



American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

**A Sustainable Food Waste Reduction System for a Greener
Tomorrow**

A Software Requirement Engineering Project Submitted
By

Semester: Spring_22_23		Section:	Group Number:	
SN	Student Name	Student ID	Contribution (CO1+CO2)	Individual Marks
7	JOYSREE DEY SUKANNA	20-41946-1		
26	S.M. SAJIBUL ISLAM	20-43042-1		
27	MD. ESHMAM RAYED	20-43044-1		
28	AUDITI ROY	20-43050-1		

Software Requirements Specification

for

A Sustainable Food Waste Reduction System for A Greener Tomorrow

Version 1.0 approved

Prepared by JOYSREE DEY SUKANNA

S.M. SAJIBUL ISLAM

MD. ESHMAM RAYED

AUDITI ROY

February 20, 2023

Table of Contents

Revision History.....	4
1. Introduction	4
1.1 Purpose	4
1.2 Document Conventions	5
1.3 Glossary.....	5
1.4 Intended Audience and Reading Suggestions.....	6
1.5 References	7
2. Overall Description	7
2.1 Product Perspective	7
2.2 Product Functions.....	9
2.3 User Classes and Characteristics	9
2.4 Operating Environment	10
2.5 Design and Implementation Constraints.....	10
2.5.1 Brand Guidelines	10
2.5.2 Project Timeline	11
2.5.3 Browser	11
2.6 User Documentation.....	11
2.6.1 Installation Guide	11
2.6.2 Product Manual	11
2.6.3 Introductory Document.....	11
3. System Requirements	11
3.1 System Features.....	11
3.2 Non-Functional/Quality Requirements	15
3.3 Project Requirements.....	17
3.3.1 Time	17
3.3.2 Budget	17
3.3.3 Resources	17
3.3.4 Programming Language.....	18
4. Design and Interface Requirements.....	19
4.1 UML Diagrams	19
4.1.1 Use Case Diagram.....	19

	MMH
4.1.2	Class Diagram 20
4.1.3	Activity Diagram 21
4.1.4	E-R Diagram 22
4.2	Data Dictionary 23
4.3	UI/UX Design Specification..... 24

Revision History

Name	Date	Reason for Changes	Version
SFWRSGT	13-03-2023	First Release	1.0

1. Introduction

1.1 Purpose

This SRS document provides details of both functional and non-functional requirements for the A Sustainable Food Waste Reduction System for A Greener Tomorrow 1.0 (**SFWRSGT** 1.0). This document intended to create a clear understanding among all stakeholders (both from client side and the service provider) regarding the system. It will also build a common platform for communication among the stakeholders.

The purpose of this project is to create a sustainable food waste reduction system that addresses the environmental impact of food waste while improving food security and economic efficiency. This system will involve the collection and distribution of edible food to those in need, while non-edible food waste will be converted into high-quality fertilizer for sale. A portion of the profits from the sale of fertilizer will be donated to charity and used for tree plantation, contributing to a greener tomorrow.

Our project focuses on reducing food waste by creating a sustainable system that minimizes the environmental impact of food waste while maximizing social and economic benefits. We aim to collect edible food that would otherwise go to waste and distribute it to those in need, such as homeless shelters, food banks, and other organizations that serve the food-insecure population. Non-edible food waste, such as vegetable peels and leftover food scraps, will be collected separately and converted into high-quality fertilizer. This fertilizer will be sold to individuals, farmers, and other organizations interested in sustainable farming practices. A portion of the profits generated from the sale of fertilizer will be donated to charity, while another portion will be used to plant trees and contribute to a greener tomorrow. By implementing this sustainable

food waste reduction system, we hope to reduce the amount of food waste produced while providing food security to those in need. At the same time, we aim to create a new revenue stream for the community, while also promoting sustainable farming practices and contributing to a greener future.

1.2 Document Conventions

The following conventions are used throughout the document -

1. All technical terms are italicized
2. All abbreviations are written with bold-faced type.
3. All figure descriptions are italicized
4. The terms System, portal and application are all used interchangeably throughout the document.

1.3 Glossary

List of term and abbreviations

Term/ Abbreviation	Definition/ Full-form
System	The web portal/application being developed
Database	Collection of all the information monitored by this system
Internet	A network connecting only devices (computers) within the company, not visible to the outside world
Stakeholder	Any Person with an interest in the project who is not a developer.
Availability	Availability of any system to the users for doing their tasks
PHP	PHP stands for Hypertext Preprocessor. It is a server-side scripting language used primarily for web development, allowing developers to create dynamic web pages and web applications.
JavaScript	JavaScript is a programming language used to create interactive and dynamic web content. It is often abbreviated as JS.
Apache Server	Apache Server is an open-source web server software that delivers web content to clients over the internet. It is the most widely used web server software and is used to host websites and web applications
IDE	IDE stands for Integrated Development Environment. An IDE typically includes a code editor, a debugger, a compiler or interpreter, and other tools that are necessary for software

	development.
CDN	CDN stands for Content Delivery Network. It is a network of servers located in different geographical locations that work together to deliver content to end-users with high availability and performance. CDNs help to reduce latency, minimize packet loss, and optimize content delivery to provide a better user experience.
Browser	A browser, short for web browser, is a software application used to access and view information on the internet.
Agile Method	Agile Method is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and customer satisfaction. It involves breaking down projects into smaller, manageable chunks called sprints, with continuous feedback and adaptation throughout the development process

