

Group 5

Food waste management system

Software Design Document

Name (s) :

19Z305 Akshara P,
19Z317 Harini S,
19Z337 Pavithra
yazhini,
19Z343 Samyuktha
A S K,
19Z345 Saranya K
,19Z314 Gokul.

Lab Section:

Software Package
Development

Date: (26/09/2021)

TABLE OF CONTENTS

| | | |
|-----------|-------------------------------|-----------|
| 1. | INTRODUCTION | 2 |
| 1.1 | Purpose | 2 |
| 1.2 | Scope | 2 |
| 1.3 | Overview | 2 |
| 2. | SYSTEM OVERVIEW | 3 |
| 3. | SYSTEM ARCHITECTURE | 4 |
| 3.1 | Architectural Design | 4 |
| 3.2 | Decomposition Description | 5 |
| 3.3 | Design Rationale | 8 |
| 4. | DATA DESIGN | 9 |
| 4.1 | Data Description | 9 |
| 4.2 | Data Dictionary | 9 |
| 5. | COMPONENT DESIGN | 14 |
| 6. | HUMAN INTERFACE DESIGN | 19 |
| 6.1 | Overview of User Interface | 19 |
| 6.2 | Screen Images | 20 |
| 6.3 | Screen Objects and Actions | 24 |
| 7. | REQUIREMENTS MATRIX | 27 |

1. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of a food waste management system. The design document acts as an input for the development process. It helps the software developers to get an overview of the design of the food waste management.. In this system donors and NGOs can have their own account. From their account donors can request for food donation, admin will verify donors applications and approve them, now NGOs can accept the food. Donors and receivers will be notified during festival seasons. Donor who donates more food during a particular week will be honored as “Achiever of the week” and they will be certified if they donate for 500 people. This system works for the people who are in need of food.

1.2 Scope

Food Waste Management System will enable donors and recipients (NGOs) to request for the food take up and accept the food respectively. Recipients can also request for the food using this system. It is mainly designed to offer food for the needy people through NGOs and notify them when the donor is available.

Main modules of this system include:

Modules to manage the needs of this system

- Donors requesting food take-up

- Application approval by admin

- NGOs accepting food take-up

- Notifying donors and receivers during festival seasons

- Displaying “Achiever of the week” to motivate donors

- Certifying donors

Modules to manage user info

- Admin managing donor/recipient info

- Donor/recipient log in

- Recipient viewing food take-up requests

- Donor/recipient updating their profile if needed

1.3 Overview

An overview of the remaining sections is listed below

Section 2 provides a general description of the functionality, context and design of our project. It also provides any background information associated with our project.

Section 3 we have the architectural design that specifies the design entities that collaborate to perform all the functions included in the system. Each of these entities has an

abstract description concerning the services that it provides to the rest of the system. In turn, each design entity is expanded into a set of lower-level design operations that collaborate to perform its services. It also provides a decomposition of the subsystems in the architectural design. Information on the rationale for selecting the architecture described earlier including critical issues and trade-offs are considered.

Section 4 explains how the information domain of our system is transformed into data structures and how the major data or system entities are stored, processed and organized. A list of databases or data storage items are also included. An alphabetical list of the system entities or major data along with their types and descriptions are listed in this section.

Section 5 gives us a closer look at what each component does in a more systematic way.

Section 6 describes the functionality of the system from the user's perspective and explains how the user will use our system to complete all the expected features and the feedback information that will be displayed for the user. Description on screen objects and actions associated with those objects are mentioned.

Section 7 provides a cross-reference that traces components and data structures to the requirements in our SRS document.

2. SYSTEM OVERVIEW

Web application has been chosen for this project that runs on a web server, unlike computer-based software programs that are run locally on the operating system of the device. Web applications are accessed by the user through a web browser with an active network connection

Admin:

Administrator has full access to the system. He / She can view all the donors and receivers information. Administrator has the access verifying donor and receiver accounts so that they get a verified account.

