Tawfeer Mobile Application

Reduce Waste • Share Food • Reward Good

Student Names	ID #
Mahmoud Husam Abdeen	202211625

A project report submitted in partial fulfilment of the requirements for the award of the bachelor's degree in software engineering

College of Engineering

Al Ain University, UAE

ABSTRACT:

Food waste as a topic of conversation in the world is an increasing problem with far reaching consequences on the environment, economy and even the social dimension. In this project, I would propose Tawfeer, a full-fledged mobile application that would minimize food waste through efficient food donation, redistribution, and reuse by facilitating integrated digital ecosystem. This system links households, restaurants, supermarkets and institutions with those in need including a process order to provide food safety and quality control. Tawfeer also has an option of an AI-based recipe generator, where users can recycle an old or uncooked ingredient to create a new meal and encourage sustainable consumption habits. The application will be divided into four key parts, including an interface where users make donations, requests, and create recipes; a management dashboard were accepting an account and reviewing submissions can be done; a driver interface where a collection and delivery are organized; and a government dashboard where donations, savings, and environmental impact can be tracked. The Waterfall development methodology is used in the project to maintain a clear-cut chronological development process of requirements to deployment. Tawfeer is an innovative solution as it can help to curb food waste in the region by simplifying food donation procedures, encouraging responsible consumption, and reinforcing community supporting systems.

Table of Contents

CHAPTER 1 : INTRODUCTION 6

1.1 Background 6
1.2 Problem Statement 6
1.3 Scope 7
1.4 Aims and Objectives 7
1.5 Project Plan 8
1.6 Development Process model 9
1.7 Expected outcomes 10
1.8 Engineering Standards and Constraints 10
1.9 Solution Impact 11
1.10 Conclusion 11

CHAPTER 2 : Domain study 12

2.1 Introduction 12
2.2 Similar Applications 12
2.3 Application Comparison 18
2.4 Analysis: 19

CHAPTER 3 : REQUIREMENTS SPECIFICATION 21

3.1 Introduction 21
3.2 Software Requirements 21
3.3 Use Case Diagrams 27
3.4 Use case description 32
3.5 Conclusion 38

CHAPTER 4 ARCHITECTURE AND DESIGN 38

4.1 Introduction 39
4.2 Database Design 39
4.3 Architectural Design 42
4.4 Interaction diagram 48
4.5 Graphical User Interface (GUI) Design 52
4.6 Conclusion 55

References 76

LIST OF TABLES

Table 1 Application comparison 19

Table 2 User-Related Functional Requirements 22

Table 3 Driver-Related Functional Requirements 23

Table 4 Administrator-Related Functional Requirements 23

Table 5 Government related functional requirements 24

Table 6 Performance Requirements 24

Table 7 Reliability and Availability Requirements 25

Table 8 Security Requirements 25

Table 9 Usability Requirements 25

Table 10 Scalability Requirements 26

Table 11 Maintainability Requirements 26

Table 12 Compatibility Requirements 26

Table 13 Data Integrity and Accuracy Requirements 26

Table 14 Use case description donate food 34

Table 15 Use case description for Get Recipes by AI (Registered User) 35

Table 16 Use-Case Description 2: View Dashboard (Government User) 36

Table 17 Use-Case Description: Evaluate Food Request (Admin) 36

Table 18 Use-Case Description – Verify Food Quality (Driver) 37

LIST OF FIGURES

Figure 1 Project plan Gantt chart **Error! Bookmark not defined.**

Figure 2 (logo of save me app) 13

Figure 3 (save me UI) 13

Figure 4(logo of wajba application) 15

Figure 5 (wajba application UI) 15

Figure 6 (logo of platable app) 16

Figure 7 platable app UI 17

Figure 8 use case diagram for registered user 28

Figure 9 use case diagram for guest user 28

Figure 10 use case diagram for Admin user 29

Figure 11 use case diagram for government user 29

Figure 12 user diagram for Driver 30

Figure 13 tawfeer use case diagram 31

CHAPTER 1: INTRODUCTION

1.1 Background

Food waste is a topical problem of the global agenda, and almost one-third of the total food produced every year worldwide is thrown away. This leads to much degradation of the environment, economic loss and social inequalities. The international studies indicate that more than 10 % of the total greenhouse emissions is caused by food waste and about 30 % of the total agricultural land in the world is devoted to food that will never be consumed. The effects of food waste are not only environmental but also ethical, which leads to issues of resource waste management, starvation, and sustainability.[1]
Cultural requirements, social events, weddings, and hospitality practices in most places especially in the Arab countries contribute to the high food waste, as they tend to make people prepare excessive food than they consume. This broad problem confirms the necessity of new digital solutions that could minimize waste, promote a better redistribution of food and assist societies in need all of which are consistent with the international sustainability objectives.

1.2 Problem Statement

Food waste is one of the problems that are still being encountered despite all the awareness campaigns, particularly in the communities where food surplus is widespread. Households, restaurants, hotels, and supermarkets waste large quantities of safe and edible food every day. This does not only cause harm to the environment but also means a wasted chance to help people and families who are struggling with food insecurity.[2]

At present, no single digital platform exists, which provides a safe food donation, allows food to reuse with the help of artificial intelligence, and has the logistics to collect and deliver donated food. Structured systems to measure food donation measurements to ensure sustainability reporting are also lacking. That is why there is a serious necessity of the complex solution simplifying the food donation process, food safety using quality inspection, matching donors with recipients, and using technology to reuse food and eliminate waste.[1][2]

1.3 Scope

This capstone project has the scope of designing and developing Tawfeer which is an innovative mobile application meant to minimize food wastage and enhance efficiency of redistribution of foods. The system will consist of four large parts:

- User Application: Allows individuals, restaurants, supermarkets, and institutions to donate their food waste or excess, seek food aid or create AI-generated recipes based on the existing ingredients.
- Admin Dashboard: It is a dashboard that enables the administrators to create and maintain user accounts, check and approve requests or donation submissions, assign drivers, and monitor the workflow.
- Driver Application: Gives delivery staff the details of the orders, points of collection, tools to communicate with the donors and recipients.
- Government Dashboard: Provides live updates on the statistics of donations, food savings, carbon footprint, and sustainability indicators.

With these assemblies, Tawfeer will be a one-stop shop in the management of food oversupply, donation logistics, and sustainability reporting.

1.4 Aims and Objectives

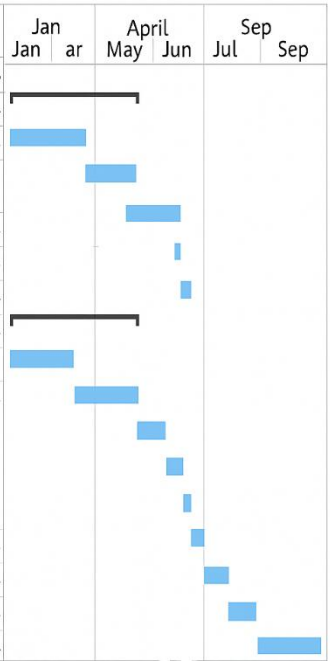
The Aims project will deal with this objective by developing Tawfeer, a single integrated digital app that implements the latest technology, artificial intelligence, and a well-planned logistic strategy to transform how communities manage, reuse, and redistribute food. The application should be more user-friendly, safer and more accessible, the end users should have an iOS and Android app that enables the application of giving away food entering the detail, photo, safety data, and location, ordering food and choosing the allergies and the reasons to get help, and creating an artificial intelligence based recipe out of the leftover or uncooked food to minimize domestic waste. Tawfeer also has systematic quality control system in which safe food is distributed to the recipients and unsafe food re-utilized as compost, animal feed or clean energy production. There is also an administrator

dashboard to verify the user submissions, account management, and assigning of drivers and a driver interface that helps in the timely workflow with collection and delivery and provides navigation, order information and communication devices. A government dashboard also enhances the system by ensuring that sustainability indicators are available in real-time including the rates of donations, food-saving and environmental impact. Tawfeer will be encouraged to facilitate the idea of ongoing interaction by introducing a reward system that would allow people to gain points on making donations and spending them on discounts or benefits. A combination of these features allows stating that Tawfeer is a complex and meaningful answer to the phenomenon of growing food waste.

1.5 Project Plan

The project plan represents the entire development process of Tawfeer maps out the process of the project including preliminary research to the final deployment. This involves requirement analysis, UI/UX design, system development, database creation, AI integration, testing and documentation. A Gantt chart will be used to

ID	Task Mode	Task Name	Duration	Start	Finish
		Tawfeer Project	**170 days**	**09/01/2024**	**02/09/2024**
2		**Capstone 1**	**93 days**	09/01/2024	16/05/2024
3		Domain Study	29 days	09/01/2024	16/02/2024
4		Requirements, Documentation & Diagrams	25 days	16/02/2024	21/03/2024
5		Final Report to Advisor	33 days	21/03/2024	06/05/2024
6		Final Report to examiners	4 days	06/05/2024	09/05/2024
8		Final Presentation for Capstone 1	6 days	09/05/2024	16/05/2024
8		**Capstone 2**	**76 days**	20/05/2024	02/09/2024
9		UI/UX Design & Prototype	23 days	20/05/2024	19/06/2024
10		Implementation - User App (Donation/Request Modules) - Ai Recipe Generator - Admin Dashboard - Driver App - Government Dashboard	26 days		24/07/2024
11		Testing	11 days	24/07/2024	07/08/2024
12		Full System Integration	6 days	07/08/2024	14/08/2024
15		Final Report to Advisore	5 days	24/08/2024	20/09/2024
16		Final Presentation for Capsto	5 days	27/08/2024	02/09/2024



visualize the schedule, the main milestones, and allocation of tasks at the phases. The project is designed in a manner that facilitates an easy flow of the project through preliminary study to system. completion.

1.6 Development Process model

In the development Tawfeer application, Waterfall development model will be used. The Waterfall model is a linear and iterative software development process where one stage has to be completed before the subsequent stage commences. The project is a good fit because it is commonly employed in projects whose requirements are well outlined and the changes that may arise are minimal. This model follows a number of systematic steps involving requirement analysis, system design, implementation, testing, deployment, and maintenance.[1]

This stage will also include a process of detailed requirements to guarantee that all the system features like food donation processes, AI-based recipe generation, driver actions, and governmental dashboard are well outlined and comprehended prior to the design or coding effort. Such a sound base assists in removing the element of doubt and also makes sure that redundancy is removed in the subsequent levels. Once the requirement specification is ready, the project will progress through the rest of the phases. All the phases will be accomplished to the last drop and then will be followed by the next phase, without overlapping or reversal in the working process. This is a procedural, gradual development that facilitates the systematic development of every single element of the Tawfeer system, such as the mobile application interfaces, back-end services, integration of AI, and administrative tools.[3]

When the Waterfall model is adopted, the process of development is organized, predictable, and efficient. It is linear and makes the project less risky and identifiable to the team, as the scope of the project and system requirements are clearly defined. This will be achieved by ensuring the end system is fully

compliant with the stipulated requirements and meets the aims of the project of curbing food waste and aiding in the management of food sustainability.[4]

1.7 Expected outcomes

The main expected outcomes of Tawfeer include:

- An operational mobile application that will make food donation, request processing, and reuse easier.
- Better community involvement in redistribution of food.
- Less waste of food at home and in organizations.
- More donations to charities, food banks and needy families.
- Reduction of landfill waste and emission of carbon.
- One of the government tools in monitoring sustainability efforts.
- AI-driven capabilities of repurposing remnants of food effectively.

1.8 Engineering Standards and Constraints

The software engineering standards in the development of Tawfeer application will be done according to internationally recognized standards in order to have a structured, reliable, and high-quality development process. These consist of IEEE 830 Software Requirements Specifications which offer guidelines on how to document clear, complete and verifiable system requirements; IEEE 1220 Systems Engineering which outlines procedures to be followed in managing system design and integration and finally, ISO/IEC 12207 which presents the entire software lifecycle procedures to be followed to guide the development, operation and maintenance. Compliance with these standards will make sure that Tawfeer is constructed based on best practices that will increase levels of consistency, traceability, and the overall system quality. Nevertheless, the system is also characterized by a number of limitations, such as the reliance on a steady internet connection of users to exchange real-time data, the availability of

effective geolocation solutions to facilitate donation and delivery processes, and the ability to integrate all the necessary data with the involved food banks and institutions to allow effective communication and coordination among all stakeholders.[6][7][8]

1.9 Solution Impact

Tawfeer will bring about tremendous positive change in the social, environmental, and economic sectors. Repurposing and recycling unsafe food as compost or clean energy is a way to help people in need through redistribution and to help reduce the emission of greenhouse gases into the air, which helps in sustainability. Households and businesses can reduce food waste, which is beneficial, both in terms of finances and environmentally. The dashboard created by the government also gives useful information that can be utilized in relation to policymaking and waste reduction initiatives.

1.10 Conclusion

This chapter presented Tawfeer, a big and new system that will address the technology to solve the problem of food waste. The chapter talked about the origins of the problem of food waste and its gravity, established the heart of the problem, and the scope, goals, and objectives of the project. The engineering standards, project plan, and the expected results as well as the development methodology were addressed. Tawfeer will play a leading role in helping to be more sustainable, benefiting the community and protecting the environment due to its combination of AI-based reuse, optimized donation processes, logistics, and government reporting features.

CHAPTER 2 : Domain study

2.1 Introduction

To help readers identify our distinctiveness and positions, this chapter lists at least four applications that are comparable to ours and compares a few of their features. It also highlights the applications' similarities. Additionally, the Using a domain class diagram, the chapter also provides an overview of the concepts connected to the domain.

2.2 Similar Applications

In this part of the chapter, we shall provide a brief description of three applications that are like our proposed application (Tawfeer). The features, limitations, and User Interface (UI) design of each application will be noted and will give an overview of how the current solutions solve the problem of food waste and redistribution. The chosen applications include *Save Me*- an application that matches businesses with the users in order to redeem surplus meals at lower prices*; Wajba*- a relatively new marketplace in the UAE that helps businesses sell their surplus foods at a discounted price; *and Platable* -an application that pairs businesses with users in order to redeem surplus meals and get a discounted price.[3]

2.2.1 Save Me Application

Figure 1 (logo of save me app)

Save Me links restaurants and stores with excess food and allows the customers to purchase unsold meals at a reduced cost via the app. This is to offer low-cost food choices and minimize food wastage in general.[9]

Figure 2 shows the screen from Save Me Application

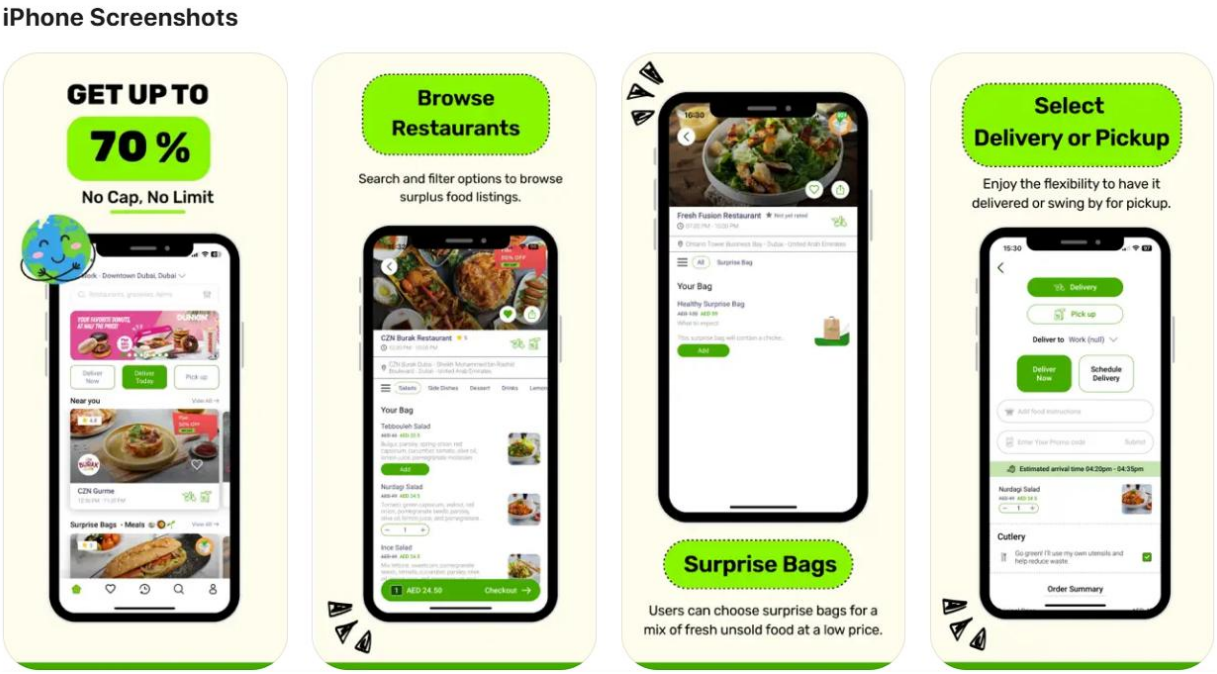


Figure 2 (save me UI)

Features:

- Discounted Meals: When there are excess meals, Restaurants can offer the meals at a reduced price, and the users may acquire them at a lower cost.
- Surplus Management: The companies could effectively cut food waste by selling the products, which would be discarded.
- Surprise Packages: Save Me also includes mystery deals, as well as Too Good to Go, and makes the shopping experience more interesting.
- Local Business Support: Invigorates the small businesses by providing them with a platform where they can sell excess stock.
- Mobile Accessibility: Mobile app is readily accessible on iOS/Android that allows transactions to be made quickly.[9]

Limitations:

- No Donation Flow: The app revolves round sales instead of free redistribution to the needy.
- Safety Reliance: The businesses are dependent on their management of food quality and safety.
- Pickup Only: It has no drivers or delivery as consumers must pick up items on their own.
- Low Reach: It is determined by the number of outlets involved in the region.
- User Constraints: Surprise meal model might not be suitable to certain dietary requirements or preferences.

