



**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

24 April 2024

Contents

[CHAPTER 1: INTRODUCTION 1](#_Toc163731657)

[1.1 Project Introduction 1](#_Toc163731658)

[1.2. Current Scenario 2](#_Toc163731659)

[1.3. Problem Domain and Project as a Solution 3](#_Toc163731660)

[1.4. Aim and objectives. 4](#_Toc163731661)

[1.5. Structure of the Report 5](#_Toc163731662)

[1.5.2. Development 5](#_Toc163731663)

[CHAPTER 2: BACKGROUND 7](#_Toc163731664)

[2.1. About the End Users 7](#_Toc163731665)

[2.2. Understanding the solution 8](#_Toc163731666)

[2.3. Comparison 12](#_Toc163731667)

[CHAPTER 3: DEVELOPMENT 17](#_Toc163731668)

[3.1. Methodology 17](#_Toc163731669)

[3.2. Survey Result 17](#_Toc163731670)

[3.4. Design (Concerning methodology what are the core design techniques. For eg: If the methodology is USDP, UML diagrams are a must) 17](#_Toc163731671)

[3.5. Implementation 17](#_Toc163731672)

[CHAPTER 4: TESTING AND ANALYSIS 18](#_Toc163731673)

[4.1. Test Plan 18](#_Toc163731674)

[4.2. Test Execution 18](#_Toc163731675)

[4.3. Critical Analysis 18](#_Toc163731676)

[CHAPTER 5: CONCLUSION 18](#_Toc163731677)

[5.1. LEGAL, SOCIAL AND ETHICAL ISSUES 18](#_Toc163731678)

[5.2. ADVANTAGES 18](#_Toc163731679)

[5.4. LIMITATIONS 19](#_Toc163731680)

[5.3. FUTURE WORK 19](#_Toc163731681)

[CHAPTER 6: References 19](#_Toc163731682)

[CHAPTER 7: Appendix 19](#_Toc163731683)

**Acknowledgement**

In closing, I would like to sincerely thank my external supervisor, Mr. Shekhar Timsina, and internal supervisor, Mr. Bishal GC, for their invaluable advice, support, and upbeat feelings during the completion of this report. I could not have finished this report with such excitement if it weren't for their joint continuous work from the beginning and the helpful feedback that motivated us. Additionally, I would like to sincerely thank all of the seniors and other students who helped me with the composition and compilation of this coursework. Their assistance both directly and indirectly inspired me to complete this coursework before the deadline. Their invaluable input and continuous commitment to editing several iterations of the report have presented me with the best chance to enhance both my report-writing abilities and the standard of my upcoming and present coursework.

Finally, I want to sincerely thank the RTE Department, London Metropolitan University, and other professors and staff members for giving us the coursework requirements and setting up a conducive environment so that we could do our coursework without any confusion.

# CHAPTER 1: INTRODUCTION

Many countries, including Nepal, grapple with the challenges of development, where significant portions of the population face unemployment and poverty. In urban areas, the plight of the homeless and impoverished individuals struggling to afford basic meals is particularly pronounced. Recognizing this pressing issue, initiatives have emerged to address food scarcity by redistributing surplus food resources to those in need. Among these initiatives is the Food Donation App, known as "Food Share," which harnesses technology to streamline the process of collecting and distributing surplus food.

## 1.1 Project Introduction

In the ever-evolving landscape of technological advancements, the Food Donation App, "Food Share," stands as a testament to the transformative power of innovation in addressing social challenges. Spearheaded by Yuvraj Tamang of ING Food Company, this project aims to revolutionise the charitable food donation sector by leveraging digital solutions. By providing a platform for efficient surplus food collection and distribution, the Food Share app bridges the gap between surplus food sources and vulnerable communities, thereby reducing food waste and alleviating hunger.

By consolidating surplus food resources and facilitating their redistribution to those in need, the Food Share app embodies a paradigm shift in charitable food donations. Its user-friendly interface and innovative features empower individuals and organizations to contribute to the fight against food scarcity and waste. Through the collaborative efforts of stakeholders, this digital solution promises to make a meaningful impact in ensuring that no one goes hungry in communities where access to food remains a daily challenge.

## 1.2. Current Scenario

The juxtaposition of surplus food waste with pervasive hunger and malnutrition presents a poignant paradox. Developed nations discard vast quantities of perfectly edible food daily, contributing to environmental degradation and resource inefficiency. Supermarkets, restaurants, and households are major contributors to this surplus, often disposing of food due to overproduction, expiration dates, or aesthetic imperfections. The resulting environmental impact, including methane emissions from landfills, underscores the urgency of addressing this issue.

Concurrently, in many developing nations like Nepal, food insecurity remains a pressing concern despite ample food resources. Limited infrastructure for storage, transportation and costly prices to some low-income sources people cannot afford food coupled with poor market linkages and awareness, exacerbates the problem of food wastage. Even as a significant portion of the population faces hunger and malnutrition, surplus food, often deemed unsuitable for sale due to cosmetic imperfections or nearing expiration dates, goes to waste. This paradox highlights the critical need for innovative solutions to bridge the gap between surplus food sources and vulnerable communities.

In Nepal, where 5.1% of the employed population lives below the poverty line of $1.90 purchasing power parity/day, the challenge of food waste exacerbates existing inequalities. Effectively managing this surplus could significantly alleviate hunger and malnutrition in communities where access to food remains a daily challenge. Addressing this issue requires a multifaceted approach, including the implementation of efficient distribution channels, community education on food preservation and utilization, and the creation of systems that redirect surplus food to those in need. By ensuring that valuable resources are not needlessly discarded, while many go without adequate nutrition, these efforts can contribute to a more equitable and sustainable food system.

## 1.3. Problem Domain and Project as a Solution

In the face of the multifaceted challenges posed by surplus food, the Food Share app emerges as a transformative solution. This visionary project leverages technology to streamline the collection and redistribution of surplus food, addressing not only the logistical hurdles of surplus food management but also the underlying issues of environmental degradation and food insecurity. By serving as a comprehensive platform, the app revolutionizes the process of surplus food distribution, ensuring that valuable resources are efficiently allocated to those in need.

The Food Share app lies in its ability to bridge the gap between surplus food providers and marginalized communities. Through its user-friendly interface, donors can easily input surplus food details, including type, quantity, and expiry dates, with location facilitating a seamless donation process. The app then matches surplus food to the specific requirements of shelters and communities, ensuring that resources reach where they are most needed. Moreover, the inclusion of educational resources and awareness campaigns empowers both donors and volunteers (recipients) with knowledge about waste reduction and responsible food utilization, fostering a culture of social responsibility.