Donor:

Donor are the people who donate food. Donors will request for food take-up. They also have the access for login, requesting for getting account verified, managing their information and logout.

Receiver:

Receivers are the people who check the app when they are in need of food. Receivers can accept the food take-up request by any donor. They also have access for login, requesting for getting account verified, managing their information and logout

3. SYSTEM ARCHITECTURE

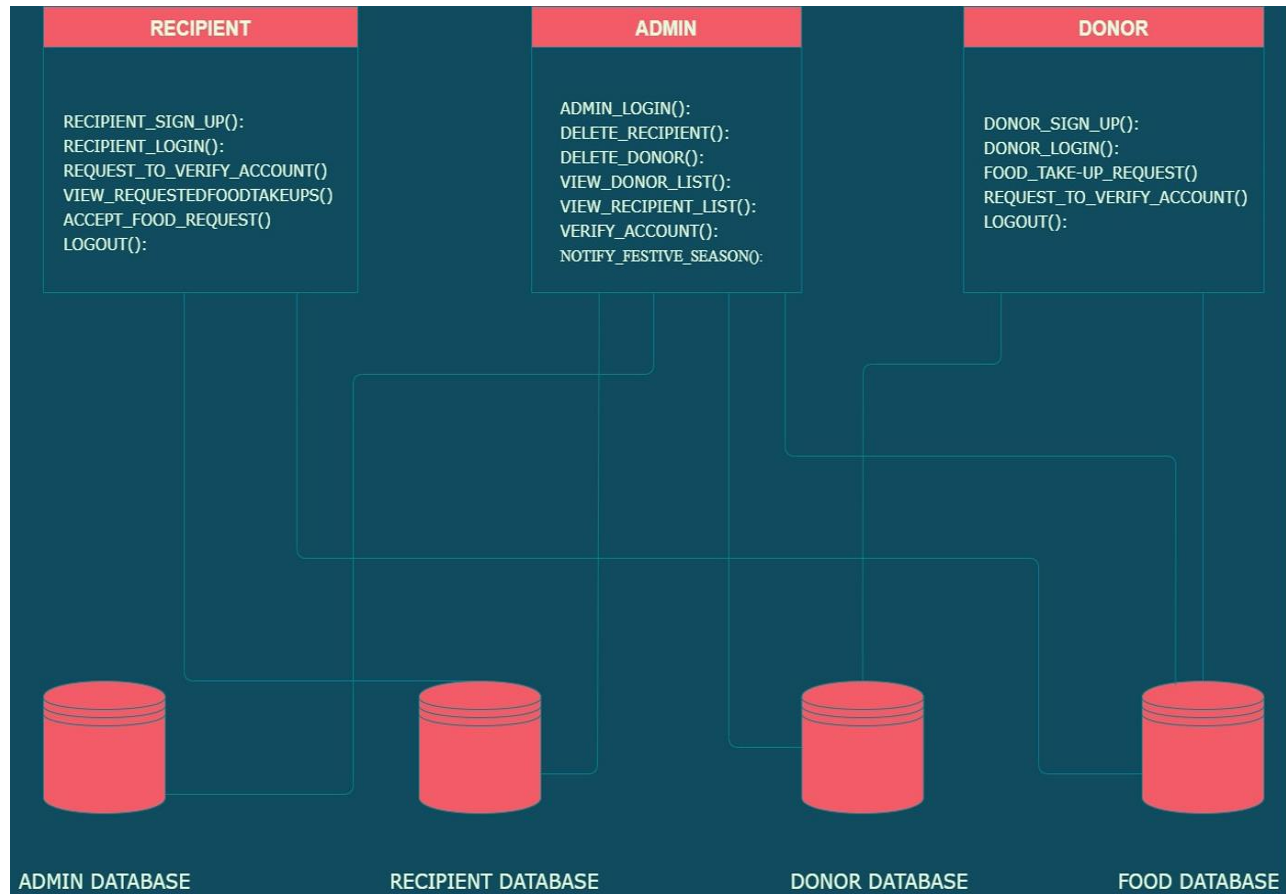
3.1 Architectural Design

This application consists of an Android application on the client side and PHPMySQL application on the server side.