1.4 Intended Audience and Reading Suggestions

The target audience for this document is Project Managers, Domain Experts, Developers, and Requirements Engineers. It is highly recommended that you read the paper to understand Food Waste Reduction and importance of tree plantation before reading this document. Audience such as:

1. Developers
2. Project managers
3. Marketing staff
4. Users
5. Testers
6. Documentation writers
7. Maintenance and Support Staff
8. Legal Staff

As per the document is intended for project managers, developers, testers, marketers, users, and document authors so, the document is divided into four parts.

1. Introduction
2. Overall Description
3. System Requirements
4. Design and Interface Requirements

All parts are independent of each other but reading the document in order will give the reader a better understanding of **SFWRSQT**. The rest of this document consists of three parts. The second provides an overview of system functionality. Introduce different types of stakeholders and how they interact with the system. It also describes the environment in which the software operates, including hardware platforms, operating systems and versions, and other software components and applications that should coexist peacefully. In addition, this part also mentions system constraints and assumptions about the project. This part is very useful for users of this system. Third part contains detailed requirements specifications and descriptions of various system interfaces. This part is written primarily for developers, as it describes the software's features in detail in technical terms. It is also written for users as it has a use case. The final part provides a numbered list of discoverable references. These remain in the SRS for tracking to completion. It also lists open issues and things that SRS still needs to do but has not yet addressed.

1.5 References

1. Wen, Z., Hu, S., De Clercq, D., Beck, M.B., Zhang, H., Zhang, H., Fei, F. and Liu, J., 2018. *Design, implementation, and evaluation of an Internet of Things (IoT) network system for restaurant food waste management*. Waste management, 73, pp.26-38.
2. C. Larman, *APPLYING UML AND PATTERNS An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3rd ed., Massachusetts: Pearson Education, 2005

2. Overall Description

2.1 Product Perspective

SFWRSQT is a new web-based system that will fasten the process of food recycling, helping the needy in more transparent way and also, will be good for environment as it focuses on the tree plantation. The system will be hosted at server. Figure 1 is a pictorial representation of the system (context diagram) that shows the system from a bird's eye view. It shows its interactions with external entities. Different types of users will use the system. Their detail descriptions will be discussed in the "User Classes and Characteristics" section.

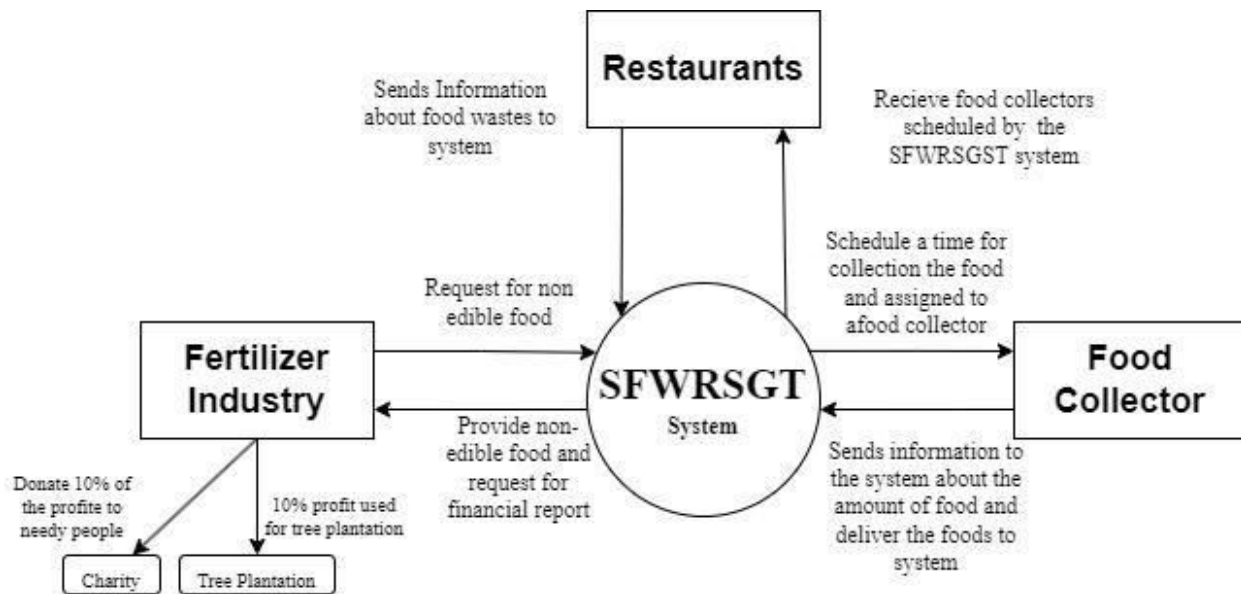


Figure 1: Context Diagram of SFWRSGT

Figure 2 shows the internal architecture of the system and how it will be deployed.