By facilitating efficient surplus food distribution and promoting education and awareness, the Food Share app embodies a holistic approach to addressing the global challenge of food. Through its innovative features and commitment to sustainability, this project promises to make a meaningful impact in ensuring that no one goes hungry while valuable resources are not needlessly discarded.

## 1.4. Aim and objectives.

### 1.4.1. Aim

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

### 1.4.2. Objectives

* Develop a user-friendly interface to facilitate surplus food input, donation matching, and recipient organization interaction.
* Establish a secure database management system to efficiently store and retrieve surplus food data, ensuring data integrity and confidentiality.
* Integrate an effective communication system between food providers and recipient organizations through API programming, enabling seamless coordination and information exchange.
* Implement innovative design elements, incorporating UI/UX principles and material design for an engaging user experience, enhancing accessibility and usability.
* Conduct comprehensive research on existing technologies and tools, identifying and integrating the most suitable ones for optimal application functionality, ensuring the app's effectiveness and efficiency.
* Ensure scalability and adaptability of the food donation platform to accommodate potential expansion and growth, allowing it to meet increasing demands and evolving needs.
* Promote community engagement and awareness through features that educate users about food waste reduction and proper food utilization, fostering a culture of responsible consumption and social responsibility.

## 1.5. Structure of the Report

### 1.5.1. Background

The development phase of the Food Share app involved the creation of both web and mobile platforms. The mobile platform, developed using Kotlin and Jetpack Compose, allows users to access the app's features seamlessly. On the other hand, the web platform, built using Django, HTML, CSS, and Bootstrap, serves as an administrative interface for managing the app's operations. Design considerations encompassed creating a user-friendly interface, establishing database management systems, and integrating communication systems for efficient food distribution. Throughout the implementation process, various challenges were encountered and addressed. These challenges ranged from technical complexities in integrating APIs to ensuring data security and scalability.

### 1.5.2. Development

The development phase of the Food Share app involved the creation of both web and mobile platforms. The mobile platform, developed using Kotlin and Jetpack Compose, allows users to access the app's features seamlessly. On the other hand, the web platform, built using Django, HTML, CSS, and Bootstrap, serves as an administrative interface for managing the app's operations. Design considerations encompassed creating a user-friendly interface, establishing database management systems, and integrating communication systems for efficient food distribution. Throughout the implementation process, various challenges were encountered and addressed. These challenges ranged from technical complexities in integrating APIs to ensuring data security and scalability.

### 1.5.3. Testing and analysis

The testing phase involved rigorous quality assurance measures to ensure the app's functionality, performance, and user experience met the desired standards. Testing methodologies included unit testing and system testing with user feedback. Testing focused on individual components of the app, verifying their correctness and robustness. Integration testing validated the interactions between different modules and systems, ensuring seamless functionality. The analysis of test results revealed areas of strength and areas for improvement. Positive outcomes included the app's intuitive user interface, efficient food donation matching, and secure data management. Areas identified for improvement included performance optimisation, user engagement features, and accessibility enhancements.

### 1.5.4. Conclusion

The Food Share app project represents a significant step forward in leveraging technology to address surplus food and food insecurity. Despite challenges encountered during development, the project has succeeded in creating a platform that streamlines the process of surplus food redistribution and fosters community engagement.

Looking ahead continued efforts will be directed towards refining the app's features, addressing user feedback, and expanding its reach to serve more communities in need. By harnessing the power of technology and collaboration, the Food Share app stands poised to make a meaningful impact in reducing food waste and alleviating hunger.

# CHAPTER 2: BACKGROUND

## 2.1. About the End Users

While most of Food Share's end users are members of the public who value and want to purchase art, the platform was originally designed to meet the requirements of a single artist who recognizes the importance of internet exposure and giving. The end users for the food share donation project encompass a variety of roles, each with specific functionalities and needs:

Client Name: **Yubaraj Tamang**

Client Information and Idea:

Yubraj Tamang is the employer of the ING Food Company's branch in Naxsal, which currently operates under ING Group. He is a beast innovator, helpful, and skilled in cooking. He has a knack for creating numerous food dishes and recipes. His passion for cooking extends beyond the kitchen, as he actively seeks solutions to minimize food wastage and contribute to the community through his culinary skills.

Currently, he works as a Cookman at Naxsal Herald College. Mr Tamang has been involved in cooking food and delivering it from Naxsal to Innovate Tech. He mentioned having an idea regarding the issue of food wastage, where sometimes food goes to waste due to overproduction or the absence of some employees. Consequently, this surplus food cannot be utilized the next day. He suggests that this excess food could be donated to poor or homeless people, thereby solving the problem of wastage. Thus, the food would not be discarded or wasted.

A letter of a contract

Description automatically generated with medium confidence

Figure 1: Client Approvable latter.

### 2.1.1 Web (Admin)

The admin interface is primarily accessed through a web browser. Administrators are responsible for managing and overseeing the entire system. Their tasks may include:

* Managing user accounts and permissions.
* Monitoring donation and distribution processes.
* Analyzing data and generating reports.
* Configuring system settings and parameters.

### 2.1.2 Mobile

Donor: Individuals or organizations who wish to donate food can use the mobile app to initiate and track their donations. Donors may perform tasks such as:

* Registering as a donor and providing necessary information.
* After verifying the account verified then log in to the system.
* Scheduling for food donation and donate, update etc.
* Viewing donation history and receipts.
* Update his/her profile account details.
* Complain the volunteers if they cannot pick up the food.
* View donates food and updates information.

Volunteer: Volunteers play a crucial role in facilitating the collection and distribution of donated food. Their responsibilities may include:

* Registering as volunteers and specifying availability.
* Receiving notifications about donation pickups or distribution events.
* Updating the status of pickups or deliveries.
* Donation completed the send the email to the donor.
* Update his/her profile account details.
* Complaint to the donor if can’t provide the food.
* View donated food details with Google Maps to the location.
* View donor profile details.

Admin: Admin who also used the mobile app for user account verification and report verification.

* Log in to the application on mobile devices.
* Verify the new user and give access or authentication for using the system.
* The donor and volunteer can complete or report verification where food donation time has any problem and give the punishment.
* Update the admin user account details.

Farmer: Farmers who produce surplus food can use the mobile app to coordinate with the platform for donations. Their tasks may involve:

* Registering as farmers and providing details about their produce.
* Indicating availability for donation.
* Communicating with donors or administrators regarding pickup logistics. the farmer can date expired food collected to be used for farming.
* The unhealthy food received from the donor.

Each end-user group has specific functionalities tailored to their role within the food share donation ecosystem, ensuring smooth operation and effective utilization of resources.

