it popular among software engineers for illustrating software architecture and behaviours. Its intuitive interface and extensive UML elements simplify diagram creation for software development projects. These tools are used for class diagrams, Used cases, DFD, ERD, Sequence Diagrams etc.



Figure 57: Draw.io tools

1.3.2. Luci Chart

The Luchi chart is a visual tool used in project management for context diagram design. It's specifically geared towards identifying and mapping out the key stakeholders, entities, or systems involved in a project. This chart helps in understanding the relationships and interactions among these elements within the project's context. It is used for UML context diagram design (DFD 0 level).

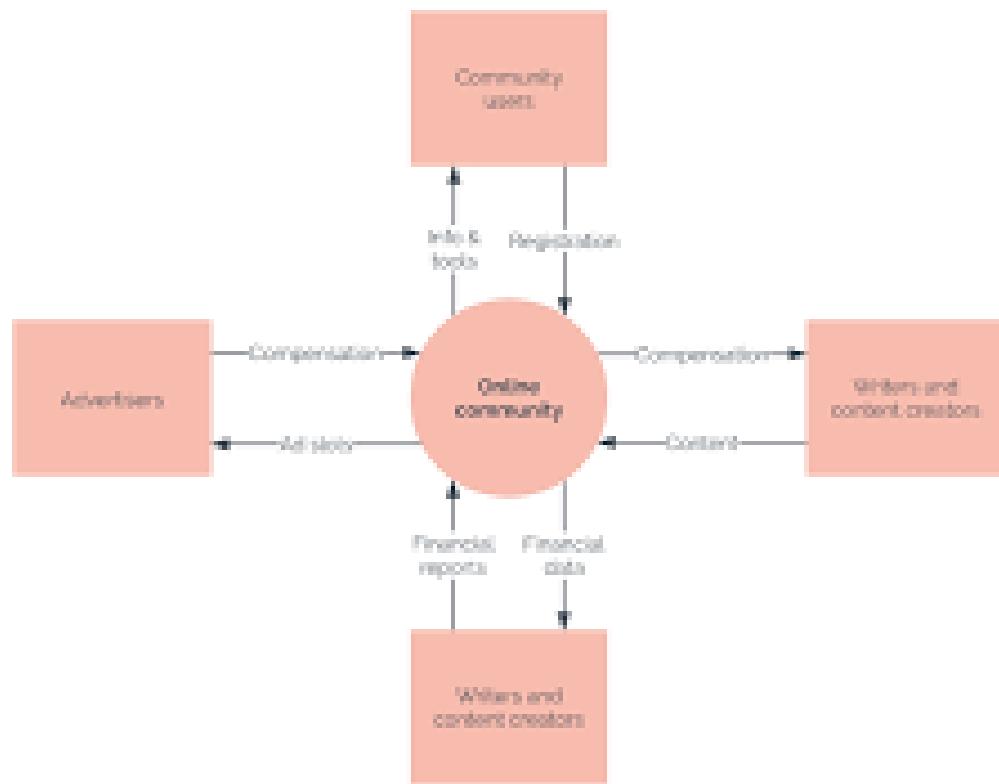


Figure 58: Luci chart tools

1.3.3. Figma

In Figma, people may collaborate to develop dynamic and interactive prototypes for user interface design. Figma is a browser-based tool. Figma has gained popularity as a tool for UI development in the project system architecture developed by the Figma tool.



Figure 59: Figma tools

[\(Go back to the same page\)](#)

7.4 Expanded Use Case

The expanded use case is given below:

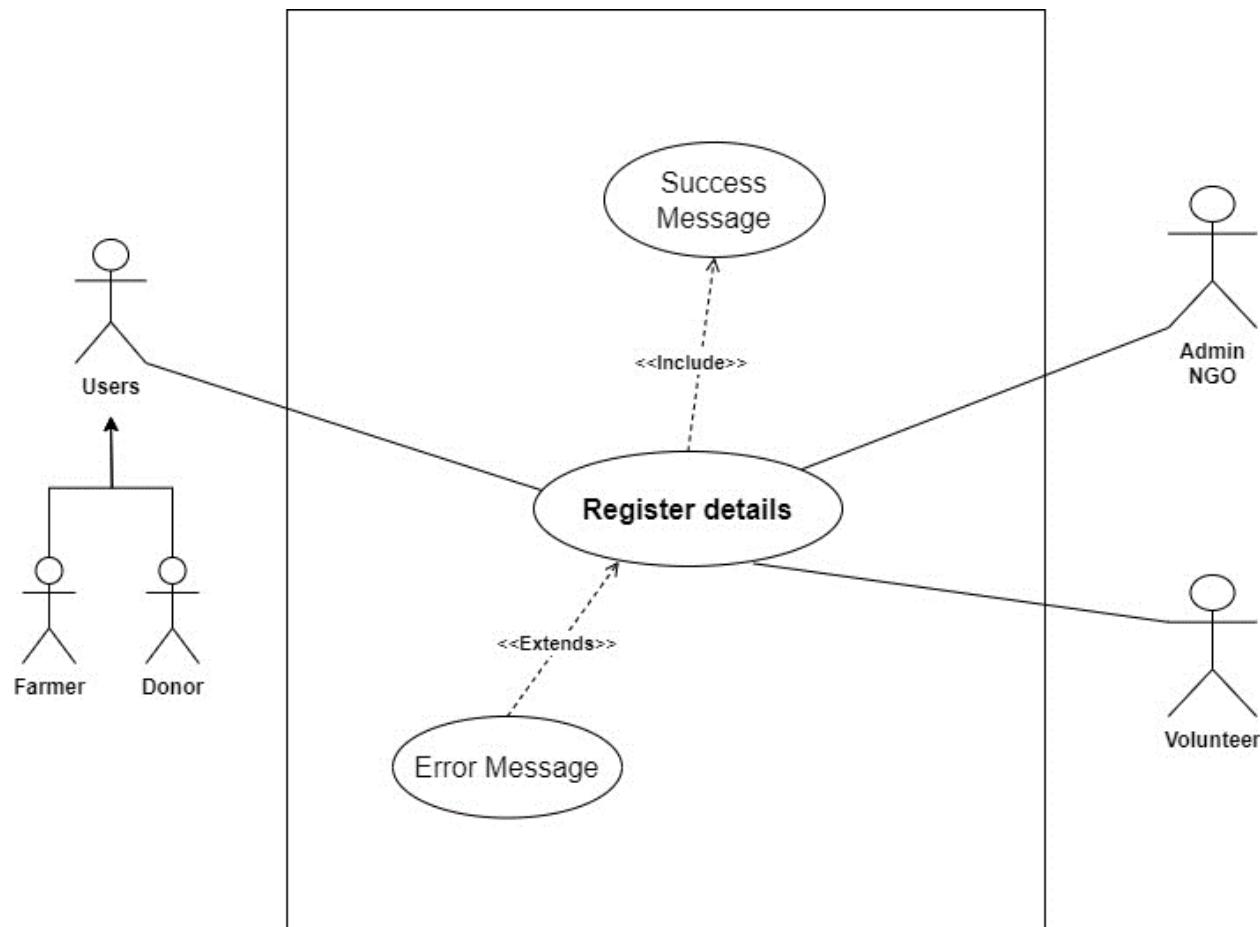


Figure 60: Register expanded use case

Use Case:	Register Details
Actors:	Donor, Volunteer, NGO, Farmer
Descriptions:	All users can input their respective details into the system. The system automatically registers the provided information upon submission, ensuring seamless integration and efficient data management.
Typical Courses of Events:	
Donor, Volunteer, NGO, Farmer, Admin	System Response
1. A user can provide the personal details for the register in the system.	
	2. The system checks whether the provided details are valid or not.
3. Request for register in the system.	
	4. Conform register with a success message if no valid data, then show the error message.

Table 27: Register user expanded use case table

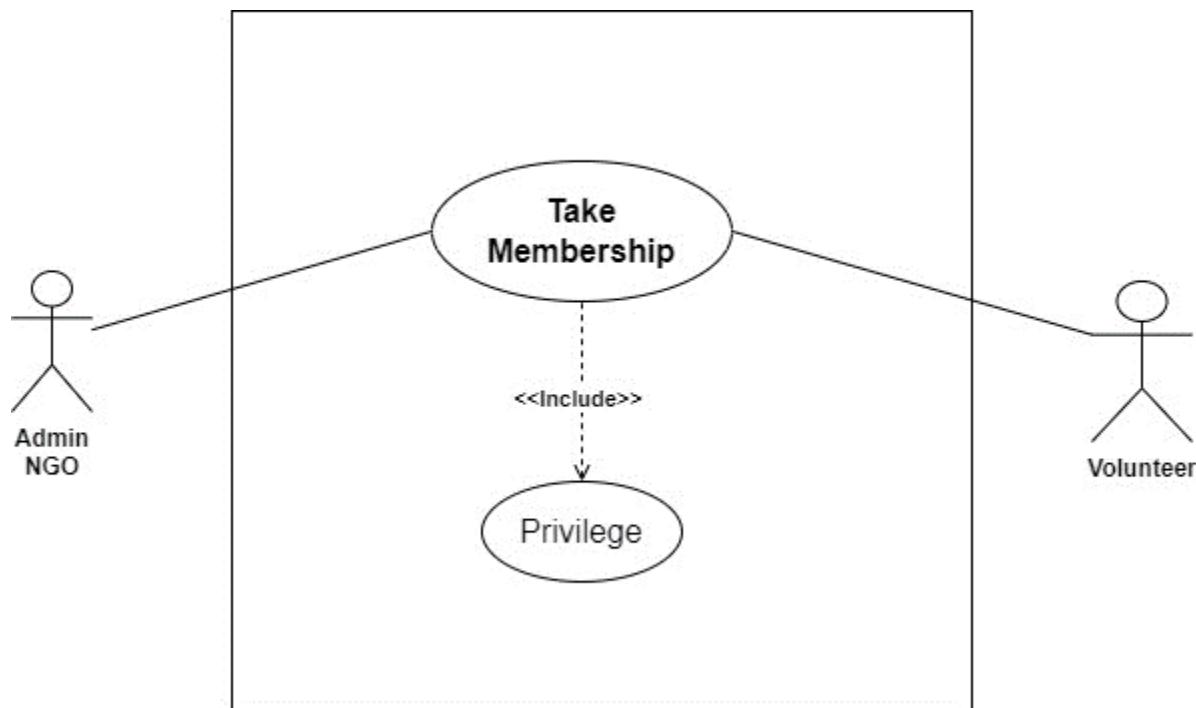


Table 28: Table membership Expanded use case

Use Case:	Take Membership
Actors:	Volunteer, NGO
Descriptions:	A new volunteer provides the personal details, and his/her details are registered with the system. The NGO provide the membership, and then volunteers take the new membership.
Typical Courses of Events:	
Volunteer, NGO	System Response
1. A new volunteer can provide the personal details for the registered membership in the NGO.	
	2. Check the Volunteer details.
3. Request to take membership in the NGO.	
	4. Confirm the registered membership and give the privilege.

Table 29: Take membership expanded use case table

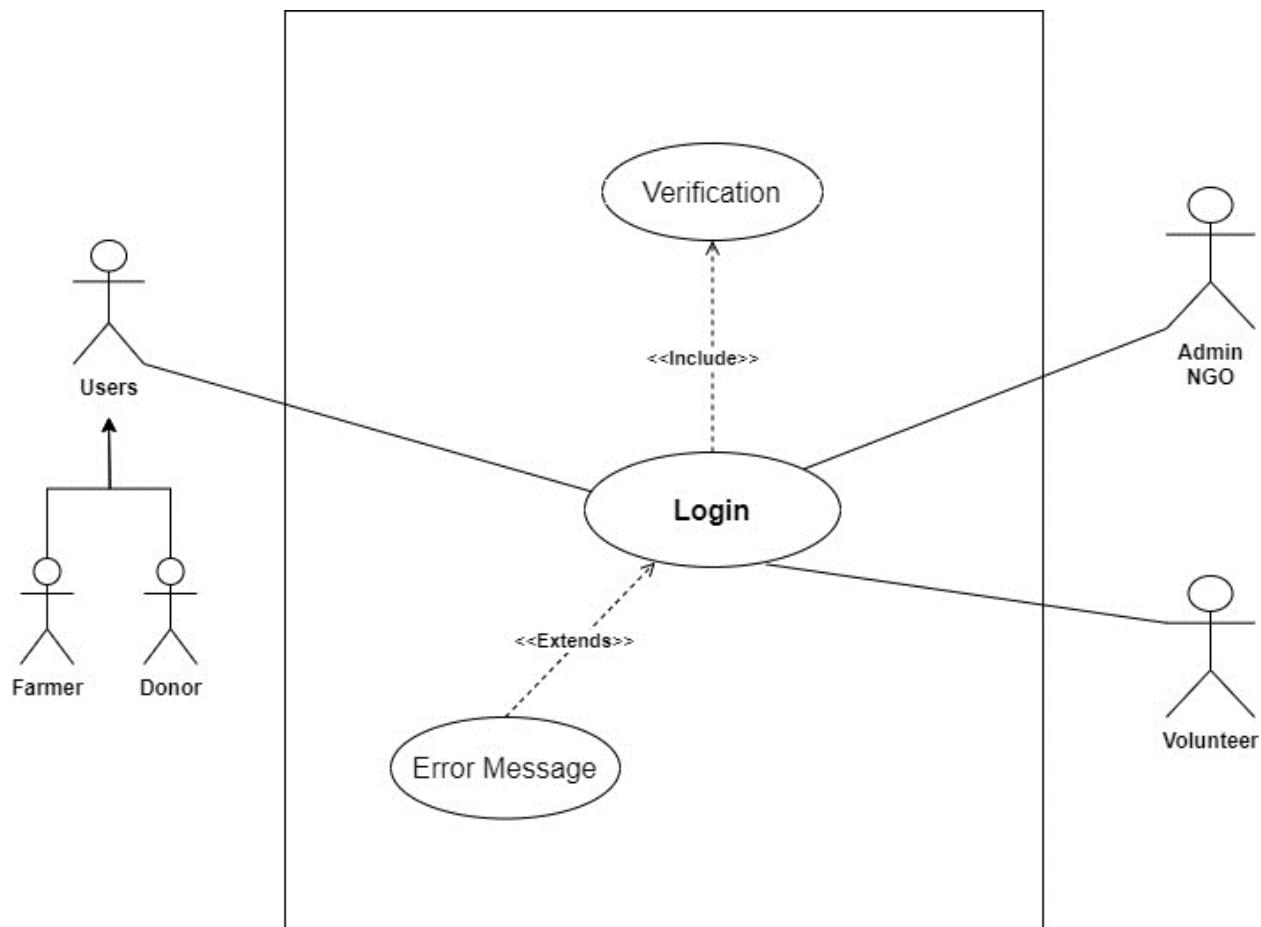


Table 30: Login expended use case diagram

Use Case:	Login
Actors:	Donor, Volunteer, Farmer
Descriptions:	After registering details in the system, all the users can provide valid details and log in to the system Then successfully log in to the system.
Typical Courses of Events:	
Donor, Volunteer, Farmer	System Response
1. A new user can provide the login details in the system.	
	2. The system checks whether the provided details are valid or not.
3. Request for login in the system.	
	4. The system can navigate to the dashboard if valid data is provided to log otherwise shows the error message.