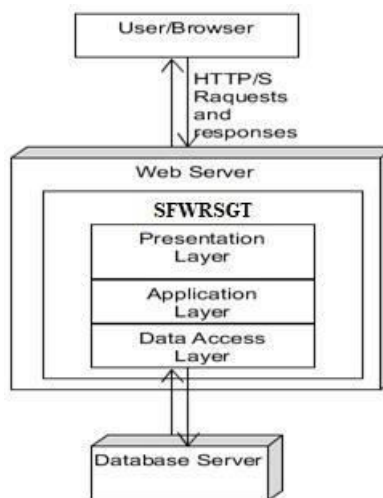


Figure 2: Deployment and Internal architecture

The system will be deployed in a web server and will use a database server for data storage. Internally, the system will be composed of three layers - the presentation layer which will be responsible for displaying/collecting information from users, the application layer will implement all the business logic as dictated by current business practices of the client and the data access layer will be responsible for dealing with the relational data store.

2.2 Product Functions

We have found different user classes for this product. According to the user classes user privileges are different. Product are listed below along with user classes:

From the Restaurant's perspective the system can do the following functionalities-

- Sign Up
- Login
- Request Food for Donation
- View Monthly Report
- View Report Details

From the Food Collector's perspective, the system can do the following functionalities-

- Sign Up
- Login
- View Donation Request
- Collect Available Food
- View Food Storage
- Divide Edible and Non-Edible Food
- Distribute Edible Food
- Distribute Non-Edible Food to Industry
- View Monthly Report
- View Report Details

From the Industry's perspective the system can do the following functionalities-

- Sign Up
- Login
- Collect Available Non-Edible Food
- Create Fertilizer
- Sell Fertilizer
- Distribute 10% Profit to Tree Plantation
- Distribute 10% Profit to Charity
- View Profit Tracking
- View Monthly Report
- View Report Details

2.3 User Classes and Characteristics

Following user classes are identified but the list is subject to change:

User Class	Characteristics
Restaurant	The user will inform the system and will schedule a time for food collector to collect the food.
Industry	The fertilizer industry is an important user of a food recycling

	software, as they rely on non-edible organic waste materials to produce high-quality fertilizers. Generated 10% profit will be donated to charity and another 10% will be used for Tree plantation.
System	Someone, probably from the Authority/ Management who can view records of all users, can authorize data entry or updates. And who also checks the food quality and categorizes the food into two types Edible and Non-edible. Then, distributes the food to Industry and needy peoples.
Food Collector	The food collector is a user who is responsible for collecting food waste from restaurants. The food collector should have access to the software system to view and update the status of the food waste collection process. The food collector should be able to generate reports on the amount of food waste collected.

2.4 Operating Environment

ID	Description
OE01	System will be web based and accessible from all types of browsers from any location and preferably at any time.
OE02	All data (user data and so forth) must be kept in a secure relational database management system.
OE03	System will run in Windows, Linux, MacOS.

2.5 Design and Implementation Constraints

1. System response to user logins must be within 5 seconds of the initiation of the request, provided network and server. *PHP* for backend.
2. Data encryption must be used
3. A CDN (Content Delivery Network) will be used to minimize the latency between the user's location and the server location which decreases the overall page load time. In addition to an extra layer of security against different kinds of attacks and an overall website caching for increasing website delivery speed.
4. The scripts will be hosted on a serverless computing network known as (Google Cloud Functions), this step ensures super-fast execution of these scripts without a single point of failure and minimizes operation cost substantially.

2.5.1 Brand Guidelines

For a business website, one of the non-negotiables for success is identifying brand guidelines and sticking to them. That is why the theme of the website matches the color of the logo. The fonts used are kept as minimal as possible in order to prevent any type of inconsistency.

2.5.2 Project Timeline

The current timeline has been planned by following the *Agile* method. The whole project has been divided between modules for planning and development teams and again into sub modules for team members. Each module will be planned, developed and then reviewed. Timeline may exceed the deadline but quality has to be prioritized over delivery.

2.5.3 Browser

Each browser has a different compiler and they interpret different codes differently and some do not support the same versions of the Markup Languages, Style Sheets or JavaScript. Therefore the appearance can be different based on the browser used to view it. The website will be developed and tested with Google Chrome, Mozilla Firefox, Safari, Microsoft Edge and Brave Browser so that almost some of the well-known browsers can support it.

2.6 User Documentation

Documentation for users will be online, available on each page of the system. Also, other options will be available:

2.6.1 Installation Guide

This covers details about how to install, set up, and use our product. It comprises step-by-step instructions that guide customers through the process installation and setup process.