## 2.2. Understanding the solution

The Food Share app addresses surplus food management by helping its collection and distribution to those in need. It runs on a web interface for administration and a mobile platform for funders, volunteers, and farmers. The system is intended to effectively connect surplus food sources with beneficiaries, reducing food waste and combating food insecurity. Here's a breakdown of its main components.

Web Interface (Admin):

Administrators control the whole system via a web browser. They handle user accounts, oversee contribution and allocation procedures, analyze data, and configure system settings.

Mobile Platform:

People or organizations may contribute food using the smartphone app. They may examine contribution history and receipts, as well as schedule pickups. Facilitate the collection and delivery of donated food, get information about pickups or distribution events, and track the status of deliveries. Coordinate surplus food contributions using the app by registering, providing produce information, indicating availability, and communicating about collection logistics. The registration procedure includes user verification by the administrator, which grants authenticated access. After logging in, users may view role-specific dashboards. The software supports password management, profile changes, and account deactivation.

### 2.2.1. Overview of the system

The Food waste food management system is the surplus food collected by the distribution system. The food provider (Donor) and volunteers (received) can communicate to pick the food from the donor and goose to distribute to the poor and homeless people. Where the web is using the admin to control the system. The mobile user is a donor and volunteer where the donor can donate the available food and the volunteer can contract to receive the food from a donor.

The new user can register user details from mobile and admin goes to verify the user and give the authenticate access for using the system. After receiving the access, the user can log in to the food share app with, his/her selection role. The donor dashboard has all user's donated latest data. The donor can go to the post screen and new food is posted with all food details. The donor after donating then goes to check the view history with details. Then the donor can home page has multiple donated food view details with a location map and contract to the donor and received food and goes to distribute the food. After completing the distribution then view history with rating. The app has additional features such as forgetting passwords or changing passwords, permanent account deactivation, profile details update and more.

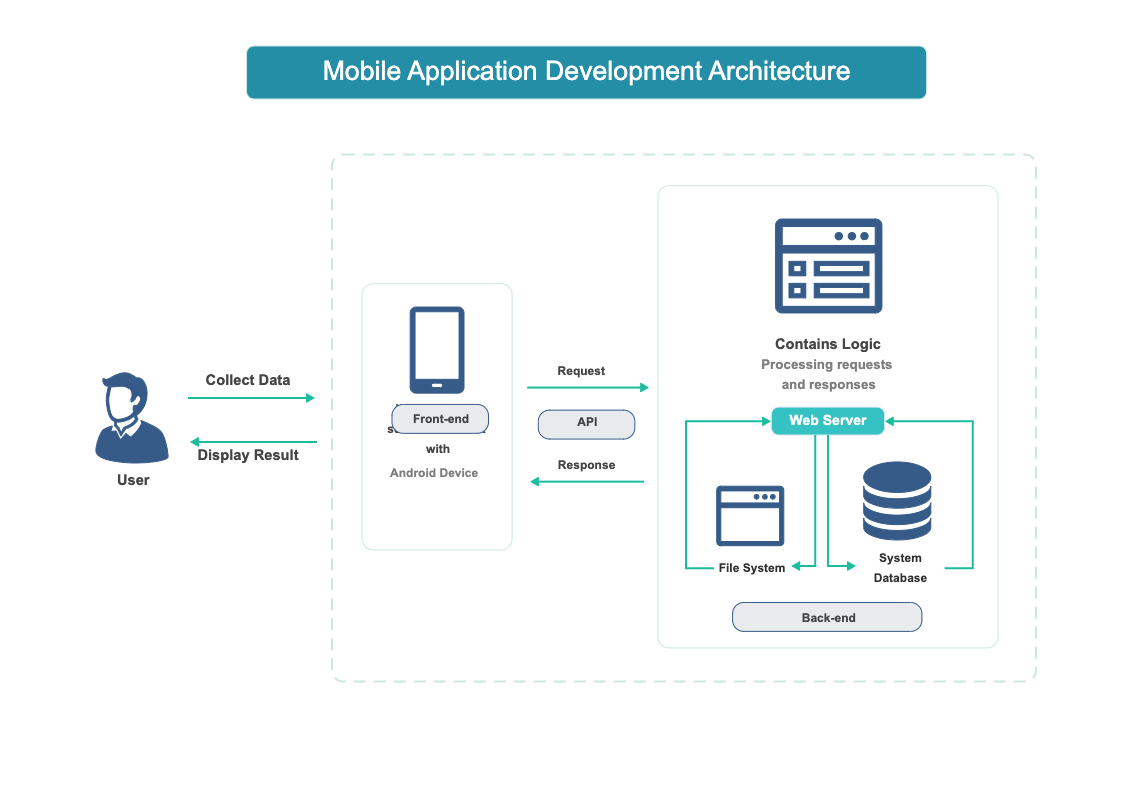


Figure 2: Overview of the system

### 2.2.2. Technical Terms and Definition (Technical Terms related to the project)

To ensure understanding, technical terms related to the project should be defined. This section elucidates terms such as API, database management system, UI/UX, scalability, framework, programming language, firebase push notification etc., providing readers with a comprehensive understanding of the technology and concepts underpinning the Food Share app.

**2.2.2.1 Django, Rest Framework**

Django is a framework used for creating the "behind-the-scenes" parts of websites. It organizes code and helps handle tasks like showing web pages, connecting to databases, and managing user accounts. With Django, developers can build these parts quickly and efficiently using the Python programming language. The rest framework is used to develop the APIs (Application Programming Interface).

**2.2.2.2 HTML, CSS, JavaScript**

The front end is what users see and interact with directly on a website. It includes things like buttons, forms, and menus, as well as the code that makes them work. Developers use technologies like HTML, CSS, and JavaScript to create these user interfaces, making sure they look good and respond smoothly to user actions.

**2.2.2.3 MySQL Database**

MySQL is a type of software used to store and organize data in a structured way. It's commonly used for websites and other applications that need to manage large amounts of information. MySQL offers features that help ensure data is stored securely, accessed quickly, and managed efficiently.

**2.2.2.4 Room Database**

Room is a tool used in Android app development to help manage data locally on a user's device. It makes it easier for developers to work with databases by providing helpful features and simplifying common tasks. The room helps ensure that data in apps is stored and accessed reliably, improving the overall performance and user experience.

**2.2.2.5 Python**

Python is a programming language often used with it. Together, they make it easier for developers to build web applications by providing tools and resources to handle common tasks. Python is known for being easy to read and write, making it a popular choice for developers.

**2.2.2.6 Jetpack Compose/Kotlin**

Jetpack Compose is a tool used in Android app development to create user interfaces (UI) more easily. It works with Kotlin, a programming language commonly used for Android development. Jetpack Compose simplifies the process of building UIs, making it faster and more intuitive for developers to create visually appealing and interactive apps.

