



**SE3040 – Application Frameworks**  
**BSc (Hons) in Information Technology**  
**3<sup>rd</sup> Year**  
**Faculty of Computing**  
**SLIIT**  
**2023 – Group Project**  
**SRS**

Group Leader: Kulasekara D.A.M.N

Group Leader's Contact Number: 0717573408

Batch: Y3.S1.WE.SE.01

Description	Student ID	Name
Group Leader	IT21028878	Kulasekara D.A.M. N
Member 2	IT21034404	Nipun P.G. I
Member 3	IT21039140	Dewasurendra S. V
Member 4	IT21071034	Wanniarachchi T. T

Application Topic	Food Wasting Reducing System (Environment and Green Energy)
Group Name	Binary Beasts (CSSE_WE_05)

## Project Declaration

We, the members of CSSE\_WE\_05 hereby declare that our group project is entirely authentic and original. We have conducted thorough research and analysis to ensure that our work is not plagiarized or copied from any other sources.

We have followed all guidelines provided by our LIC and have complied with all ethical and academic standards.

We take full responsibility for the authenticity of our work and understand the implications of academic dishonesty. We have worked collaboratively to produce this project, and each member has contributed to the best of their abilities.

We hereby affirm that our project represents our honest effort and commitment to academic integrity, and we take pride in presenting it as our own.

A handwritten signature in black ink, appearing to read 'Norma', is written over a circular stamp or seal.

---

Group Leader (Signature)

Kulasekara D.A.M. N

## Table of Content

SE3040 – Application Frameworks .....	1
BSc (Hons) in Information Technology .....	1
3 <sup>rd</sup> Year .....	1
Faculty of Computing .....	1
SLIIT .....	1
2023 – Group Project .....	1
SRS.....	1
Table of Content.....	3
1. Introduction .....	5
1.1 purpose.....	5
1.2 Intended Audience .....	6
1.3 Definitions .....	6
1.4 Scope .....	6
1.5 System Interfaces .....	6
1.6 Constraints, Assumptions, and Dependencies.....	7
1.7 Individual Contributions .....	7
Kulasekara D.A.M.N.....	7
Nipun P.G.I .....	7
Dewasurendra S. V .....	7
IT21039140 .....	7
2. Functional Requirements .....	8
Kulasekara D.A.M.N (IT21028878) – Donor management .....	8
Nipun P.G.I (IT21034404) – Recipe management .....	8
Dewasurendra S. V (IT21039140) – Meal plan management.....	9
Wanniarachchi T. T (IT21071034) – Receiver management.....	9
3. Non-functional Requirements.....	10
Kulasekara D.A.M.N (IT21028878) .....	10
Nipun P.G.I (IT21034404) – Recipe management .....	10

	Dewasurendra S. V (IT21039140) – Meal plan management.....	10
	Wanniarachchi T. T (IT21071034) – Receiver management.....	11
4.	User Interface Requirements.....	12
	Kulasekara D.A.M.N (IT21028878) – Donor management .....	12
	Nipun P.G.I (IT21034404) – Recipe management .....	13
	Dewasurendra S. V (IT21039140) – Meal plan management.....	14
	Wanniarachchi T. T (IT21071034) – Receiver management.....	15
5.	System Architecture.....	16
6.	Data requirements.....	17
	Kulasekara D.A.M.N (IT21028878) – Donor management .....	17
	Nipun P.G.I (IT21034404) – Recipe management .....	18
	Dewasurendra S. V (IT21039140) – Meal plan management.....	18
	Wanniarachchi T. T (IT21071034) – Receiver management.....	18
7.	Performance requirements.....	18
	Kulasekara D.A.M.N (IT21028878) – Donor management .....	18
	Nipun P.G.I (IT21034404) – Recipe management .....	19
	Dewasurendra S. V (IT21039140) – Meal plan management.....	19
	Wanniarachchi T. T (IT21071034) – Receiver management.....	19
8.	Security requirements.....	20
	Kulasekara D.A.M.N (IT21028878) – Donor management .....	20
	Nipun P.G.I (IT21034404) – Recipe management .....	20
	Dewasurendra S. V (IT21039140) – Meal plan management.....	20
	Wanniarachchi T. T (IT21071034) – Receiver management.....	21
9.	Testing requirements.....	21
	Kulasekara D.A.M.N (IT21028878) – Donor management .....	22
	Nipun P.G.I (IT21034404) – Recipe management .....	22
	Dewasurendra S. V (IT21039140) – Meal plan management.....	22
	Wanniarachchi T. T (IT21071034) – Receiver management.....	23

# 1. Introduction

## 1.1 purpose

The primary purpose of the system our team hopes to develop is to **reduce food wastage** which is common in many homes, supermarkets, fruit, and vegetable shops.

Here we focus on dry foods and uncooked vegetables and fruits. The reason for choosing dry food is the possibility that parties in need of food may not receive cooked food parcels on time due to the quality of cooked food, preservation, and parceling methods, climatic conditions, and other circumstances beyond the control of an online system.

Due to Sri Lanka's current acute economic crisis, half of Sri Lankan families are cutting their children's meals, and 27 percent of the more than 2,300 households surveyed report that adults are skipping meals to feed their children.