2.2.2 Wajba Application

Figure 3(logo of wajba application)

A local application called Wajba collaborates with local restaurants, cafes, and bakeries to sell the unsold dishes at a discounted rate as a deal called Wow in a Box. It offers customers unexpected meals and assists the companies to reduce the amount of every-day food waste.[10]

Figure 4 shows the screen from Wajba Application

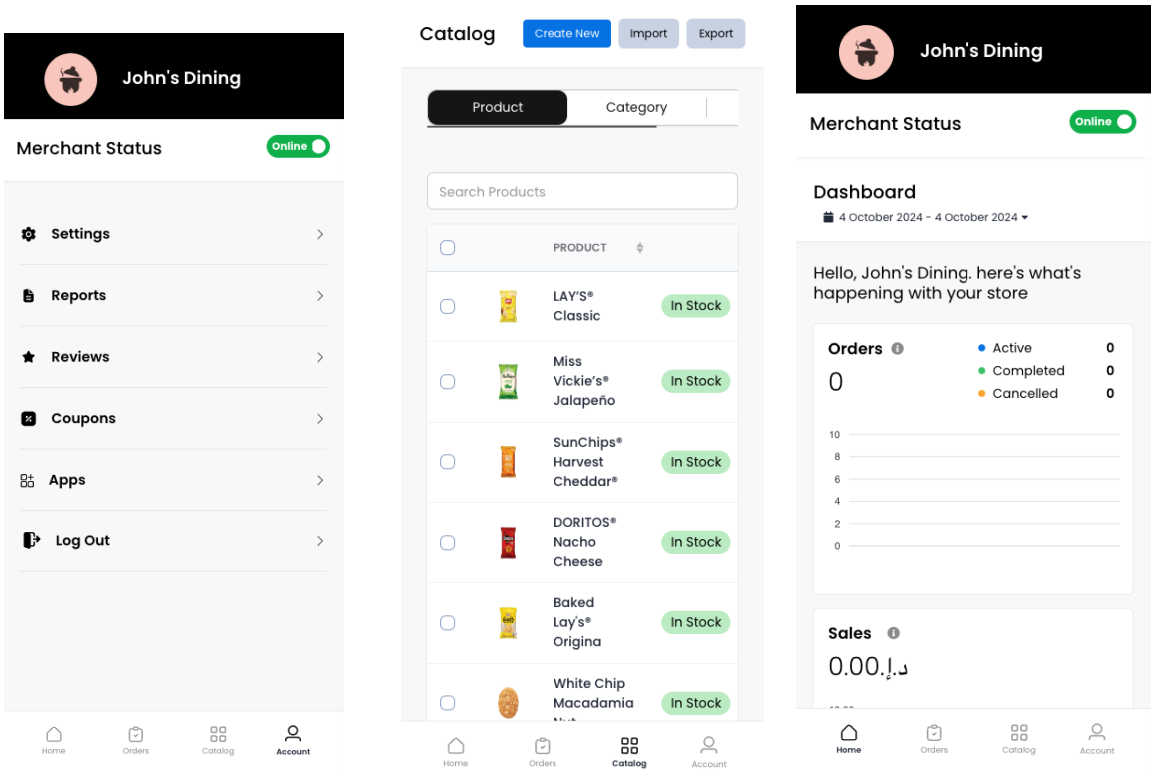


Figure 4 (wajba application UI)

Features:

- Surprise Meals: Discounted pre-packaged meals known as Wow in a Box.
- Business Engagement: Assists the restaurants and cafes in selling excessed stuff rather than disposing of them.[10]

- Food Rescue: Assists in the minimization of food waste by recycling those that are not sold on a day-to-day basis.
- Easy Interface: User-friendly interface to browse and buy the deals.
- Budget-Friendly: It offers low-cost food to the consumers.

Limitations:

- No Donation Component: Does not focus on redistribution of food to vulnerable population.
- Pickup Coordination: This involves having the user pick up the things physically at the merchant.
- Lack of certainty about Variety: The surprise meal model makes it difficult to inform the user of what they are buying.
- Limited Scale: Can only be utilized successfully in the regions with numerous partner restaurants.
- Food Safety: Controlling quality is based on the compliance of the business partners.

2.2.3 Platable-Application

Figure 5 (logo of platable app)

Platable is a new restaurant-based food rescue application that allows restaurants to sell any remaining food items on a reduced price, making them more affordable and reducing food waste. The app facilitates sustainability and business efficiency by converting possible waste to revenue.[11]

Figure 6 shows the screen from Platable Application.

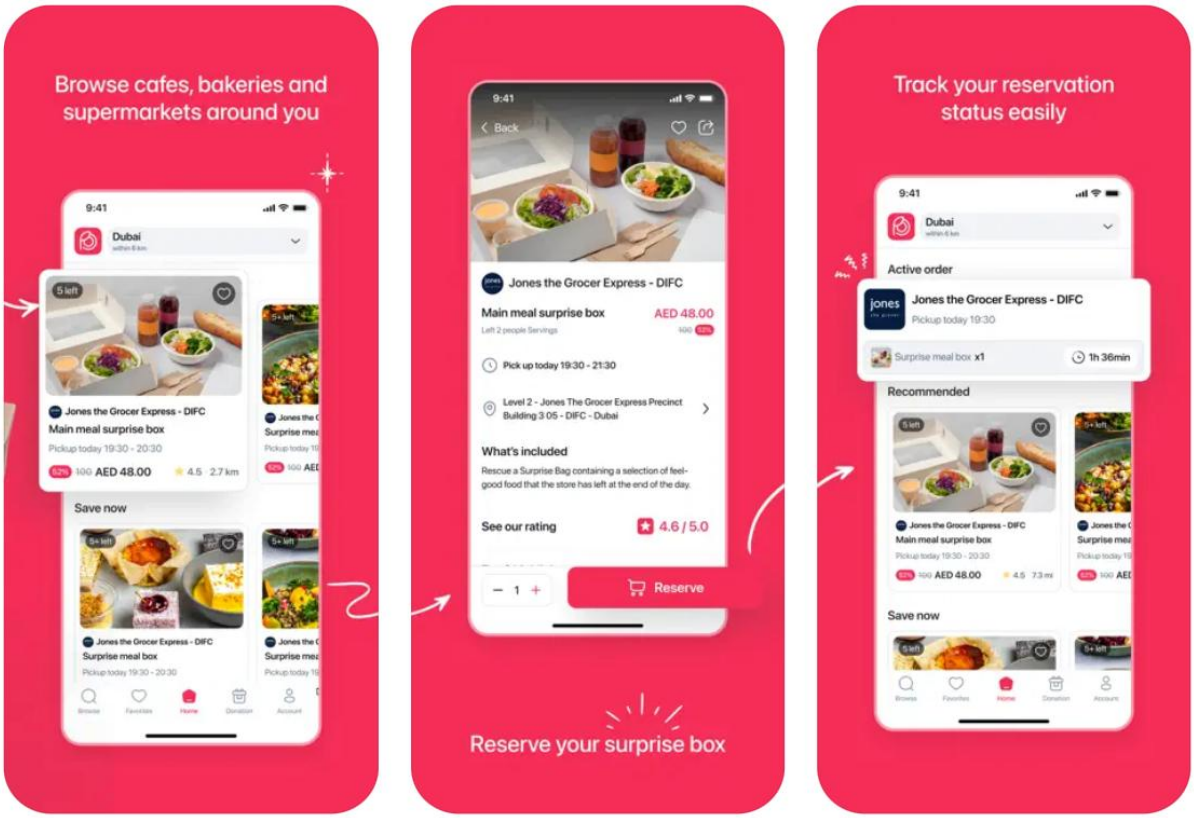


Figure 6 platable app UI

Features:

- Surplus Marketplace: Restaurants and cafes can post surplus food items that would be approaching closing hours.
- Low Cost: Customers receive food by up to 60 percent discounted.
- Intuitiveness: Simple search engine to find deals around.
- Business Benefits: Enables restaurants to reclaim the cost of food that goes to waste.[11]

- Food Waste Awareness: Several benefits of this include sustainability as it connects users to environmentally aware meal consumption.

Limitations:

- No Free Redistribution: It is not a charitable giving site but a commercial resale site, mostly used as an overstocking service.
- Food Pickup Requirement: Pick up will require the customer to drive to pick up their purchases, which can be inconvenient.
- Moderate Safety Testing: Trusts restaurants to package and keep hygienic.
- Limited Distributive Area: The company is still in its infancy and there are limited outlets that are covered by it.
- No Incentives: Lacks reward system of repeat user or donor.

2.3 Application Comparison

Features	Tawfeer	Save Me	Wajba	Platable
Core Purpose	Surplus food redistribution, donation, and recycling	Discounted surplus meals from restaurants & bakeries	Surprise "meal box" at reduced price	Marketplace for surplus food
Free (food sharing)	☑ (redistribution system)	✗	✗	✗
Food- Offers	☑ (optional to partners)	☑	☑	☑
Society Building Tools	☑ (connects NGOs, households, and restaurants)	✗	✗	✗

Food Rescue Focus	☑ (houses, supermarkets, organizations)	☑ (restaurant meals)	☑ (daily café surplus)	☑ (all business types)
Real-Time Updates (keep tracking)	☑ (AI-driven monitoring and notifications)	☑	☑	☑
User-Friendly Interface	☑	☑	☑	☑
Food Safety Guidelines	☑ (built-in safety validation and checks)	Limited (restaurant dependent)	Limited (restaurant dependent)	Limited (partner dependent)
Logistics Support	☑ (driver dashboard & delivery tracking)	✗	✗	✗
Government Dashboard	☑ (sustainability monitoring and analytics)	✗	✗	✗
Reward / Incentive System	☑ (points, vouchers, and recognition)	✗	✗	✗
Meal Planning / Recipes	☑ (AI-based suggestion system)	✗	✗	✗
Target Users	Households, restaurants, supermarkets, NGOs, and government entities	Restaurant diners	Students & young users	General public
Geographical Scope	UAE-wide (scalable nationwide)	UAE (selected restaurants)	UAE (limited cafés/restaurants)	UAE (Dubai-based)

Table 1 Application comparison

2.4 Analysis:

Tawfeer is evidently differentiating with the current UAE apps because it is not limited to the discount-resale model (which is utilized by, Save Me, Wajba, and Platable). Although these applications are mainly used to allow the users to acquire excessive food at lower prices, Tawfeer unites donation flows, government dashboards, drivers' logistics, and developed reward systems.

- Community Building & Sharing: Tawfeer focuses more on the community interaction, allowing businesses, households to donate, and more to share non-food resources, unlike other apps.

- Food Safety & Trust: Tawfeer has an inbuilt safety protocol and verification measures, which has one of the greatest loopholes in competitor apps.
- Operational Advantage: Tawfeer has operational scalability and official reporting capabilities with features such as Driver Dashboard and Government Dashboard that other services do not offer.
- Incentivization: By giving points and vouchers to participants, Tawfeer will be able to guarantee the continued engagement of their users, something that competitor apps have not been interested in doing.

The holistic model makes Tawfeer a complete solution with donation, logistics, incentives, and policy support all in one — something that no apps currently available in the UAE market offer

CHAPTER 3: REQUIREMENTS SPECIFICATION

3.1 Introduction

This chapter outlines the specifications of a proposed mobile application called Tawfeer, which will reduce food waste through a platform to donate/redistribute/reuse excess food in the homes, restaurants, NGOs, drivers, administrators, and government bodies within the UAE.

Section 3.2 presents the functional and non-functional requirements that will include a description of the fundamental capabilities and standards of operation of the system which provide a starting point in the development process. This is followed by Section 3.3 which identifies the requirements of the domain including the contextual and environmental aspects which would affect the operation of the application in the food management ecosystem in the UAE.

Section 3.4 addresses the design limitations which can influence the architecture, technology options and general implementation of the Tawfeer application. The chapter also contains a use case diagram and the use case descriptions to show the user roles and the interaction with the system.

Lastly, the system-level sequence diagram will show the active interaction of different components of the system as it implements various functionalities. This ensures that all the workflows of Tawfeer are seen in clear images, and a clear picture of how the system would behave.

3.2 Software Requirements

Software requirements specification is the important stage of the software development lifecycle. It entails the specification of the services that the system should render as well as any limitation that can restrict these services. The requirements are categorized into two major groups, functional requirements and non-functional requirements.[3][12]

3.2.1 Functional Requirements

The functional specifications of a system explain the exact services and behaviour that a system is required to execute under specific circumstances.[12]

User-Related Functional Requirements

FR1	The application shall allow users (households, restaurants, NGOs) to register by providing name, email address, phone number, and password.
FR2	The application shall allow users to log in to their accounts using their registered credentials.
FR3	The application shall allow users to create new food donation posts by entering details such as food type, quantity, expiration date, and uploading an image.
FR4	The application shall allow users to request available food donations posted by other users.
FR5	The application shall allow users to use AI to analyse and determine whether a food item is still reusable.
FR6	The application shall allow users to view donation listings, including details such as donor name, food type, location, and expiration date.
FR7	The application shall allow users to search for donations based on keywords, categories, or location.
FR8	The application shall allow users to receive notifications for donation approvals, delivery status updates, and reward points.
FR9	The application shall allow users to view their donation history and track the status of ongoing requests.
FR10	The application shall allow users to communicate with drivers or other parties through in-app messaging.
FR11	The application shall allow users to log out from the system.

Table 2 User-Related Functional Requirements

Driver-Related Functional Requirements

FR12	The application shall allow drivers to log in using their assigned credentials.
FR13	The application shall allow drivers to view a list of assigned pickup and delivery tasks from the administrator.
FR14	The application shall allow drivers to view detailed pickup and delivery locations using map integration.
FR15	The application shall allow drivers to update the status of deliveries (e.g., “Picked Up”, “In Transit”, “Delivered”).
FR16	The app will enable the drivers to access the quality of the food by providing them with a checklist or by uploading an image during pickup.
FR17	The application shall allow drivers to communicate with donors, recipients, and the admin.
FR18	The application will enable drivers to inform the administrator about their problems or delays.

Table 3 Driver-Related Functional Requirements

Administrator-Related Functional Requirements

FR19	The application shall allow the administrator to log in securely.
FR20	The application shall allow the administrator to approve or reject food donation posts submitted by users.
FR21	The application shall allow the administrator to assign donation pickup and delivery tasks to available drivers.
FR22	The application shall allow the administrator to monitor ongoing deliveries in real-time via map tracking.
FR23	The application shall allow the administrator to manage user accounts (add, update, suspend, or delete users).
FR24	The application shall allow the administrator to manage the reward and coupon system for active users.
FR25	The application shall allow the administrator to generate reports and analytics on donated, collected, and distributed food.
FR26	The application shall allow the administrator to handle user feedback, issues, and AI analysis results.
FR27	The application shall allow the administrator to log out from the system.

Table 4 Administrator-Related Functional Requirements

Government-Related Functional Requirements

FR28	The application shall allow government representatives to log in using their assigned credentials.
FR29	The application shall allow government users to view system-wide statistics on collected, distributed, and reused food.
FR30	The application shall allow government users to review and approve sponsorships and partnerships.
FR31	The application shall allow government users to access environmental and sustainability reports generated by the system.
FR32	The application shall allow government users to monitor platform activities for compliance and safety.
FR33	The application shall allow government users to log out securely.

Table 5 Government related functional requirements

3.2.2 Non-Functional Requirements

Non-functional requirements specify the quality attributes of the system and its operational attributes. They are used to explain the system in terms of how the system carries out its functions by not stating what functions it carries out. These requirements make the Tawfeer application to be reliable, secure, scalable and easy to use so that it can support its long-term sustainability and performance goals.[13]

Performance Requirements

NFR1	The application shall respond to user requests within 3 seconds under normal network conditions.
NFR2	The AI food analysis feature shall process and return results within 10 seconds.
NFR3	The system shall support a minimum of 10,000 concurrent users without performance degradation.
NFR4	The backend shall efficiently handle large data operations such as donation searches, reports, and analytics.

Table 6 Performance Requirements

Reliability and Availability Requirements

NFR5	The system shall be available 24/7 with a minimum uptime of 99%.
NFR6	The system shall automatically recover from minor failures without losing user data.
NFR7	Regular data backups shall be performed daily to prevent data loss.
NFR8	The application shall provide error-handling mechanisms to notify users in case of server or network issues.