**2.2.2.7 Firebase Push Notification**

Firebase is a platform provided by Google that offers various tools and services to help developers build and manage apps more efficiently. It includes features like authentication (managing user accounts), real-time database (storing and syncing data across devices in real-time), cloud messaging (sending messages to users), and hosting (hosting web content). Firebase helps developers focus on creating great user experiences without having to worry about managing server infrastructure.

### 2.2.3. Function and Features

2.2.3.1. Login and Registration:

New users can create or register for the system with valid data and select their role. Authenticated users can access the app's functionalities. Once authenticated, users cannot log in to the system again.

2.2.3.3. Forgot/Reset Password:

Users can change or update their password before logging in if it's forgotten, and they can recover or reset forgotten passwords. The option to reset the password is available both before and after logging into the system.

2.2.3.4. Food Donation:

Enables donors to post surplus food details for donation. Donors can access information about the surplus food they've donated and notify all users.

2.2.3.5. View Food Details:

Provides comprehensive information about available food items. Users can view donated food items, the details of the donating user, the donation location, and the food's current status. After completing the donation, users can view who will receive the food for distribution.

2.2.3.6. View Donation Locations:

Displays the locations for donation pickups. Both donors and volunteers can view donation locations via Google Maps. The map shows both the current user location and the food donation locations.

2.2.3.7. View Profile:

Users can access and update their profile information. All authenticated users can view user profile details. Also, users can view the NGO profiles.

2.2.3.8. Account Deactivation:

Allows users to permanently deactivate their accounts. If users are no longer interested in keeping their accounts active, they have the option to permanently deactivate them.

2.2.3.9. Report/Complaint:

If a donor donates food and the volunteer accepts it for distribution, but the donor cannot give the food to the volunteer, the volunteer can report to the admin. Similarly, if a volunteer cannot pick up the donated food, the donor can report to the admin.

2.2.3.10. Notification (Firebase push notification and Local notification):

When a donor donates food, all user devices receive a notification. Additionally, local notifications are triggered for actions such as forgetting passwords and account deactivation.

2.2.3.11. Rating

Volunteers can collect feedback from food consumers and provide ratings for the donated food. They can assign a rating ranging from a minimum of 0 to a maximum of 5 stars. This rating system helps to evaluate the quality of donated food and ensures continuous improvement in the donation process. Additionally, it encourages transparency and accountability within the food distribution network.

These functions and features ensure a seamless user experience, facilitating efficient surplus food management and distribution while promoting user engagement and accountability.

## 2.3. Comparison

The Food system is so many available in different countries on different platforms where similar projects to some features are different in food share applications. A similar project is Flashfloods, Food Reduce No Waste Food etc.

### 2.3.1. Similar Projects

* + - 1. **Flashfoods**

Author: local government bodies, NGOs,

Flashfoods is a mobile application that connects users with surplus food from local restaurants and grocery stores at discounted prices. Users can browse available food items, purchase them directly through the app, and pick them up from designated locations. Flashfoods aims to reduce food waste by offering discounted surplus food to consumers while providing additional revenue streams for businesses. Flashfood gives unbeatable deals on groceries at peak deliciousness. This application provides the save big on fruit, vegetables, meat, milk, cheese, pantry staples and more but is not fully free distributed. (local government bodies, 2024)

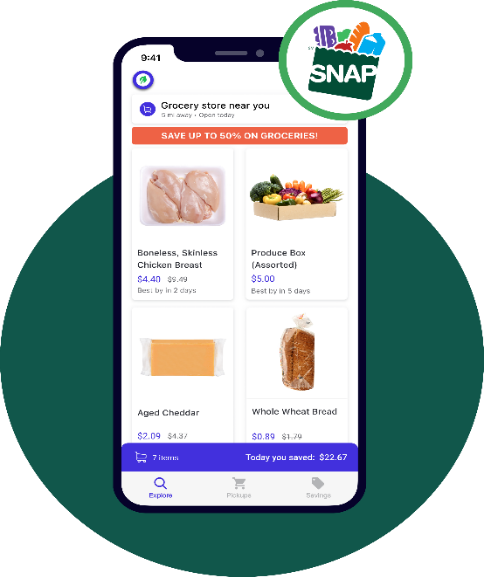


Figure 3: Flash floods similar app image

* + - 1. **Food Rescue**

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

Jeff Schacher and Kevin Mullins founded Food Rescue as Community Plates in Fairfield County in 2011, in response to serious concerns about food insecurity and waste. The system's main goals were to reduce food waste and provide meals to people in need. This innovative approach garnered widespread support from volunteers, food donors, and social service organizations. Over time, they expanded and eventually renamed the nonprofit Food Rescue. It reduces food waste by enabling food donors to donate their surplus food to social service agencies. It also provides meals to those facing food insecurity by allowing social service agencies to join and receive food. To collect and share surplus food seven days a week, and there is no cost to the food donor or the recipient. (Dave Lampert, Now Fair Food NZ, 2024)



Figure 4. Food Rescue similar project image

* + - 1. **No Food Waste**

Author: Anne Frank

"No Food Waste" also collaborates with businesses, schools, and government agencies to implement strategies for reducing food waste in various settings. This includes initiatives such as food rescue programs, composting initiatives, and policy advocacy for standardized date labelling and food donation regulations. That aims to foster a culture of mindfulness and responsibility towards food consumption, ultimately contributing to a more sustainable and resilient food system. Through collective efforts and partnerships, the program seeks to minimize food waste and maximize the utilization of resources, ensuring that everyone has access to nutritious food while reducing the environmental impact of food production and disposal. (Dave Lampert, Food Reduces, 2024)



Figure 5: No Food Waste similar projects image.

### 2.3.2. Comparisons (Comparing the features and critical evaluation of the solution)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Flashfood** | **Food Rescue** | **No Food Waste** | **Food Share** |
| Food Post/Donation | 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 |
| Rating | No | No | No | Yes |
| User View History | No | No | No | Yes |

Table 1: Comparison to a similar project

when comparing the features of Flashfood, Food Rescue, No Food Waste, and my system "Food Share", I find that all platforms support basic functionalities like food posting and donation, as well as push notifications. However, Food Share distinguishes itself with additional features such as local notifications, donation complaint functionality, rating, and user view history. Flashfood lacks donation-compliant capabilities, while No Food Waste does not offer location viewing on Google Maps. Food Rescue lacks a rating feature and user view history. Overall, Food Share provides a more comprehensive set of features, enhancing user experience and administrative capabilities.

# CHAPTER 3: DEVELOPMENT

## 3.1. Methodology

The methodology used the develop the system where it can be helpful to the development process. The different types of projects are suitable for different methodologies.