Table 31: Login user expanded use case table

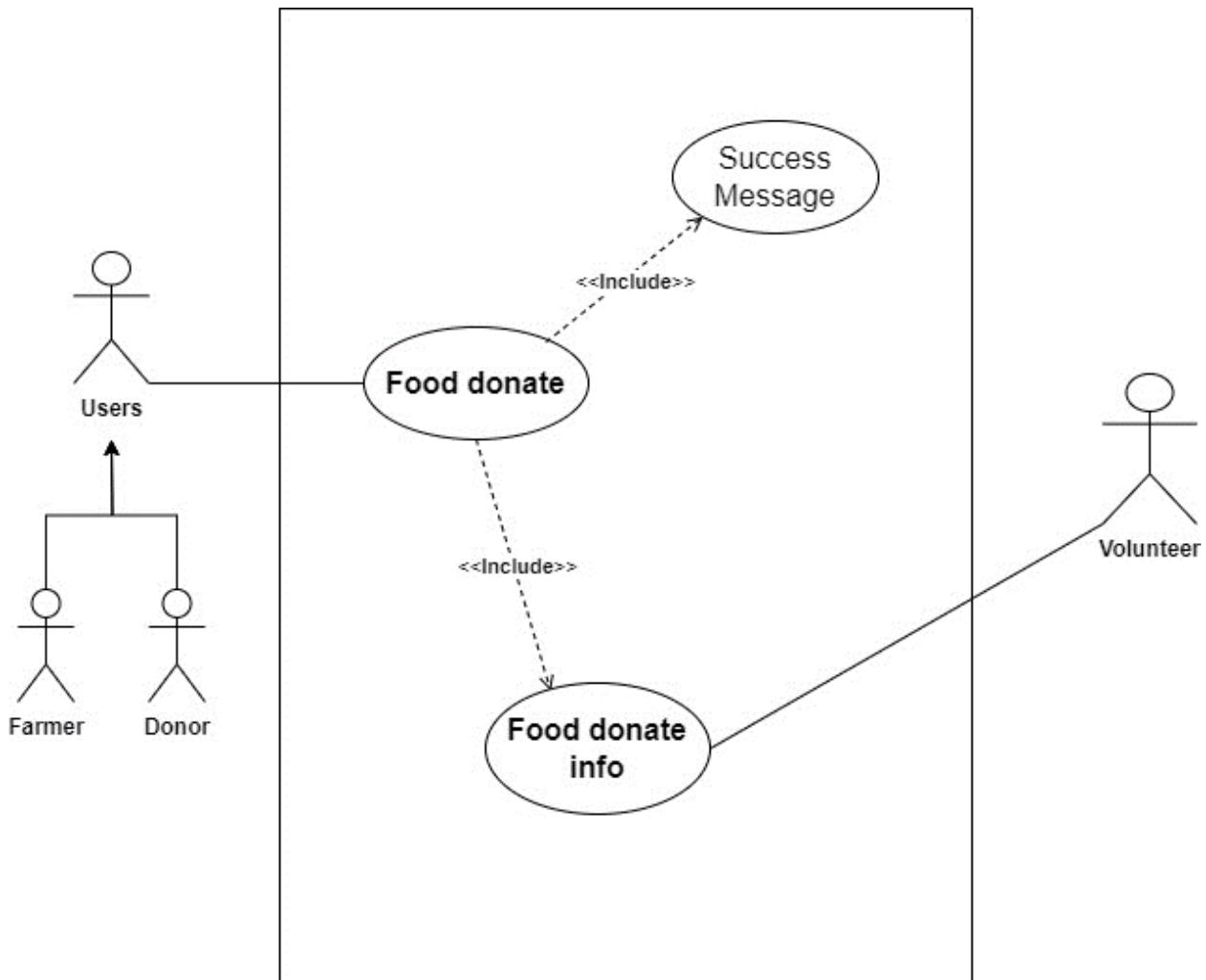


Table 32: Food donation expended use case diagram

Use Case:	Food Donation
Actors:	Donor, Farmer, volunteers
Descriptions:	The Donor or Farmer can donate the proper food information and details with location. The system can show the donation food details in the history after posting the donated food.
Typical Courses of Events:	
Donor, Farmer	Volunteers
1. A donor can post the food details for donation.	
	2. After getting the donation information volunteers respond to the request to accept the food.
3. Donors confirm to provide the food when contacted with the donor.	
	4. Volunteers receive the food and go to distribute it.

Table 33: Food donation user expanded use case table

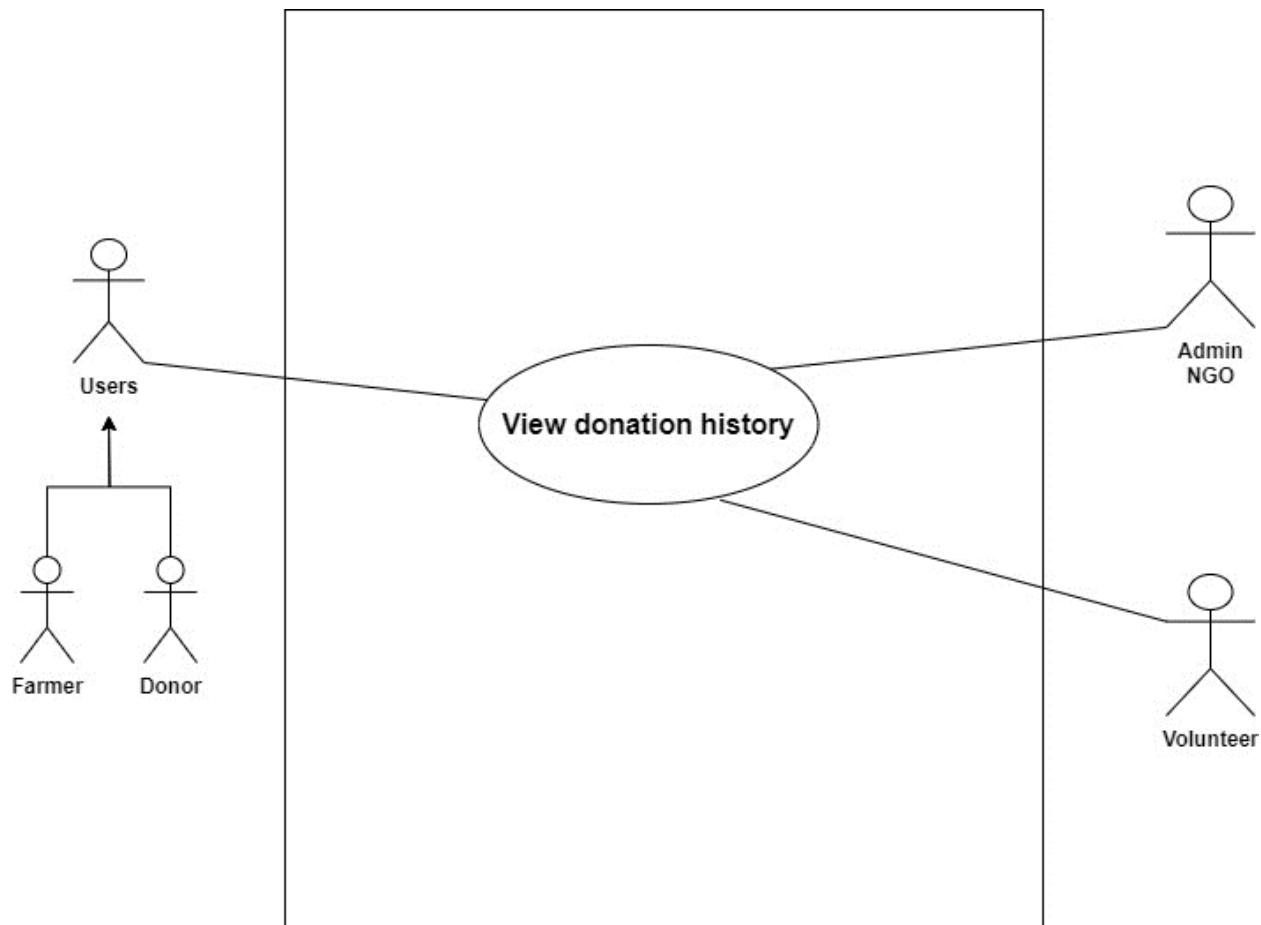


Table 34: View Donation history expended use case diagram

Use Case:	View Donation History
Actors:	Volunteer, Farmer
Descriptions:	All users can input their respective details into the system. The system automatically registers the provided information upon submission, ensuring seamless integration and efficient data management.
Typical Courses of Events:	
Volunteer, Farmer	System Response
1. A volunteer can view the donation post food details.	
	2. show the donation food details with all information like food, donor, and location details.
3. View the donated location.	
	4. The system can show the food donated location with Google Maps.

Table 35: View history user expanded use case table.

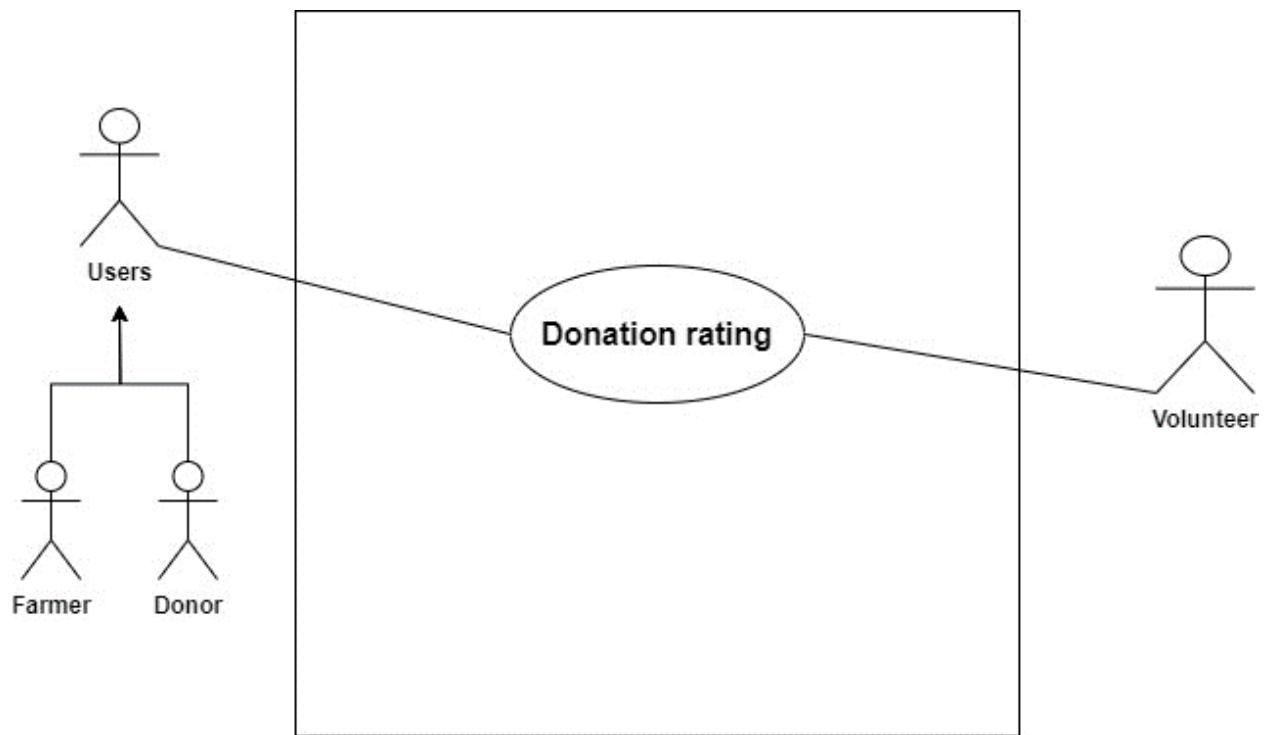


Table 36: Donation rating expanded use case diagram.

Use Case:	Donation Rating
Actors:	Donor, Volunteer, Farmer
Descriptions:	After the food is completely donated to some people the volunteer can give the donation rating to the donor with food distributed information.
Typical Courses of Events:	
Volunteer	Donor
1. After the food received to distributed give the donation rating with all the information.	
	2. Get the donated rating info.
3. Not possible to accept the donation of food if the volunteer cannot give a response.	
	4. Volunteers do not respond to receive the food and when the expiration date is over the donor gets the expiration notification.

Table 37: Donating rating expanded use case table

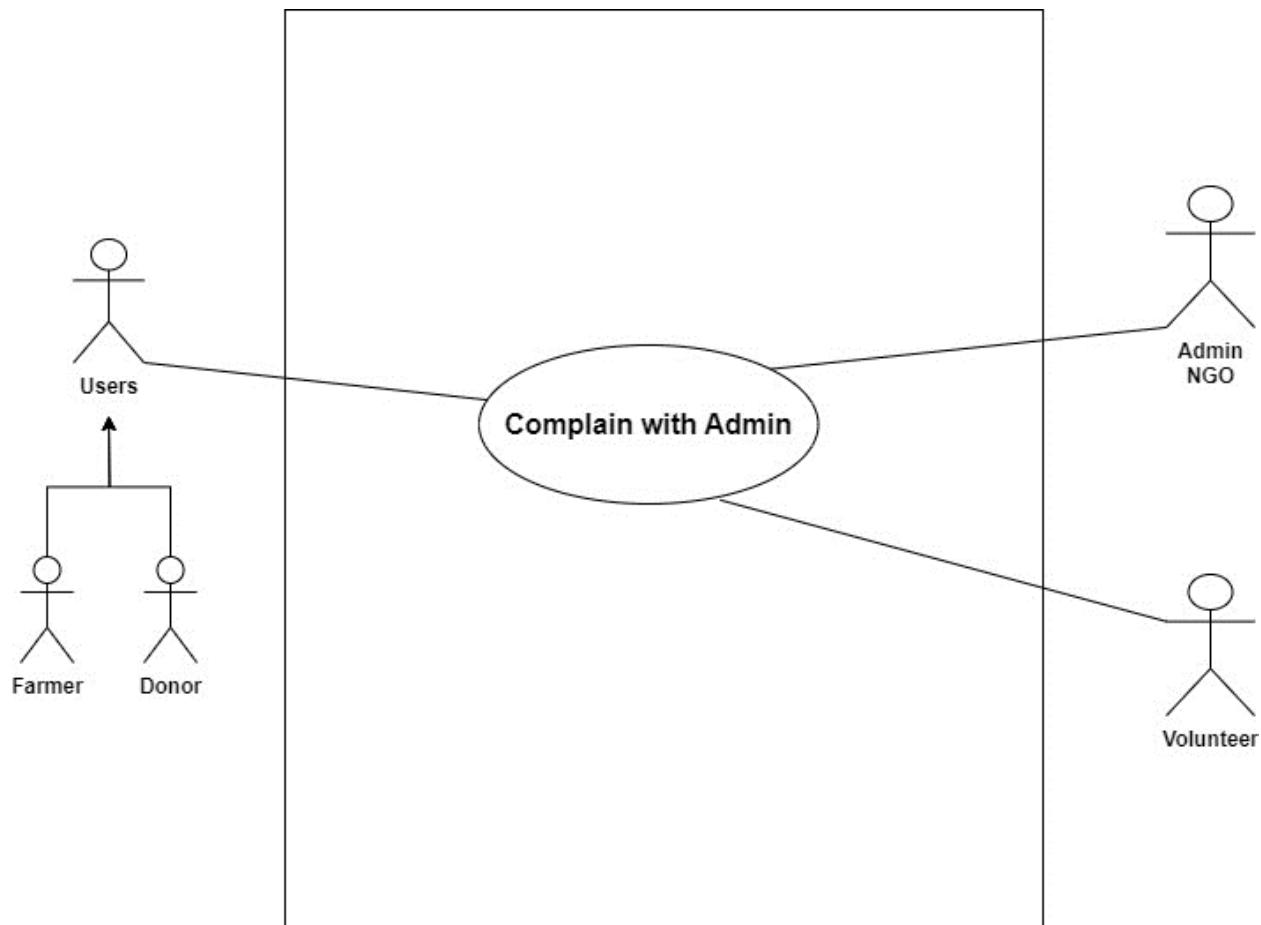


Table 38: Complaint with admin expanded use case diagram

Use Case:	Complain with Admin
Actors:	Donor, Volunteer, Farmer
Descriptions:	All users can input their respective details into the system. The system automatically registers the provided information upon submission, ensuring seamless integration and efficient data management.
Typical Courses of Events:	
Donor, Volunteer, Farmer	Admin Response
1. The food donation time donor volunteer and farmer can complain to the admin.	
	2. Admin verifies the complaint and gives the warning.
3. Received warning.	