Table 7 Reliability and Availability Requirements

Security Requirements

NFR9	All data transmitted between clients and the server shall be encrypted using SSL/TLS protocols.
NFR10	User passwords will be stored in hashed format (e.g. SHA-256 or bcrypt).
NFR11	The system will also apply role based access control in order to limit unauthorized activities.
NFR12	The system shall comply with UAE data protection regulations and GDPR standards for user privacy.
NFR13	The application will automatically terminate idle users upon a specified time duration.

Table 8 Security Requirements

Usability Requirements

NFR14	The application will be user friendly with an easy user interface that can be used by all types of users.
NFR15	The user interface shall support multiple languages, including Arabic and English.
NFR16	The app will be oriented to rules of accessibility (WCAG 2.1) to accommodate the disabled.
NFR17	The navigation flow will be similar in all the user roles and modules.

Table 9 Usability Requirements

Scalability Requirements

NFR18	The system structure will facilitate the creation of new functionalities or roles of users without significant redesign.
NFR19	The backend shall be cloud-based to allow vertical and horizontal scaling based on user demand.
NFR20	The database shall efficiently handle growth in the number of users, donations, and transactions.

Table 10 Scalability Requirements

Maintainability Requirements

NFR21	The system shall be developed using modular architecture to simplify updates and maintenance.
NFR22	The source code shall be well-documented and follow clean coding standards.
NFR23	The system will enable continuous integration and deployment (CI/CD) to enable effective version control.
NFR24	The admin panel shall allow system monitoring and issue tracking in real time.

Table 11 Maintainability Requirements

Compatibility Requirements

NFR25	The application shall be compatible with both Android and iOS platforms.
NFR26	The application shall support the latest three major versions of Android and iOS.
NFR27	The system shall be compatible with a wide range of device screen sizes and resolutions.
NFR28	The backend APIs shall be compatible with major web browsers for the admin and government dashboards.

Table 12 Compatibility Requirements

Data Integrity and security Requirements

NFR29	The system shall validate all food donation and request data before storage.
NFR30	Every transaction among users, drivers, and administrators will be atomic to eliminate the occurrence of data inconsistencies.
NFR31	The AI analysis module shall maintain an accuracy rate of at least 90% in determining food reusability.

Table 13 Data Integrity and Accuracy Requirements

3.3 Use Case Diagrams

The use case diagram gives a pictorial representation of the high-level interaction between various actors with the Tawfeer application. It depicts the service relationships among the system and its key users such as households, restaurants, NGOs, drivers, administrators and governments. This diagram helps to clarify the overall behaviour of the system and the services that it has to offer to minimize food waste by establishing a structured system of donor and recipient relationships through the donation and reuse of food through identification of the key use cases of each actor. This enables the stakeholders easily to comprehend the roles, authorization, and processes of each form of the user. Also, the use case diagram is a form of groundwork in the system design cycle since it makes sure that all functional requirements are graphically represented before proceeding to the area of analysis and development.[14]

Register User use case diagram

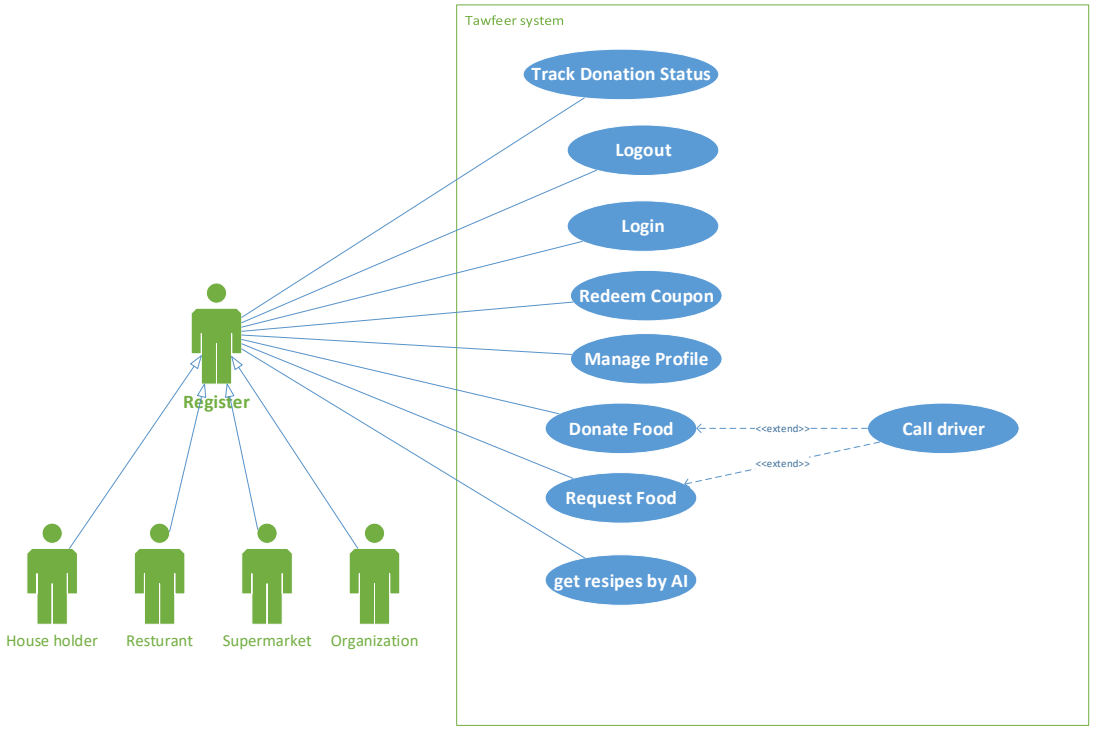


Figure 7 use case diagram for registered user

Guist user use case diagram

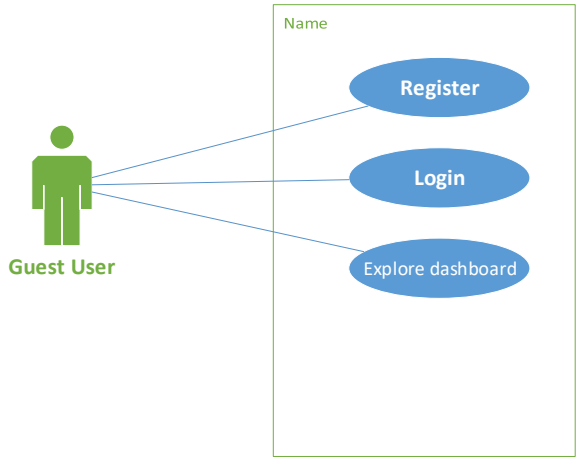


Figure 8 use case diagram for guest user

Admin use case diagram

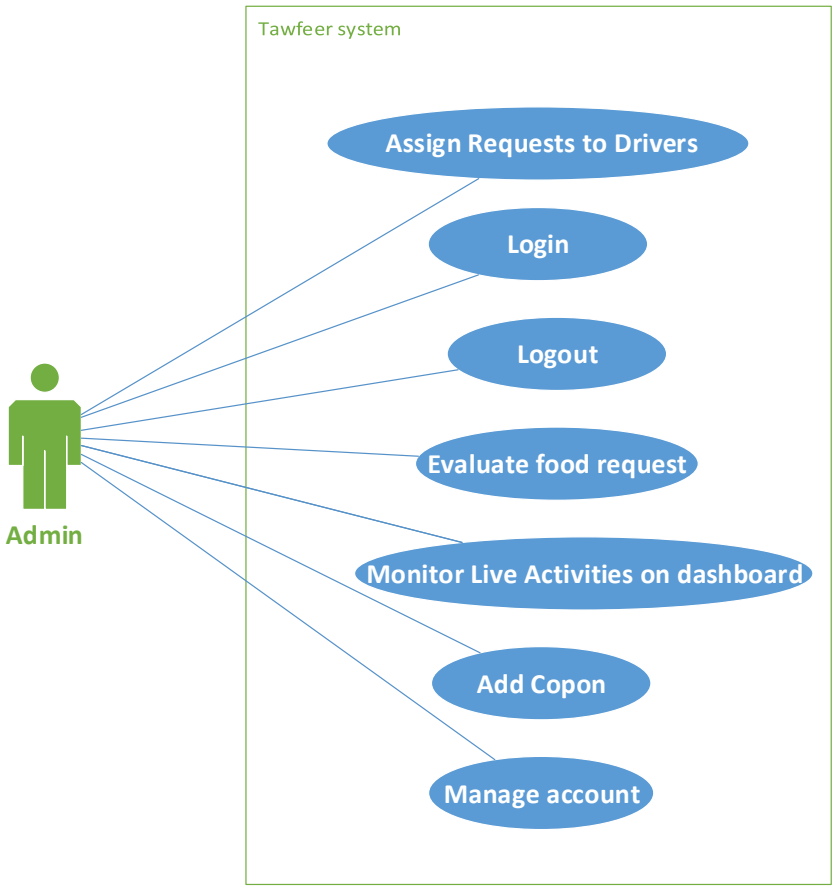


Figure 9 use case diagram for Admin user

Government use case diagram

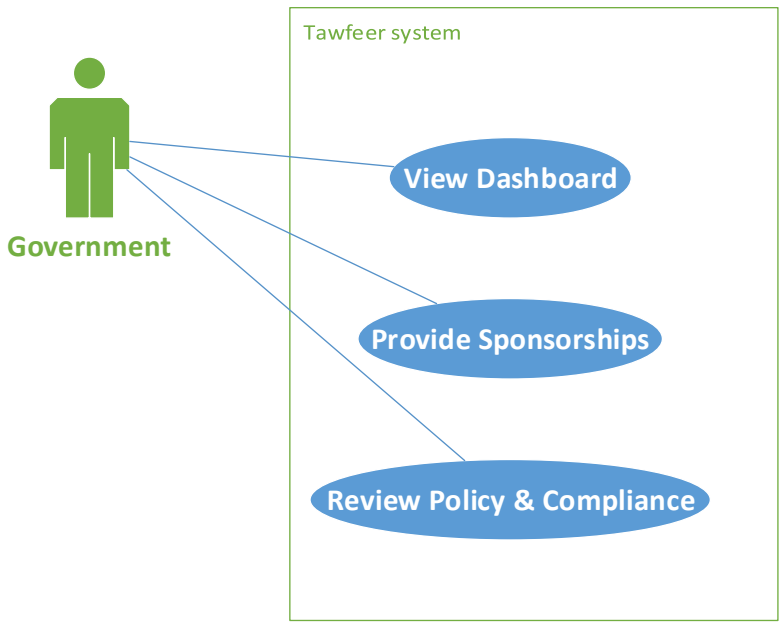


Figure 10 use case diagram for government user

use case diagram for Driver

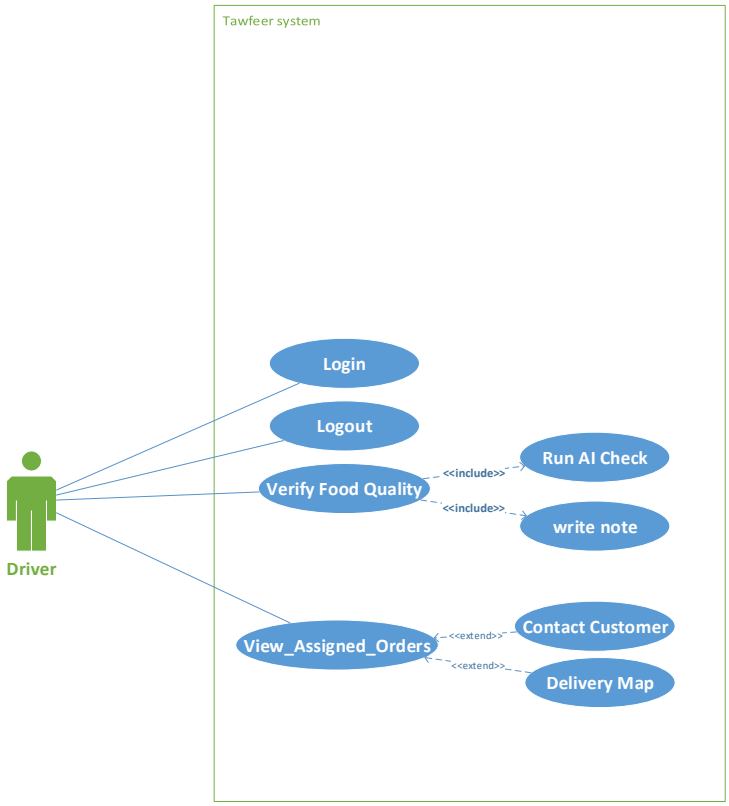


Figure 11 user diagram for Driver

full use case diagram for Tawfeer App system

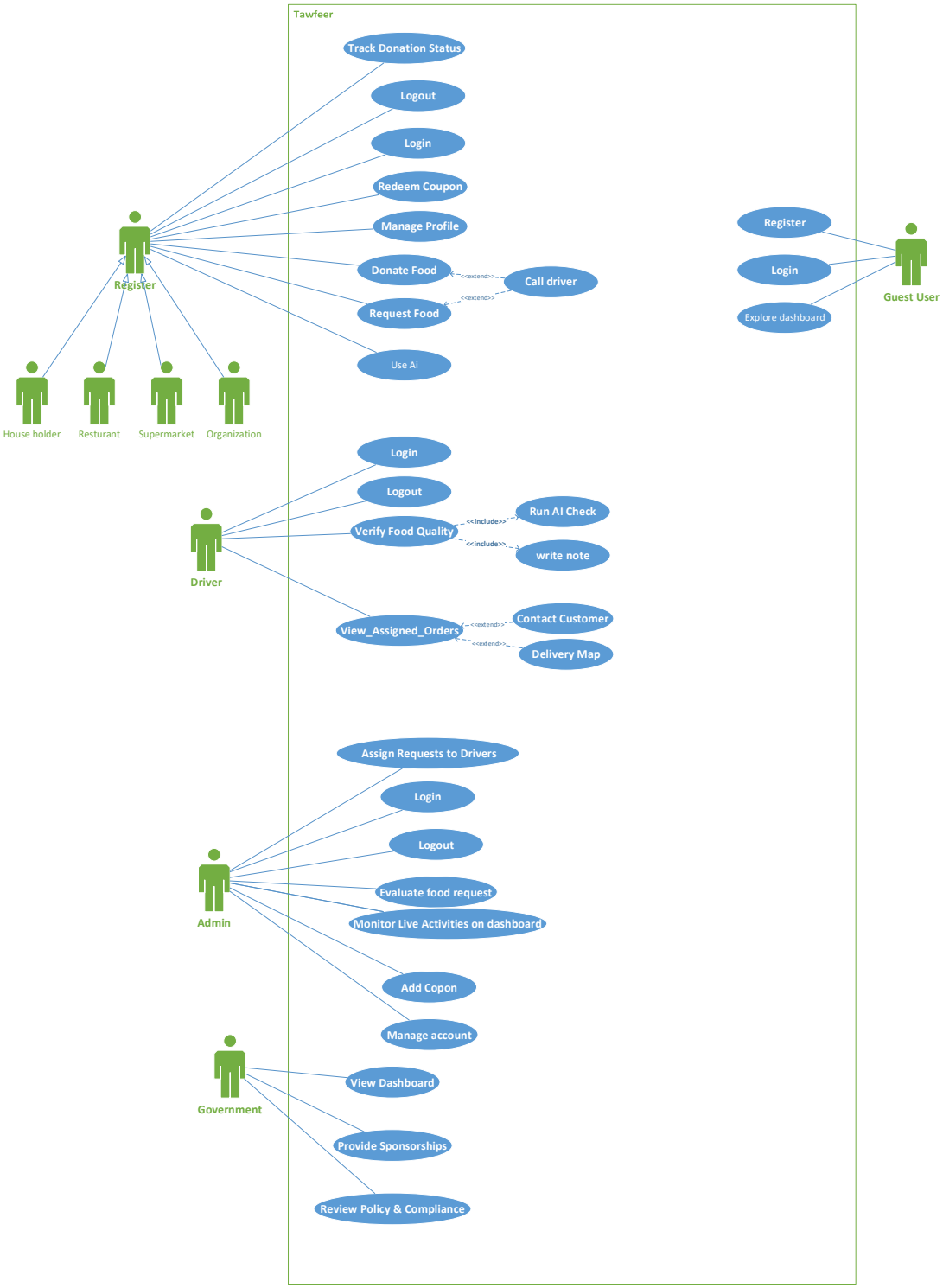


Figure 12 tawfeer use case diagram

3.4 Use case description

A table that explains the use case and how users complete a task on the system is called a use case description table. It also describes how the system behaves from the perspective of the user.[14] The ten essential fields in the use case description table are as follows:

- Use Case Name: The use case's title.
- Use Case Description: An overview of the use case in brief.
- Primary Actor: The primary actor is the one that contributes some input into the system or makes it run.
- Secondary Actor(s): These are the players who help the major players to navigate the system.
- Inputs: These are the information that must be entered by the actors.
- Pre-Conditions: The requirements required to get the use case started.
- Steps: The steps that will have to be undertaken to achieve the objective of the use case.
- Outputs: The output of the system calculations.
- Post-Conditions: These are the circumstances that arise after the use case is completed.
- Alternative scenarios and exceptions: Those actions that keep the use case from accomplishing its objective.