### 3.1.1. Considered Methodologies

It is a broad topic that can be explored in various contexts, especially in the realm of research and project development. For a final-year project, understanding and explaining considered methodologies typically involves discussing the approaches, frameworks, or techniques that were evaluated and chosen for the project's development or research process. The three methodologies are research like waterfall, scrum and prototype.

**3.1.1.1 Water Methodology**

The waterfall methodology is a sequential, linear approach to project management in which a project's phases must be finished one after the other. It places a strong emphasis on comprehensive documentation, clearly stated project requirements, and an organized, set plan (Theil, 2024). The waterfall methodology has some phases available where planning, designing, development testing and deployment.

Its structured nature can be beneficial for projects with well-defined requirements and stable scope. However, the waterfall methodology also has limitations. For instance, it may not accommodate changes or updates well once the project has progressed beyond the initial planning phase. This rigidity can lead to delays or difficulties in adapting to evolving requirements or unforeseen challenges. Additionally, the waterfall approach may result in longer development cycles, as each phase must be completed before moving on to the next, potentially slowing down the overall project timeline. Therefore, while the waterfall methodology can be effective for certain types of projects, it may not be suitable for those that require flexibility, iterative development, or frequent stakeholder involvement.

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

Advantages of waterfall methodology:

* Clear project organization and control.
* Early error detection, saving development time.
* Efficient time and cost estimation.
* Defect recognition before advancing phases.

The disadvantage of waterfall methodology:

* Requires upfront clear requirements.
* Limited flexibility for changes.
* Lack of risk and security management.
* Not suitable for large projects due to sequential completion.

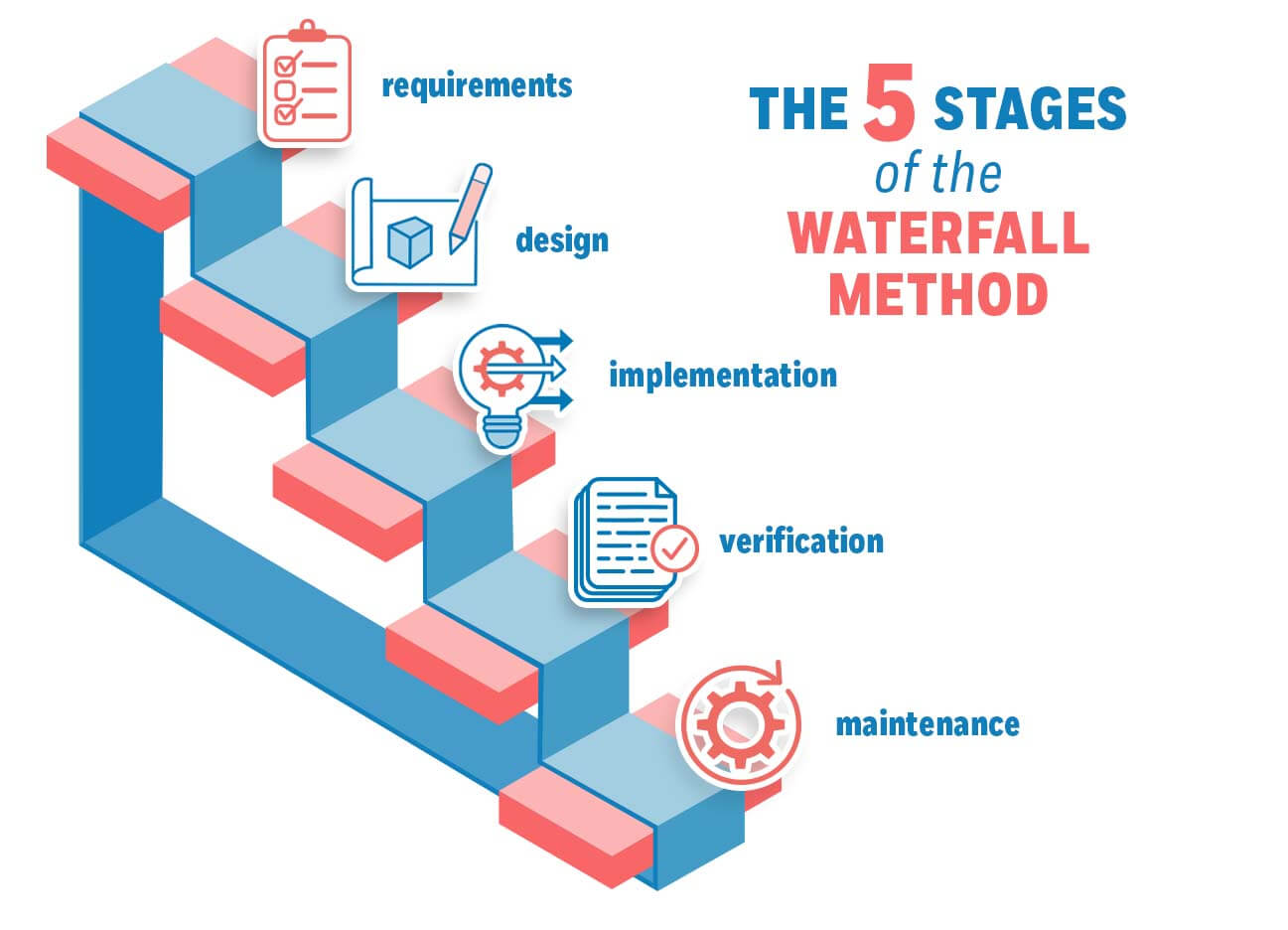


Figure 6: Waterfall methodology.

**3.1.1.2 Scrum Methodology**

Scrum is, in fact, an advancement of Agile Management. The Scrum technique is predicated on well-defined roles and processes that need to be adhered to throughout the software development lifecycle. This adaptable approach incentivizes the use of the 12 agile principles inside a framework that has been decided upon by every member of the product team (S, 2024). Scrum is a management methodology used to self-organize and work together toward common goals. It outlines a sequence of meetings, materials, and duties to expedite project completion. Scrum approaches enable teams to manage themselves, learn from prior experiences, and adjust to changing conditions, much like a sports team prepares for an important competition (AWS, 2024).

While Scrum offers flexibility and responsiveness, it necessitates ongoing management and engagement to ensure smooth progress throughout iterative cycles. In the dynamic realm of software development, Scrum orchestrates agile processes akin to a conductor leading a symphony.

Advantages of Scrum Methodology:

* Ensures efficient utilization of time and resources.
* Breaks down large projects into smaller, manageable sprints, enhancing focus and productivity.
* Ideal for rapid development projects, facilitating quick iterations and delivery.
* Enables concurrent coding and testing during sprint reviews, promoting continuous improvement.
* Embraces agility by actively soliciting feedback from stakeholders and customers, fostering collaboration and alignment.