Table 39: Complaint with admin expanded use case table

[\(Go back to the same page\)](#)

7.5 DFD Level 2

7.5.1. Register Details

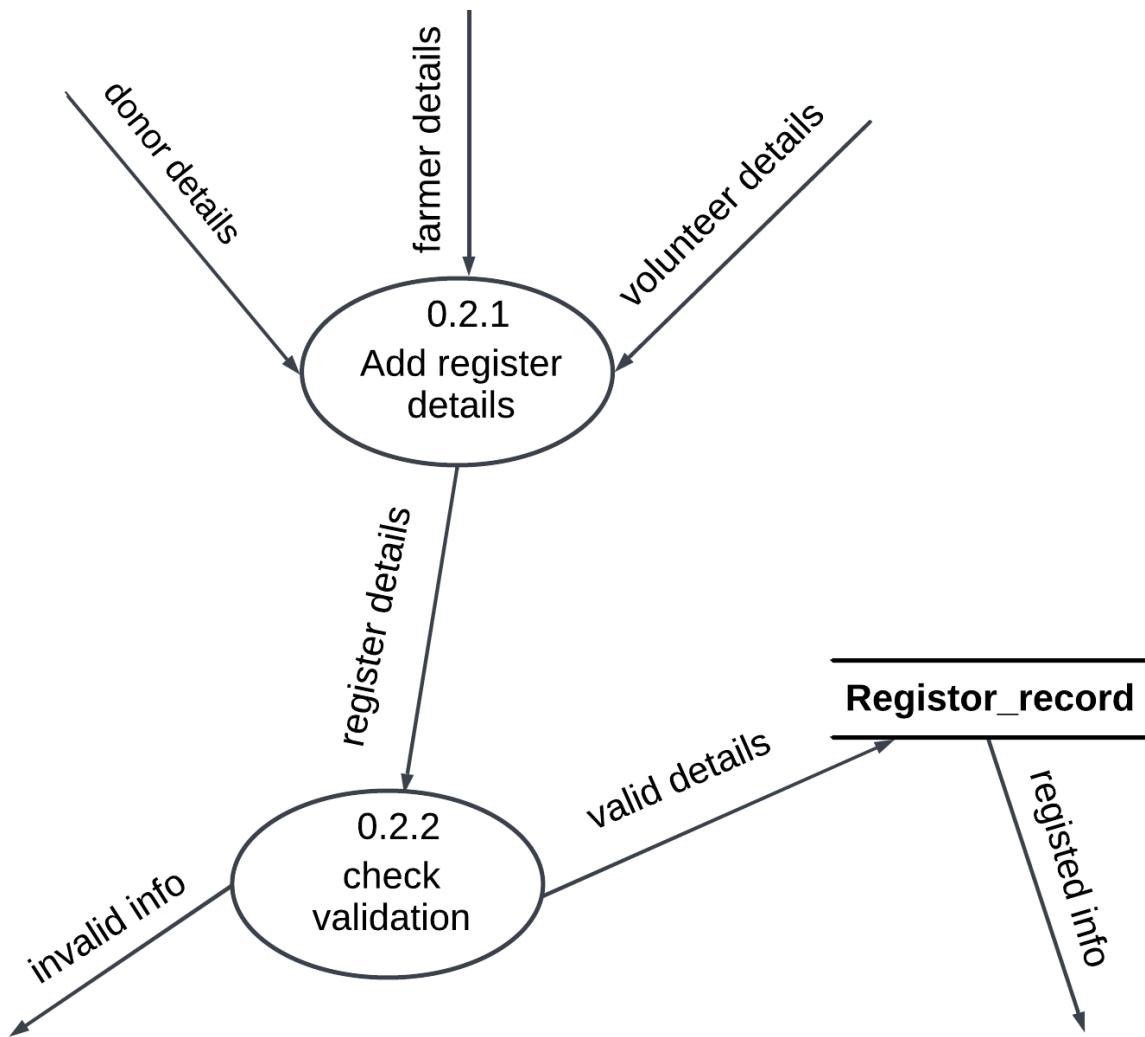


Figure 61: Register details DFD Level-2

7.5.2. Login System

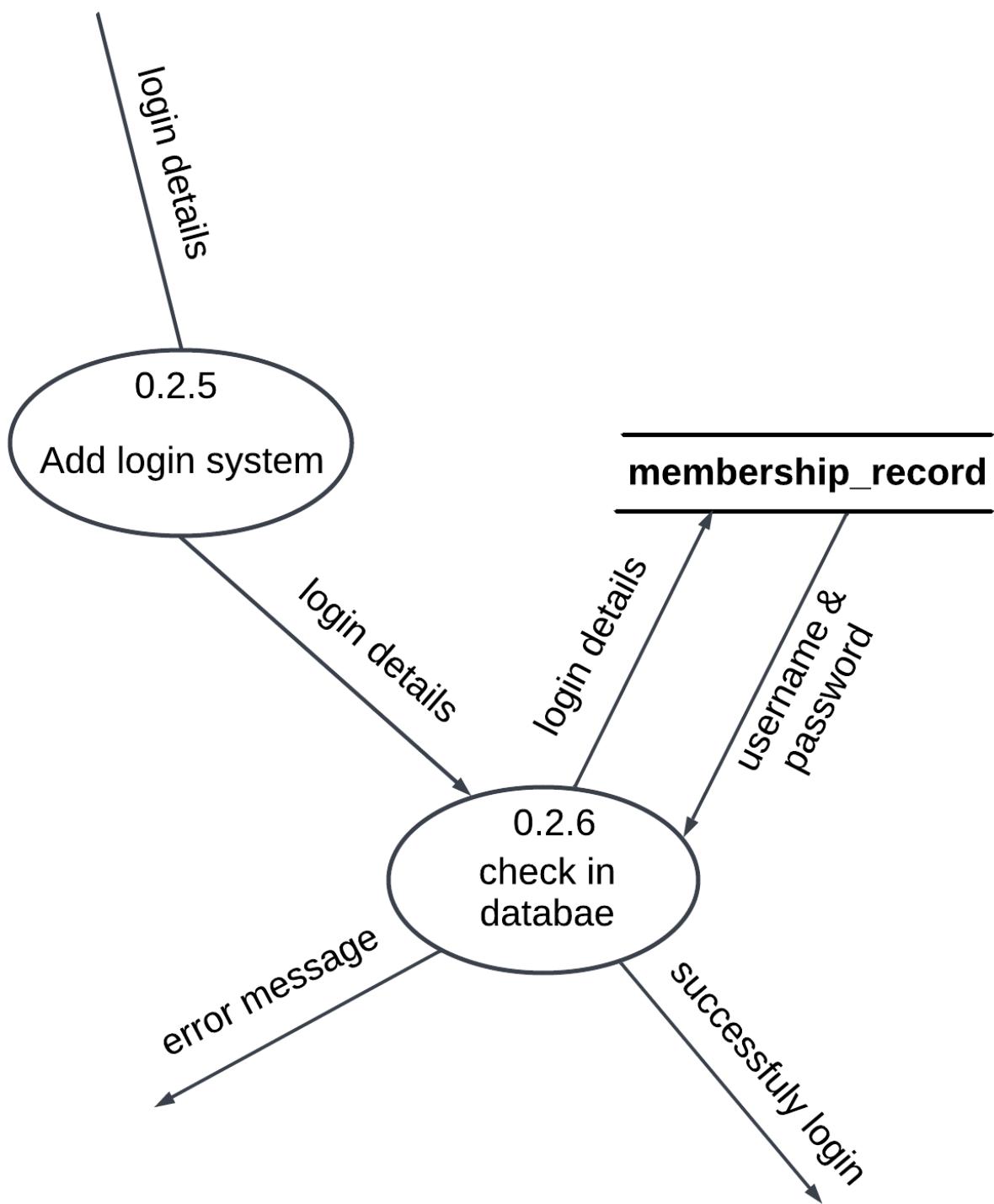


Figure 62: Login details DFD Level-2

[\(Go Back to the same page\)](#)

7.6 Activity Diagram

7.6.1. Login Activity

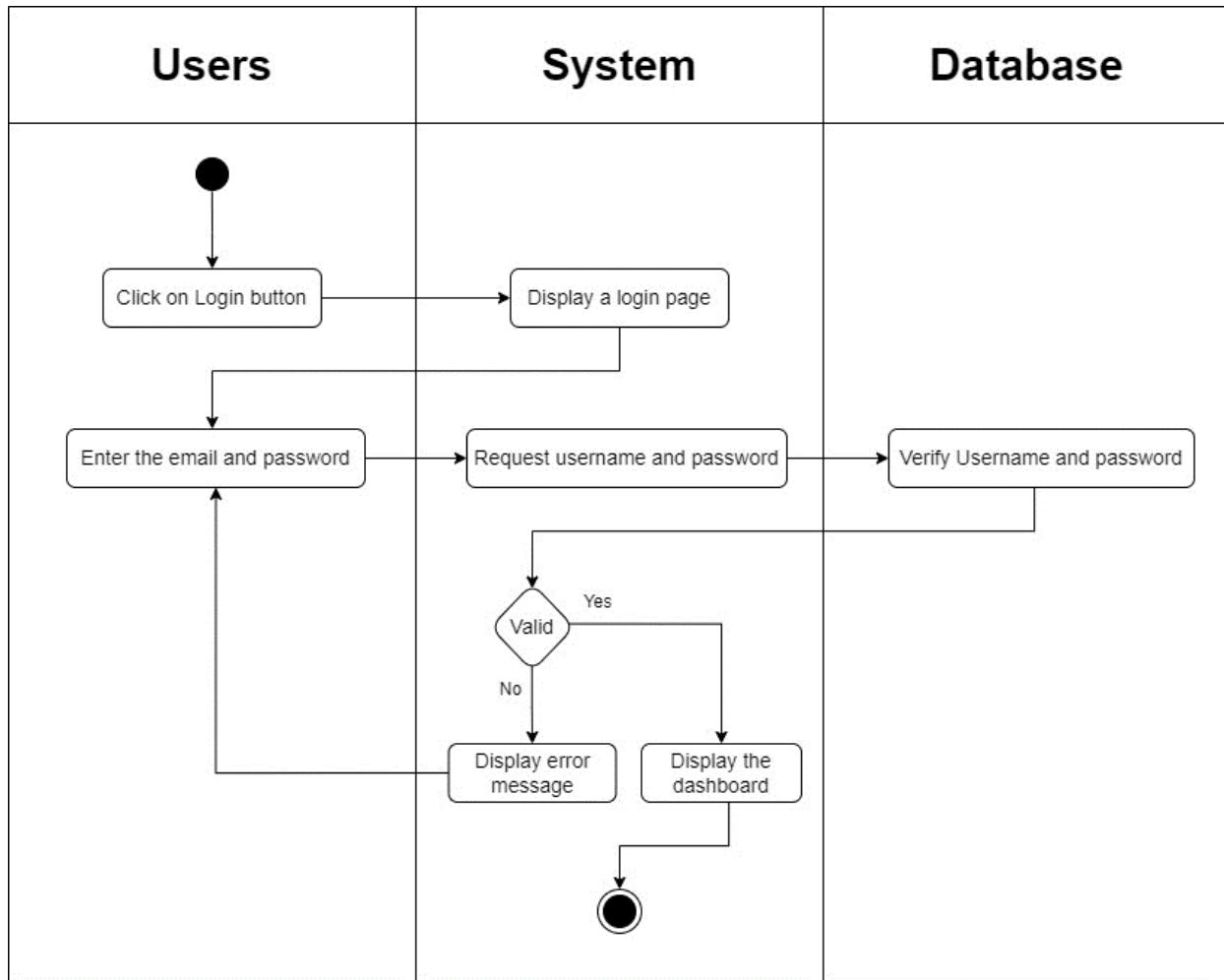


Figure 63: Login system activity diagram

7.6.2. Register Activity

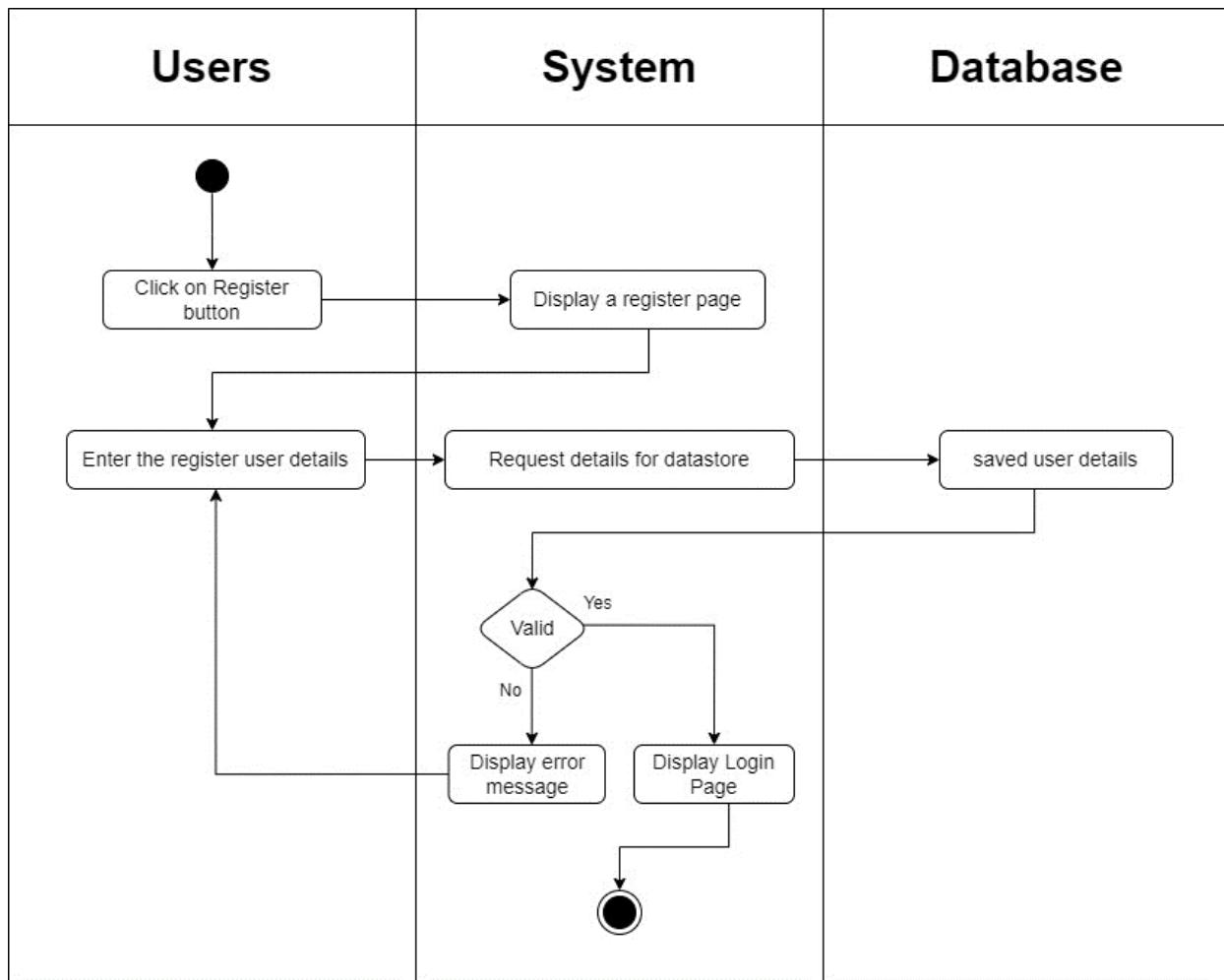


Figure 64: Register system activity diagram

7.6.3. Logout Activity

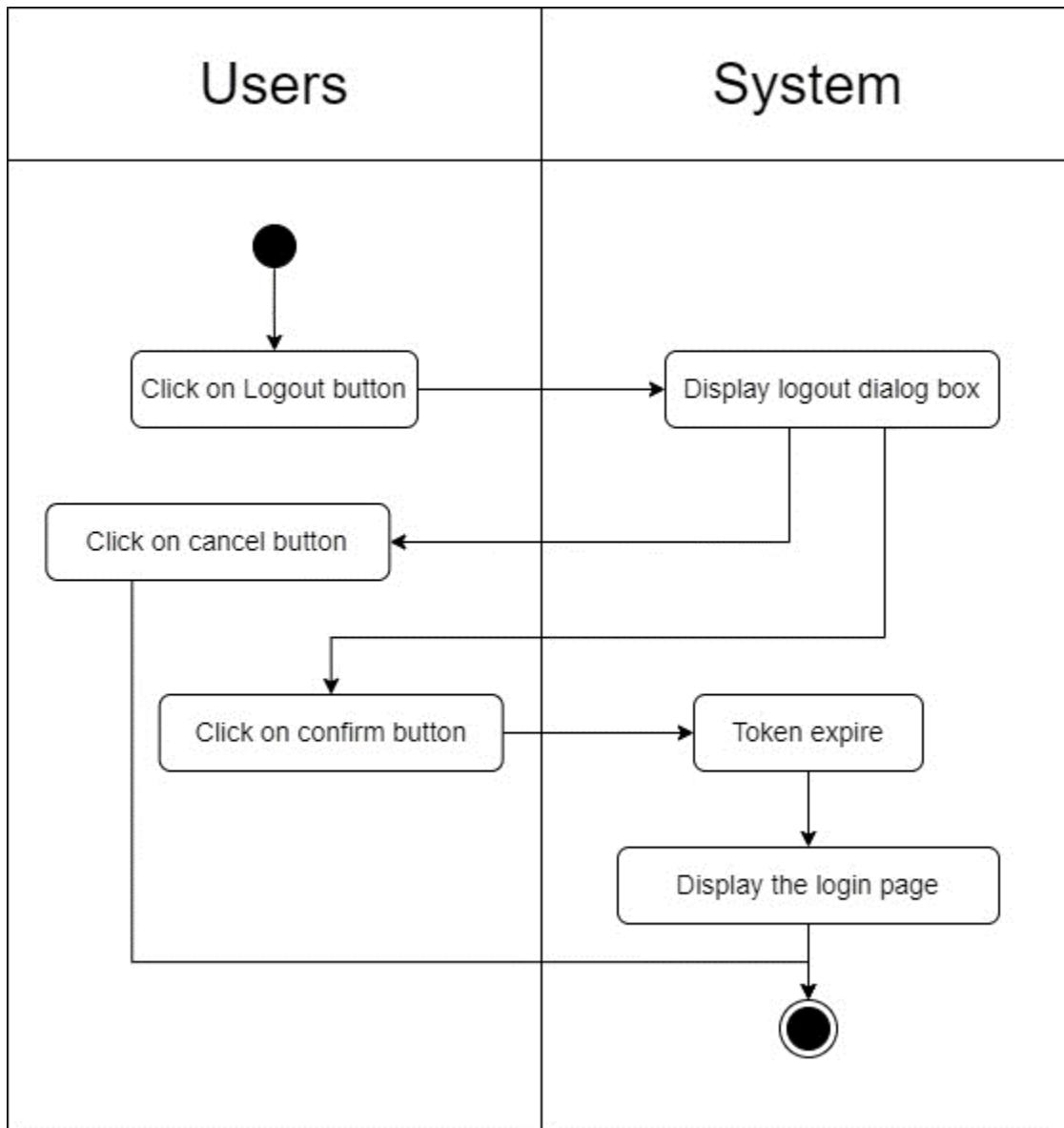


Figure 65: Logout system activity diagram

[\(Go back to the same page\)](#)