2.6.2 Product Manual

If you want documentation that demonstrates every step regarding how to use and make the most of your offerings, the product manual is the right choice. It comprises examples, video tutorials, and how-to articles that explain how our product works.

2.6.3 Introductory Document

This document gives an overview of what our product is about, what are its functions, use cases, and what customers can expect from it. It's like a short and quick guide to inform customers with our product.

3. System Requirements

3.1 System Features

1. Login

Functional Requirements (FRs)

- 1.1 The software shall allow users to login with their given username and password, email address, phone number, etc.
- 1.2 The login credentials (username and password) shall be verified with database records.
- 1.3 If the login successful, the home page of the user account shall be displayed.
- 1.4 If the username and/or password has been inserted wrong, the random verification code shall be generated and sent to the user's email address by the system to retry login.
- 1.5 If the number of login attempt exceed its limit (3 times), the system shall block the user account login for one hour.

Priority Level: High

Precondition: user have valid user id and password

2. Monthly Report

Functional Requirements (FRs)

- 2.1 Users shall be able to see the total food donated to the needy people for the current month.
- 2.2 Users shall be able see how many trees were planted this month.
- 2.3 Users shall be able to see the top 3 donators of the month.

Priority Level: Medium

Precondition: User have valid user id and password. Also, must be a User.

Cross-references: N/A

3. Food Donation from Restaurant

Functional Requirements (FRs)

- 3.1 The restaurant shall be able to give the quantity of food.
- 3.2 The restaurant shall be able to give the condition of the food.
- 3.3 Shall provide an expiry time for the food.
- 3.4 Information about food quantity shall be saved in the database.

Priority Level: Medium

Precondition: User must have to login successfully.

4. Account Setting

Functional Requirements (FRs)

- 4.1 The system shall allow users to update their credentials.
- 4.2 Restaurants shall able to turn on anonymous mode to hide their profile.
- 4.3 Updated credentials shall be saved in database.

Priority Level: High

Precondition: User must have to login successfully.

Cross-references: 1.1

5. Available Edible Food

Functional Requirements (FRs)

- 5.1 The amount of food given to each person shall be defined by the system.
- 5.2 A process shall be in place for distributing the donated food to those in need.
(Examples- homeless peoples, beggars, and other charitable organizations who works for needy peoples.)
- 5.3 The donated food shall be subject to inspection to ensure that it meets safety and consumption standards. The system shall incorporate mechanisms to prevent the distribution of spoiled or contaminated food, including automated quality checks, real-time tracking of inventory to detect and isolate any potential issues.
- 5.4 The system shall maintain an inventory of all donated food items and track their expiration dates to prevent the distribution of expired food.

Priority Level: High

Precondition: Only System will use this feature.

Cross-references: N/A

6. Available Food Collectors

Functional Requirements (FRs)

- 6.1 System shall allow the restaurants to check the availability of distributor.
- 6.2 Restaurant shall be able to view the available food collector and arrival time.
- 6.3 The system must provide a unique serial number of that food collection process before assigning the food collector, which will help to keep track on the foods and the food collector.

Priority Level: Medium

Precondition: Must be a food collector type of user. User must check the assigned task first with the details of location, food amount.

Cross-references: N/A

7. Food Quality Checking

Functional Requirements (FRs)

- 7.1 The system shall be able to categories the food after it is collected from the restaurant.
- 7.2 The system must allow the option to notify the fertilizer industry if it's non-edible.

- 7.3 If the food turns out to be non-edible, then the system shall provide an option for tagging the collected food with a unique serial number for keep track of the food. The unique serial number must be saved into database.
- 7.4 If the food is edible then there shall be an alert notification in the system.
- 7.5 Edible foods shall be distributed and the amount of food distributed every day shall be counted, also must be saved into the database.
- 7.6 The system shall allow the fertilizer industry interface to see these non-edible food alerts, after system updates the non-edibility notification to the database.

Priority Level: High

Precondition: Food collector must collect the food from the restaurant.

Cross-references: 6

8. Check Request for Non-edible Food

Functional Requirements (FRs)

- 8.1 The system shall display requests for non-edible food collection from industrial sources upon receiving alert posts from the system.
- 8.2 If the system accepts the request of the industry, an acceptance message shall be sent to the industry, and a specific time slot shall be provided for the pick-up of inedible food.

Priority Level: Medium

Precondition: N/A

Cross-references: N/A

9. Tree Plantation Monitoring

Functional Requirements (FRs)

- 9.1 Based on the generated 10% profit, the system shall count how many trees must be planted.
- 9.2 Every tree that is planted shall be assigned a unique serial number, which will be saved in a database.
- 9.3 The fertilizer industry shall have the authority to choose the location for tree plantation.
- 9.4 This record shall be essential in generating the monthly report and tracking the trees.

Priority Level: Low

Precondition: Based on the financial report this requirement will execute.