We will provide excess food to such families with the help of this system. But it is difficult to identify the economic status of those families by registering them in the system as individual families. To prevent this, various committees in the villages with food shortage families can register in the system. Then through those committees, the necessary food can be distributed to the families.

We are developing this system to establish the relationship between the food doner and the receiver. So, food waste, common in urban areas, can be reduced and dry food can be provided to hungry families for free.

And in some houses, they cook more than the required amount in daily cooking. This can also happen during festivals. To minimize this, we hope to design the system so that by inputting the type of food to be cooked and the number of people into the system, a suitable meal plan and suggestion can be obtained.

In That way, we can achieve our main purpose.

## 1.2 Intended Audience

Our target audience can be identified as households where food waste occurs regularly, fruit and vegetable shops and supermarkets. Simply put, any place or organization and person that wastes food falls under our target audience. Similarly, charities, child orphanages, adult orphanages, and any organizations and committees where there is a food shortage, on the other hand, are our target audience.

## 1.3 Definitions

**Donor** - Vegetables and fruit shops, Grocery shops Supermarkets, any shops, and any individuals where there is food excess.

**Receiver** - Charities, child orphanages, adult orphanages, and any organizations or committees where there is a food shortage.

**Administrator** – The person who manages the system.

**Donations** - Posts published in the system about food excess.

## 1.4 Scope

After registering the system, every type of user has an account. Users can login to the system. According to the type of users, they can publish their donation details. Then these donations should be in the right condition. After checking donations details by an administrator, the donation details will be published in the system.

Parties in need of donations can search for surplus food through the system. They can search for them by name and location. Those parties can request food by clicking the request button. If the donor is willing to give food to the person requesting the food, the request can be accepted. After this process, the respective donation will no longer be displayed in the system.

The system suggests a meal plan based on the size of the group to reduce food wastage caused by excessive cooking. By entering the food to be cooked and the number of people into the system, a suitable meal plan and suggestion can be obtained.

## 1.5 System Interfaces

This system will be developed using MERN Stack.

MERN Stack is a popular, open source, more accessible, and faster JavaScript-based full-stack web development stack. It contains four leading technologies.

**M** – *Mongo* – No SQL Database (Document based database)

**E** – *Express.js* – Server-side web application framework

**R** – *React.js* – Client-Side JavaScript framework

**N** – *Node.js* – Core Server side JavaScript environment

### Integrated Development Environment

- As an integrated development environment, Visual Studio Code will be used

### REST API

- The REST API will be used with the help of HTTP methods (GET, POST, DROP, DELETE), allowing clients to interact with the server-side over the Internet.

### GitHub

- This project is done as a group, it is also done to allow everyone in the group to collaborate on the same code. Therefore, GitHub will be used as Git-based repository hosting systems.

## 1.6 Constraints, Assumptions, and Dependencies

- Cooked foods are not allowed to distribute via the system due to the possibility that parties in need of food may not receive cooked food parcels on time due to the quality of cooked food, preservation, and parceling methods, climatic conditions, and other circumstances beyond the control of an online system.
- Delivery service is not available in the system.
- Donors can donate only excess foods.
- Receivers can only receive excess food.
- The food to be donated must last at least one week.
- Donated food appears in the system after approval by an admin. The maximum time it takes is only 24 hours.
- If one of the stakeholders is not active within 1 month, a user profile will be canceled by admin.

## 1.7 Individual Contributions

Name	IT Number	Contribution
Kulasekara D.A.M.N	IT21028878	Donor management
Nipun P.G.I	IT21034404	Recipe management
Dewasurendra S. V	IT21039140	Meal plan management
Wanniarachchi T. T	(IT21071034)	Receiver management

## 2. Functional Requirements

### Kulasekara D.A.M.N (IT21028878) – Donor management

- To be a donor, the relevant party should register to the system with accepting terms and conditions. If any required field is empty, those kinds of parties cannot use the system.
- After registering the system, a donor account will be created.
- Using a username and password, the donor can log in to the system. If login fails donors cannot login to the system.
- Then the donor can post a donation post. The post should match the relevant conditions. No limits for donations. Donors can post any number of donation posts per day.
- Donations appear in the system after approval by an admin. If donation post is not the right conditions, post is deleted by admin.
- Donors can view their own and other donation details.
- Donors can edit their own donation details. But all details are not allowed to be edited. Because receivers face various problems with that.
- When a donor receives a request from a receiver, the donor can accept or reject it.
- When a donor accepts a request, both parties can connect via a phone call.
- Donors can view, edit, and delete their own profile.
- Donors can use recipe service and meal plan service.
- After the whole task is done donors can Logout of the system.

### Nipun P.G.I (IT21034404) – Recipe management

- To view recipes, the relevant party should register to the system with accepting terms and conditions. If any required field is empty, those kinds of parties cannot use the system.
- After registering the system, an account will be created.
- Using a username and password, the user can log in to the system. If login fails users cannot login to the system.
- Hence all receivers are an organization or committee, recipes are only visible for donors.
- Then the donor can navigate to the recipe page and enter the main ingredient.
- After, recipes according to the entered ingredient are displayed by the system.
- Donors can view those recipes.
- Then donors can filter recipes again according to the sub ingredients in the above recipes.
- When a donor tries to enter an invalid input, an error message will be displayed.