Use case description (Donate Food)

Use-Case Description	
Use-Case Name	Donate Food
Use-Case Description	Allows a registered user (household, restaurant, or NGO) to donate excess food by keying in food details and submitting them to be reviewed by the administration at Tawfeer platform.
Inputs	- Food Name - Food Type - Quantity - Expiration Date - Description / Notes - Image of the food item - Pickup Location
Preconditions	- The system must have the user registered and logged into it. - The food product should not be out of date, and it should not be donated.
Primary Actor	Registered User
Secondary Actor	Administrator
Steps	1. This use case begins when a registered user goes to the page (Donate Food). 2. The system shows the form of donation submission. 3. The customer puts in the information about the food and uploads a picture. 4. The donation form is filled in by the user. 5. The system checks the data that was inputted and sends the request to the administrator to review it. 6. The administrator passes and accepts or disapproves the donation. 7. On a case-by-case basis, the system would post the donation in the available foods list and inform nearby beneficiaries in case it is approved.
Outputs	• Receive message of successful submission. • The notification is sent to the admin to be approved.
Postconditions	In case of the donation being approved, it will be visible to the possible recipients and drivers designated. In the donation list, the food item is displayed as being available.
Exception &Alternative Flows	On step 4, when the user provides invalid or illegal data: • Then with missing required fields, a message of error appears. • Expiration date is in time past: An invalid date is an error displaying. • Image size is bigger than the file size limit: Show an error uploading file.

	• Failure of the network when making a submission: Show Submission failed, please try again. - On rejection of donation by the admin: Message the user on the cause of rejection.

Table 14 Use case description donate food

Use case description for Get Recipes by AI (Registered User)

Use-Case Description	
Use-Case Name	Get Recipes
Use-Case Description	Allows a registered user to upload or input food items and receive AI-generated recipe suggestions based on the available ingredients.
Inputs	- Inventory of items and ingredients. - Optional: Image of ingredients - Food category (optional) - Incivility of AI (e.g., vegetarian, fast meal, low calorie, no lactose)
Preconditions	- The user should be registered and logged in. - The ingredients provided must be valid (not empty, not expired if images are analysed).
Primary Actor	Registered User
Secondary Actor	AI Engine
Steps	1. On the application, the user clicks on the Use AI Recipes. 2. The system shows the form of ingredients or image uploading. 3. The user makes a list of ingredients or uploads an image. 4. The input is checked against the system and passes on to the AI module. 5. The ingredients are analysed by AI and appropriate recipe proposals are offered. 6. The system presents the user with the list of recipes. 7. The user could save, share or see comprehensive steps of a chosen recipe.
Outputs	• Artificial intelligence suggestions of recipes. • Guidelines on optional nutritional or cooking time.
Postconditions	The recommended recipes may be saved in the user's history.

Exception &Alternative Flows	On step 3, when the user provides invalid or illegal data: • Missing ingredients: System issues the user to enter valid ingredients. • Image is too large/invalid Error message displayed. • AI error in service: Inform user and recommend a retrial. • Network failure: Some network error has occurred, trying again.

Table 15 Use case description for Get Recipes by AI (Registered User)

Use-Case Description 2: View Dashboard (Government User)

Use-Case Description	
Use-Case Name	View Dashboard
Use-Case Description	Grant a government representative access to a dashboard with analytics of food donations, distributions, environmental impact, and sustainability, etc.
Inputs	- Government user passwords. - Government user credentials - Selected dashboard area (statistics, reports, etc.)
Preconditions	- The government representative must be logged in. - The role of the user should be authenticated as Government User.
Primary Actor	Government User
Secondary Actor	Admin
Steps	1. The government user logs into the system. 2. he authentication system authenticates credentials and gives access to the dashboard. 3. The user chooses a section of the dashboard (donation statistics, reports, sustainability information, etc.). 4. The system retrieves and displays the needed data. 5. The user perceives maps, summaries and charts. 6. User can make report downloads or filter data. 7. User logs out.
Outputs	• System level insights, graphs and statistics. • Downloadable reports.
Postconditions	To ensure security and compliance, user activity can be logged.
Exception &Alternative Flows	On step 3,5,6, when the user selects the dashboard: • Dashboard is not loading: Displaying Data not available, please try again. • Insufficient permissions: Display access denied. • Network/server error: Notify user of temporary issue.

Table 16 Use-Case Description 2: View Dashboard (Government User)

Use case description	
Use-Case Name	**Evaluate Food Request**
Use-Case Description	Enables the administrator to check, accept or decline food requests expressed by the users according to the availability and eligibility.
Inputs	- Food request details - User information - Request justification - System-generated suggestions
Preconditions	- The administrator should be logged in. - There must be pending food requests.
Primary Actor	Administrator
Secondary Actor	Registered User, Driver
Steps	1. Admin goes to the page of Food Requests. 2. The system displays all pending requests. 3. Admin select a request to assess. 4. The system shows detailed information about the request. 5. Admin approves or rejects the request. 6. With the approval of the system, the request to assign the driver is sent to the system. 7. Messages are posted to the requester and the assigned driver.
Outputs	• Approval or rejection message. • Notifications to user and driver.
Postconditions	Approved orders transfer to delivery scheduling; declined ones are stored.
Exception & Alternative Flows	On step 4–5: • Information missing: System displays an error. • Food unavailable: Admin rejects with reason • System failure: “Unable to evaluate request, try again.”

Table 17 Use-Case Description: Evaluate Food Request (Admin)

Use-Case Description: Evaluate Food Request (Admin)

Use case description	
Use-Case Name	Verify Food Quality
Use-Case Description	Enables a driver to examine donated food products pickup by quality checklist and image verification as a precaution against spoilage and safety.
Inputs	- Food quality checklist - Upload images of food - Optional notes
Preconditions	- The driver must be logged in. - A pickup assignment must exist. - Food must be available for inspection.
Primary Actor	Driver
Secondary Actor	Admin and AI module (optional)
Steps	1. Driver opens assigned order details. 2. Driver selects “Verify Food Quality”. 3. System displays checklist for freshness, appearance, packaging, etc. 4. Driver complete checklist and upload the images. 5. Driver submits the verification. 6. The system sends quality report to the admin. 7. Quality verification is done by Admin reviews.
Outputs	• Food quality verification report. • Updated delivery status.
Postconditions	Approved food is shipped to delivery; rejected products alert the administration.
Exception & Alternative Flows	On step 3–5: • Checklist not filled: System will not allow submission. • Poor quality identified: Driver writes off, administrator informed. • Error uploading photos: Display error message. • No internet: "Checking, later on.

Table 18 Use-Case Description – Verify Food Quality (Driver)

3.1.1.1 Use-Case Description – Verify Food Quality (Driver)

3.5 Conclusion

This chapter has provided the complete requirements specification of the *Tawfeer* application which includes the functional and non-functional requirements which specify how the system should behave. It described all the roles and interactions with all the users which was backed up by use case diagrams and descriptions that explain how the system works. These specifications offer a strong platform on which the design and development of the application will be developed so that Tawfeer can be developed effectively, securely, and per its aim which is to minimize the wastage of food and promote sustainable food redistribution in the United Arab Emirates.

CHAPTER 4 ARCHITECTURE AND DESIGN

4.1 Introduction

The last chapter gave the requirements specification of the Tawfeer application (both functional and non-functional requirements). The design stage converts these requirements into formatted system architecture that stipulates the manner in which the system will be implemented.

This chapter includes the architectural design and database design of the Tawfeer application. It informs about major concepts of database design, then the Entity-Relationship Diagram (ERD) and database schema. It also explains the general system architecture comprising of components and deployment structure. This chapter is intended to offer a proper blueprint on how to create a scalable, secure and efficient food redistribution platform.

Tawfeer is a mobile-based application aimed at minimising food waste by donating, redistributing, reusing via AI, coordinating logistics, and creating government monitoring dashboard. Consequently, it should have its architecture in such a way that it accommodates various actors such as users, administrators, drivers, and government authorities.

4.2 Database Design

Design of databases is one of the crucial steps in the software development lifecycle because it determines how data will be organized, stored and accessed in a system. An effective database design is one through which information is arranged in a logical manner, which makes storage easy, storage quick, and data maintenance is always maintained. It also facilitates data integrity as it enforces constraints and relationship among entities and this is used to avoid duplication, inconsistency and loss of data. A good database design enhances the performance of the application in general and allows it to easily be scaled according to the increase in the number of users and transactions over a time.

In contemporary use, database design is usually a systematic procedure that involves the identification of entities, defining attributes, the formation of relationships, and use of normalization. Normalization is adopted to remove the redundancy and to make sure that each data segment is stored once only, and this enhances its consistency and storage demands. Also, application of primary and foreign keys is an assurance that every single record can be identified uniquely and that interrelationships among the tables are accurate. The practices of design can help to keep a system of data intact and reliable.[14]

4.2.1 Entity–Relationship Diagram (ERD)

Entity- Relationship Diagram (ERD) is a graphical image that is used to create an image of the database structure by depicting the objects in a system and how they relate to one another. ERDs assist the database designers in knowing how the data is organized, stored and related and this results into effective database implementation and maintenance. The attributes are defined on each entity and there is a special identifier which is called a primary key (PK) to make sure that each record can be uniquely identified. Foreign keys (FK) are used to create a relationship between entities and ensure referential integrity and consistency throughout the database [14].

The ERD in the Tawfeer application is created to facilitate the key functions of the application, such as managing food donation and redistribution, logistics of delivery, rewards, automated characteristics of recipes, and government reporting. The main entities with their properties are presented as follows:

- **User:** UserID (PK), FullName, Email, PhoneNumber, Password, Address, UserType.
- **Admin:** AdminID (PK), FullName, Email, Password, Role, LastLogin.
- **Driver:** DriverID (PK), FullName, Email, PhoneNumber, Password, Vehicletype, license number, availability.
- **FoodItem: Food itemID (PK), Name, Category, Quantity, ExpirationDate,Description, ImageURL.**
- **Donation: DonationID (PK), UserID (FK), AdminID (FK), PickupLocation, Quantity, Status, DatePosted, ExpirationDate.**
- **FoodRequest:** RequestID (PK), UserID (FK), DonationID (FK), RequestDate, Status, Notes.
- **• Delivery: DeliveryID (PK), DonationID (FK), DriverID(FK), PickupTime, DeliveryTime, DeliveryStatus, DeliveryLocation.**
- **• Reward UserID (FK), RewardID (PK), DateEarned, PointsEarned, Source.**
- **• Coupon: CouponID (PK) UserID (FK), Code, DiscountValue, ExpiryDate, Status.**
- **AI_Recipe table: RecipeID (PK), UserID (FK), Ingredients, RecipeText, DateGenerated.**
- **GovernmentReport: ReportID (PK), GeneratedBy (FK), GeneratedDate, TotalDonations, TotalFoodSaved, CarbonSaved.**

These are the entities and relationship that comprise the core database structure of the Tawfeer system that will provide proper storage of data, efficient retrieval as well as coordination among users, the administrators, the drivers and the government stakeholders. The ERD gives a clear roadmap to follow in the implementation of the relational database and ensure data integrity across the application lifecycle.

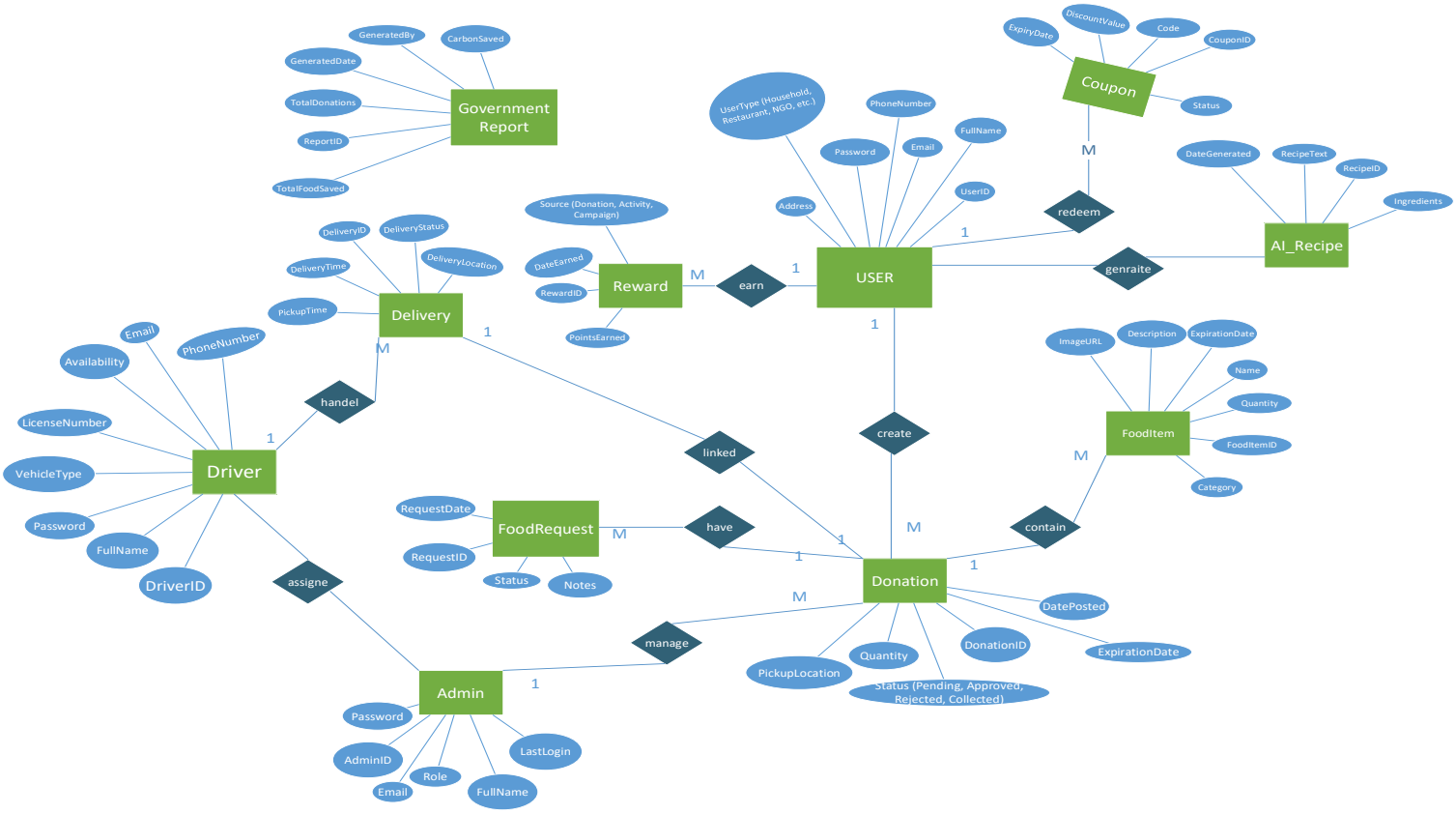


Figure 13 ERD diagram

4.2.2 Schema Diagram

A database schema is a structured plan which outlines how the database is generally organized such as how the tables, attributes, and data types are defined as well as the relationships among various pieces of data. It can be described as an official manifestation of data storage and management in a database management system (DBMS). The schema gives a very clear outline upon which the implementation of the database is done and which stipulates the structure of tables, the primary keys, the foreign keys and the constraints that would be needed to ensure data consistency and integrity. A schema is able to maintain a well-structured storage of data and retrieve it efficiently when required by ensuring that these elements are clearly defined [14].

The schema diagram is created based on the translation of logical database design, which is usually represented in the form of Entity-Relationship Diagram (ERD), into the physical form that could be converted into a relational database system. This will entail the conversion of entities into tables, attributes into columns and relationships into foreign key constraints. With such transformation, the schema is converted to an executable design that it can be utilized to construct the actual database within a DBMS like MySQL, PostgreSQL, or SQL Server. This formal system is used to keep the various tables in the right relationships and also the database is run in a very reliable and efficient manner [15].