Cross-references: N/A

10. Available Recycle Food

Functional Requirements (FRs)

- 10.1 The system shall provide information to the fertilizer industry about the quantity of available food.
- 10.2 Fertilizer Industry shall be given the option to select their preferred quantity.

Priority Level: Low

Precondition: N/A

Cross-references: N/A

3.2 Non-Functional/Quality Requirements

QA1: Availability: The system should be available for 23.5 hours a day and all 7 days of the week, with a 30 minutes max downtime per day for server reset, database backup & data integrity check.

Priority Level: High

Precondition: N/A

Cross-references: QA8, QA9

QA2: Efficiency: At the desired peak load circumstances, at least 25% of the processing capacity and RAM available to the program must be idle.

Priority Level: Low

Precondition: N/A

Cross-references: QA3, QA5, QA6

QA3: Integrity: The database must have robust cyber security measures to safeguard against any unauthorized access, theft, or modification of data. Only authorized personnel with proper access credentials or administrator privileges will be permitted to perform maintenance tasks. Moreover, the database must be regularly backed up during server downtime as a backup measure.

Priority Level: High

Precondition: N/A

Cross Reference: QA2, QA5, QA10, QA11

QA4: Flexibility: The software should be adaptable to minor changes and have the ability to produce additional outputs using existing data in the database.

Priority Level: High

Precondition: N/A

Cross Reference: QA2, QA10

QA5: Interoperability: The software should be capable of integrating with other systems or devices used in the process, such as scales, tracking devices, and waste disposal equipment. The software should also be able to exchange information with other software systems used by stakeholders, such as suppliers, food collectors, and system facilities. This interoperability will enable the system to operate more efficiently and effectively, minimizing waste and maximizing the use of resources. Additionally, interoperability will enable the software to be easily upgraded or scaled as the needs of the business evolve over time.

Priority Level: Low

Precondition: N/A

Cross Reference: QA2, QA4, QA7

QA6: Maintainability: The software should be easy to maintain and update over time, with minimal disruption to operations. The system should be designed with a modular architecture, which enables developers to modify or add functionality without affecting the overall structure of the software. The code should be well-documented, with clear instructions on how to maintain and troubleshoot the system. Additionally, the software should be compatible with common programming languages and platforms, making it easy to find skilled developers to work on the system. Any data loss should be easily recoverable in a short time due to automated daily backups.

Priority Level: High

Precondition: N/A

Cross Reference: QA8, QA10

QA7: Portability: The software should be able to run on different platforms, such as desktops, laptops, and mobile devices.

Priority Level: Medium

Precondition: N/A

Cross Reference: QA3 QA11

QA8: Reliability: The software must be designed and developed to minimize the risk of data loss, system crashes, or other technical issues that could disrupt the food distributing process.

Priority Level: Medium

Precondition: N/A

Cross Reference: QA10

QA9: Robustness: The software should be able to handle unexpected situations and recover from errors or failures quickly and smoothly.

Priority Level: Low

Precondition: N/A

Cross Reference: QA8, QA12

QA10: Testability: The software should have built in mechanisms for logging errors and providing detailed information on errors encountered during testing, enabling developers to

quickly identify and fix issues. Testing should be done by 10 individual testers over a week long period with each distributing food to 100 peoples, the data volume should provide sufficient sampling for any potential errors to be found in either write and fetch request.

Priority Level: High

Precondition: N/A

Cross Reference: QA2, QA3, QA7

QA11: Performance: The software should be able to process large volumes of data quickly and accurately. The software should also be able to handle a high volume of concurrent users without any lag or downtime.

Priority Level: High

Precondition: user should be using a good internet line.

Cross Reference: N/A

QA12: Usability: The system should be easy to use and navigate with a user friendly interface. Every page will have clear instruction of what to do. The software should also be accessible and compatible with various devices and operating systems to enable users to access it from anywhere at any time.

Priority Level: Medium

Precondition: N/A

Cross-references: QA5, QA6

3.3 Project Requirements

3.3.1 Time

A *four-month* time frame can be both a blessing and a challenge for a project. On one hand, four months provide a decent amount of time to plan, execute and deliver a project. It allows the team to work on a project in a structured way, with the ability to adjust and fine-tune their approach. On the other hand, four months may not be enough time to accommodate for any major roadblocks or unexpected issues that could arise during the project lifecycle. As such, it is important to establish clear timelines and priorities at the outset of the project, with regular check-ins to ensure that the project stays on track. Overall, a four-month time frame can be a suitable and manageable time frame for many projects, as long as the team remains flexible, communicative, and focused on delivering high-quality results

3.3.2 Budget

The project requires the development of a software solution within a budget of 8 lakh taka. The software must meet the specified requirements and deliverables, which will be determined through consultation with the stakeholders

3.3.3 Resources

The project necessitates a team of seven individuals, comprising one *project manager*, four *software developers*, and two *software testers*. To facilitate software development and testing, the team will require five *desktop computers* and two *Laptops*. Access to the internet is essential for research and effective communication. The project will also entail the use of reusable components, subsystems, modules, and data management systems to optimize software development and testing.