Disadvantages of Scrum Methodology:

* Lack of a set completion date can lead to scope creep, potentially affecting project timelines and deliverables.
* Project failure is more likely in the absence of commitment or cooperation from team members.
* Implementing the Scrum framework can be challenging in large teams, requiring additional coordination and communication.
* Projects may face significant setbacks if key team members depart midway, impacting continuity and progress.

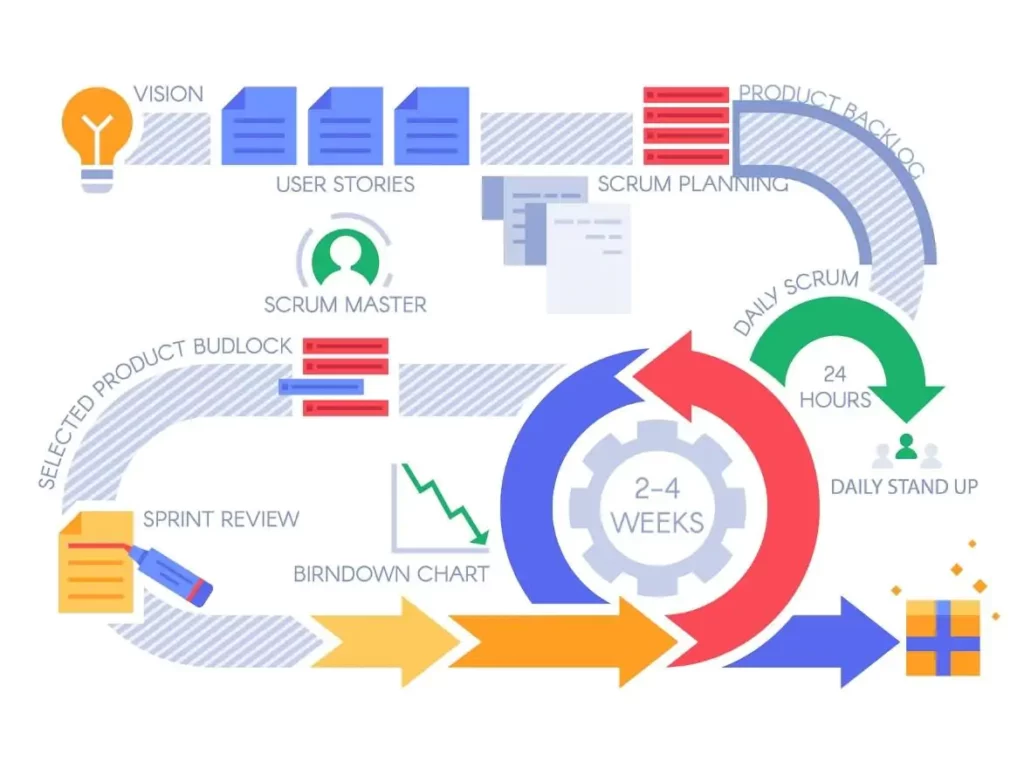
****

Figure 7: Scrum Methodology

**3.1.1.3 Prototype Methodology**

According to the prototype process, a functional system prototype must be constructed before any real software development is done. A model that serves as an implementation of the system is called a prototype. In most cases, a prototype is a relatively unfinished version of the real system that may have few functional features, poor dependability, and ineffective performance when compared to the real software. Often, the client's expectations of the software product are only partially clear. The prototype approach may be used in this situation if there is a lack of precise information about the system's input, processing requirements, and output requirements (JavaTpoint, 2024).

The prototyping methodology is an iterative software development approach where a prototype (a preliminary version of a product) is created to validate and refine design concepts before full-scale development. In this methodology, a basic version of the software or product is quickly developed and presented to stakeholders or end-users for feedback and evaluation. The prototype methodology allows for rapid development, early validation of ideas, and continuous improvement, leading to the development of successful and user-friendly products. Prototyping helps in refining requirements, understanding user needs, and validating design concepts before investing heavily in full-scale development. (LUMITEX, 2024)

Advantages of Prototype Methodology:

* Prototype Methodology provides flexibility in design, allowing for adaptable design processes.
* It aids in the easy identification of errors during development.
* The opportunity for refinement facilitates the smooth integration of new requirements.
* It enhances customer satisfaction and engagement through active user participation.
* User involvement in the development phase promotes collaboration and user-centric solutions.

Disadvantages of Prototype Methodology:

* The implementation of this approach may lead to high development costs.
* Variations in requirements can result in inconsistencies and frequent modifications.
* Rapid prototype development may lead to rushed solutions, compromising quality.
* Increasing complexity may arise within the system as prototypes evolve.
* Inadequate problem analysis may result in deficiencies within the model.