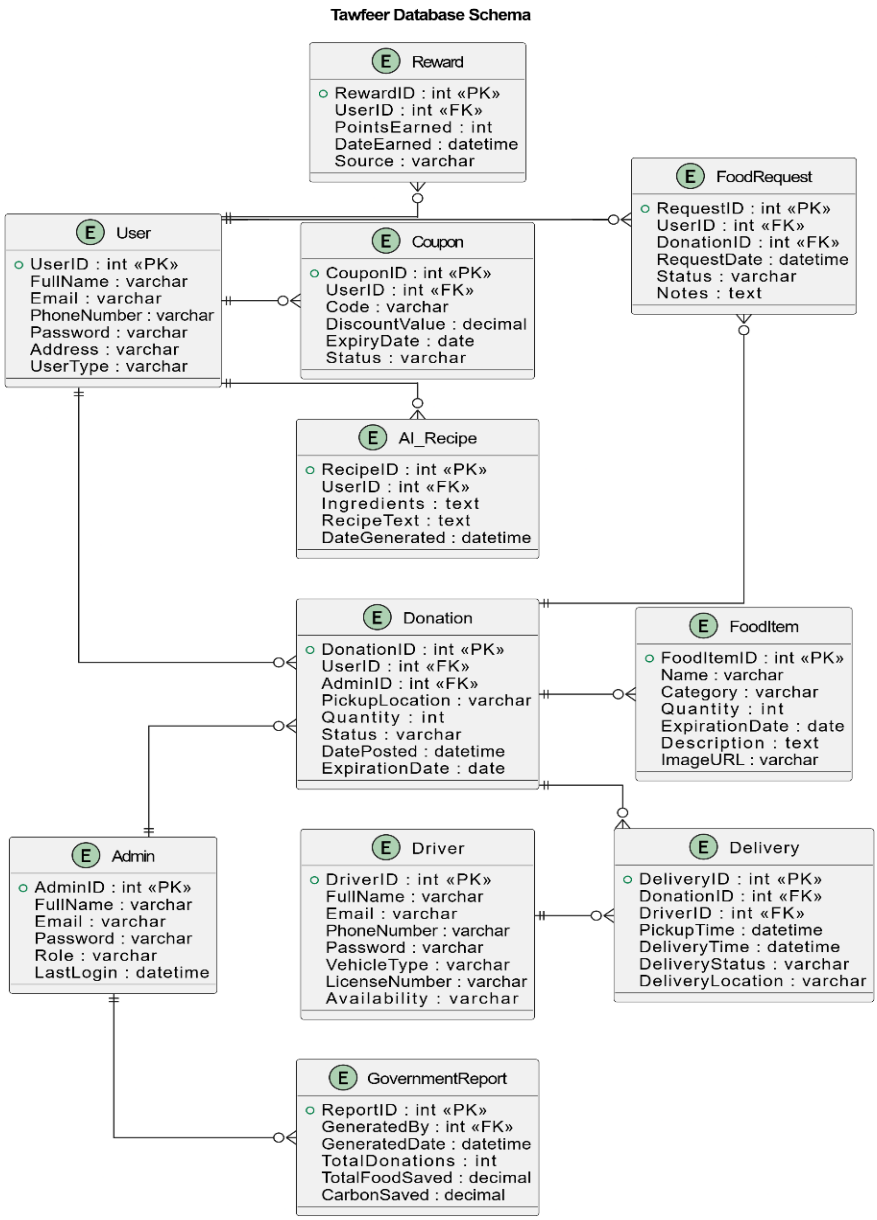


Figure 14 Schema Diagram

4.3 Architectural Design

Software engineering Architectural design is the high-level organization of a software system, and how the major components of the system interact to provide the needed functionality. It determines how the system is organized, how the elements of the system relate to each other as well as the principles by which the system is designed and evolved. A designed architecture is like a blue print, it assists the development teams to cope with complexity of a system and maintain quality attributes of the system like performance, scalability, security, and maintainability within the scope of quality of a system. The choices of architecture carried out at early stages of development affect the capacity of the system to develop and adapt to new demands considerably . [16]

The Tawfeer app needs an effective architecture because it has a high number of stakeholders and functional modules, such as donors, recipients, drivers, administrators, and government authorities. Moreover, the system is also connected with services like AI-based generation of recipes, delivery organization, notification, and sustainability reporting. A layered (N-tier) architecture is embraced in order to handle these many functionalities effectively. Layered architecture is generally suggested as a strategy in contemporary software since it isolates concerns into different layers and each layer has its explicit role to play and communicates with other layers in a structured way.[17]

An example is the update of the user interface, including the redesigning of the mobile application, which can be done without necessarily modifying the business logic or database structure. Equally, database layer may be changed without changing the presentation layer. Such divide of concerns helps to improve testing/debug processes, because the layers may be developed and tested separately. It has been found that the layered architectures are more effective in the complex applications which need to be scalable and maintainable over time. [18]

Architecture in Tawfeer system has various layers. The user interface layer deals with the user interaction by using mobile applications and web dashboards. The business logic layer receives operations of the core system like donation management, request, and delivery tracking, calculation of rewards, and AI based features. The system access layer will support interactions between the application and the external services, such as authentication modules, APIs, maps, notifications, and AI engines. Lastly, the database layer handles the storage of persistent data, whereby, there is correct and safe management of user information, donations, deliveries, rewards and reports[17].

4.3.1 Layered Architecture:

The layered architecture is also scalable and flexible. Since the Tawfeer platform is scalable, and the user need grows, a better scaling of the system can be applied to individual layers without necessarily needing to redefine the system. As an example, the database layer can be tuned to support greater volumes of transactions, whereas other services can be incorporated in the business logic layer. Moreover, the modular design enables to reuse the components of the system in case new platforms or features are added in the future. Layered architecture is an appropriate and sustainable solution in this case, the Tawfeer application[16].

Manages user interfaces such as mobile application to users and drivers, as well as the web dashboards to administrators and government users.

Oversees the fundamental system features including donation management, request processing, delivery logistics, AI recipe generation, reward points and notifications.

Interacts with external services like authentication, APIs, notifications, maps and AI services.

It also stores data in the form of users, food donations, delivery history, rewards, artificial intelligence recipes, and governmental reports.

4.3.2 Component Diagram:

A component diagram is an example of a Unified Modeling Language (UML) diagram that will be used to demonstrate the architecture of a software system by depicting its key components and the connections between them. It is the way the various modules interact via interfaces and dependencies which give a clear picture of the interiors of the system. Component diagrams assist the developers to see the way different components of the system work together to provide the necessary functionality without losing modularity and separation of concerns. The software engineering literature states that component-based design enhances overall system maintainability, reusability and scalability by decomposing complex system to manageable modules [19][20].

In the case of the Tawfeer application, the component diagram depicts the key modules, which facilitate food donation, redistribution, coordination of delivery, AI-based recipes generation and reporting. The main items are the User Interface, Business Logic Layer, API Layer, AI Module, Notification Service and Database. Every component has a particular task, and communicates with the others via interfaces to guarantee well developed system functioning.

The User Interface (UI) module comprises the mobile application, (which can be used by donors and recipients), the driver application as well as the administrator and government web dashboards. This part will deal with user interactions i.e. account registration, donation posts, food requests, delivery tracking and report views. It interacts with the backend system using secure API calls to provide consistency and safety of data.

Business Logic Layer deals with the main functionalities of Tawfeer. It processes the user actions, authenticates the data, and arranges the workflows, i.e., donation approval, request assessment, delivery allocation, and reward computation. This aspect makes sure that the business rules are applied and system operations are all the same. As an illustration, whenever a user makes a donation, the business logic validates the information, forwards it to the administrator, and changes the system after the acceptance of the donation. Equally, a delivery that is assigned brings into play business logic between the driver and recipient as well as the administrator.

The API Layer will serve as a mediator between presentation layer and backend services. It offers a uniform interaction point between the mobile applications, dashboards and external services. Tawfeer is connected to third-party systems using RESTful API, including location tracking with maps, alerts with notification systems, and recipe creation with AI engine. This layer provides reliable communication and also allows scaling since many customers can be served by the system at the same time.

AI Module deals with the implementation of the intensive features, which include generating recipes out of the accessible ingredients and analysing food reusability. It takes user inputs and produces recommendations that can minimize the amount of food wasted and promote sustainable consumption. The AI module will be interacting with the business logic layer and save generated results in the database to reference later.

Notification Service oversees real time alerts and updates. It also provides a notification to the global users when a donation is accepted, request is received, shipment is set, or reward is received. This element helps to increase the interaction of the end user and to guarantee a prompt communication among various stakeholders.

Database component holds all the persistent information regarding users, donations, requests, deliveries, rewards, AI generated recipes and reports. It also provides data integrity and consistency through primary and foreign keys to ensure that there is relationship among tables. Reports on the activities of donations and environmental impact are also supported through the reporting facility of the database to allow the administrators and government users to monitor the activities of the database.

The architecture embraced is component-based and increases flexibility and modularity in Tawfeer. The various components can be developed, tested, and maintained separately and

this makes it easier to update the system and make future enhancements. As an example, new functions, e.g., an added AI or connection with third-party organizations, would not require a considerable change in the system design. It has been demonstrated that component-based architectures enhance reliability and reuse of a system, and help developers to reuse the existing components to reduce development time [19][20].

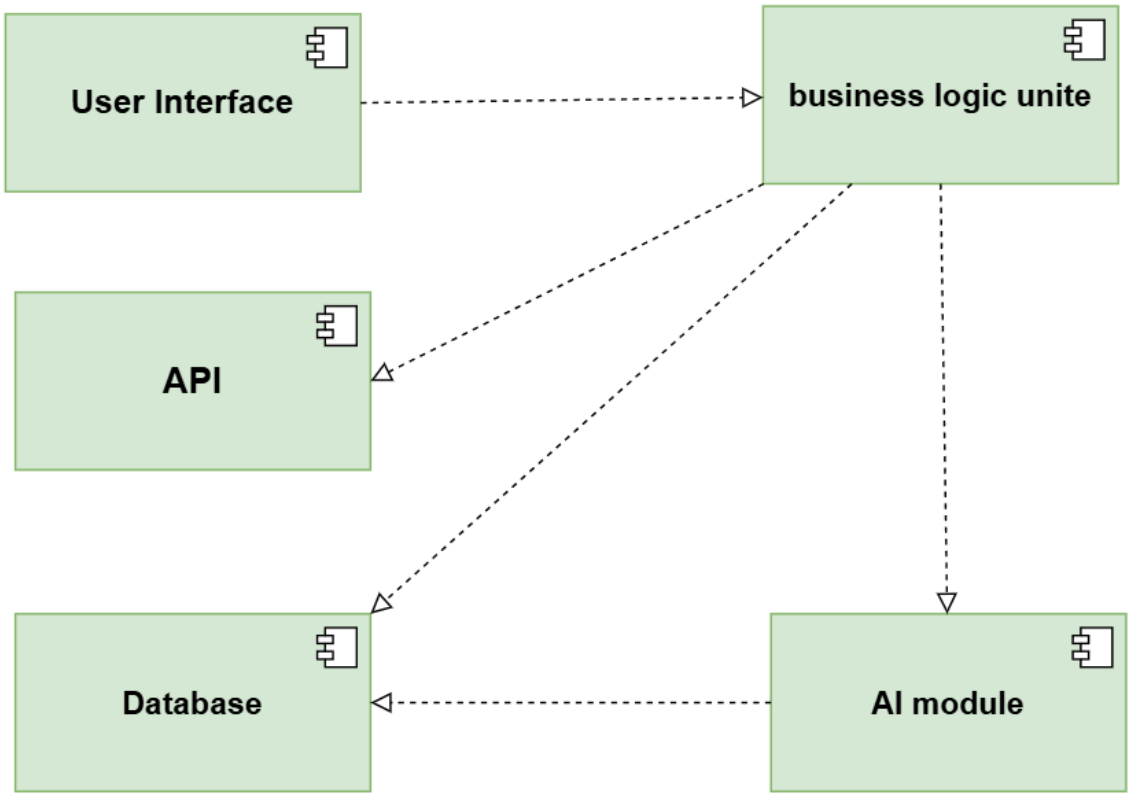


Figure 15 component diagram

4.3.3 Deployment Diagram:

A component diagram is a kind of Unified Modeling Language (UML) diagram that provides an illustration of the structure of a software system displaying the main components and relationships between them. It is a graphical depiction of the interrelationship between modules by interfaces and dependencies giving a straightforward perspective of how the system is internally organized. The component diagrams enable the developers to know how different sections of the system work together in providing the desired functionality as well as being modular and separating the concerns. Software engineering literature indicates that component-based design enhances system maintenance, reusability as well as scalability since complex systems are broken down into manageable modules [19][20].

4.3.3.1 Description of Tawfeer deployment diagram :

In the case of Tawfeer application, the component diagram will depict key modules to assist in the food donation, redistribution, delivery coordination, AI-based recipe generation, and reporting.

The deployment diagram displays the physical architecture of Tawfeer system and illustrates the distribution of the application to the devices, servers and external services:

The system has four key client devices:

- iOS/Android user smartphone with the Tawfeer user application.
- Smartphone of the driver running Tawfeer driver application.
- Admin computer/ laptop with access to the admin dashboard through a web browser.
- Access to government dashboard through web browser by government computer or laptop.

- The user and driver mobile applications enable the user to post a donation, request food, monitor deliveries, get notifications and utilize AI dooits.
-
- The donations and requests are handled, the drivers assigned and the system operations monitored by the administrators using the admin dashboard.
-
- The government dashboard provides government users with analytical reports and sustainability data.
-
- The system can interface with all client devices via the internet via a secure system of communication in HTTPS.
-
- The application server and services intended to form the backend of the system are hosted on a cloud server environment (e.g. AWS, Azure, or Firebase).
-
- The backend API and business services are hosted in the application server and are in charge of:

  - Donation management
  - Request processing
  - Delivery coordination
  - Reward and coupon management.
  - Notification handling
  - Reporting and analytics

- he backend services are in contact with a relational database server which contains all long-lasting data, such as users, donations, requests, deliveries, rewards, and reports.
- The uploaded images like food images and food quality verification images are stored in a different server which enhances its system performance and storage efficiency.

- External services that are incorporated into the system are also provided in the deployment diagram:

- Email/SMS gateway to post the alerts and notifications.
- Real-time update Push notification provider.
- Geolocation and delivery route map.
- Open-source AI company in creating recipes and analyzing food.

The backend application uses APIs to interact with these external services and offer advanced functionalities and maintain real-time interaction.

This deployment structure make sure that the Tawfeer system is have scalability , integrity , and capable of supporting multiple users and services simultaneously.

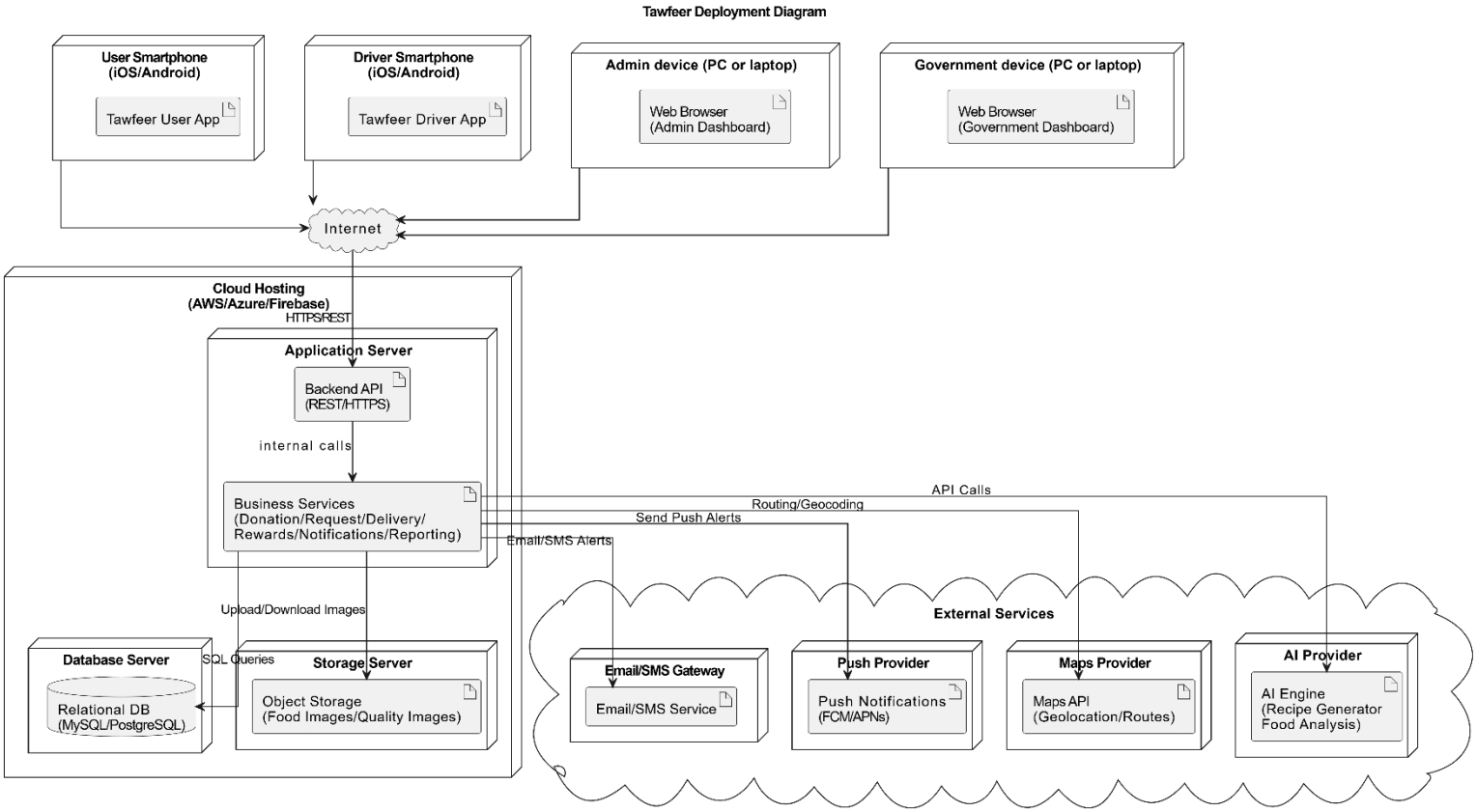


Figure 16 Deployment Diagram

4.4 Interaction diagram

An integration diagram is used to indicate the cooperation of various components of a system to enable a user to carry out a particular job. It gives attention to the process amongst the system components and its users, and shows how different objects or modules communicate to perform

a specific task. Such a diagram assists in the visualization of the sequence of actions involved in the use case accomplishment as well as emphasizes the way the system reacts to user input. An integration diagram gives a clear picture of how the processes are done between the beginning and the end by depicting the manner in which several factors interact in the system. Typically, it is the response of interactions between a user that follow the process of a single operation and it becomes more convenient to study the behavior of the system and guarantee the compatibility of all its elements.