The Android application is the part visible to the user and one it interacts with, while the PHP/MySQL-based server-side component serves as an interface between the Android application and the database on the server.

The Food waste management system uses 3 main functions or modules:

1. Admin module
2. Donar module
3. Receiver module



3.2 Decomposition Description

1. Admin Module:

This module is for both donor and receiver to access the website. In this module, according to the type of user, i.e., donor or receiver further sub-modules are called.

SUB-MODULES associated with ADMIN:

- (i) Admin login
- (ii) Delete recipient/donor
- (iii) View recipient/donor
- (iv) Verify account
- (v) Notify festive season

DATABASES associated with ADMIN:

- (i) Donor Database
- (ii) Recipient Database
- (iii) Admin Database

(iv) Food Database

2. Receiver Module:

Once the receiver logs in to the website, they can check for the date, time, availability of food, quantity and book the food. Separate sub-modules perform each of these operations.

SUB-MODULES associated with RECIPIENT:

- (i) Recipient sign up
- (ii) Recipient login
- (iii) Request to verify account
- (iv) view/ accept food takeups
- (v) Logout

DATABASES associated with RECIPIENT:

- (i) Recipient Database
- (ii) Food Database

3) Donor Module:

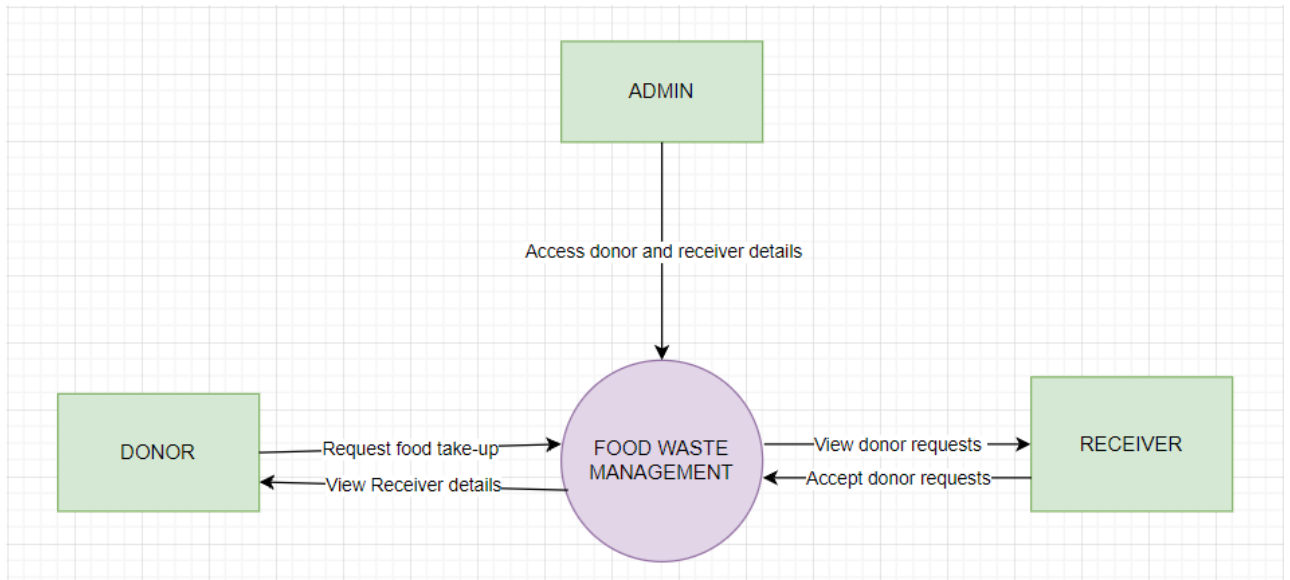
The various functions performed by admins like adding a new food with all its details, remove food once they are donated, changing the location and change the time and updating the information and maintaining the receiver's information are taken care of their respective sub-modules—