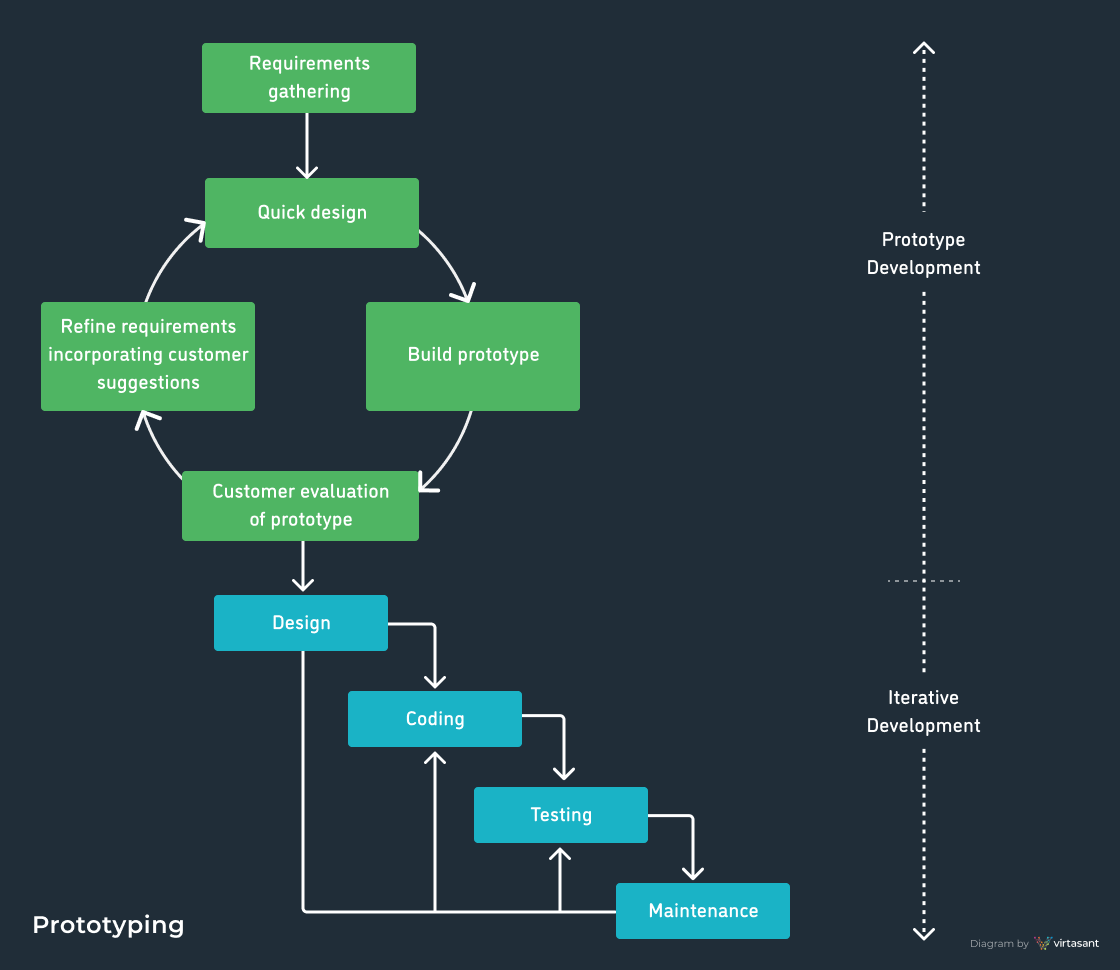


Figure 8: Prototype methodology.

### 3.1.2. Justification for Not Selected Methodologies

The three methodologies are researched to determine which is better suited for final-year project development. The waterfall and prototype methodologies are not selected due to their limitations, making them unsuitable for the project. Therefore, neither of these methodologies is chosen for the development of the FYP (Waste Food Management System).

3.1.2.1 Reason for not choosing Waterfall methodology.

|  |  |
| --- | --- |
| Scenario | The food donation project requires a flexible approach due to evolving requirements and frequent adjustments during implementation. |
| Methodology | Waterfall Methodology |
| Justification | The Waterfall Methodology's linear progression may struggle to accommodate evolving project needs and frequent adjustments. In dynamic projects like food donation, where requirements evolve and continuous improvement is vital, a more adaptable methodology is preferred. Waterfall's rigid structure could hinder the project's ability to respond effectively to changing circumstances, potentially compromising its success. |

Table 2: Justification scenario 1 for waterfall methodology

|  |  |
| --- | --- |
| Scenario | Active stakeholder involvement and feedback are essential for the success of the food donation project, but the Waterfall methodology limits stakeholder engagement until the project's final stages. |
| Methodology | Waterfall Methodology |
| Justification | Waterfall typically restricts stakeholder involvement until late project stages, limiting opportunities for crucial feedback and collaboration throughout the project lifecycle. In projects like food donation, where stakeholder input is crucial for alignment with project goals and user needs, this restricted engagement could lead to misunderstandings and misaligned expectations. A more iterative methodology would better facilitate ongoing stakeholder involvement and feedback, enhancing project success. |

Table 3: Justification scenario 2 for waterfall methodology

|  |  |
| --- | --- |
| Scenario | The food donation project requires rapid iteration and learning from ongoing experiences to ensure its effectiveness. |
| Methodology | Waterfall Methodology |
| Justification | Waterfall's sequential approach does not support rapid iteration and learning from ongoing experiences. In projects like food donation, where continuous improvement and responsiveness are essential, a more iterative methodology is preferable. Waterfall's lack of flexibility and adaptability may hinder the project's ability to incorporate learnings and make necessary adjustments quickly, potentially affecting its effectiveness and success. |

Table 4: 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.

3.1.2.2 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 | While Prototype Methodology emphasizes building and refining a working model based on clear initial requirements, it may struggle to keep pace with rapid changes in a project with continually shifting needs. The iterative nature of prototyping could lead to frequent modifications, creating inefficiencies and making it challenging to maintain a stable prototype aligned with evolving project needs. |

Table 5: 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 environments with constrained time and resources, the iterative nature of prototypes may lead to delays or inefficiencies. This methodology may not be suitable when fixed deadlines must be met or when resource utilization needs to be optimized within a confined timeframe. |

Table 6: Justification scenario 2 for Prototype methodology

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

Table 7: 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.

### 3.1.3. Justification for Selected Methodology

The scrum methodology is selected for project development where the justification and reasons for Choosing the Scrum Methodology are given.

|  |  |
| --- | --- |
| Scenario | The project encounters frequent changes in community needs or regulatory requirements. |
| Methodology | Prototype 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. |

Figure 9: Justification scenario 1 for Scrum methodology.

|  |  |
| --- | --- |
| Scenario | New features or functionalities need to be incorporated swiftly into the project. |
| Methodology | Prototype 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. |

Figure 10: Justification scenario 2 for Scrum methodology.

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

Figure 11: Justification scenario 3 for Scrum methodology.

|  |  |
| --- | --- |
| Scenario | Thorough testing for each feature or component completed in the project is essential. |
| Methodology | Prototype 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. |

Figure 12: Justification scenario 4 for Scrum methodology.

|  |  |
| --- | --- |
| Scenario | Continuous improvement is essential to enhance project outcomes, where Scrum aids in the development phase. |
| Methodology | Prototype 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. |

Figure 13: Justification scenario 5 for Scrum methodology.

|  |  |
| --- | --- |
| Scenario | The project may encounter sudden priority shifts due to unforeseen emergencies. |
| Methodology | Prototype 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. |

Figure 14: 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 FYP project. Therefore, I chose the Scrum methodology.

## Survey Result

### 3.2.1. Pre-Survey Results

### 3.2.2. Post-Survey Results

## 3.3. Requirement Analysis

## 3.4. Design (Concerning methodology what are the core design techniques. For eg: If the methodology is USDP, UML diagrams are a must)

## 3.5. Implementation

The implementation phase of the system development involves translating the finalized design into a functional system.

### 3.5.1. System Development (Important screenshots of development core features and architecture)

3.5.1.1 Database model implementation

This is the just database model implemented where more core features implementation helpful. This phase is crucial as it lays the foundation for storing and managing data efficiently. The user model contains details about users, such as their personal information, login credentials, and any other relevant data. The food model stores information about food items, including their type, quantity, expiration date, and any additional attributes. The history model tracks the history of actions or events within the system, providing a log of user interactions, food donations, or other relevant activities. The notification model manages device tokens, enabling the system to send notifications to users' devices for important updates or events. This phase focuses on structuring the database schema to accommodate the core features of the application, ensuring seamless data management and retrieval throughout the system.