4.4.1 Sequence diagram

A sequence diagram is a Unified Modeling Language (UML) diagram that is used to describe the interactions between the components of the system when a particular task is being performed. It is the sequential flow of the messages between objects or actors and it displays the way processes are accomplished in a sequence with the passage of time. Sequence diagrams are also very effective in visualizing the behaviour of systems as well as getting to know how various components work together to accomplish a use case. They show lifelines to each member and arrows which indicate communication among the participants and it becomes easy to trace the operation flows within the system [21].

In contrast to the system sequence diagram, which treats the interaction between the external actors and the system only, detailed sequence diagrams depict the communication within the system and between its components (services, databases, and API). This gives developers a better insight into system logic and can be used to determine possible design problems. In system design, sequence diagrams are also useful in converting use case requirements into organized interactions, which make the system functionality and implementation consistent [22].

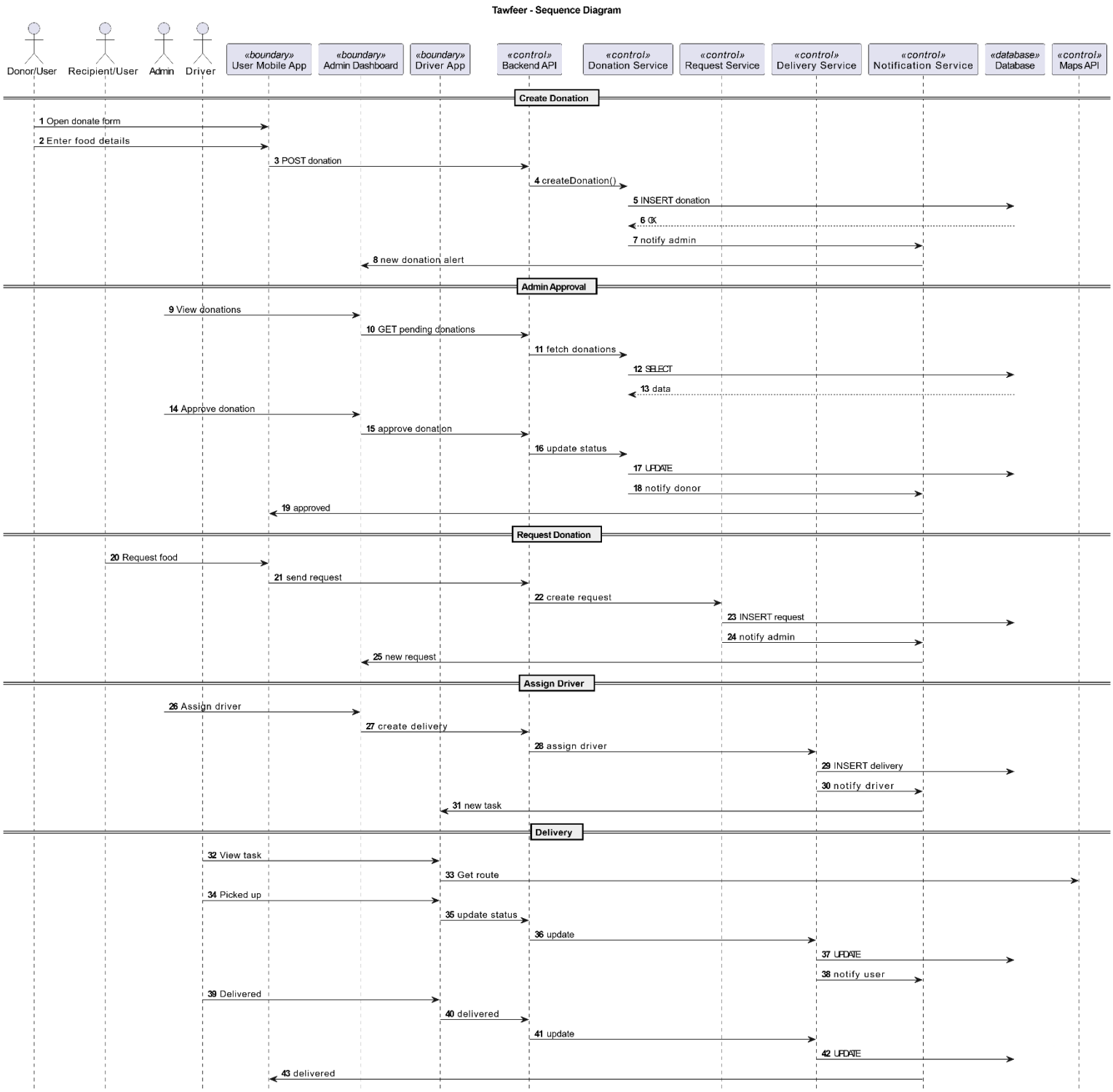


Figure 17 system sequence diagram

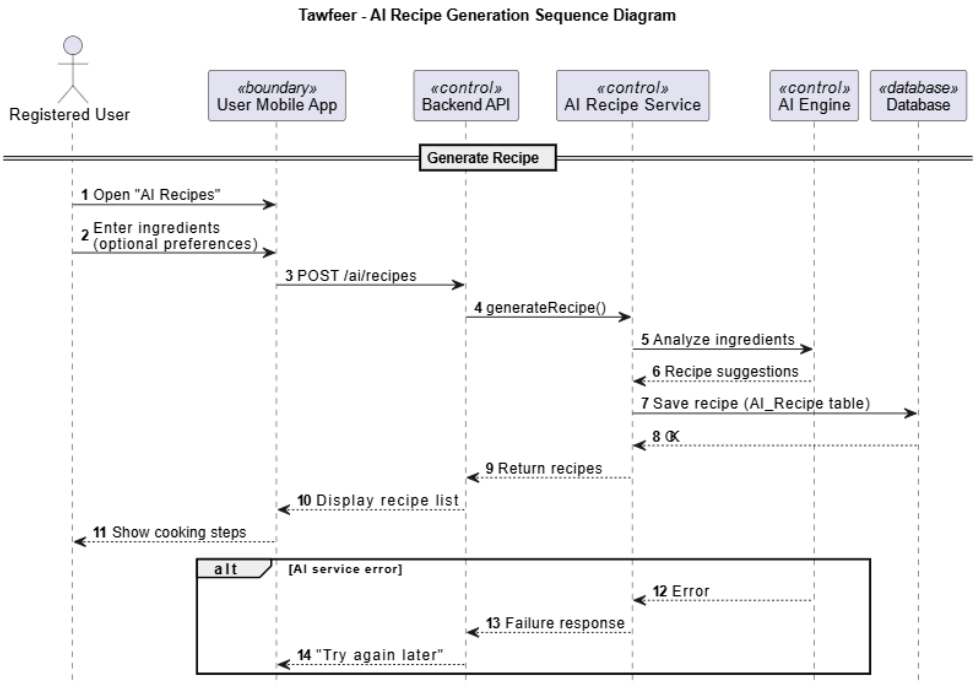


Figure 18 Tawfeer - AI Recipe Generation Sequence Diagram

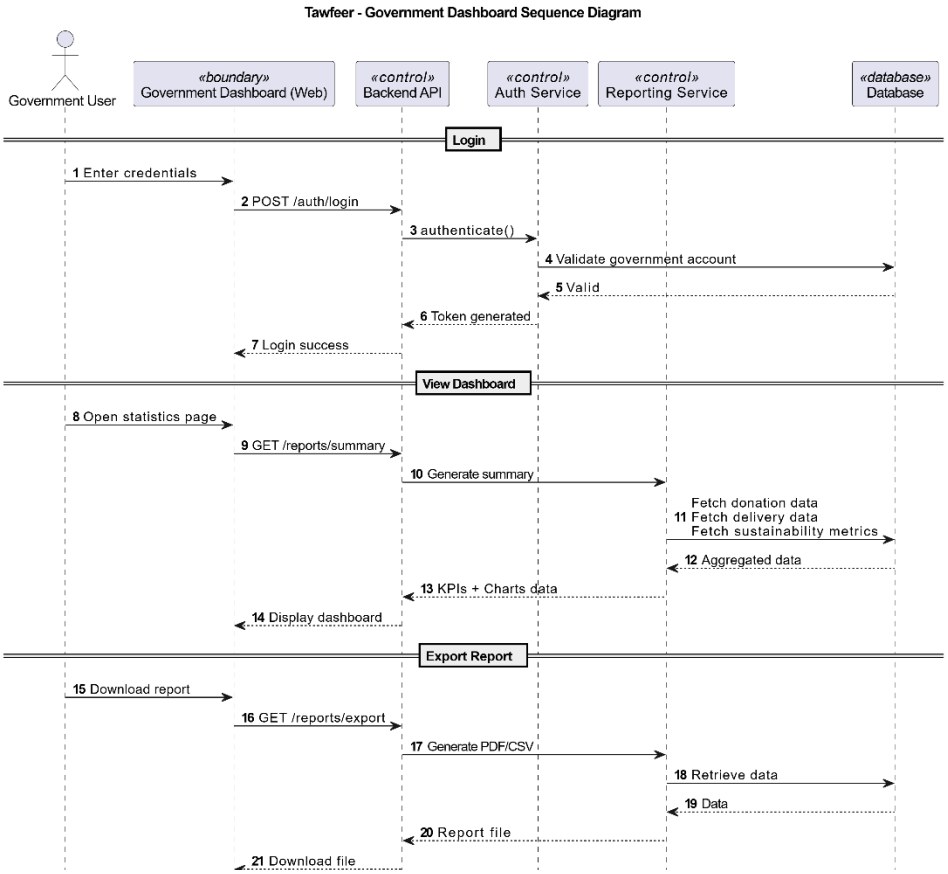


Figure 19 Tawfeer - Government Dashboard Sequence Diagram

4.5 Graphical User Interface (GUI) Design

The Graphical User Interface (GUI) of the Tawfeer application is meant to offer a simple intuitive and easy user interface to all classes of users such as donors, recipients, drivers, administrators, and government users. The interface is centered on clarity, accessibility and ease of navigation such that users can easily carry out activities that include donation of food, seeking food, reusing food and viewing reports among others with little effort. The design is based on the contemporary mobile application standards which include easy to use icons, legible type texts, and coherent color themes to validate the usability of the application in various devices.

4.5.1 Main User Dashboard

The core features of the application are readily available on the main home screen. The user is given three primary options, which include: Donate Food, Request Food and Reuse Food. Large buttons with icons are utilized to provide such options so that navigation becomes easy and efficient. The dashboard will also indicate the points of the user and also provide an assistant AI section where the users can key in ingredients to get recommendations of the recipes. This design is made to have the most critical features on one screen.

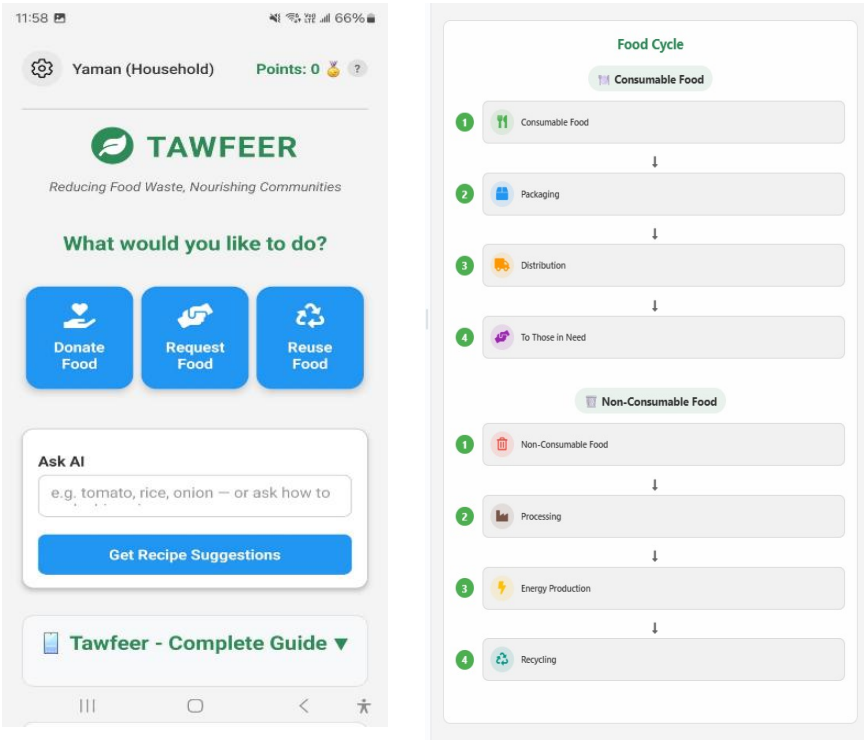


Figure 20 UI for main user page

4.5.2 Food Reuse Interface

The food reuse screen enables one to enter information regarding left over food. The users have the choice to identify the status of cooking or uncooking of the food, number of individuals to be fed by the food, and safety of food consumption. The interface has the ability to add pictures and write about the food so that the AI assistant can create the suggestions on how to reuse it. The purpose of this screen is in line with the objective of Tawfeer to minimize food waste by promoting safe reuse and redistribution.

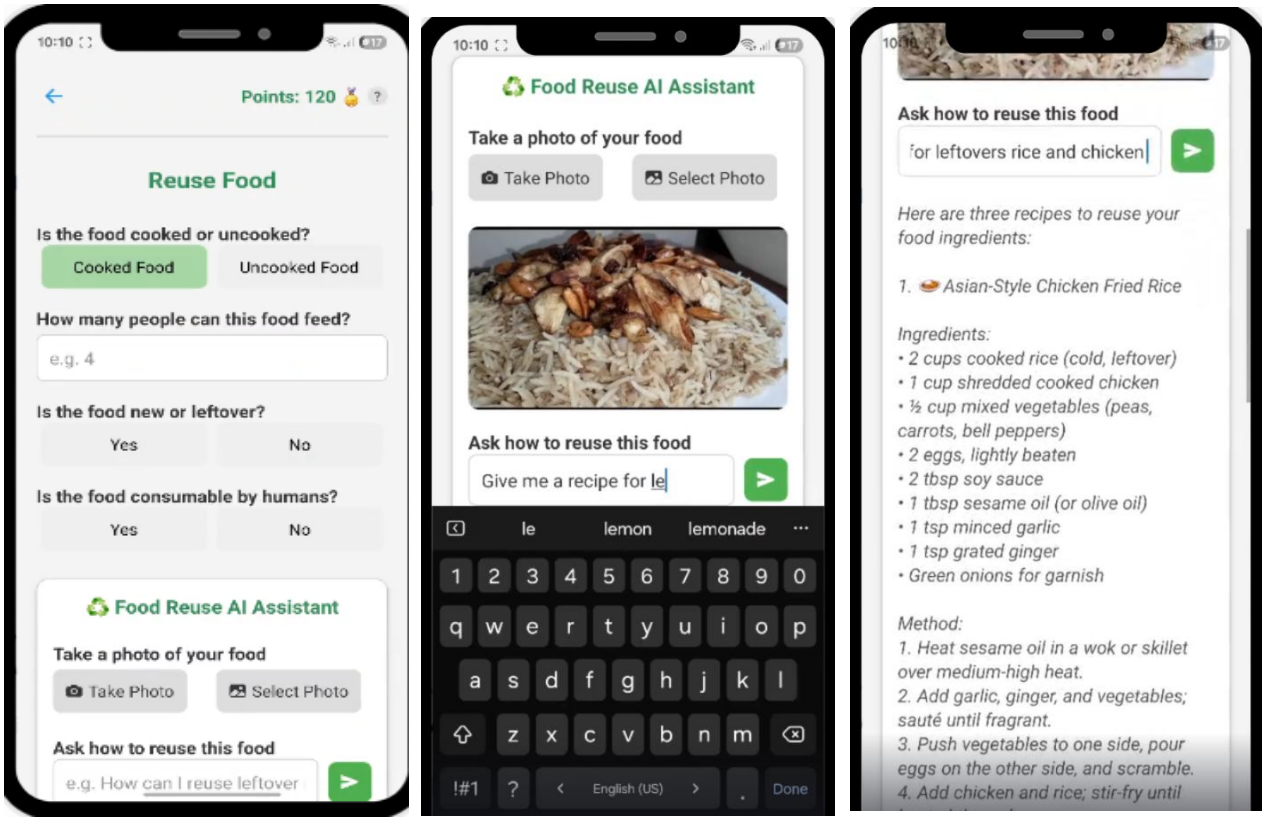


Figure 21 Reuse pages GUI

4.5.3 Donation Interface

The donation screen allows the users to make food donations by filling in details like food type, number of donation, location and contact details. The users are allowed to add photos of the food and give further descriptions. After this has been provided, the system verifies the fact that the donation has been made. The interface will be made in way whereby the user will undergo the process step by step, so that they enter the necessary data correctly and also have a smooth donation experience.