- (i) Donor sign up
- (ii) Donor login
- (iii) Food take up request
- (iv) Request to verify account
- (v) Logout

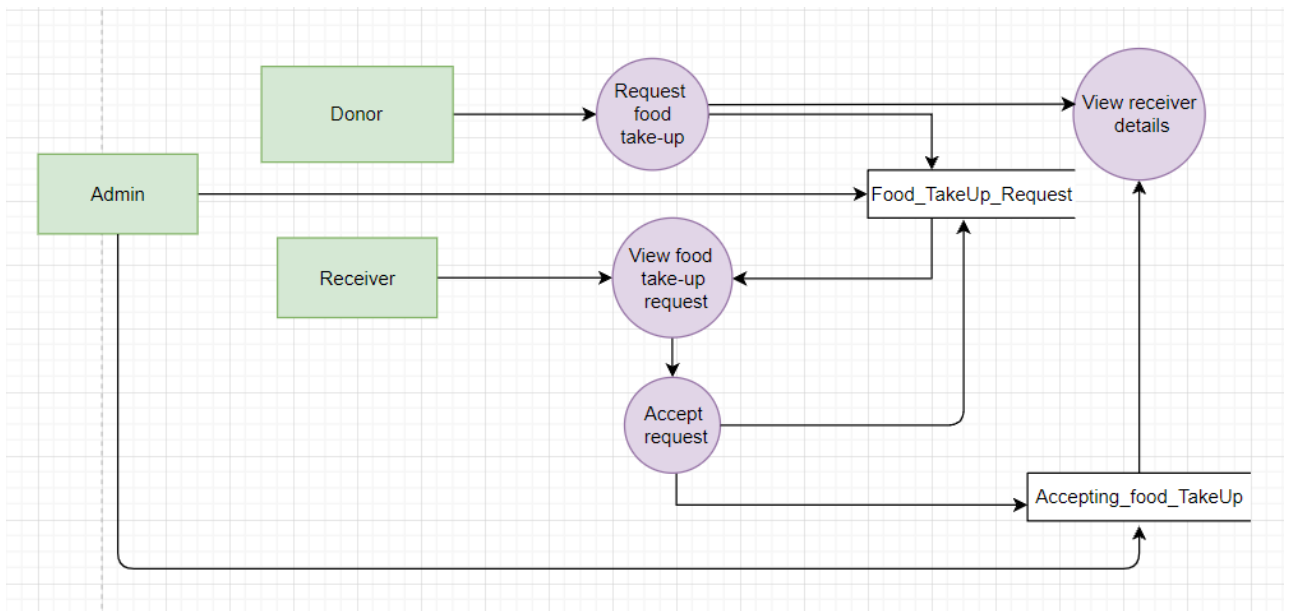
DATABASES associated with DONOR

- (i) Donor Database
- (ii) Food Database

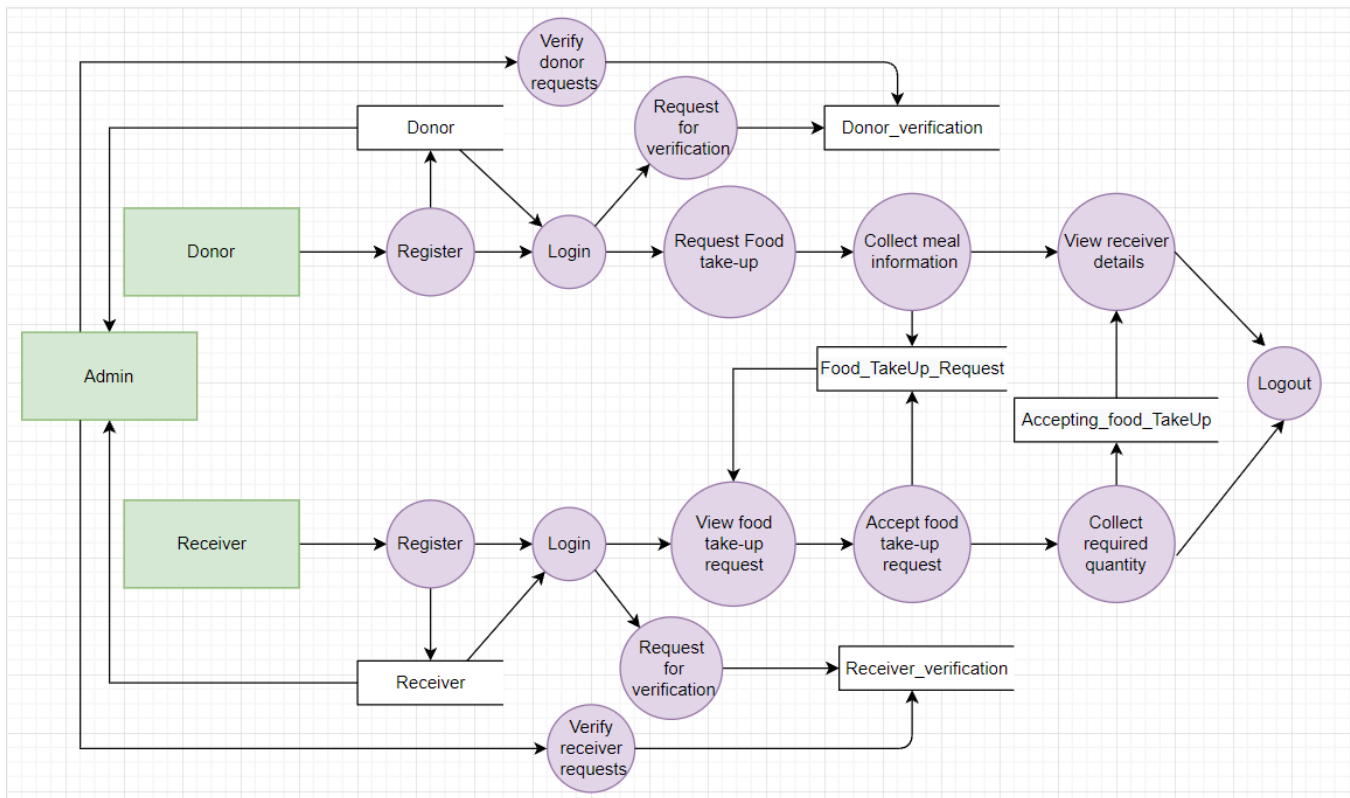
0-level Data Flow Diagram



1-level Data Flow Diagram



2-level Data Flow Diagram



3.3 Design Rationale

We use call and return architecture for the food waste management system as it supports system modifiability, scalability and performance. Under this we are working with the sub-style –

Main program or Subprogram architectures. The main program structure decomposes into a number of subprograms or functions into a control hierarchy. Main program contains a number of subprograms that can invoke other components.

Some critical issues that we faced are –

1. Difficulty in parallel processing
2. Difficulty to distribute across traditional machines.
3. Exceptions to normal operation are awkward to handle.

4. DATA DESIGN

4.1 Data Description

Data flow diagram is used to explain the entities and its relationships with other entities. MySQL database can be used to store a database of food waste management systems.