A screen shot of a computer program

Description automatically generated

Figure 15: Implementation of the database user model

A screen shot of a computer program

Description automatically generated

Figure 16: Implementation of database food model.

A screen shot of a computer program

Description automatically generated

Figure 17: Implementation of database report model.

A screen shot of a computer code

Description automatically generated

Figure 18: Implementation of database history model.

A screen shot of a computer code

Description automatically generated

Figure 19: Implementation of database notification device token.

3.5.1.1 Implementation of Push Notification Frontend (Mobile) and Backend (Django)

The system integrates Firebase push notifications for both the front end (mobile) and back end (Django). The system utilizes Firebase Cloud Messaging (FCM) to enable push notifications between the mobile application and the Django backend. Firebase provides a robust and scalable platform for sending messages to devices in real time. On the front (mobile) side, the application leverages the Firebase Messaging library to send and receive push notifications. This allows the application to deliver timely updates and alerts to users' devices, keeping them informed about important events or changes. On the backend (Django) side, the system integrates Firebase Admin SDK to send push notifications to mobile devices. This allows the backend server to trigger notifications based on certain events or conditions, such as new updates or notifications from users. Overall, the implementation of Firebase push notifications enhances the system's communication capabilities, enabling seamless and efficient delivery of notifications to users across both frontend and backend components.

A computer code with many colorful text

Description automatically generated with medium confidence

Figure 20: Implementation of Firebase service account key- backend

A screenshot of a computer program

Description automatically generated

Figure 21: Implementation of service account key in Firebase admin.

A screenshot of a computer program

Description automatically generated

Figure 22: Implementation of notification send function.

A screen shot of a computer

Description automatically generated

Figure 23: implementation for receiving the notification message

A screen shot of a computer program

Description automatically generated

Figure 24: Implementation for generating notification.

A screen shot of a computer program

Description automatically generated

Figure 25: Implementation of notification in custom view

3.5.1.2 Implementation of Donate/Post Food

In the "Food Share" application, users, referred to as donors, are provided with the functionality to donate food items. This feature allows individuals or organizations who are interested in contributing to food donation initiatives to participate actively. Donors who have surplus food available can utilize this feature to post details about the food they intend to donate. This information may include the type of food, quantity, expiration date (if applicable), and any other relevant details. By utilizing this functionality, donors can easily contribute to the food donation process, thereby helping to reduce food waste and support those in need within the community.

A screen shot of a computer program

Description automatically generated

Figure 26: Implement to Food donations (post) - backend.

A screen shot of a computer code

Description automatically generated

3.5.1.3 Implementation of View History

Users, whether donors or volunteers, can view the history of donations made or actions taken within the system. This includes details such as the type and quantity of donated food, the date and time of donation, and any additional relevant information. Donors can access volunteer profile details, allowing them to view information about the volunteers who have participated in the donation process. Similarly, volunteers can view donor profile details, providing transparency and fostering a sense of community within the platform. This feature enhances user experience by providing access to comprehensive donation history details and facilitating interaction between donors and volunteers within the system.

3.5.1.4 Implementation of Rating

The system incorporates a functionality where volunteers can provide feedback and rating points for donors who have donated food. Volunteers can give ratings and provide feedback regarding the donation experience to donors. This feedback may include factors such as the quality of the donated food, the donor's communication and cooperation, and overall satisfaction with the donation process. The system allows volunteers to collect and manage this feedback, ensuring that donors receive constructive input on their contributions. Additionally, volunteers can assign rating points to donors based on their evaluation of the donation experience. This feature promotes accountability and transparency within the donation ecosystem, allowing volunteers to recognize and appreciate donors for their contributions while also providing valuable feedback to enhance the donation process.

A screen shot of a computer program

Description automatically generated

Figure 27: Implementation of Donation rating - backend.

A screen shot of a computer program

Description automatically generated

Figure 28: Implementation of donation rating frontend.

3.5.1.6 Implementation of Complaint/Report

The system incorporates a feature that allows donors and volunteers to report issues or complaints to the admin. Both donors and volunteers have the option to report issues or file complaints to the system administrator. These reports may pertain to various scenarios, such as instances where a donor is unable to provide the promised food donation or when a volunteer fails to pick up the donated food. By providing this functionality, the system ensures that any discrepancies or problems encountered during the donation process can be promptly addressed and resolved. The admin can review the reported issues and take appropriate actions, such as facilitating communication between the parties involved or addressing any underlying issues within the system. The complaint/report feature enhances transparency and accountability within the donation platform, fostering trust and reliability among donors, volunteers, and administrators alike.

A screen shot of a computer code

Description automatically generated

Figure 29: Implementation of report - backend

A screen shot of a computer program

Description automatically generated

Figure 30: Implementation of report-frontend.

3.5.1.7 Implementation of Map

The system incorporates a feature that utilizes maps to display donation locations. The system integrates mapping functionality to enable users to visualize the locations where food donations are made. Users, such as donors or volunteers, can access a map view within the application or platform to see the geographic distribution of donation points. By leveraging mapping technology, users can easily identify nearby donation locations and plan their donation or pickup activities accordingly. The map may display markers or pins indicating specific donation sites, allowing users to navigate and locate them with ease. Overall, the map implementation enhances the user experience by providing a visual representation of donation locations, facilitating efficient navigation, and promoting active participation in the donation process.

A screen shot of a computer

Description automatically generated

Figure 31: Implementation of map key

A computer screen shot of a program

Description automatically generated

Figure 32: Implementation of map access to the current location late value

A screen shot of a computer program

Description automatically generated

Figure 33: Implementation of map state man tent

A screen shot of a computer program

Description automatically generated

Figure 34: Implementation of Google Maps.

### 3.5.2. System Architecture

# CHAPTER 4: TESTING AND ANALYSIS

## 4.1. Test Plan

### 4.1.1. Unit Testing

### 4.1.2. System Testing

## 4.2. Test Execution

### 4.2.1. Unit Testing

### 4.2.2. System Testing

## 4.3. Critical Analysis

# CHAPTER 5: CONCLUSION

## 5.1. LEGAL, SOCIAL AND ETHICAL ISSUES

### 5.1.1. LEGAL ISSUES

### 5.1.2. SOCIAL ISSUES

### 5.1.3. ETHICAL ISSUES

## 5.2. ADVANTAGES

## 5.4. LIMITATIONS

## 5.3. FUTURE WORK

### 5.3.1 Dark Mode

### 5.3.2 Chat (Messaging)

### 5.3.3. Kotlin multi-platform

# CHAPTER 6: References

# CHAPTER 7: Appendix