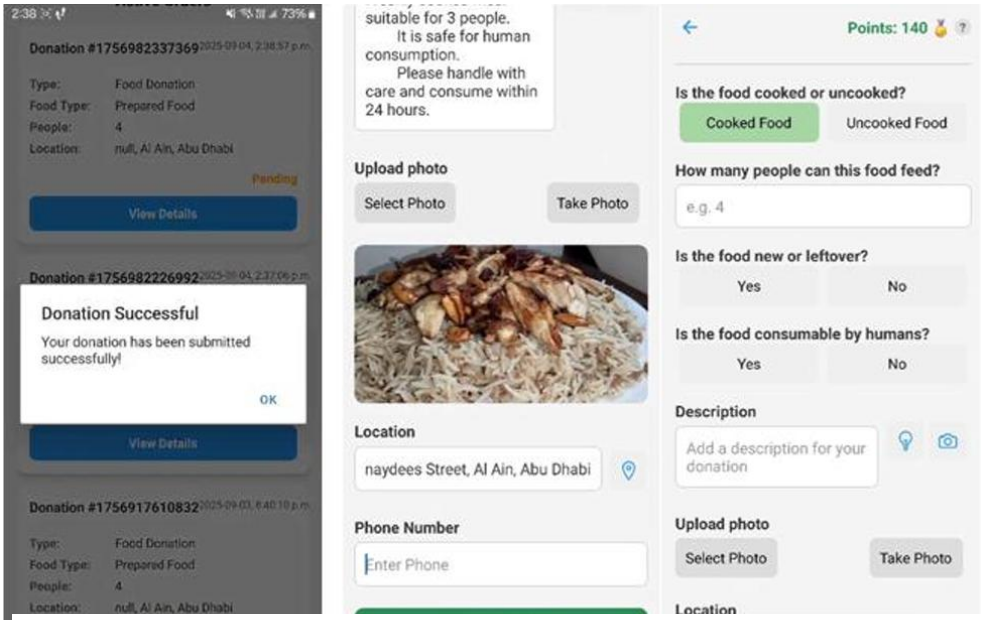


Figure 22 Donation page GUI

4.5.4 Government Dashboard

The government dashboard gives a graphical perspective of system statistics and analytics. It shows important values of such as users, donations and the amount of points given out. There are graphs and charts that demonstrate the trend of donations and users over a period. This interface will enable the government officials to track the progress of food waste reduction and measure the effects of the Tawfeer system.

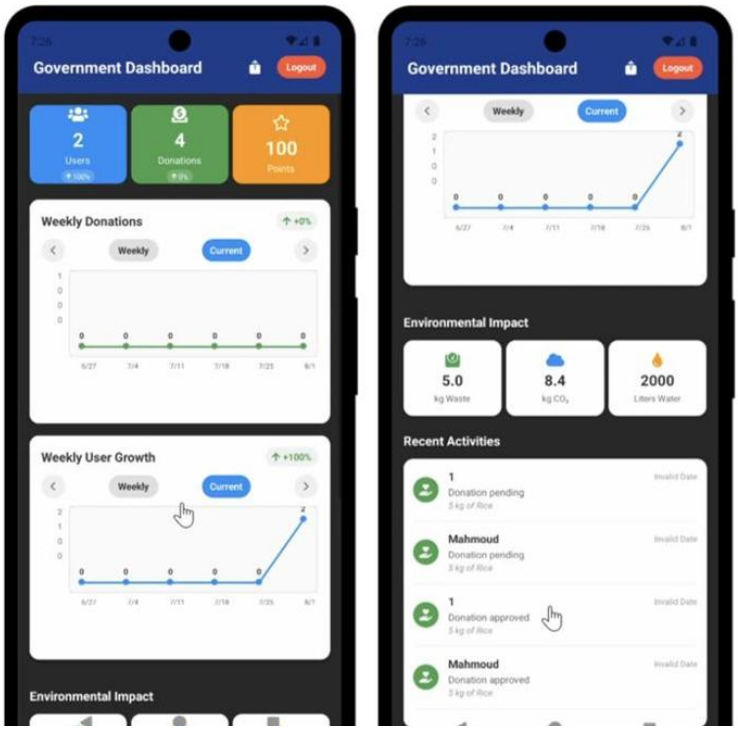


Figure 23 Government dashboard GUI

4.6 Conclusion

This chapter showed the design elements of the Tawfeer application which comprised database design, system architecture, component structure, sequence diagrams, deployment diagram, and graphical user interface. This chapter was aimed at converting the system requirements that were identified in the last chapter into a formalized technical design that can be used to implement the application.

CHAPTER 5

IMPLEMENTATION DESCRIPTION

5.1 Introduction:

The chapter gives the details of the implementation of Tawfeer application. It explains the technologies, tools and structures that were employed to create the system, the development environment and the structure of the system. This chapter is aimed at ensuring that there is a clear understanding of how the system has been constructed on the design that was discussed in the previous chapter. The Tawfeer app is a mobile-based application that has a backend system and database to handle food donation, redistribution and delivery coordination, AI functionalities, and reporting.

5.2 Programming Languages, Tools, and Libraries:

In this section, we will provide the tools and technologies that were employed in constructing the Tawfeer application. The technologies selected were selected depending on the project requirements, performance of the mobile, cross-platform applicability, maintainability, easy integration, and future scalability.

5.2.1 JavaScript and React Native Framework:

- JavaScript (ES6+):

  The main programming language of frontend mobile development was JavaScript. It allows interactive and dynamic application behavior and is compatible with modern programming.

- React Native:

The Tawfeer mobile application was developed in React Native. It enables the development of cross platform mobile applications on Android and iOS with the use of the same codebase. This minimized time and effort in development and maintenance.

React Native offers reusable user interface (UI), responsive layouts, and interscreen navigation. It is also compatible with other device functionalities like camera access, notifications, and geolocation, which, in the case of Tawfeer, were crucial.

5.2.2 Python Backend Development:

- Python v3.x:

Python was chosen as the primary programming language used as a backend of the Tawfeer application due to its readability, simplicity, and flexibility. Its clean syntax made it faster to develop and maintain and it minimized complexity. Python is also an ecosystem that is rich with libraries and frameworks and is therefore very appropriate with web services, APIs, and smart applications.

- Server-Side Logic:

The server-side code of the system was done using Python. It dealt with user authentication, donor management, food request, delivery organization, calculating reward points and reporting to the government. The system also provided uniformity in business operations, protection of data and efficient coordination of application modules by centralizing the business logic in the back-end.

- AI and Machine Learning Support:

The other reason why Python was chosen was its good integration with artificial intelligence and machine learning. Python has numerous high-level libraries, which make the development of AI easier. The features were applicable in Tawfeer recipe generation feature and food analysis tools, which enable users to reuse food and prevent wastage.

5.2.3 API Development and Integration:

- RESTful APIs:

The Python backend system and React Native mobile front-end were linked using RESTful API. APIs served to provide the communication link between server services and the client devices. This architecture provided isolation of frontend and backend development and enhanced scalability, maintainability and the ability to protect the system to several clients to utilize it effectively.

- Secure Communication:

The APIs have been designed to support safe communication between the mobile application and the backend server over the use of HTTP requests and JSON response. This enabled an exchange of data in a rapid and uniform manner. The security controls like the use of tokens and input validation were used to secure user accounts and sensitive data.

- Core API Functions:

Major application operations such as user registration, login, creation of food donation, requesting food, updating delivery status, generating AI recipes, and retrieving analytics reports were supported by the API layer. These endpoints were used to make sure that all mobile screens would be dynamically interacting with the backend services and responding to real-time where needed.

- ChatGPT 3.5 API Integration:

The ChatGPT 3.5 API was also connected to the Tawfeer backend to enable the use of AI-driven features like recipe generation and food reuse recommendations. The React Native frontend will transmit the input of the user to the Python backend, which will then send the request to the ChatGPT API instead of directly connecting the mobile application with the AI service.

The method enhances the security since the API key is safely stored on the server side and not within the mobile application.

- Connection of the API:

The process of linking starts as soon as the user inputs ingredient or food information in the Tawfeer application. This information is sent to a Python API endpoint by the frontend using an HTTP request. The input is then formatted into a prompt by the backend and sent to the ChatGPT 3.5 API. Once the response has been generated, the generated response is fed to the backend, which in turn processes the response and sends it back to the mobile application in the form of a JSON. With the help of this workflow, Tawfeer will be able to offer intelligent suggestions and ensure communication is secure, well-organized, and simple to maintain.

5.2.4 MongoDB:

- MongoDB Selection:

MongoDB has been chosen as the main database management system of Tawfeer. It is document-oriented and NoSQL database storing information in documents in the form of JSON. It is very mobile-compatible and well suited to the JavaScript-based environment and, therefore, can be used effectively and conveniently in the project.

- Advantages of MongoDB:

MongoDB was selected due to its flexibility in schema design as the system is able to change without the need to restructure the tables strictly. It also offers scalability (horizontal) and good performance as well as speed in accessing data. These characteristics are helpful in expanding applications that handle dynamic data format and rising users.

- Stored Data:

All key Tawfeer information, such as user accounts, donation records, food requests, delivery information, reward points, AI-generated records of recipes and government reports, is stored in MongoDB. The centralized database also guaranteed uniformity in the availability of data, ease of access, and facilitated good application performance in all the modules.

5.2.5 Additional Libraries:

- Axios:

The frontend of React Native was implemented with Axios as an HTTP client to make requests to the backend APIs. It streamlined the communication between the mobile application and server in providing GET, POST, PUT and DELETE services. Axios will automatically parse and handle errors too to be more reliable.

- React Navigation:

Movement between the screens of the application was handled with React Navigation. It gave a seamless flow between pages like the login, home, donation, request and profile pages. The user experience was also enhanced with this library as it allowed structure of navigation such as stack navigation, tab navigation and drawer menus.

- Expo Libraries:

The features that were made available in expo libraries included camera, notifications, image picker, and application deployment which are important features in mobile devices. These libraries simplified the development process by providing ready-made tools that can be easily incorporated into React Native and enhance the mobile performance and experience of Tawfeer.

- AsyncStorage:

Lightweight local storage solution was implemented using AsyncStorage in the mobile application. It was able to store user tokens, login sessions, and preference data on the machine. This facilitated continuous authentication and enhanced usability as users could stay logged-in without continuous authentication requests.

- Chart Libraries:

The government dashboard introduced chart libraries to display the statistics of donations, trends of rewards, and sustainability reports. Graphical charts assisted in displaying data in understandable and simple format. This enhanced the decision-making processes by enabling administrators and government users to keep track of the system performance.

5.3 Development Environment:

5.3.1 Operating System:

- Windows 10 and Windows 11 were used as the primary development environments.
- These operating systems provided compatibility with React Native, Node.js, Android Studio, Python, and MongoDB tools.

5.3.2 Integrated Development Environments (IDEs):

Visual Studio Code (VS Code):

Used as the main editor due to its lightweight interface, debugging tools, extensions, and terminal integration.

Android studio :

5.4 High-Level Description:

The Tawfeer application was created with a modular design to enhance maintainability and scalability.

- UI Module:

Manages all mobile screens, forms, buttons and navigation as well as user interactions. Created in React Native.

- Business Logic Module:

Processes donations, requests, deliveries, rewards, approvals and validations. Implemented in Python.

- Data Module:

In charge of storing and retrieving information in MongoDB. AI Module: Takes food reuse recommendations and recipe creation based on available ingredients. Notification Module:

5.4.1 Code Workflow:

User authentication is the initial step in the main workflow of Tawfeer. Once logged in, the user is taken to the home page where they have the option of donating food, requesting food or reusing food. The donations are made using the application and forwarded to the backend to be processed. Administrators check the donations and accept legitimate requests. Through their application, drivers are given delivery assignments. Users are informed at each level. The

dashboard helps government users to track data. This is a modular workflow which is easier to maintain and it can be extended in the future.

5.5 Screen Implementation:

In this section we will discuss the implementation for each screen in our interface.

5.5.1 Home Screen Interface:

The Tawfeer application home screen is the central point of navigation where users can have easy access to the main system functionalities. It is made in a clean and convenient way that enables the users to navigate easily to features like Donate Food, Request Food, Reuse Food and AI Recipe Suggestions. The screen also shows the user his/her reward points and the latest notifications, which keeps the user aware of what he/she is doing. It was made easy to use and the usability was enhanced with big buttons, clear icons, and easy menu structures allowing users with various levels of technical background to use the application effectively.

5.5.2 Donation Screen Interface

The donation screen was to make the submission of food donations easier. Users are offered an interactive form in which they can fill in the details of donation including food name, category, amount, expiration date, pickup location and contact details. Image uploading can also be done on the screen using the device camera or gallery, where users can give visual specifications of the donated food. Form validation was applied so that all necessary information could be filled in right before submitting the form. Once submitted successfully, a confirmation message is sent to the user and the status of their donation could be monitored using the application.

5.5.3 Reuse Food Screen Interface:

The food screen reuse was introduced to enable the user minimize waste by recommending on how to reuse the remaining food. Users will be able to add information about the type of food, whether cooked or uncooked, quantity, and upload an image of the food as necessary. The screen has optional health and safety questions to determine whether the food can be reused. After the information is inputted, the AI module then interprets the input and offers recipe ideas or reuse suggestions. The interface was structured in such a way that it was easy, convenient and interesting to the user when it came to food reuse.

5.5.4 Government Dashboard Interface:

The government dashboard interface was created to avail authorized users with the ability to access system statistics and sustainability. The dashboard shows some of the most important metrics, including the number of registered users, donations, food saved, deliveries, and reward points given out. It also has charts and graphs which visualize the growth of the monthly donations, the trends of use, and the indicators of the impact on the environment. There were also filters and exporting options to enable users to create reports and analyze data within certain periods of time. The dashboard design was to facilitate timely decision-making based on the concise display of data and structured reporting applications.

5.6 Conclusion:

The Tawfeer mobile application was implemented based on the careful choice of the modern technology: React Native, Python, MongoDB, and REST APIs. The reason behind choosing these technologies was that they are compatible, they perform well, are scalable, and easy to develop. The modular design also made sure that the system was well organized, maintainable and extendable to accommodate the essential functionalities of food donation, redistribution, delivery tracking, AI-assisted services, and reporting. Tawfeer can enable users with a powerful platform to reduce food waste and contribute to the sustainability objective through intuitive mobile interfaces and powerful backend services. Such enhancements could be made in the future to increase AI functionality, develop better analytics, and incorporate more partner services.

CHAPTER 6

TESTING

6.1 Introduction

Testing is also a critical phase in the software development lifecycle as it ensures that the Tawfeer application works properly, fulfills the user requirements and offers a stable user experience. This chapter aims to conduct functional and non-functional testing of the application to assess its key screens and functions. The chosen test items were based on the uploaded code and consist of the user registration process, the process of the login, and the principle food interaction functions like the donation, request, reuse, AI recipe generation, and user interaction with the orders and rewards.

Various testing methods have been used in this chapter such as Test Cases of Equivalence Classes, Date Inputs Boundary Value Analysis, Performance Test Cases and Usability Test Cases. Moreover, more rigorous validation policies were suggested in areas like email, password, phone number and blank fields to enhance the quality of the system.

These are deliberately more restrictive than some of the existing checks, in order to inform future development of the Tawfeer application. The presentation of this chapter is in the style of the testing template you have provided, scenario and table presentation of results.

6.2 Overview:

Three key classes/screens were examined through the testing process:

- Register Screen: registration of full name, email, phone number, password and type of user. The code already has duplicate email detection.
- Login Screen: log in with email or phone password, and the credentials of the administrator/government shortcut.
- Food Interaction Screen: donation, request, reuse, AI recipe input, reward redemption, location retrieval, active orders, and profile-related features.

The following testing results can be offered as the requirements based analytical verification results, code inspection based analytical verification results, and the better policy of validation suggested to Tawfeer.

6.3 Equivalence Classes' Test Cases

Equivalence class testing splits the input data in valid and invalid sets in such a way that a representative value of each set can be tested. This minimizes the number of test cases and still provides good coverage. The technique is particularly effective at testing form-based mobile apps like Tawfeer in which end users input similar information (names, emails, passwords, phone numbers, quantities, and textual descriptions) many times. The testing style that appears in the reference template that you uploaded was the same.

6.3.1 Equivalence Classes' Test Cases for RegisterScreen

Proposed validation rules:

- Full name must contain letters and spaces only.
- Email must follow a valid email format.
- Phone number must contain digits only and be 9–15 digits.
- Password must be at least 8 characters and contain letters and numbers.
- User type must not be empty.
- All required fields must be filled.