- Recipes are created by admin.
- Admin allows to view, edit and delete recipes.

### **Dewasurendra S. V (IT21039140) – Meal plan management**

- To view a meal plan, the relevant party should register to the system with accepting terms and conditions. If any required field is empty, those kinds of parties cannot use the system.
- After registering the system, an account will be created.
- Using a username and password, the user can log in to the system. If login fails users cannot login to the system
- Hence all receivers are an organization or committee, meal plans are only visible for donors.
- Donors can enter the number of people and food as inputs.
- Then meal plans are displayed by the system.
- If a donor enters an invalid inputs error message will be displayed by the system.

### **Wanniarachchi T. T (IT21071034) – Receiver management**

- To be a receiver, the relevant party should register to the system with accepting terms and conditions. If any required field is empty, those kinds of parties cannot use the system.
- After registering the system, a receiver account will be created.
- Using a username and password, the receiver can log in to the system. If login fails receivers cannot login to the system.
- Receivers can search for donations according to the locations. Then receivers can request for wanted foods. If the donor accepts or rejects a request, a notification will be received from the receiver.
- After the donor accepts the request both parties can connect via a phone call.
- Receivers can view, edit, and delete their own profile.
- After the whole task is done receiver can Logout of the system.

### **3. Non-functional Requirements**

**Kulasekara D.A.M.N (IT21028878)**

#### **Performance**

The system should respond quickly to requests. Because food donors must be able to identify before expiration dates of the foods. Users should be able to quickly find the locations of people who can donate food and the type of food needed within the system. To spend less time and find the desired search system, the performance of the system should be increased.

#### **Scalability**

The scale of the system must be flexible so that the desired amount of food donations can be added to the system without problems.

**Nipun P.G.I (IT21034404)**

#### **Usability**

People of any level should be able to use the system without any difficulty. For that, the usability of the system should be at an important level.

#### **Maintainability**

A system that contributes to food distribution must be capable of quick and accurate maintenance in the event of breakdowns.

**Dewasurendra S. V (IT21039140)**

#### **Reliability**

The system should display food donations properly in the system. Otherwise, the relevant parties should not be able to see the donations properly.

#### **Security**

Because the personal details of the users are in the system, the system should be high in terms of security.

**Wanniarachchi T. T (IT21071034)**

### **Portability**

It is necessary to switch the system's platform or environment.

### **Resilience**

Backup and disaster recovery procedures should both be supported by the system's ability to recover from faults or failures.