7.7 Wireframe UI

7.7.1 Splash screen

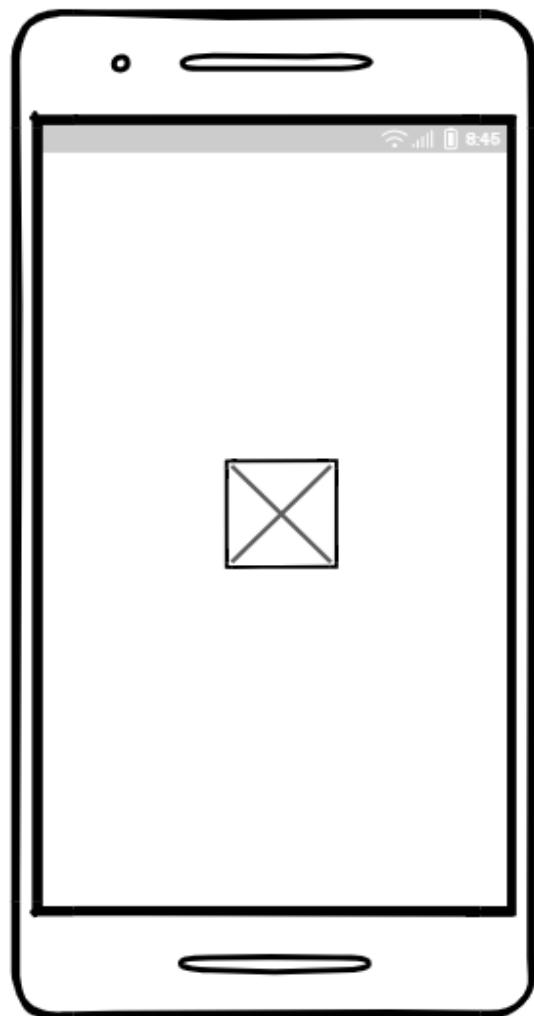


Figure 66: Splash screen mobile UI

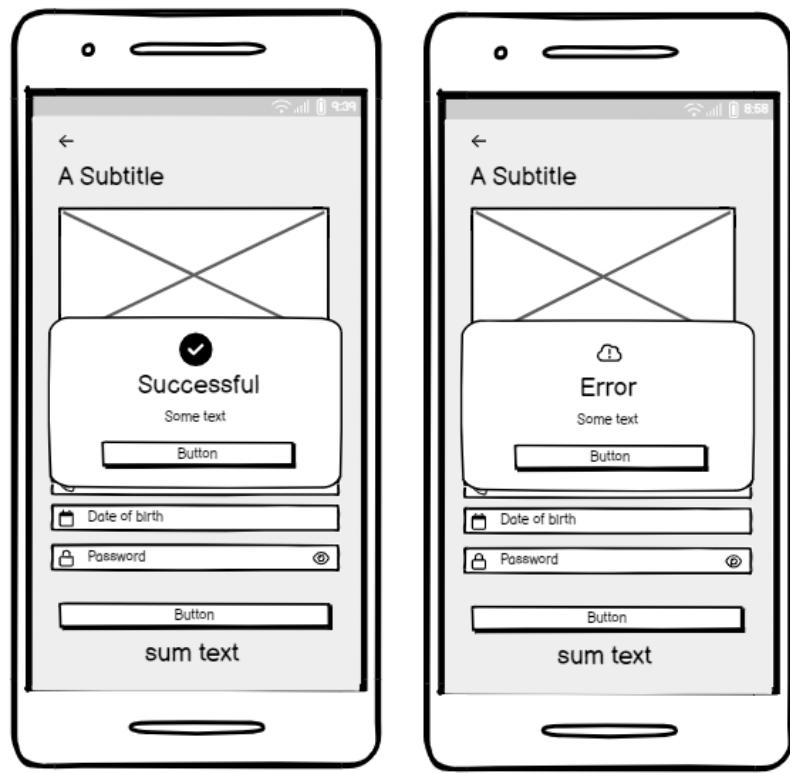


Figure 67: Register response message mobile UI



Figure 68: Donor Profile screen mobile UI



Figure 69: Donor Setting screen mobile UI



Figure 70: Volunteer Setting screen UI

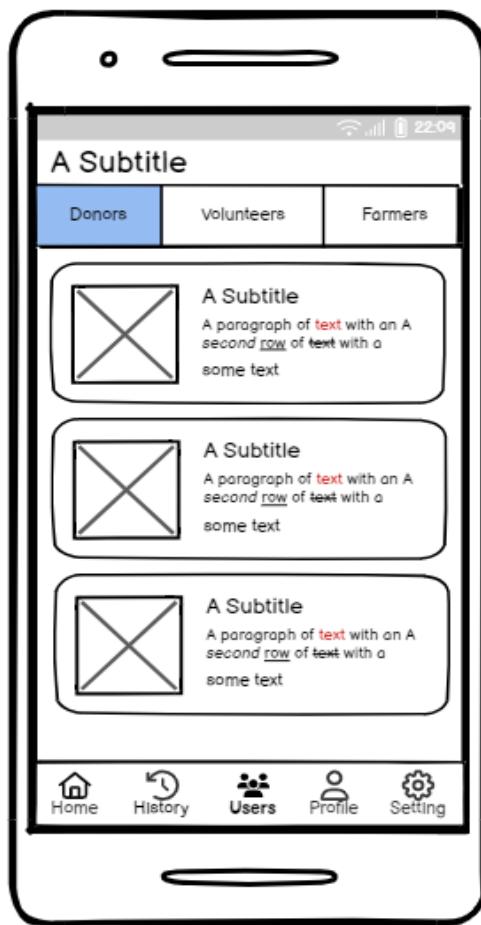


Figure 71: NGO Users screen UI

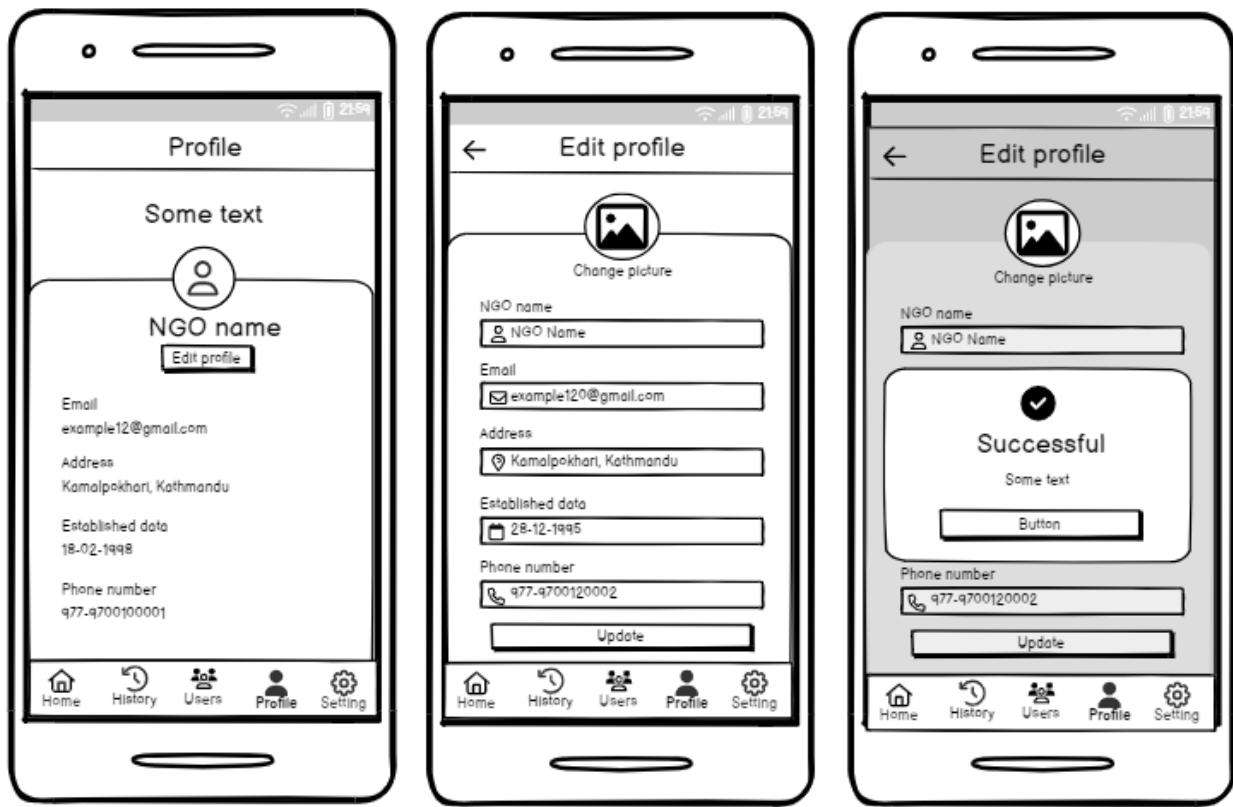
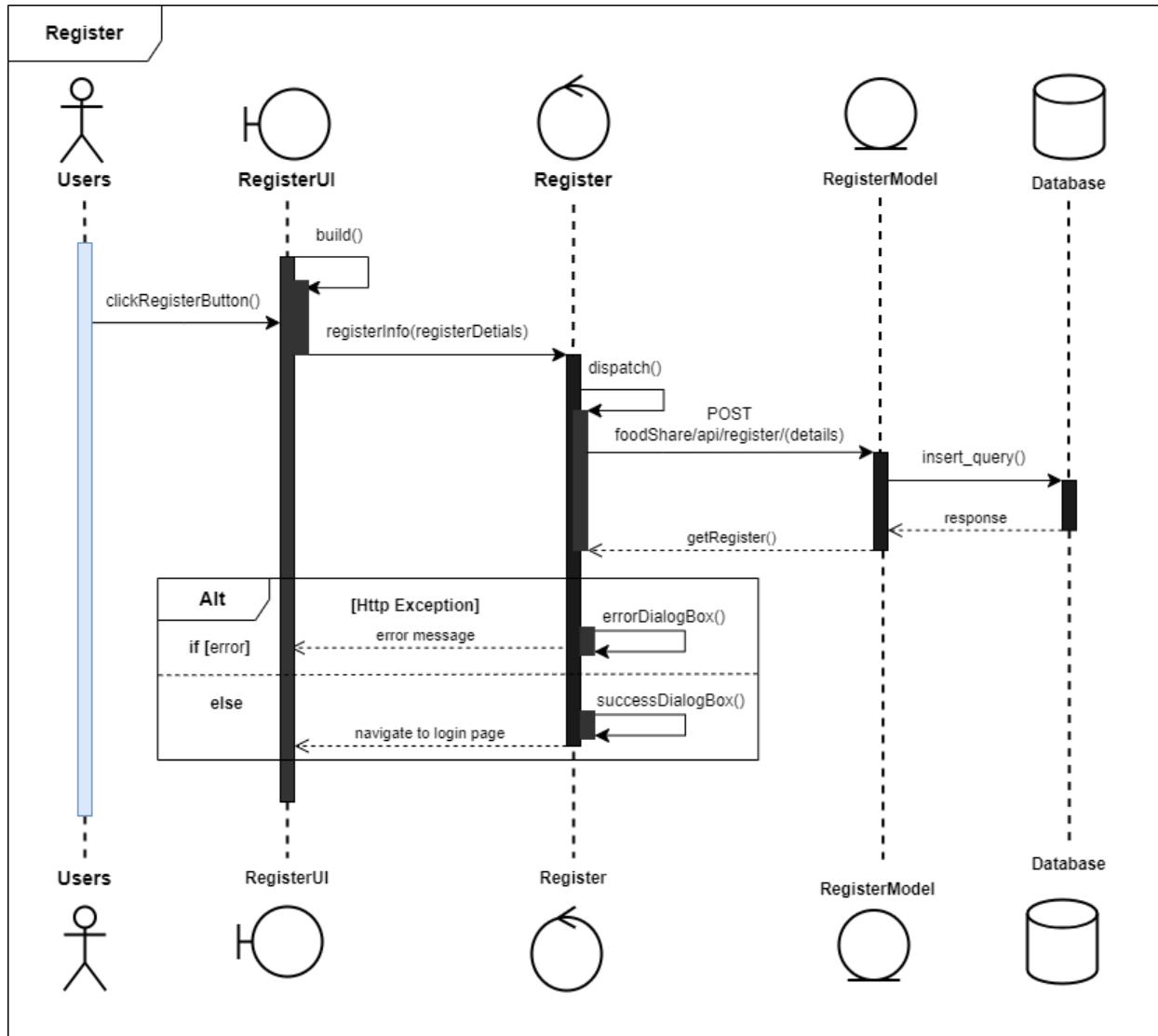
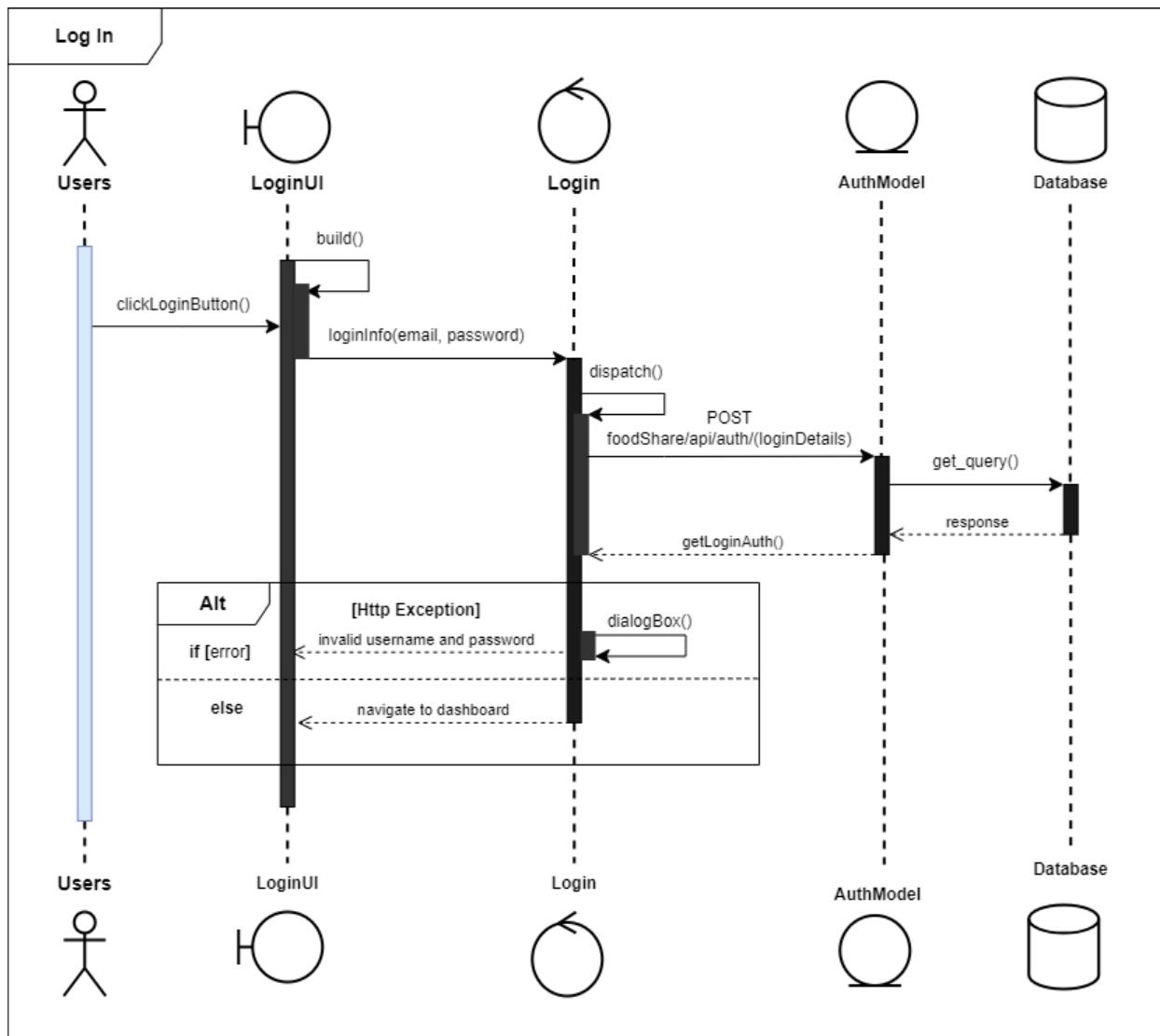


Figure 72: NGO Profile screen UI

[\(Go back to the same page\)](#)

7.8 Sequence Diagram





[\(Go back to the same page\)](#)

7.9 Survey form

This is the survey form for project requirements collection and finding the current problem phase of society. The system has included some features by survey form. The form's main purpose is to waste food and to manage surplus food to donate to poor or homeless people. The system has what the requirement is more needed it is found by the survey.