3.3.4 Programming Language

The system developer needs *PHP*, *JavaScript* to develop Backend Application and run it on *Apache* server and to run *PHPMyAdmin* SQL server for *MySQL* database, *ReactJS* to develop Frontend Application and any type of IDE for code writing. Operating system can be flexible such as Windows, Mac, Linux.

4. Design and Interface Requirements

4.1 UML Diagrams

4.1.1 Use Case Diagram

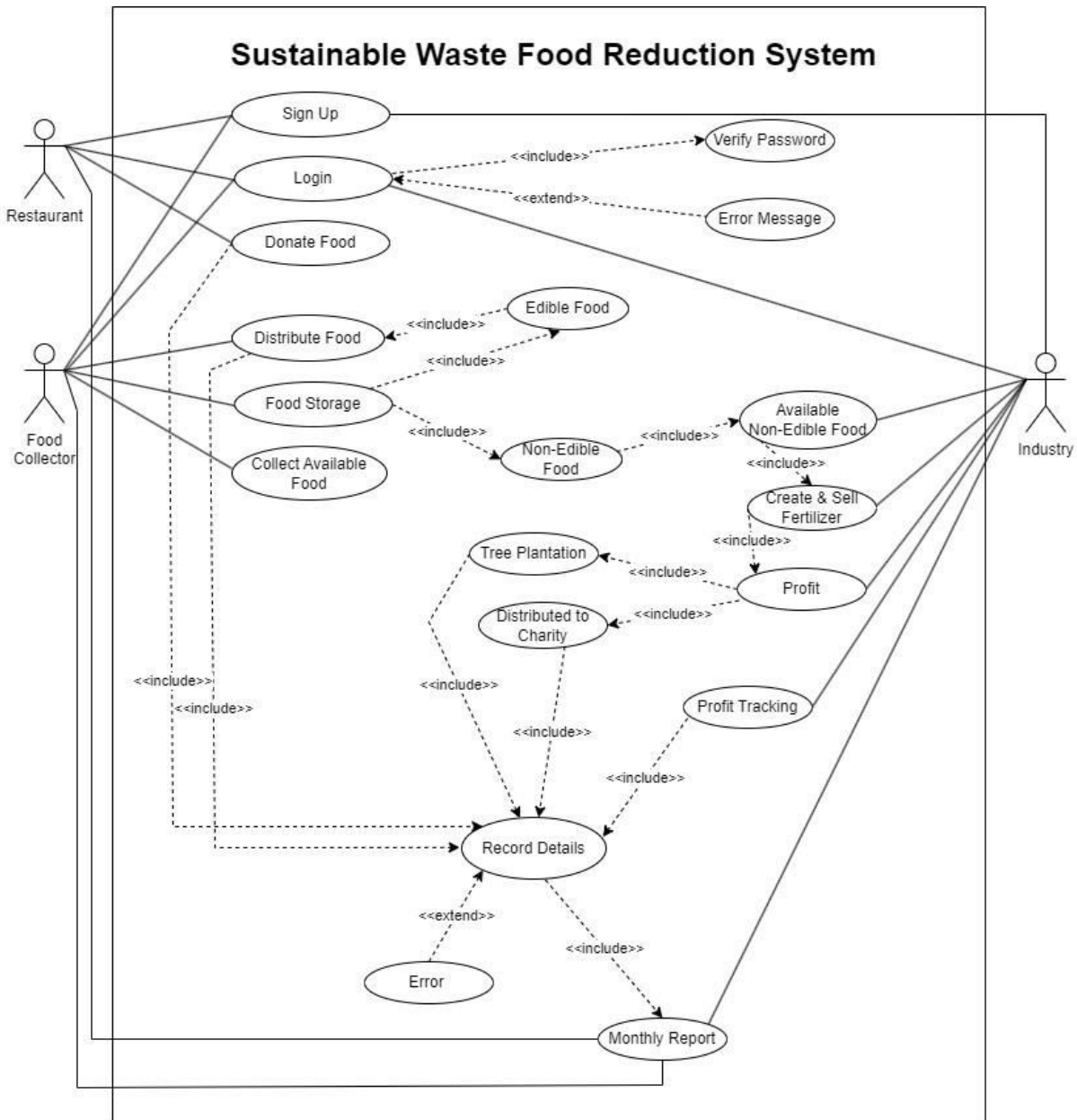


Figure: Use Case diagram of the System

4.1.2 Class Diagram

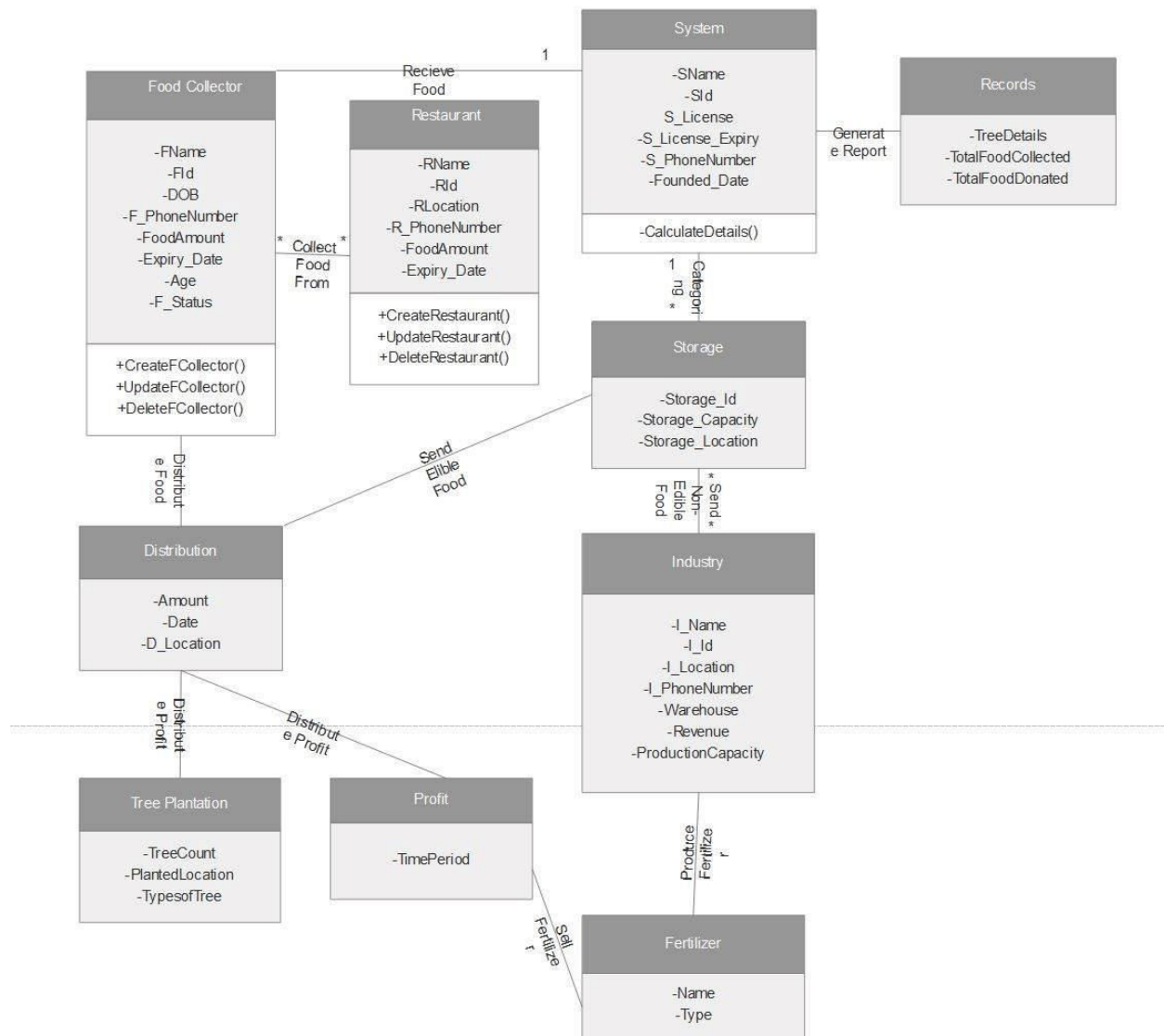


Figure: Class Diagram of the system

4.1.4 E-R Diagram

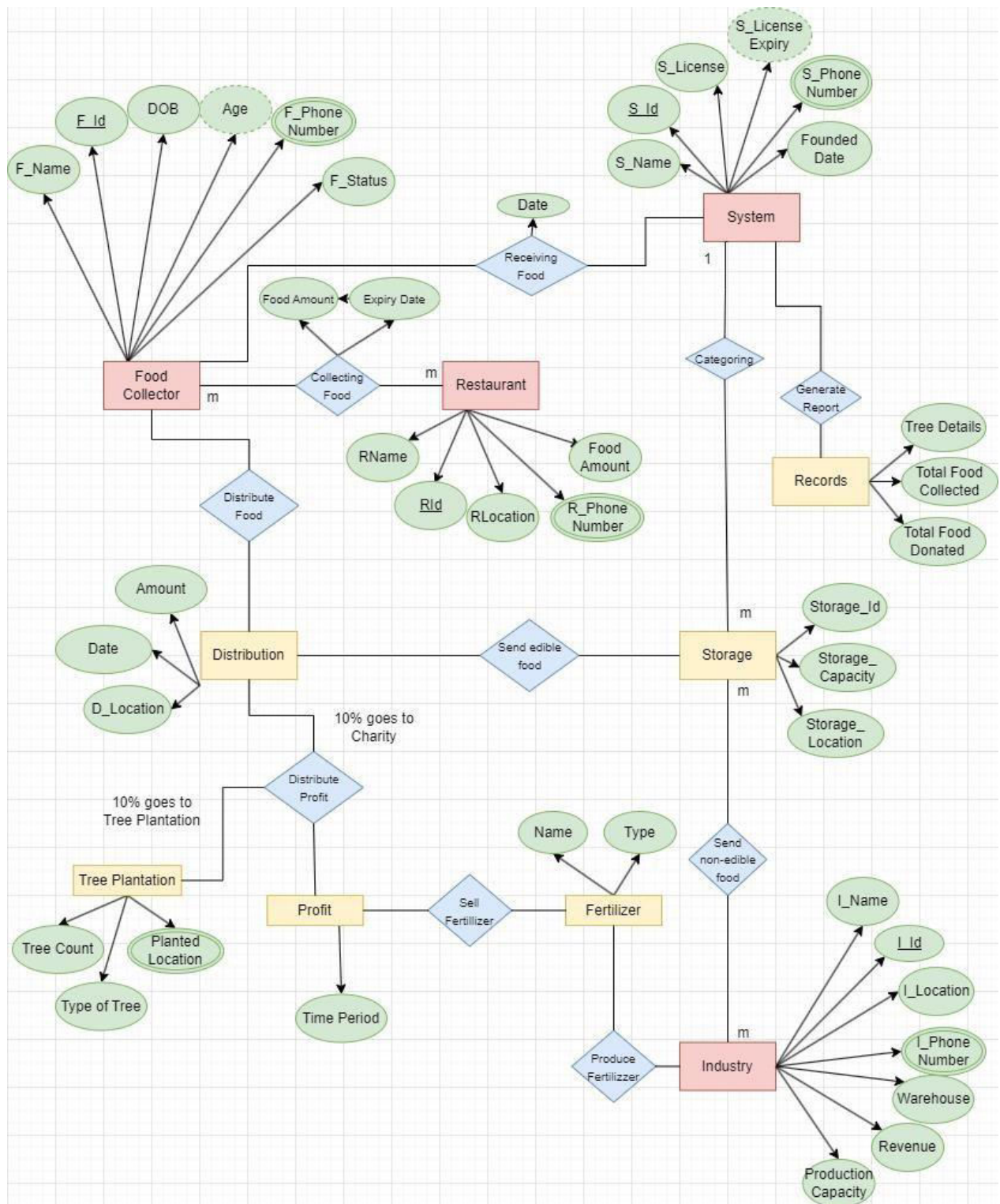


Figure: ER Diagram (Entity Relationship)

4.2 Data Dictionary

Entity	Attribute	Type/Size	Validation	Key
System	S_Name	Text(100)	Required	
System	S_id	Int	Required	Primary
System	S_License	Text(100)	Required	
System	S_PhoneNumber	Text(50)	Required	