## 4. User Interface Requirements

Kulasekara D.A.M.N (IT21028878) – Donor management

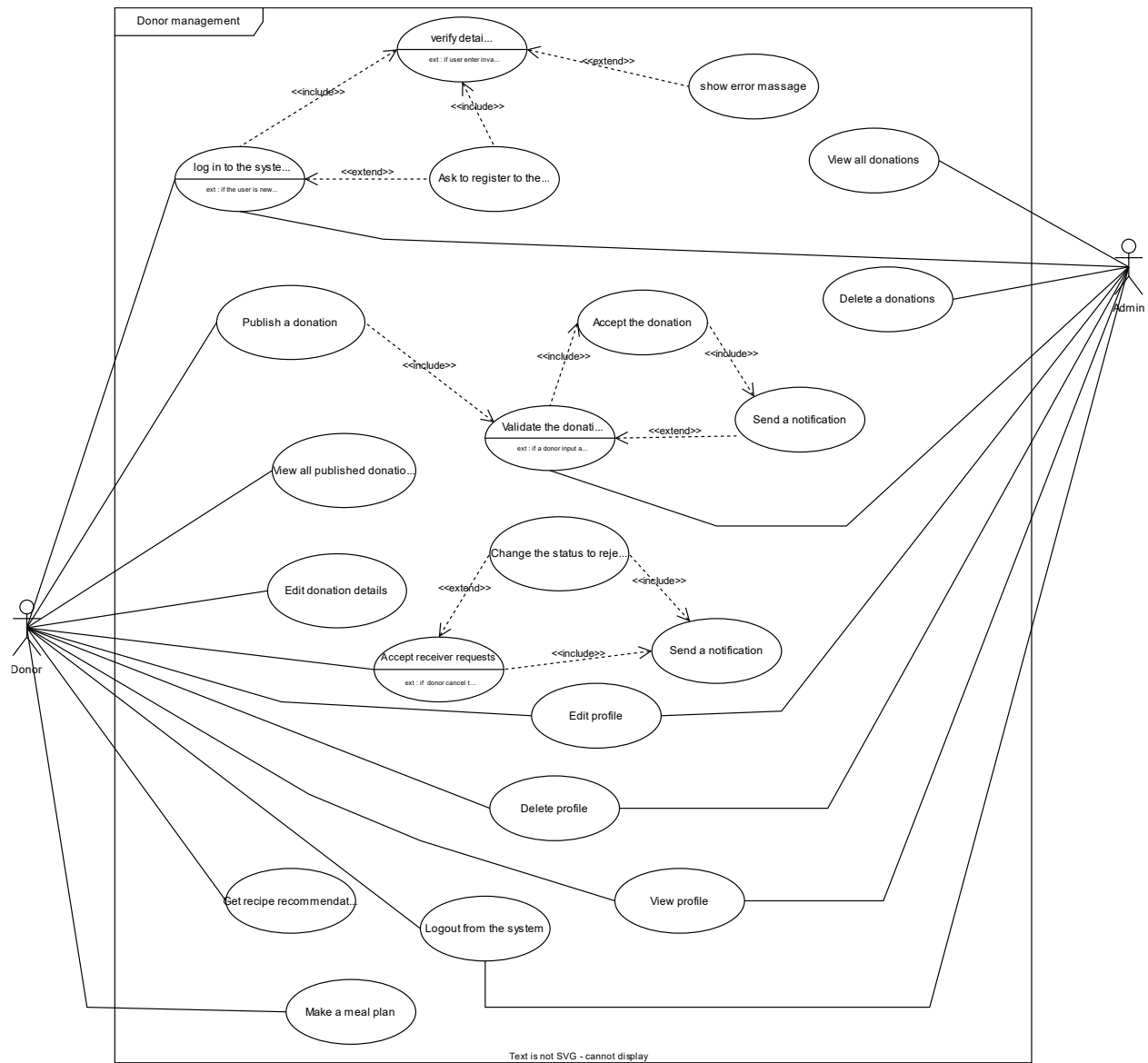


Figure 1: Use Case diagram – Donor Management

## Nipun P.G.I (IT21034404) – Recipe management

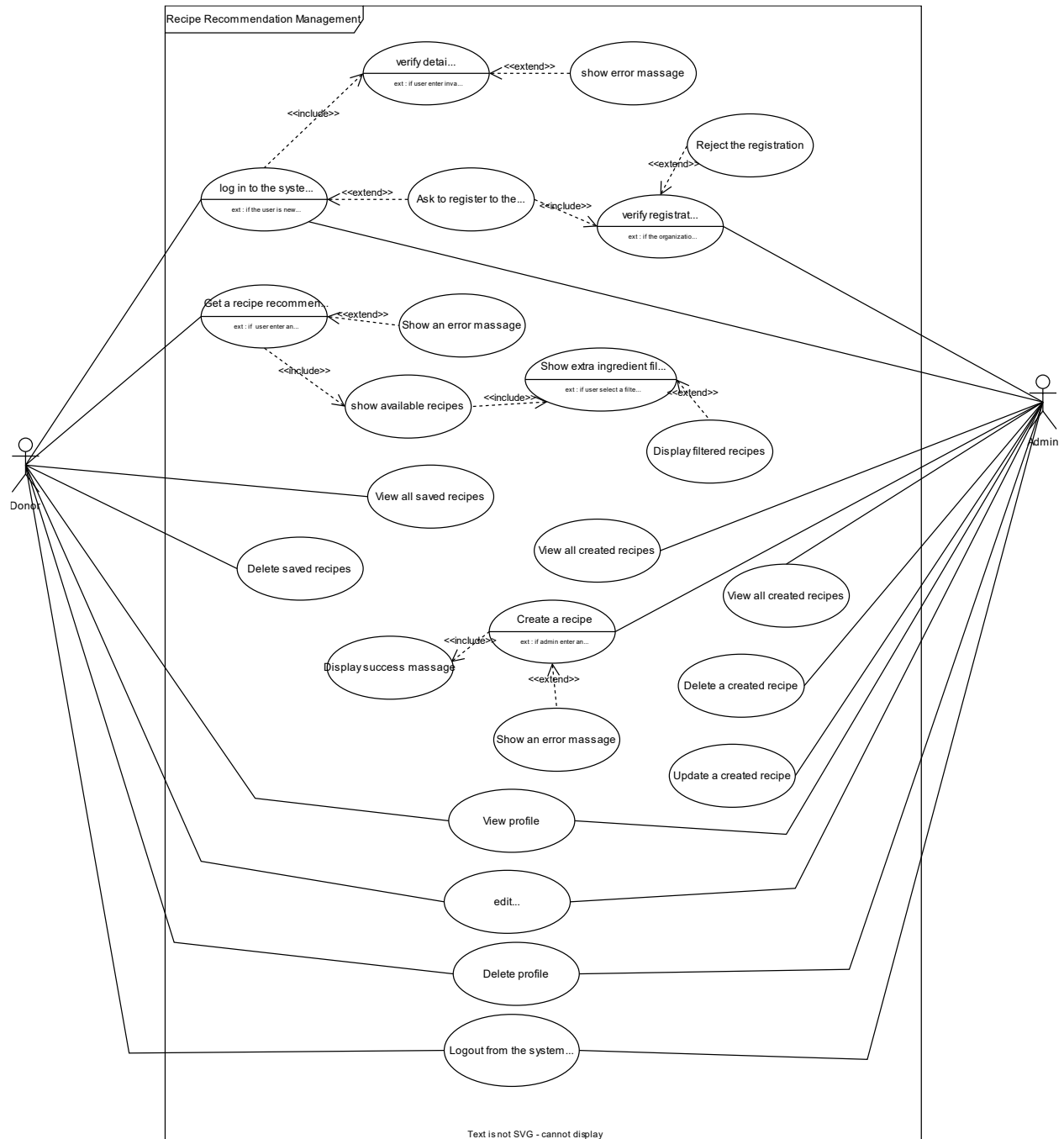


Figure 2 – Use case diagram – Recipe managemen

## Dewasurendra S. V (IT21039140) – Meal plan management

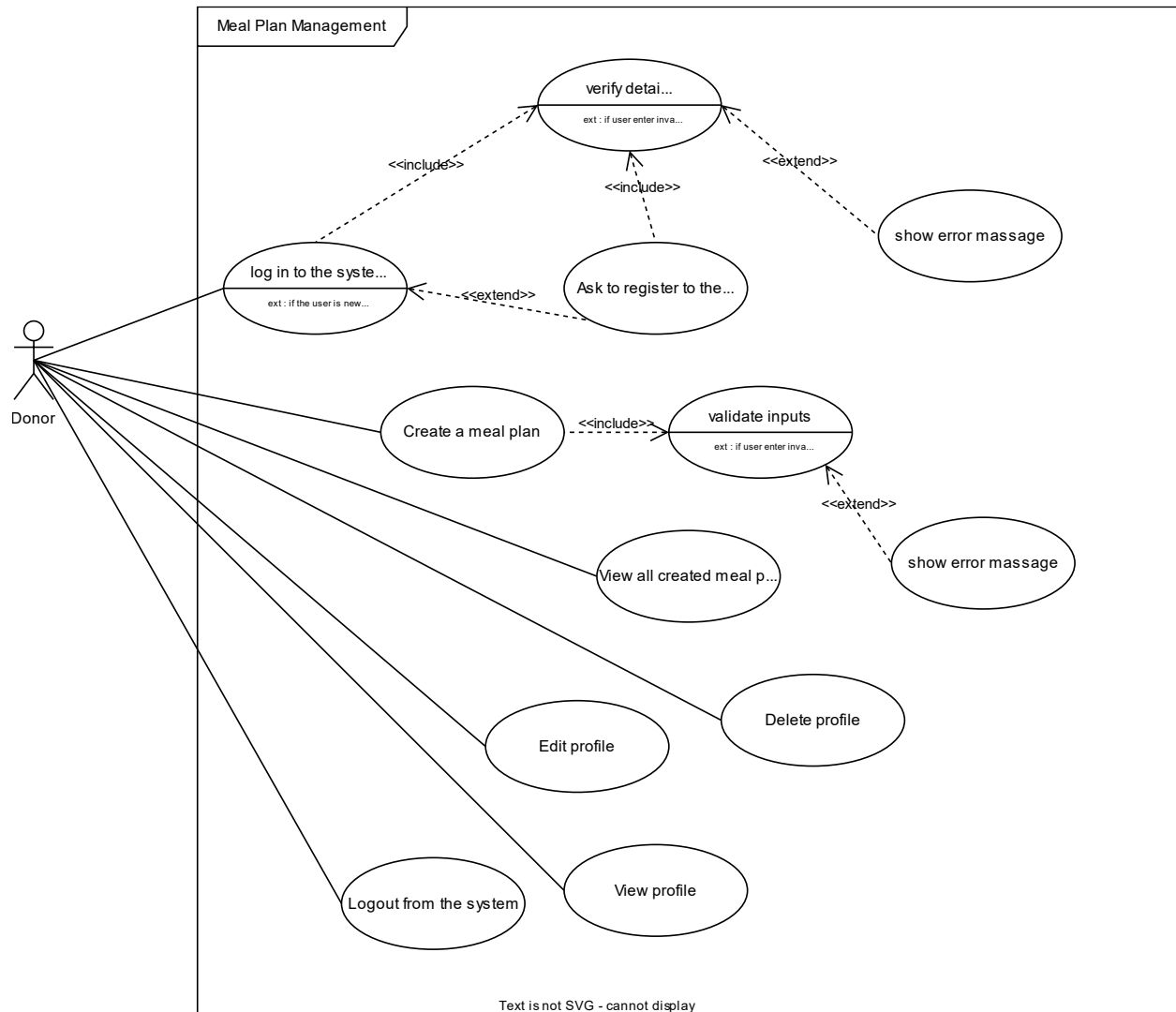


Figure 3 – Use case diagram – Meal plan management

## Wanniarachchi T. T (IT21071034) – Receiver management

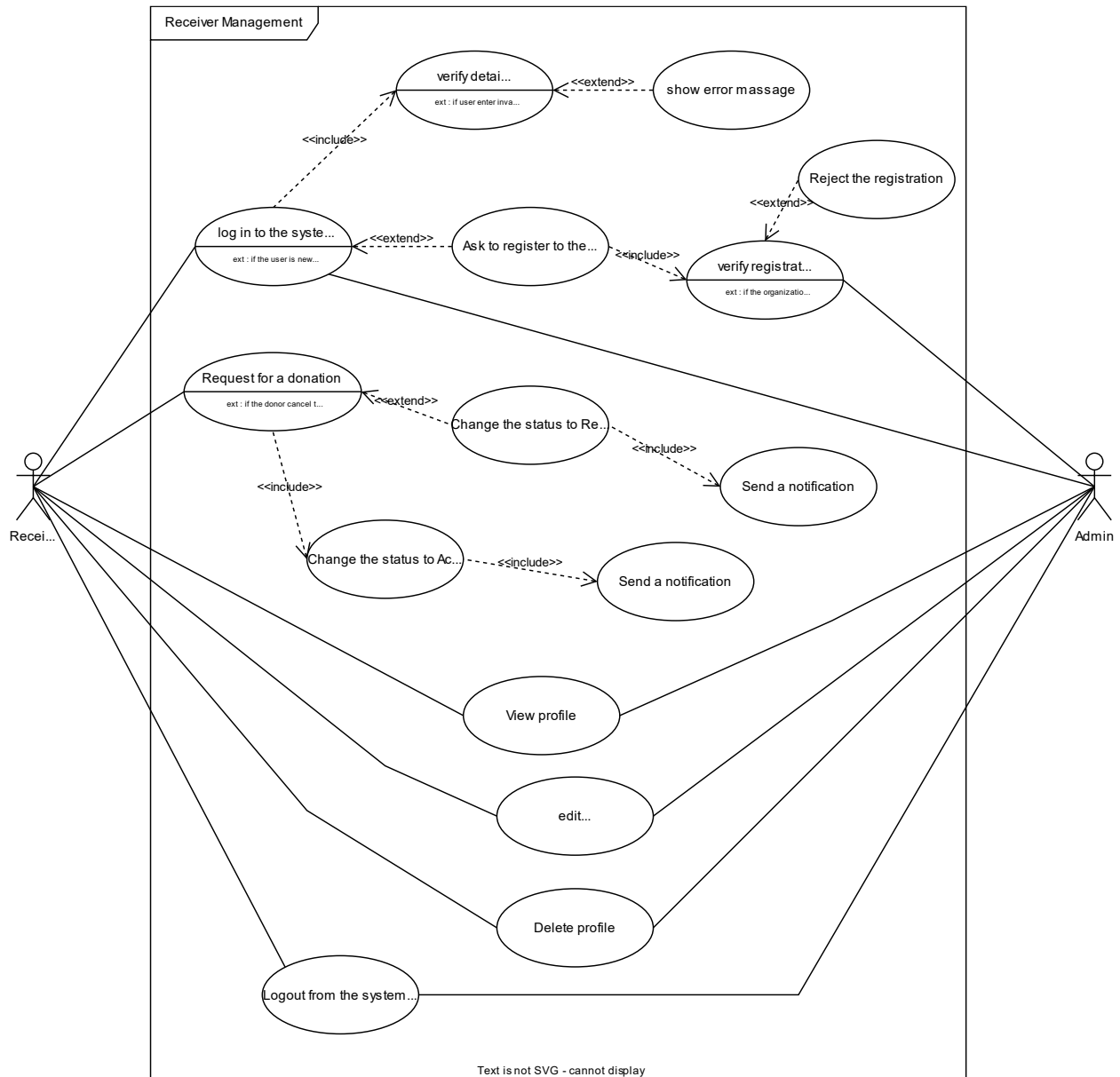


Figure 4 – Use case diagram – Receiver management

## 5. System Architecture

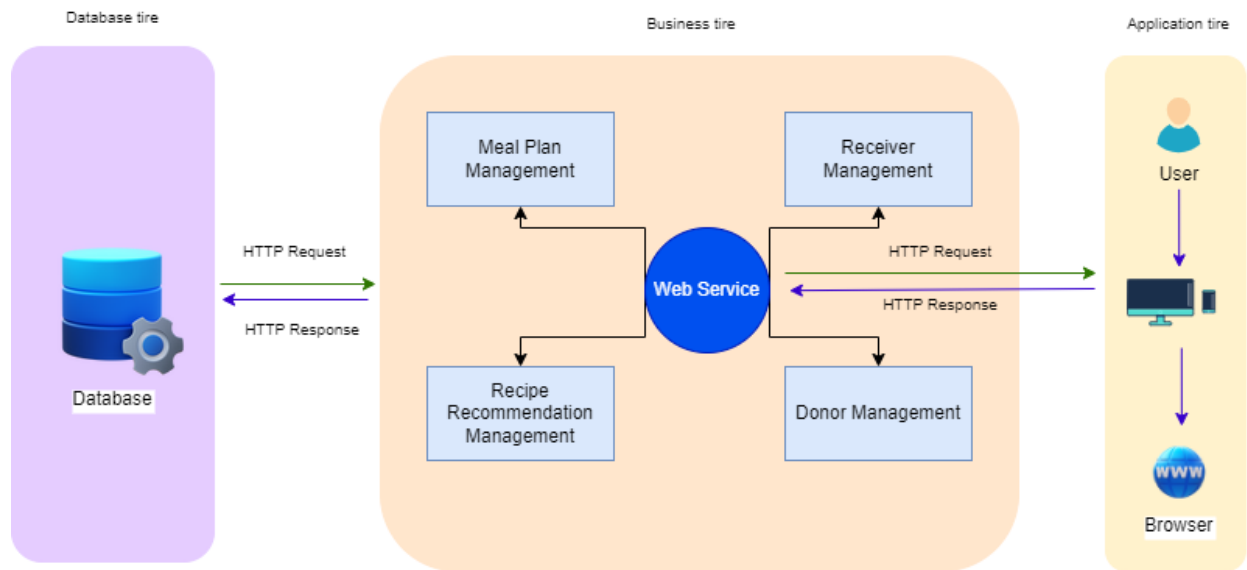


Figure 5 – System Architecture