The server form response details are given below:

subigyan.adhikari@innovatetech.co

chirayurana@yahoo.com

karkisangam131@gmail.com

sabinadhanuki2@gmail.com

prabhat.adhikari@innovatetech.co

anmol@innovatetech.co

prabhash.thakuri@innovatetech.co

np01nt4s220035@islingtoncollege.edu.np

sujan.poudel@innovatetech.co

Figure 73: Survey other response email

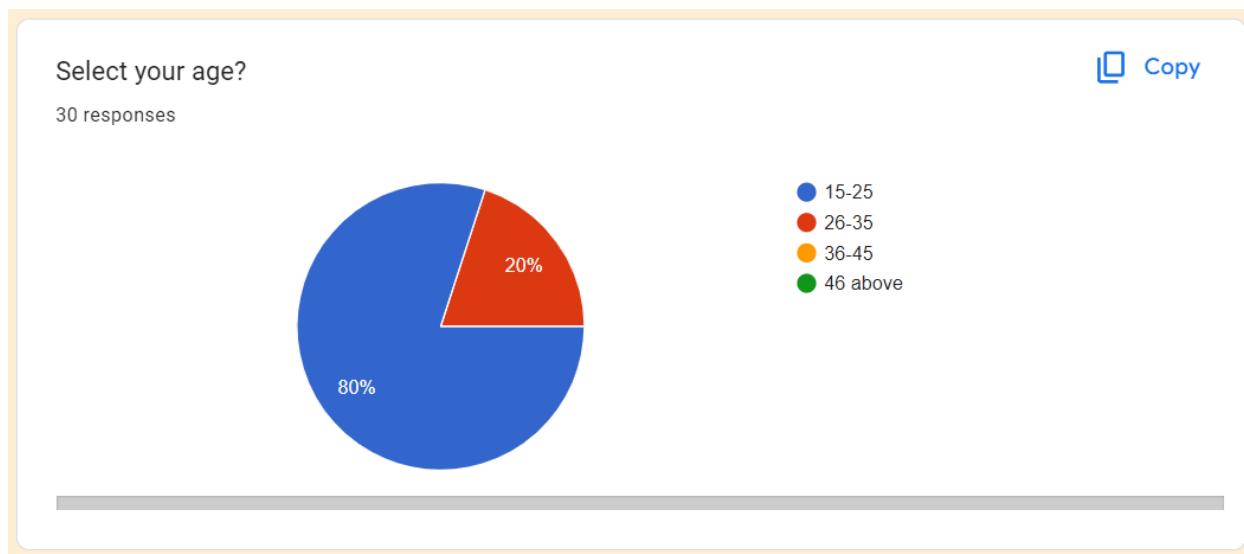


Figure 74: User age status in Pai chart

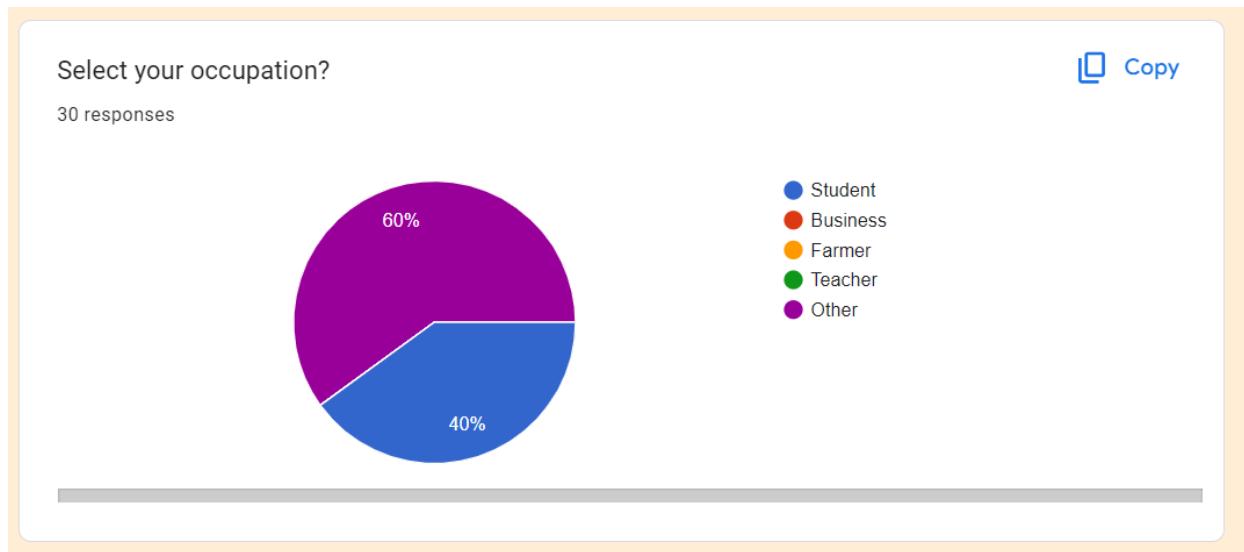


Figure 75: Select your occupation. response in chart



Figure 76: find the excess food that could be donated. response in a pie chart

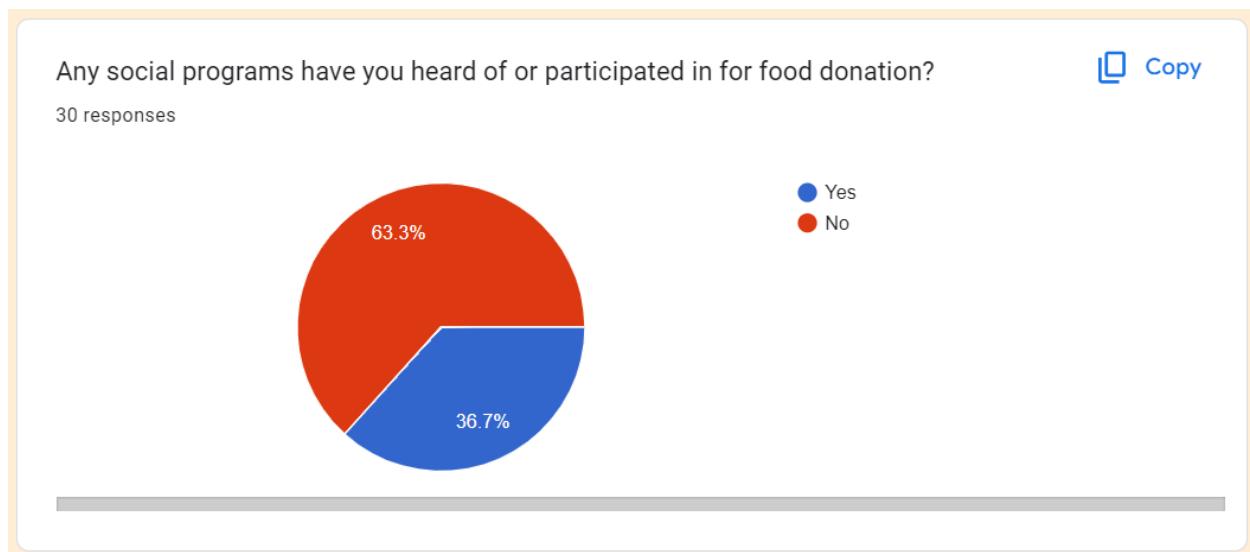


Figure 77: participated in food donation? response in the pai chart.

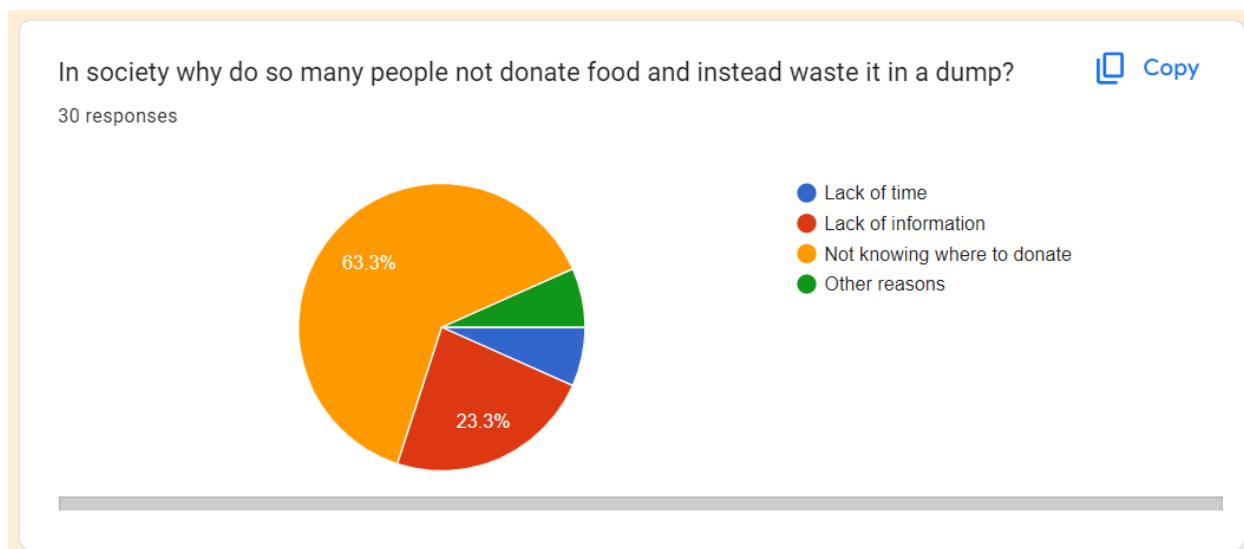


Figure 78: Not donation reason and why waste it in a dump? response in a pie chart

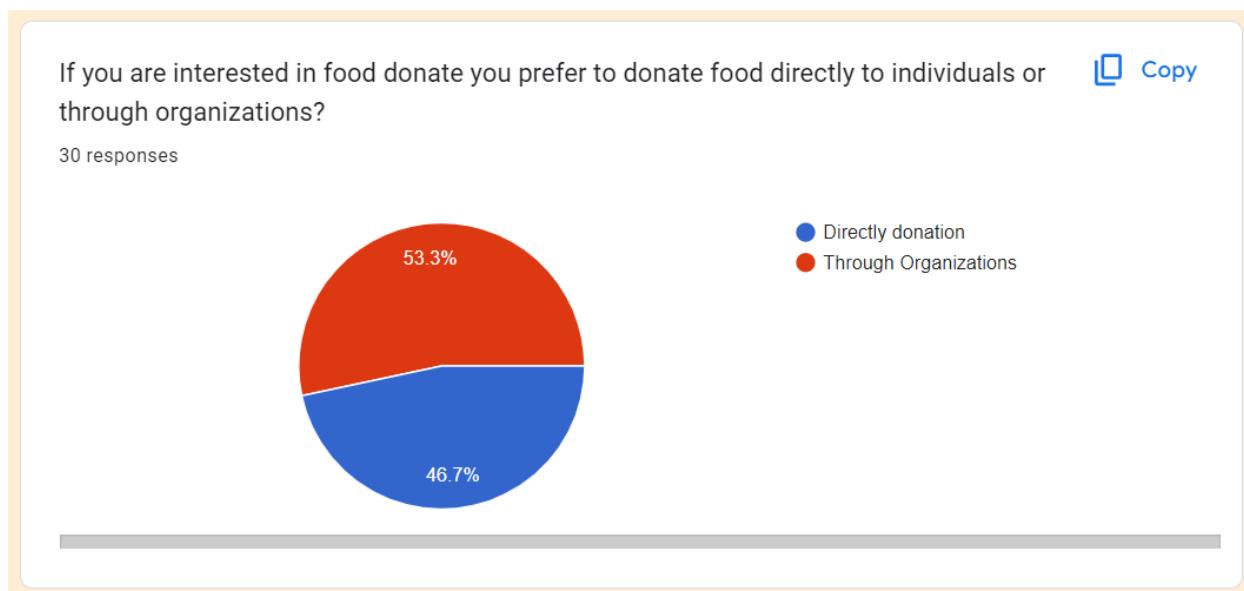


Figure 79: Donation options individual or through the organization? Response pie chart