System	Founded_Date	Date	Valid Date	
Restaurant	RName	Text(150)	Required	
Restaurant	Rid	Int	Required	Primary
Restaurant	RLocation	Text(200)	Required	
Restaurant	R_PhoneNumber	Text(50)	Required	
Restaurant	FoodAmount	Float	Required	
FoodCollector	F_Name	Text(100)	Required	
FoodCollector	F_id	Int	Required	Primary
FoodCollector	DOB	Date	Valid Date	
FoodCollector	Age	Int	Required	
FoodCollector	F_PhoneNumber	Text(50)	Required	
FoodCollector	F_Status	Text(200)	Required	
Industry	I_Name	Text(100)	Required	
Industry	I_Id	Int	Required	Primary
Industry	I_Location	Text(200)	Required	
Industry	I_PhoneNumber	Text(50)	Required	
Industry	Warehouse	Text(100)	Required	
Industry	Revenue	Float	Required	
Industry	ProductionCapacity	Float	Required	
Records	TreeDetails	Text(500)	Required	
Records	TotalFoodCollected	Float	Required	
Records	TotalFoodDonated	Float	Required	
Distribution	Amount	Float	Required	
Distribution	Date	Date	Valid Date	
Distribution	Location	Text(200)	Required	
TreePlantation	TreeCount	Long int	Required	
TreePlantation	PlantedLocation	Text(200)	Required	
TreePlantation	TypeOfTree	Text(100)	Required	
Profit	TimePeriod	Timestamp	Valid Time	
Fertilizer	Name	Text(100)	Required	
Fertilizer	Type	Text(100)	Required	
CollectingFood	FoodAmount	Float	Required	
CollectingFood	ExpiryDate	Date	Valid Date	
ReceivingFood	Date	Date	Valid Date	
Storage	Storage_Id	Int	Required	Primary
Storage	Storage_Capacity	Float	Required	
Storage	Storage_Location	Text(200)	Required	

4.3 UI/UX Design Specification

- Select and appropriate tools for prototyping
- Design a prototype of your proposed solution using the selected tools based on the above-mentioned UI/UX design principles.

Text Format:

- Style: Times New Roman
- Size: 12
- Space: 1.0
- Alignment: Justify