## 6. Data requirements

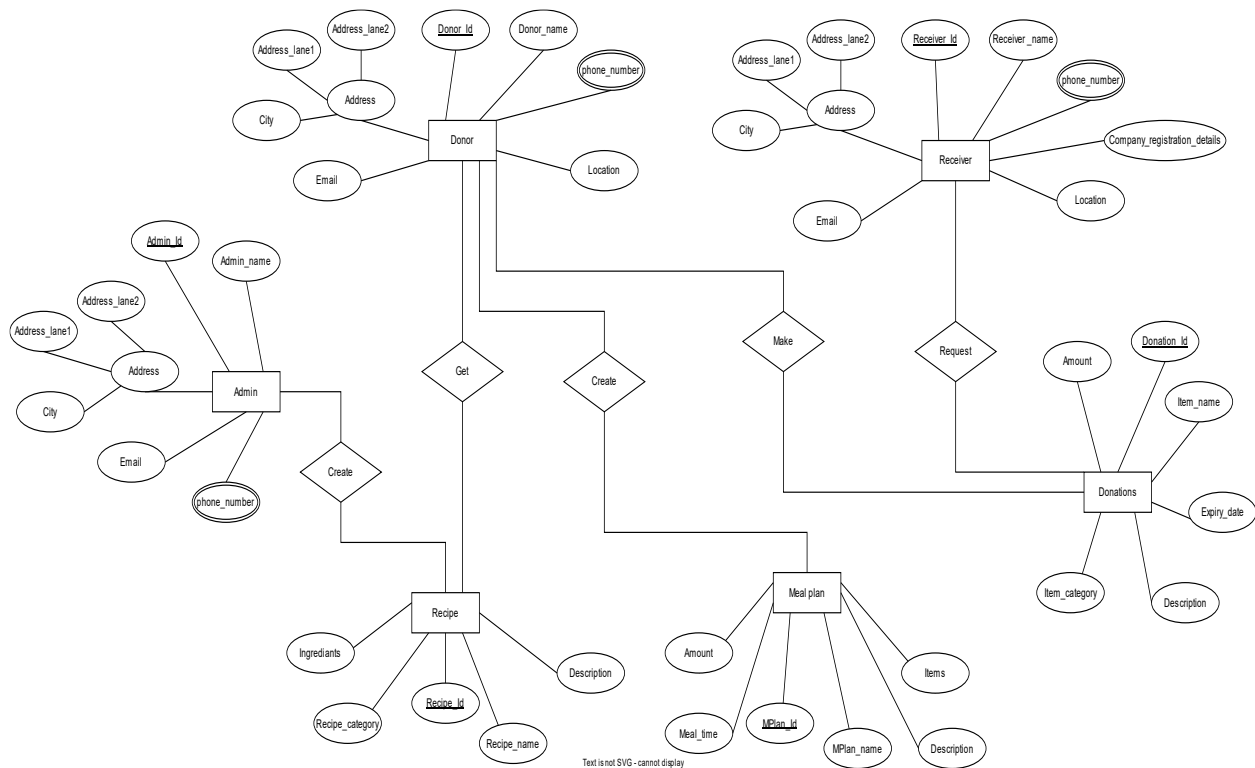


Figure 6 – High level ER Diagram

### Kulasekara D.A.M.N (IT21028878) – Donor management

When an individual, shop, or business enters the system as a donor, they must enter their name, telephone number, age, address, National Identity Number, and email address into the system. While entering the system as a donor, most sensitive data is not captured. It makes it easy for anyone to join the system and share their extra food.

But when posting food into the system, it is required to collect significant details. The reason is to ensure food quality. Food name, expiration date, current date, quantity, and storing method will be stored.

## Nipun P.G.I (IT21034404) – Recipe management

The data obtained here is not sensitive data. Common vegetables or fruits and spices required to prepare a recipe are entered as data. Although not sensitive data, this data should also be well protected.

When an admin creates a recipe admin needs to provide the list of ingredients needed to prepare the recipe, category of the recipe, recipe name and the necessary steps to make the recipe.

## Dewasurendra S. V (IT21039140) – Meal plan management

The data obtained here is not sensitive data. Although not sensitive data, this data should also be well protected.

## Wanniarachchi T. T (IT21071034) – Receiver management

Some sensitive data is required while registering in the system as a receiver. The reason for that is to prevent the misuse of food given by donors. The entities such as the name of the organization or committee, addresses, telephone numbers, names of trustees, registration numbers, and a copy of the registration certificate are required.

## 7. Performance requirements

### Kulasekara D.A.M.N (IT21028878)

**Response time:** The time it takes for the system to respond to a system user's request, should be in seconds or milliseconds.

**Concurrency:** The system should execute multiple tasks simultaneously without any blocking or conflict.

**Scalability:** The system should handle an increasing number of tasks without compromising performance.

**Extensibility:** The system should support additional features, functions, or integrations in the future without changing the main scope.