Figure 80: Donation application rating? response in a pie chart

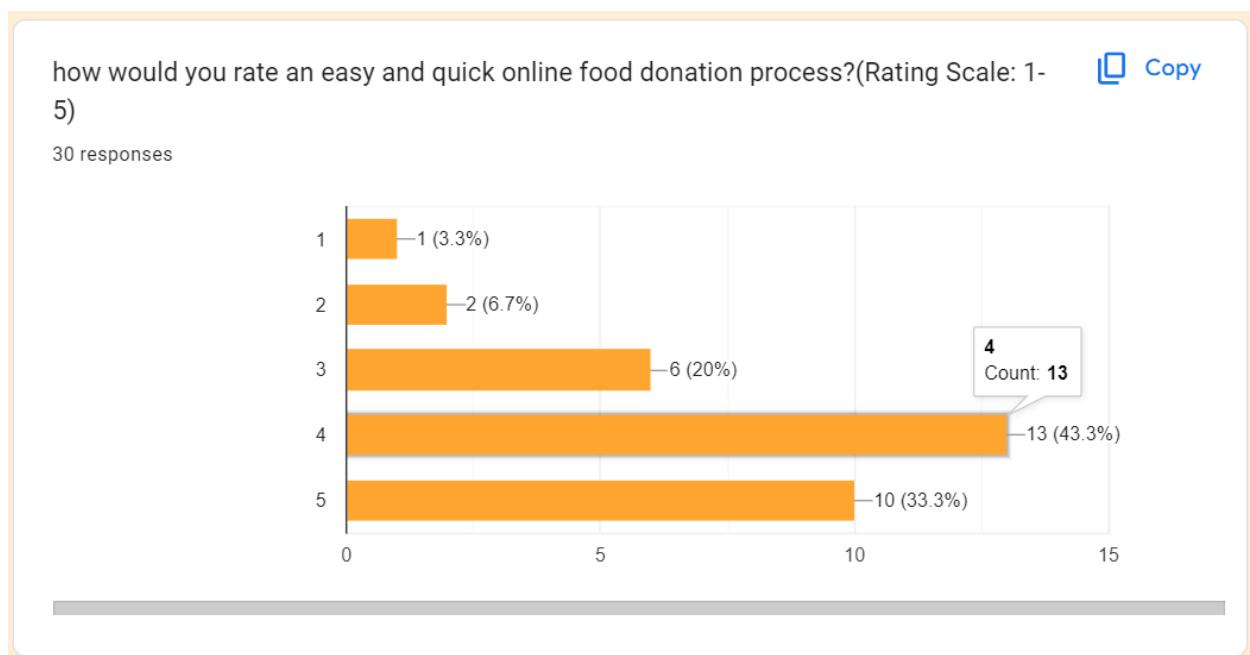


Figure 81: Ration for online food donation app? response in a pie chart

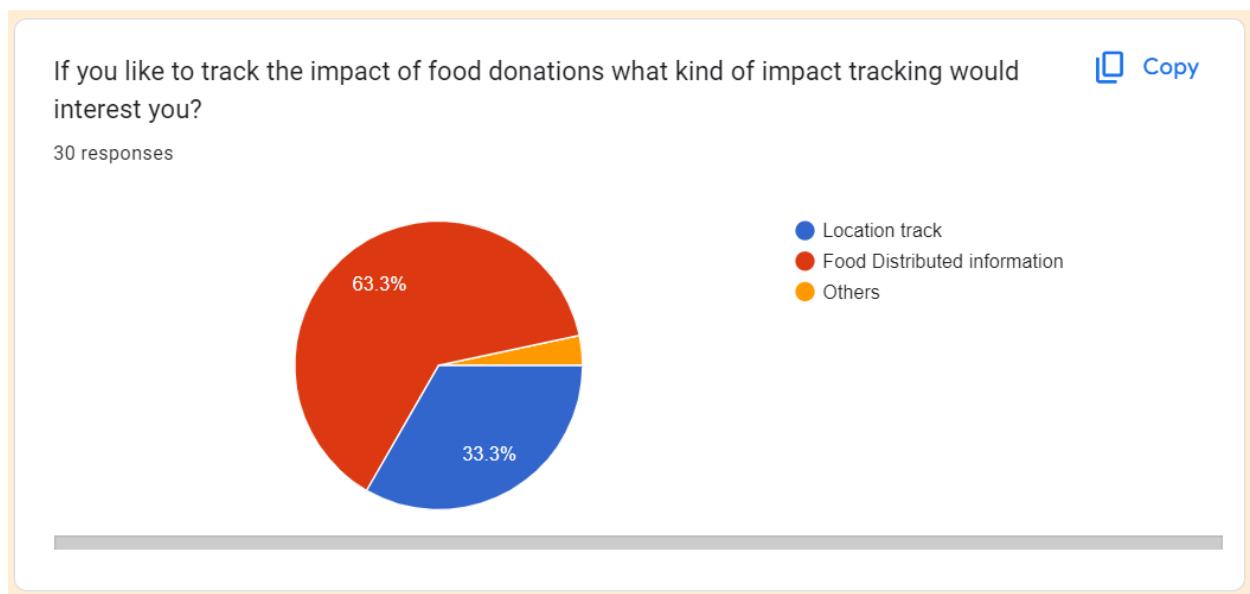


Figure 82: How to track the donation food where distributed? response in pain chart

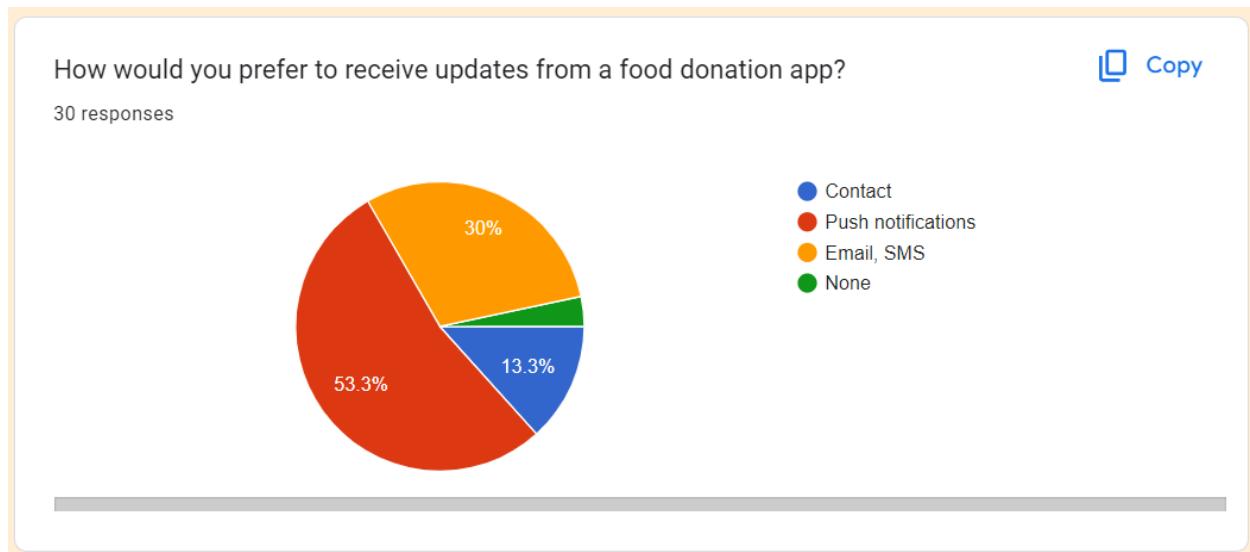


Figure 83: Donation received update form app? response in a pie chart.

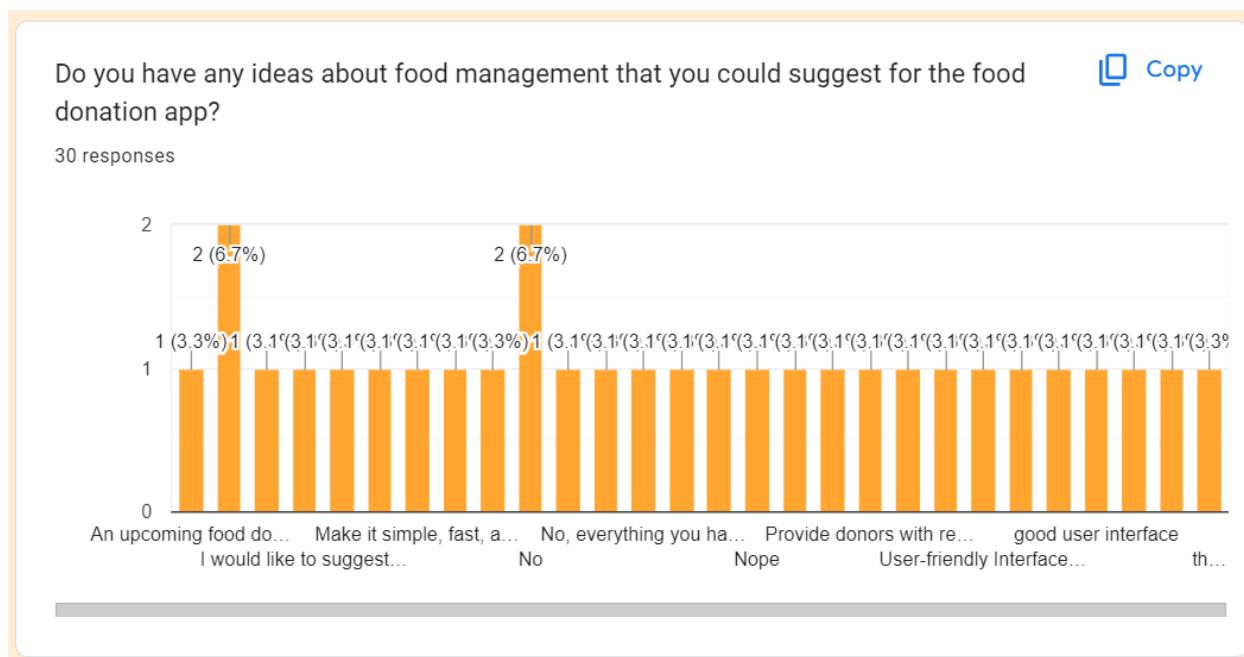


Figure 84: Suggestion for a food donation app? response in a pie chart

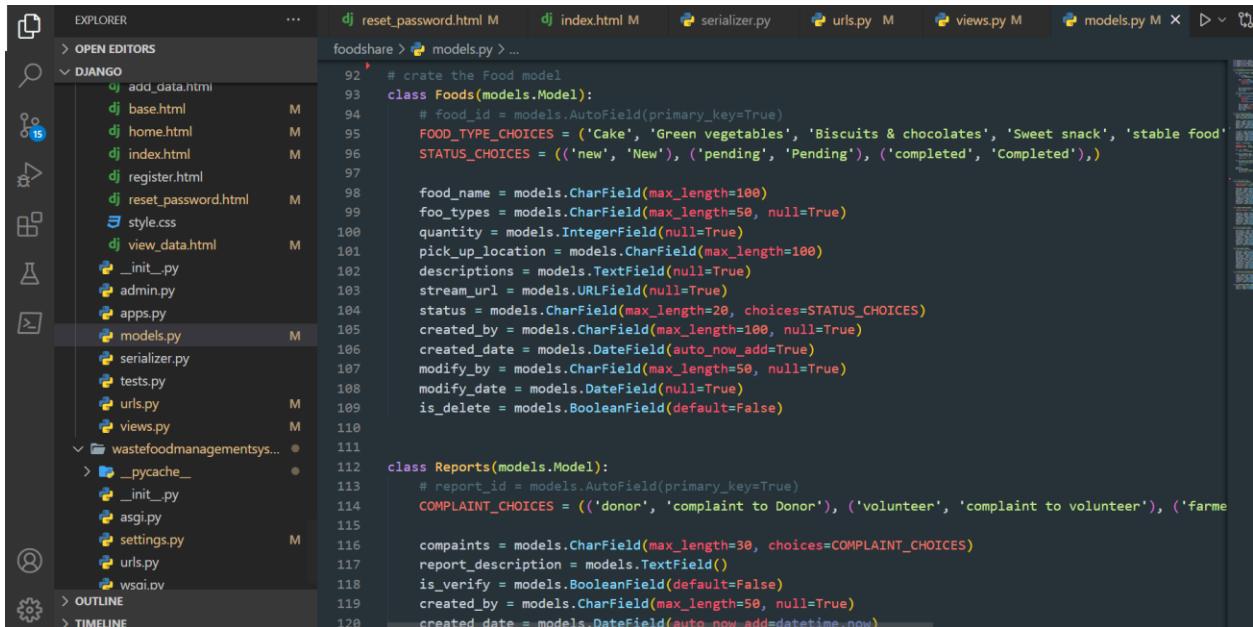
Google form response: https://docs.google.com/forms/d/1nvGA9vr6h--5PxgHTYFd_Ycq6iJtVw_YeNFvg3xbwVQ/edit#responses

[\(Go back to the same page\)](#)

7.10 Work to date

7.10.1. Backend

7.10.1.1. Database development



```

EXPLORER          dj reset_password.html M   dj index.html M   serializer.py   urls.py M   views.py M   models.py M X D V
OPEN EDITORS
DJANGO
> add_data.html
dj base.html M
dj home.html M
dj index.html M
dj register.html
dj reset_password.html M
style.css
dj view_data.html M
__init__.py
admin.py
apps.py
models.py M
serializer.py
tests.py
urls.py
views.py M
wastefoodmanagementsys... ●
> __pycache__
__init__.py
asgi.py
settings.py M
urls.py
wsai.py
OUTLINE
TIMELINE
foodshare > models.py > ...
92 # create the Food model
93 class Foods(models.Model):
94     # food_id = models.AutoField(primary_key=True)
95     FOOD_TYPE_CHOICES = ('Cake', 'Green vegetables', 'Biscuits & chocolates', 'Sweet snack', 'stable food')
96     STATUS_CHOICES = (('new', 'New'), ('pending', 'Pending'), ('completed', 'Completed'))
97
98     food_name = models.CharField(max_length=100)
99     foo_types = models.CharField(max_length=50, null=True)
100    quantity = models.IntegerField(null=True)
101    pick_up_location = models.CharField(max_length=100)
102    descriptions = models.TextField(null=True)
103    stream_url = models.URLField(null=True)
104    status = models.CharField(max_length=20, choices=STATUS_CHOICES)
105    created_by = models.CharField(max_length=100, null=True)
106    created_date = models.DateField(auto_now_add=True)
107    modify_by = models.CharField(max_length=50, null=True)
108    modify_date = models.DateField(null=True)
109    is_delete = models.BooleanField(default=False)
110
111
112 class Reports(models.Model):
113     # report_id = models.AutoField(primary_key=True)
114     COMPLAINT_CHOICES = (('donor', 'complaint to Donor'), ('volunteer', 'complaint to volunteer'), ('farmer', 'complaint to farmer'))
115
116     complaints = models.CharField(max_length=30, choices=COMPLAINT_CHOICES)
117     report_description = models.TextField()
118     is_verify = models.BooleanField(default=False)
119     created_by = models.CharField(max_length=50, null=True)
120     created_date = models.DateField(auto_now_add=datetime.now)

```

Figure 85: Database model development

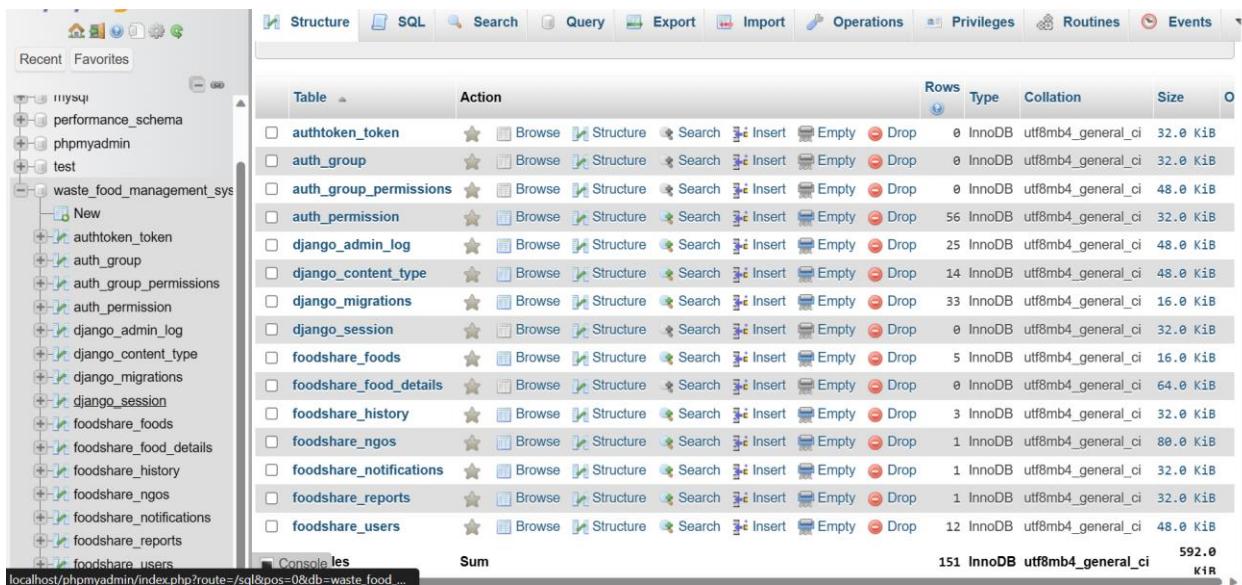
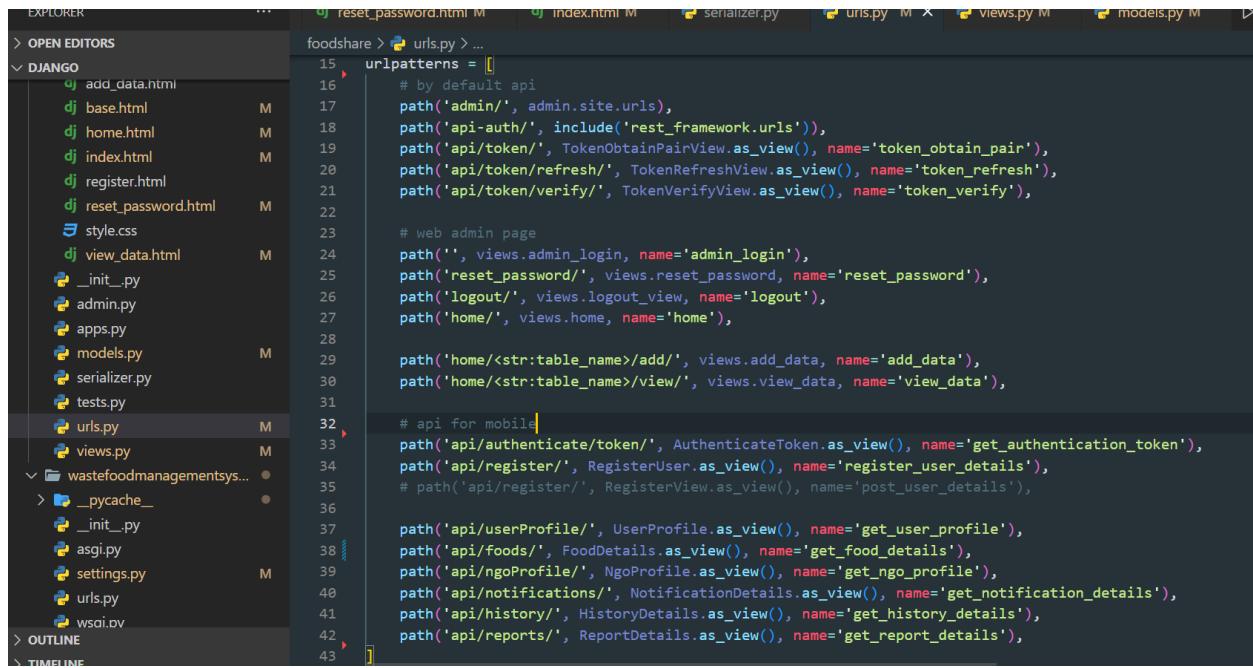