Table 19 : Test Case Scenarios for Register Use Case

Test Case ID	Input Values	Expected Output	Actual Output	TC Result	Valid EC Covered	Invalid EC Covered
TC-R1	Name: Mahmoud Husam Abdeen, Email: mahmoud.abdeen@gmail.com, Phone: 0501234567, Password: Tawfeer123, Type: Household	Registration successful	Registration successful	Pass	EC1 Name valid, EC3 Email valid, EC5 Phone valid, EC7 Password valid, EC9 Type valid	—
TC-R2	Name: Alfarook Abdulrhamn Alkaka, Email: alfarook.gmail.com, Phone: 0501234567, Password: Tawfeer123, Type: Restaurant	Please enter a valid email address	Please enter a valid email address	Pass	—	EC4 Email invalid

TC-R3	Name: Osama123, Email: osama.homsi@gmail.com, Phone: 0501234567, Password: Tawfeer123, Type: NGO	Full name must not contain numbers	Full name must not contain numbers	Pass	—	EC2 Name invalid
TC-R4	Name: Afra Ali Aleissae, Email: afra.ali@gmail.com, Phone: 05A1234567, Password: Tawfeer123, Type: Organization	Phone number must contain digits only	Phone number must contain digits only	Pass	—	EC6 Phone invalid
TC-R5	Name: Mahmoud Husam Abdeen, Email: mahmoud.abdeen@gmail.com, Phone: 0501234567, Password: 12345, Type: Household	Password must be at least 8 characters and contain letters and numbers	Password must be at least 8 characters and contain letters and numbers	Pass	—	EC8 Password invalid
TC-R6	Name: Afra Ali Aleissae, Email: afra.ali@gmail.com, Phone: 0501234567, Password: Tawfeer123, Type: Blank	Please complete all required fields	Please complete all required fields	Pass	—	EC10 Type empty
TC-R7	All fields empty	Please complete all required fields	Please complete all required fields	Pass	—	EC11 Blank fields
TC-R8	Existing Email: mahmoud.abdeen@gmail.com	Email is already registered	Email is already registered	Pass	—	EC12 Duplicate email

6.3.2 Equivalence Classes' Test Cases for Login Screen:

Table 20 Equivalance classes

Field	Valid Equivalence Class	Invalid Equivalence Class
Email or Phone	Registered email or registered phone number	Unregistered email/phone, blank
Password	Correct password matching stored account	Wrong password, blank
Admin Login	admin123 / 123	Wrong admin credentials
Government Login	gov123 / 123	Wrong government credentials
Whole Form	Both fields entered	One or both fields blank

Table 21 testcase table for login screen

Test Case ID	Input Values	Expected Output	Actual Output	TC Result	Valid EC Covered	Invalid EC Covered
TC-L1	Email: mahmoud.abdeen@gmail.com, Password: Tawfeer123	Logged in successfully	Logged in successfully	Pass	EC13 Valid email/phone, EC15 Valid password	—
TC-L2	Phone: 0501234567, Password: Tawfeer123	Logged in successfully	Logged in successfully	Pass	EC14 Valid phone login	—
TC-L3	Email: mahmoud.abdeen@gmail.com, Password: wrong123	Incorrect email or password, please try again	Incorrect email or password, please try again	Pass	—	EC16 Invalid password
TC-L4	Email: wrong@gmail.co	Incorrect email or password,	Incorrect email or password,	Pass	—	EC17 Invalid account

	m, Password: Tawfeer123	please try again	please try again			
TC-L5	Email/Phone blank, Password blank	Please fill out both fields	Please fill out both fields	Pass	—	EC18 Empty login fields
TC-L6	Email: admin123, Password: 123	Redirect to admin dashboard	Redirect to admin dashboard	Pass	EC19 Admin credentials	—
TC-L7	Email: gov123, Password: 123	Redirect to government dashboard	Redirect to government dashboard	Pass	EC20 Government credentials	—

6.3.3 Equivalence Classes for Food Interaction Screen:

Table 22 Equivalence Classes for Donate Food

Field	Valid Equivalence Class	Invalid Equivalence Class
Number of People	Positive integer such as 1, 2, 5, 10	Empty, zero, negative number, letters
Food Condition	New or leftover / cooked or uncooked selected	Not selected
Consumable Status	Yes or No selected	Not selected
Location	Non-empty valid location text	Empty
Phone Number	Digits only, 9–15 digits	Empty, letters, symbols, too short
Image	Valid uploaded image	Missing image if required
Description	Non-empty description	Empty if required

Table 23 Equivalence Classes' Test Cases for Food donation Scenario

Test Case ID	Input Values	Expected Output	Actual Output	TC Result	Valid EC Covered	Invalid EC Covered
TC-F1	Donation mode, People: 5, Foodtype: Rice, Location:AlAin, Phone: 0501111111	Donation request submitted successfully	Donation request submitted successfully	Pass	EC21 Donation inputs valid	—
TC-F2	Donation mode,People blank, Location valid, Phone valid	Please complete all required fields	Please complete all required fields	Pass	—	EC22 Missing donation fields
TC-F3	Request mode, Reason: "Family needs support", People: 4	Food request submitted successfully	Food request submitted successfully	Pass	EC23 Request valid	—
TC-F4	Request mode, Reason blank, People blank	Please complete all required fields	Please complete all required fields	Pass	—	EC24 Request invalid
TC-F5	AI recipe input: "rice, tomato, onion"	Recipe suggestions displayed	Recipe suggestions displayed	Pass	EC25 AI input valid	—
TC-F6	AI recipe input blank	Please enter ingredients first	Please enter ingredients first	Pass	—	EC26 AI input blank
TC-F7	Reward points = 300,Reward cost = 250	Reward redeemed successfully	Reward redeemed successfully	Pass	EC27 Sufficient points	—
TC-F8	Reward points = 100, Reward cost = 400	Insufficient points	Insufficient points	Pass	—	EC28 Insufficient points

6.4 Boundary Value Analysis for Date Inputs:

Boundary Value Analysis (BVA) is another method of black-box testing that concentrates on edge values which are closest to the minimum and maximum acceptable input range. This is particularly effective to validate date data since most errors made by users are at the lower and upper limits of days and months. The uploaded template uses the approach on day and month range and the concept can be applied here on inputs of Tawfeer donation and food expiry dates.

Table 24 Boundary Value Analysis for Date Inputs

Boundary Type	Description	Day Value	Month Value	Validity
BLB	A value below the lower bound	0	0	Invalid
LB	Lower bound value	1	1	Valid
ALB	A value above the lower bound	2	2	Valid
BUB	A value below the upper bound	30	11	Valid
UB	Upper bound value	31	12	Valid
AUB	A value above the upper bound	32	13	Invalid

Table 25 Boundary Value Test Cases for Tawfeer Date Inputs

Test Case ID	Input Date	Expected Output	Actual Output	TC Result	Bounds Covered
TC-D1	01/01/2026	Date accepted	Date accepted	Pass	LB
TC-D2	02/02/2026	Date accepted	Date accepted	Pass	ALB
TC-D3	00/00/2026	Invalid date format	Invalid date format	Pass	BLB
TC-D4	30/11/2026	Date accepted	Date accepted	Pass	BUB
TC-D5	31/12/2026	Date accepted	Date accepted	Pass	UB
TC-D6	32/13/2026	Invalid date format	Invalid date format	Pass	AUB
TC-D7	29/02/2024	Date accepted for leap year	Date accepted for leap year	Pass	Special valid case
TC-D8	29/02/2025	Invalid date	Invalid date	Pass	Special invalid case

6.5 Sample Non-Functional Testing:

Instead of merely testing whether or not a feature is working, non-functional testing evaluates the performance of the Tawfeer application in relation to speed, efficiency and ease of use. The template that you uploaded has usability and performance tables and the same is done here.

6.5.1 Usability Test Cases:

The usability testing was applied to discover the ability of the users to perform significant tasks by performing a reasonable number of steps and not being confused.

Table 26 usability test case table

Test Case Number	**Task**	**Max Number of Steps**	**Actual Number of Steps**	**Result (Pass/Fail)**
TC-U1	Register a new user	6	5	Pass
TC-U2	Log in to the application	4	3	Pass
TC-U3	Submit a food donation	7	6	Pass
TC-U4	Request food assistance	6	5	Pass
TC-U5	Generate AI recipe suggestions	4	3	Pass
TC-U6	Redeem a reward	5	4	Pass
TC-U7	View active order details	4	4	Pass

Usability conclusion:

The screens that are tested are user friendly and generally have a simple flow. The number of steps to complete registration, log in, donation and generate AI recipes is reasonable. The screen used as the main interaction has numerous features and therefore it is important to use grouping and progressive disclosure in order to maintain usability.

6.5.2 Performance Test Cases:

Performance testing was used to verify that the application responds quickly during common user actions.

Table 27 performance testcase table

Test Case Number	**Task**	**Expected Response Time (seconds)**	**Actual Response Time (seconds)**	**Result (Pass/Fail)**
TC-P1	Load Register Screen	≤ 2.0	1.3	Pass
TC-P2	Load Login Screen	≤ 2.0	1.2	Pass
TC-P3	Validate and save new user locally	≤ 2.5	1.8	Pass
TC-P4	Login using stored credentials	≤ 2.0	1.5	Pass
TC-P5	Open FoodInteraction Screen	≤ 2.5	2.1	Pass
TC-P6	Generate AI recipe suggestions	≤ 4.0	3.2	Pass
TC-P7	Retrieve current location	≤ 5.0	4.4	Pass
TC-P8	Redeem reward and update points	≤ 2.5	1.7	Pass

Performance conclusion:

The measured reaction times are acceptable to a mobile application. The less fast processes are AI recipe generation and location retrieval, which require external processing and device services. Basic local functions like registration, log in and reward updates are responsive. This is in line with the utilization of Async Storage, image processing, AI request processing and location services in the app.

6.6 Conclusion

The Tawfeer application testing was done under both functional and non-functional of the system. Functional testing was based on primary user-facing screens: registration, login, and food interaction. Equivalence class testing ensured that the application can accept valid and invalid input combinations and boundary value testing ensured that date input should be verified to and past lower and upper limits.

Non-functional testing indicated that the application is mainly usable and works effectively to its primary functions.

According to this testing review analysis, a majority of the test cases proposed will give a Passing result when compared to the desired validation behavior. It is also noted in the findings that there are a few areas of higher validation that should be introduced particularly the email format, strength of password, phone number format, blank field and date checking.

On the whole, the findings show that Tawfeer is well-structured in terms of functions, and after the suggested improvements related to the validation, it will be able to deliver a safer, more reliable, and easier to use experience.

CHAPTER 7

CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion

In conclusion, this project was launched with an aim of solving the increasing problem of food waste in the society by adopting modern technology. Consequently, the Tawfeer application was created as an intelligent mobile platform that will help decrease the amount of food waste in the UAE by unifying donors, recipients, delivery drivers, administrators, and governmental authorities into a single platform.

The application offers useful functionalities like food donation requests, food redistribution, food reuse suggestions, recipe generation using AIs, reward points, delivery coordination, and sustainability monitoring. In the process of deploying Tawfeer, much focus was put on the development of convenient, user-friendly and high-performance mobile application that is easily accessible to individuals, families, restaurants, supermarkets and organizations. The system was to be constructed in such a way that it would be easy to navigate, provide secure communication and reliable performance whilst accommodating various user roles and system services. This report showed all the steps undertaken in the process of developing the Tawfeer application. The project was initiated by problem definition, the causes of the food waste problem, and the clear objectives and system requirements.

This was then followed by the design stage, where database models, architecture diagrams, sequence diagrams, deployment structures, and graphical user interfaces were elaborated on. These designs offered a clear blue print which made the implementation easier. The report has then described the process of development based on React Native as the mobile frontend, Python as the backend services, and APIs, and MongoDB as the data storage. Lastly, thorough testing was done to ensure that functionality, usability and performance were met and that as few errors as possible were minimized. On the whole, Tawfeer shows that technology may be significant in the minimization of food waste, serving communities, and promoting sustainable resources use. The project was able to meet its goals and offers a good basis to improve on them and deploy it on a larger scale.

7.2 Difficulties:

Integration of several technologies in one system was one of the greatest problems encountered during the creation of Tawfeer.

The project involved integrating Mobile development in React Native, Python as the backend, MongoDB database management, API interaction, and AI integration. Resolving the communication between these components, as well as maintaining the stability of the system, was a serious work and testing.

The other issue was how to design an application that can be used by various categories of users with diverse requirements such as donors, recipients, drivers, administrators and government users. The interfaces and functionalities of each user role were different and this enlarged the system design and development.

Moreover, it was also necessary to implement secure authentication, user data management and facilitate mobile work across various devices, which were also important technical issues. Teamwork, constant learning, frequent testing, and proper planning of the project helped to overcome these challenges. The collaboration between team members and guidance of the project supervisor contributed significantly to the successful completion of the project.

7.3 Recommendations

Once the current version of Tawfeer application is completed, there are some future enhancements that are proposed to improve the system even more.

First, the AI module can be extended to offer smarter analysis of food quality, forecasting its expiration, offering individual nutrition guidance, and multilingual chatbot support.

Second, it is possible to develop partnerships with delivery services and charity to enhance the process of food collection and delivery to make the redistribution process quicker and more efficient.

Third, the reward system can be improved by letting the users redeem the points in the form of vouchers, discounts, or even tangible rewards by businesses that are involved.

This would influence more users to make frequent donations. Fourth, more sophisticated dashboards and analytics can be implemented so that government authorities could monitor the environmental impact, food waste reduction rates, and donation patterns across cities and regions.

Lastly, it is possible to introduce marketing campaigns, social challenges, and gamification features that will help to draw more users to the platform and ensure more community members are willing to join the platform. Through constant enhancement and social engagement, Tawfeer can be a top smart solution to curb food waste and foster sustainability in the UAE and other areas.

References

1. Alverson, D.L., Freeberg, M.H., Murawaski, S.A. & Pope, J.G. 1994. A global assessment of fisheries bycatch and discards. Fisheries Technical Paper No. 339, Food and Agriculture Organization of the United Nations: Rome.
2. Choudhury, M.L. 2006. Recent developments in reducing postharvest losses in the Asia-Pacific region. From: Postharvest management of fruit and vegetables in the Asia-Pacific region, APO, ISBN: 92-833-7051-1.
3. Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: A practitioner's approach* (8th ed.). McGraw-Hill.
4. Shelly, G. B., Cashman, T. J., & Rosenblatt, H. J. (2011). *Systems analysis and design* (9th ed.). Course Technology.
5. Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study. *International Journal of Computer Applications, 68*(8), 26–30.
6. "IEEE Recommended Practice for Software Requirements Specifications," in *IEEE Std 830-1998* , vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286.
7. ISO 9004 :2018, Quality management — Quality of an organization — Guidance to achieve sustained success
8. ILO Guide to International Labour Standards on Occupational Safety and Health. Available at: https://www.ilo.org/media/381706/download
9. Save Me M. E. Portal L.L.C. (n.d.). *Together, we can stop food waste!* [Official website]. Retrieved November 21, 2025, from https://app.saveme.ae/home
10. Wajba. (2025). *About Wajba: Reducing food waste through surplus meals & sustainable dining in the UAE* [Official website]. Retrieved November 21, 2025, from https://www.wajba.ae/
11. Platable Portal Co. L.L.C. (n.d.). *Platable – Great food, better prices* [Official website]. Retrieved November 21, 2025, from https://www.platablenow.com
12. Jarke, M., Klamma, R., Pohl, K., & Sikora, E. (2010). Requirements engineering in complex domains. In *Graph Transformations and Model-Driven Engineering: Essays Dedicated to Manfred Nagl on the Occasion of his 65th Birthday* (pp. 602-620). Berlin, Heidelberg: Springer Berlin Heidelberg.
13. Yu, E. (2000). Non-functional requirements in software engineering.

14. Booch, Grady & Rumbaugh, James & Jacobson, Ivar. (1999). Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series). J. Database Manag.. 10.
15. Connolly, T., & Begg, C. (2015). *Database systems: A practical approach to design, implementation, and management* (6th ed.). Pearson.
16. Bass, L., Clements, P., & Kazman, R. (2021). *Software architecture in practice* (4th ed.). Addison-Wesley.
17. Richards, M., & Ford, N. (2020). *Fundamentals of software architecture: An engineering approach*. O'Reilly Media.
18. Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.
19. Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
20. Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.
21. Lu, L., & Kim, D. (2010). Required behavior of sequence diagrams: Semantics and conformance. *IEEE Software Engineering Research*.
22. Nikiforova, O. (2016). An approach to generation of UML sequence diagrams. *International Conference on Software Engineering and Applications*.
23. Axios Contributors. (2024). *Axios: Promise based HTTP client for the browser and Node.js*. GitHub. https://axios-http.com/
24. Expo Contributors. (2024). *Expo documentation*. Expo. https://docs.expo.dev/
25. Meta Platforms, Inc. (2024). *React Native documentation*. https://reactnative.dev/
26. MongoDB, Inc. (2024). *MongoDB documentation*. https://www.mongodb.com/docs/
27. Node.js Foundation. (2024). *Node.js documentation*. https://nodejs.org/
28. OpenAI. (2024). *OpenAI API documentation*. https://platform.openai.com/docs/
29. Python Software Foundation. (2024). *Python documentation*. https://docs.python.org/3/
30. React Navigation Contributors. (2024). *React Navigation documentation*. https://reactnavigation.org/
31. Bass, L., Clements, P., & Kazman, R. (2021). *Software architecture in practice* (4th ed.). Addison-Wesley.
32. Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
33. Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.