## Nipun P.G.I (IT21034404)

**Extensibility:** Without requiring significant changes, the system needs to be able to be expanded with new features or functionality.

**Adaptability:** The system must be flexible enough to accommodate new technologies or requirements that change.

**Standards compliance:** The system should strictly adhere to organizational or industry standards like ISO or IEEE.

**Maintainability:** The system's simplicity in terms of updating, repairing, or expanding.

## Dewasurendra S. V (IT21039140)

**Interoperability:** This term emphasizes a system's capacity to operate seamlessly with other systems or applications.

**Portability:** The ability of a system to be transferred between platforms or environments.

**Resilience:** This term describes the system's capacity to bounce back from mistakes or malfunctions, including backup and disaster recovery procedures.

**Testability:** This term describes how simple it is to test a system, including how simple it is to develop test cases and track results.

## Wanniarachchi T. T (IT21071034)

**Flexibility:** The system needs to be adaptable and able to change as the user's needs and specifications do.

**Robustness:** The program must be capable of coping with unforeseen inputs and mistakes without crashing or malfunctioning.

**Resilience:** The system needs to be able to bounce back from failures or interruptions.

**Auditable:** Detailed logs and recordings of all user interactions and system activities must be provided by the system, which must also be auditable.

## 8. Security requirements

### Kulasekara D.A.M.N (IT21028878)

- Based on a user's identity, role, or privileges, access control allows you to restrict who has access to sensitive information, systems, and networks.
- Before allowing access to resources, a person, device, or program must authenticate their identity.
- Depending on a user's identification, role, or rights, access to particular resources is granted or denied through the authorization process.
- Protecting private information from illegal access, disclosure, or change.
- Data integrity is the defense of data against illegal erasure, corruption, or change.

### Nipun P.G.I (IT21034404)

- Physical security refers to safeguarding tangible assets, such as servers, data centers, and other machinery, using techniques like access controls and security cameras.
- To detect, contain, and respond to security issues, procedures and processes are established as incident response.
- Monitoring detects unusual activities on networks and systems, such as unauthorized access or data espionage.
- Vulnerability management entails routinely checking apps and systems for vulnerabilities and applying patches or other countermeasures as necessary.
- Threat intelligence: Information gathered about possible security threats to an organization that is used to guide security choices and actions.

### Dewasurendra S. V (IT21039140)

- Controlling who has access to systems, apps, and data as well as how they can interact with them is done through access controls.
- Disaster recovery is the process of creating plans and methods, such as backup and restore procedures, for recovering from disasters or system failures.
- Making sure that essential business operations can continue in the case of a disruption, like a natural disaster or cyberattack.

- Risk management is the process of identifying and evaluating potential threats to the security of an organization and then taking action to reduce such threats.
- Privacy: Ensuring the security of sensitive and personal data, such as customer information, financial data, and medical records.

### Wanniarachchi T. T (IT21071034)

- User Education: Informing staff members and other users about recommended security procedures and potential risks.
- Establishing protocols for reporting security incidents, together with who to report them to and what information to include, is known as incident reporting.
- Testing disaster recovery processes to make sure they work as intended and can be rapidly put into action in the case of a disaster.
- Data backup: Frequently archiving important data to make sure it can be recovered in the case of corruption or loss.
- Least Privilege: Giving users the least amount of access required for them to carry out their duties in order to lessen the consequences of a security breach.

## 9. Testing requirements

### **Kulasekara D.A.M.N (IT21028878)**

- Functional requirements testing entails examining how well the features and functionalities of the software program adhere to the demands of the client.
- Usability testing entails evaluating the software application's usability and user-friendliness.
- Performance testing examines an application's capacity to process numerous users or transactions without experiencing any lag or slowdown.
- Testing the application's security features helps to guarantee that it is safe and secure from online threats.
- Testing for compatibility entails determining whether a program is compatible with various hardware and software setups.

### **Nipun P.G.I (IT21034404)**

- Legal requirements: Verify that the system conforms with all rules, regulations, and specifications that may be relevant.
- Test the system's ability to function well under stress or heavy loads.
- Test the system's ability to gracefully manage faults and exceptions so it doesn't crash or lose data.
- Response time: Verify that the system answers user queries promptly and without creating unneeded delays.
- Test the system's ability to maintain data integrity and ensure that no data is lost or corrupted.

### **Dewasurendra S. V (IT21039140)**

- Requirements for testability: Verify that the system is built in a way that makes testing and debugging easy.
- Test the system's installation and configuration to ensure that they go smoothly and without mistakes.
- Test the system's capacity to be relocated from one environment to another without any problems.
- Requirements for documentation: Verify that the system documentation is accurate, current, and comprehensive.

## Wanniarachchi T. T (IT21071034)

- Test that the system complies with accessibility standards and can be used by individuals with impairments.
- Test the system's capacity to manage growing loads and scale up as the user base expands.
- Test the system's interoperability with other systems and platforms to see if it can exchange data and communicate with them.
- Requirements for maintainability: Verify that the system is simple to update, modify, and maintain.
- Test the system's ability to recover from errors or crashes and restore data to its original condition.