Table	Action	Rows	Type	Collation	Size
auth_token_token		0	InnoDB	utf8mb4_general_ci	32.0 Kib
auth_group		0	InnoDB	utf8mb4_general_ci	32.0 Kib
auth_group_permissions		0	InnoDB	utf8mb4_general_ci	48.0 Kib
auth_permission		56	InnoDB	utf8mb4_general_ci	32.0 Kib
django_admin_log		25	InnoDB	utf8mb4_general_ci	48.0 Kib
django_content_type		14	InnoDB	utf8mb4_general_ci	48.0 Kib
django_migrations		33	InnoDB	utf8mb4_general_ci	16.0 Kib
django_session		0	InnoDB	utf8mb4_general_ci	32.0 Kib
foodshare_foods		5	InnoDB	utf8mb4_general_ci	16.0 Kib
foodshare_food_details		0	InnoDB	utf8mb4_general_ci	64.0 Kib
foodshare_history		3	InnoDB	utf8mb4_general_ci	32.0 Kib
foodshare_ngos		1	InnoDB	utf8mb4_general_ci	80.0 Kib
foodshare_notifications		1	InnoDB	utf8mb4_general_ci	32.0 Kib
foodshare_reports		1	InnoDB	utf8mb4_general_ci	32.0 Kib
foodshare_users		12	InnoDB	utf8mb4_general_ci	48.0 Kib
Console	Sum	151	InnoDB	utf8mb4_general_ci	592.0 Kib

Figure 86: Database table in MySQL

7.10.1.2. Django Rest API development



```

EXPLORER ... reset_password.html M index.html M serializer.py urls.py M views.py M models.py M
OPEN EDITORS
DJANGO
    add_data.html
    base.html M
    home.html M
    index.html M
    register.html
    reset_password.html M
    style.css
    view_data.html M
    __init__.py
    admin.py
    apps.py
    models.py M
    serializer.py
    tests.py
    urls.py M
    views.py M
wastefoodmanagementsys... ●
    __init__.py
    asgi.py
    settings.py M
    urls.py
    wsai.ov
OUTLINE
TIMELINE
foodshare > urls.py > ...
15     urlpatterns = []
16         # by default api
17         path('admin/', admin.site.urls),
18         path('api-auth/', include('rest_framework.urls')),
19         path('api/token/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
20         path('api/token/refresh/', TokenRefreshView.as_view(), name='token_refresh'),
21         path('api/token/verify/', TokenVerifyView.as_view(), name='token_verify'),
22
23         # web admin page
24         path('', views.admin_login, name='admin_login'),
25         path('reset_password/', views.reset_password, name='reset_password'),
26         path('logout/', views.logout_view, name='logout'),
27         path('home/', views.home, name='home'),
28
29         path('home/<str:table_name>/add/', views.add_data, name='add_data'),
30         path('home/<str:table_name>/view/', views.view_data, name='view_data'),
31
32         # api for mobile|
33         path('api/authenticate/token/', AuthenticateToken.as_view(), name='get_authentication_token'),
34         path('api/register/', RegisterUser.as_view(), name='register_user_details'),
35         # path('api/register/', RegisterView.as_view(), name='post_user_details'),
36
37         path('api/userProfile/', UserProfile.as_view(), name='get_user_profile'),
38         path('api/foods/', FoodDetails.as_view(), name='get_food_details'),
39         path('api/ngoProfile/', NgoProfile.as_view(), name='get Ngo_profile'),
40         path('api/notifications/', NotificationDetails.as_view(), name='get_notification_details'),
41         path('api/history/', HistoryDetails.as_view(), name='get_history_details'),
42         path('api/reports/', ReportDetails.as_view(), name='get_report_details'),
43

```

Figure 87: Rest API Url

```

class AuthenticateToken(APIView):
    def post(self, request):
        email = request.data.get('email')
        password = request.data.get('password')
        if email and password:
            # Retrieve the user based on the provided email
            user = get_user_model().objects.filter(email=email).first()
            if check_password(password, user.password) or password == user.password:
                # Password is valid
                serialized_user = UserSerializer(user).data
                if serialized_user['is_active'] == True and serialized_user['is_delete'] == False:
                    refresh = RefreshToken.for_user(user)
                    response_data = [
                        'username': serialized_user['username'],
                        'role': serialized_user['role'],
                        'isSuccess': True,
                        'accessToken': str(refresh.access_token),
                        'message': 'Login successful',
                    ]
                    return Response(response_data, status=status.HTTP_200_OK)
                else:
                    return Response({'error': 'The account is not active', 'isSuccess': 'false'}, status=status.HTTP_401_UNAUTHORIZED)
            else:
                return Response({'error': 'Invalid password', 'isSuccess': 'false'}, status=status.HTTP_401_UNAUTHORIZED)

        return Response({'error': 'Please provide email and password', 'isSuccess': 'false'}, status=status.HTTP_400_BAD_REQUEST)
    
```

Figure 88: Rest API login authentication to get access token

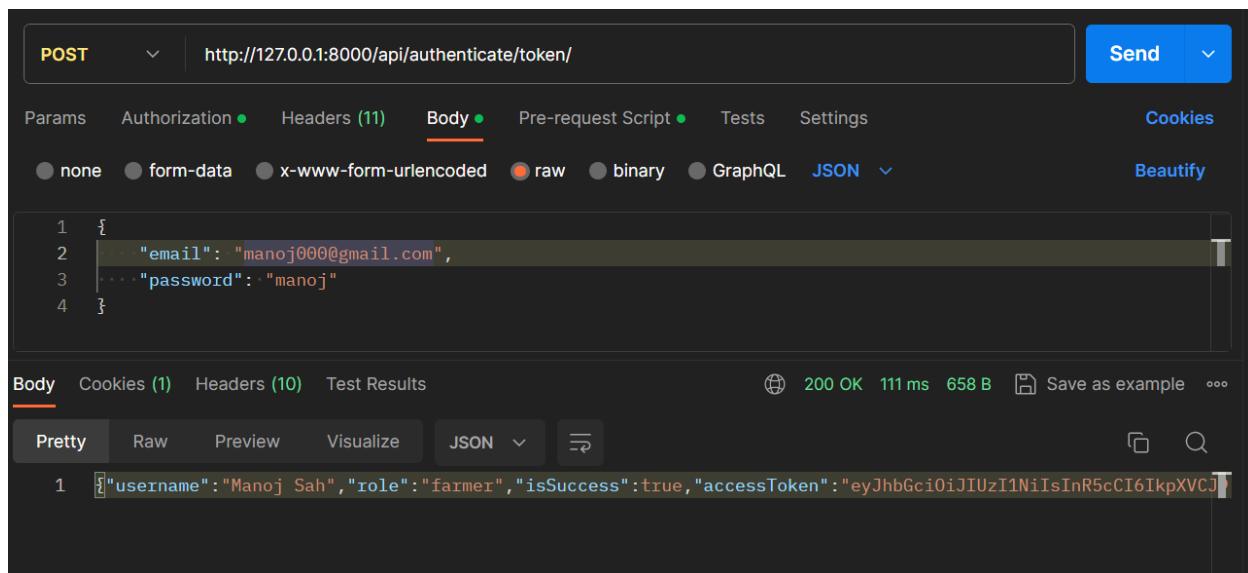


Figure 89: Login authentication API test

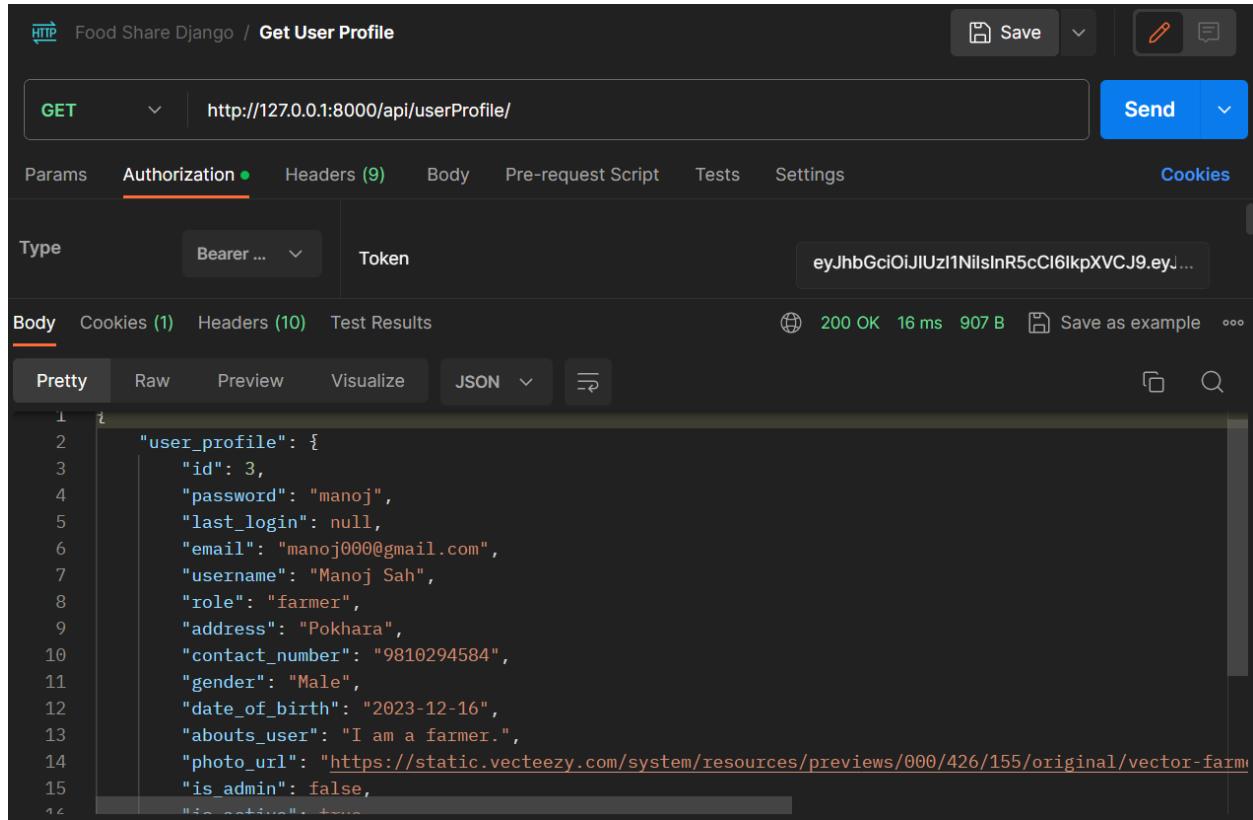


Figure 90: Access token to get the user profile

7.10.2. Frontend

7.10.2.1. Web UI

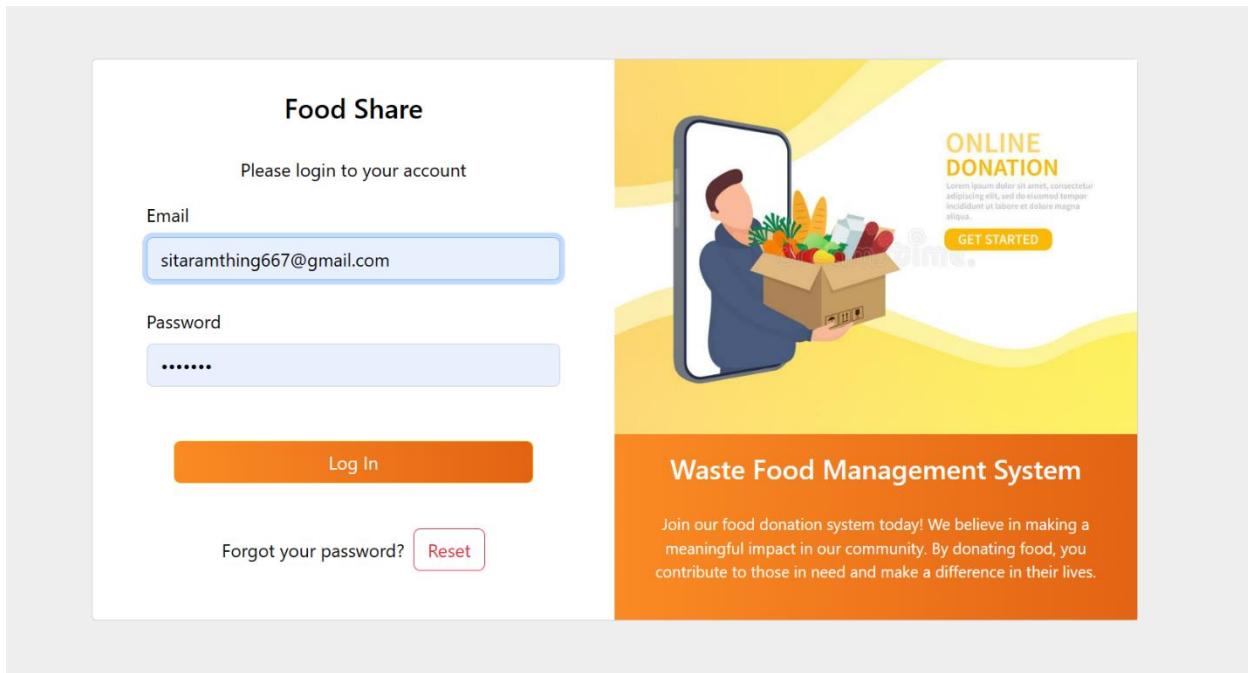


Figure 91: Login page web frontend development

Figure 92: Dashboard web frontend development

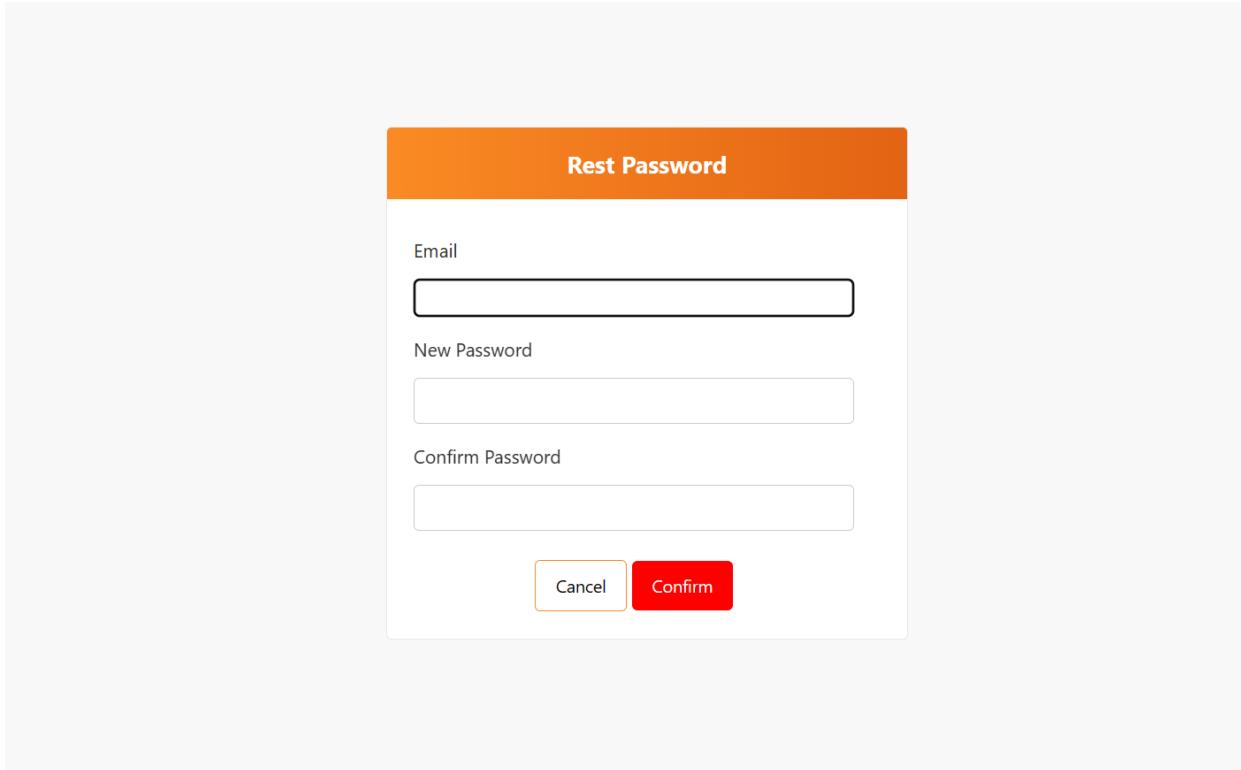


Figure 93: Reset password web frontend development

7.10.2.2. Mobile UI

Figure 94: Mobile Walkthrough Screen 1

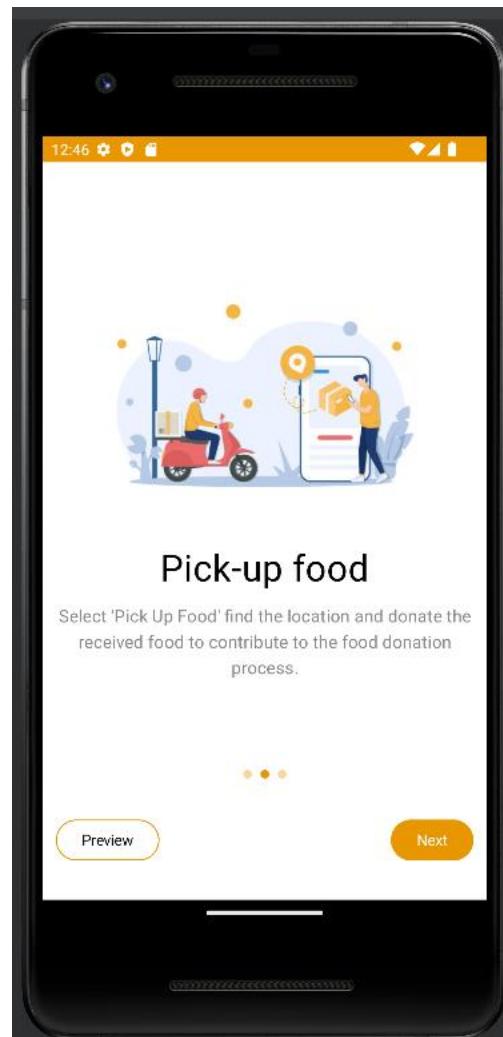


Figure 95: Mobile Walkthrough Screen 2

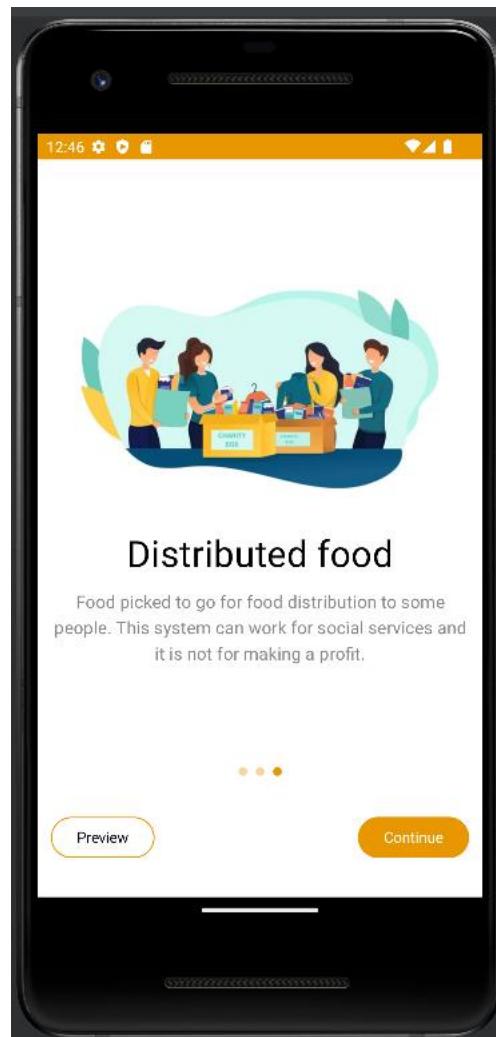


Figure 96: Mobile Walkthrough Screen 3

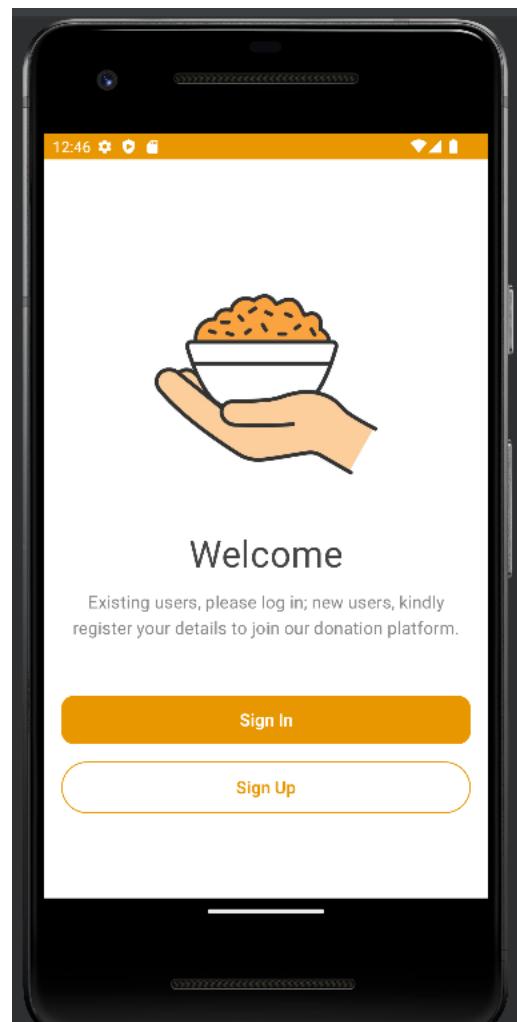


Figure 97: Mobile welcome screen

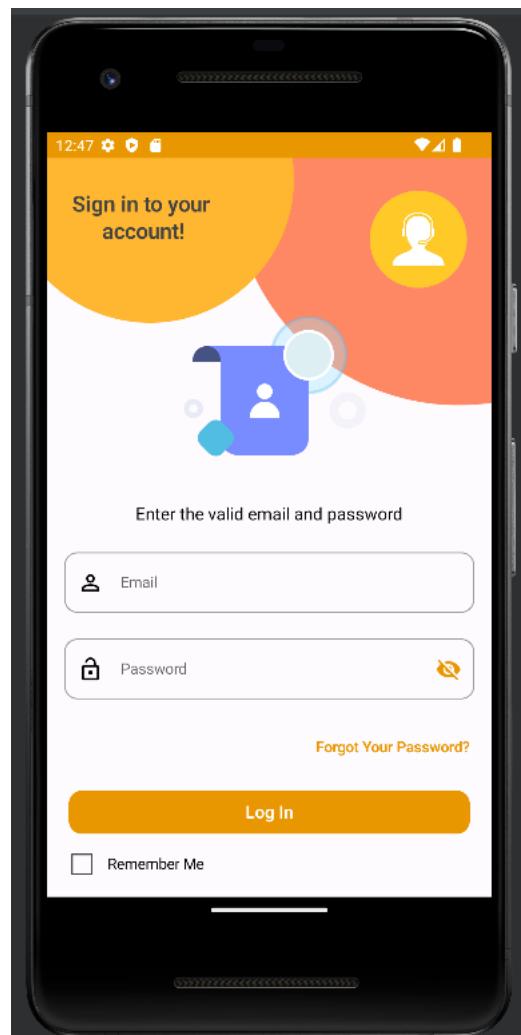


Figure 98: Mobile login screen

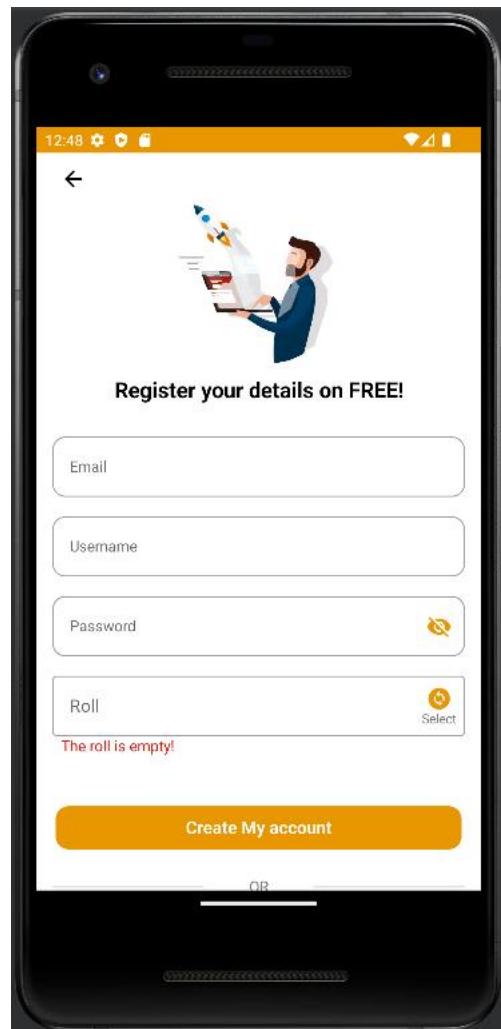


Figure 99: Mobile register screen

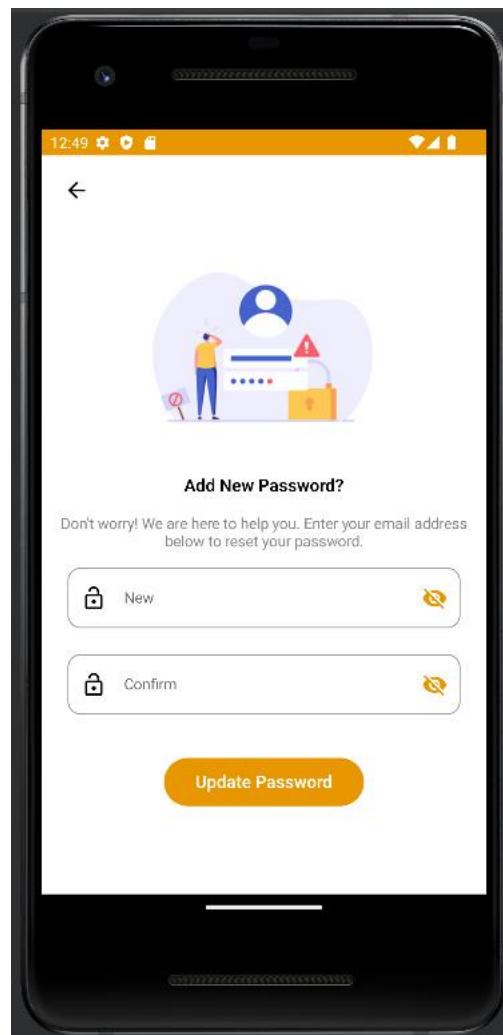


Figure 100: Mobile reset password screen

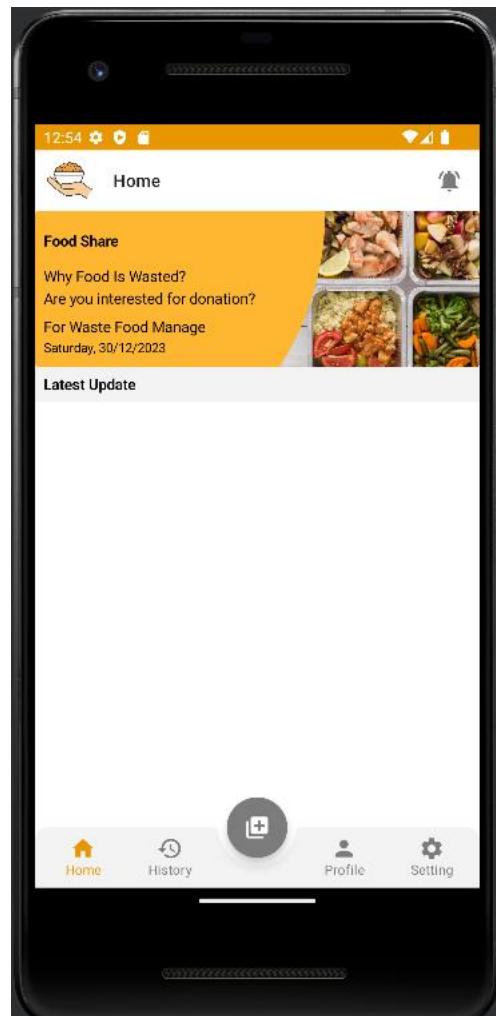


Figure 101: Mobile dashboard screen

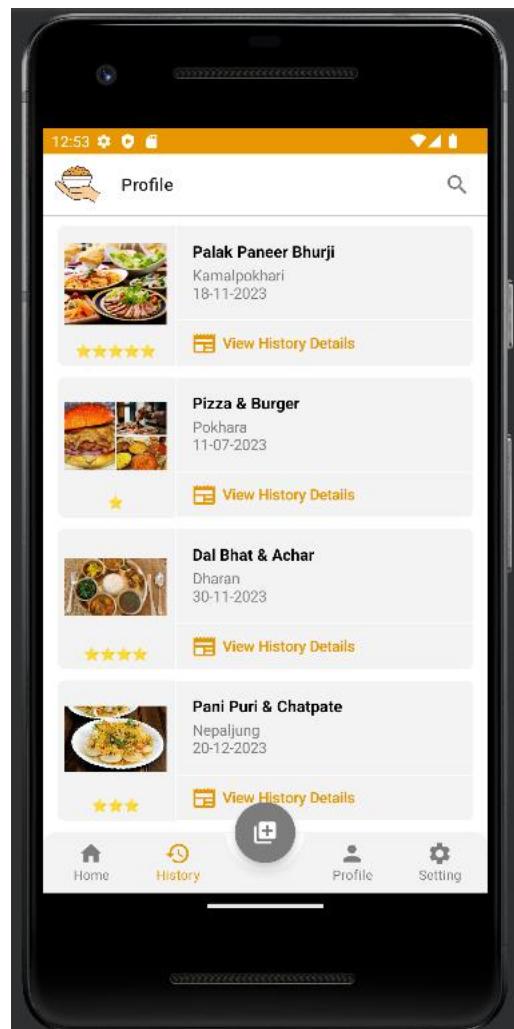


Figure 102: Mobile Profile screen

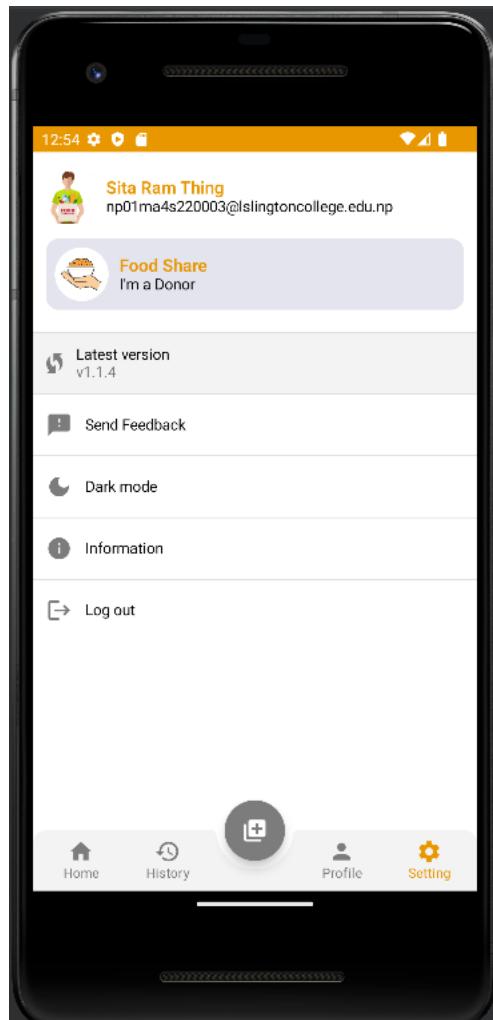


Figure 103: Mobile setting screen

[\(Go back to the same page\)](#)