4.2 Data Dictionary

Donor : Contains general information of a donor

| Data | Type | Description |
|------------|-------------|------------------------------------|
| d_username | varchar(10) | Unique key to identify each donor |
| d_name | varchar(20) | Donor name for log into the system |
| d_phone | Int(10) | Contact number of donor |
| d_addr | varchar(70) | Email of donor |
| d_email | varchar(30) | Address of donor |
| d_password | varchar(30) | Password of donor |

Receiver : Contains general information of receiver

| Data | Type | Description |
|-------------|-------------|---------------------------------------|
| r_username | varchar(10) | Unique key to identify each receiver |
| r_name | varchar(20) | receiver name for log into the system |
| r_phone | Int(10) | Contact number of receiver |
| r_addr | varchar(70) | Email of receiver |
| r_email | varchar(30) | Address of receiver |
| r_password | varchar(30) | Password of receiver |

Admin : Contains general information of Admin

| Data | Type | Description |
|-------------|-------------|------------------------------------|
| ad_username | varchar(10) | Unique key to identify admin |
| ad_name | varchar(20) | admin name for log into the system |
| ad_phone | Int(10) | Contact number of admin |
| ad_password | varchar(30) | password of admin |

| | | |
|----------|-------------|----------------|
| ad_email | varchar(70) | email of admin |
|----------|-------------|----------------|

Food_TakeUp_request : Donors requesting for food take up

| Data | Type | Description |
|---------------------|-------------|---|
| d_username | Varchar(10) | Unique key to identify donor |
| d_addr | Varchar(70) | Donor address |
| d_phone | Int(10) | Contact number of donor |
| req_id | Int(10) | Id for food request |
| no_of_people_served | Int(500) | Indicates total number of people to be served |
| items | varchar(50) | Items available |
| food_takeup_addr | Varchar(80) | Address of food take up |
| f_validity | Int | Food validity in hours |

Accepting_food_TakeUp: Receivers accepting food take up

| Data | Type | Description |
|-------------|-------------|---------------------------------|
| r_username | Varchar(10) | Unique key to identify receiver |
| r_name | Varchar(10) | Receiver name |
| r_addr | Varchar(70) | Address of receiver |
| d_username | Varchar(10) | Unique key to identify donor |
| req_id | Int(10) | Id for food request |

Donor_Verification : Admin verifying the donor's request

| Data | Type | Description |
|--------------|-------------|---|
| req_id | Int(10) | Id for food request |
| Ver_req_type | Varchar(20) | Request from donor |
| d_username | Varchar(10) | Donor username |
| status | Varchar(10) | Status shows Whether the request accepted or received |

Receiver_Verification : Admin verifying the receiver's request

| Data | Type | Description |
|--------------|-------------|---|
| req_id | Int(10) | Id for food request |
| Ver_req_type | Varchar(20) | Request from receiver |
| r_username | Varchar(10) | receiver username |
| status | Varchar(10) | Status shows Whether the request accepted or received |

5. COMPONENT DESIGN

In this section, we take a closer look at what each component does in a more systematic way. If

1.FOOD_TAKE-UP_REQUEST()

Redirects to a page to enter meal information

Get meal quantity,description

IF(“Post” is clicked)

Store the food take-up request

Return to previous page

ELSE IF(“Discard” is clicked)

Return to previous page

2.ACCEPT_FOOD_REQUEST()

Display the requests from donors

Recognize the request accepted by receiver

Pop up a window to enter quantity

IF(“OK” is clicked)

IF(entered quantity=donated quantity)

Remove the food take-up request from database

ELSE

donated quantity=donated quantity-entered quantity

Close the pop up window

ELSE IF(“Discard” is clicked)

Close the pop up window

3.NOTIFY_FESTIVE_SEASON()

Pop window appears to notify the festive season.

4.REQUEST_TO_VERIFY_ACCOUNT()

Redirects to a page to enter the details of donor/receiver

IF(“Submit” is clicked)

Data is stored in database for verification

Return to previous page

ELSE IF(“Discard” is clicked)

Return to previous page

5.VERIFY_ACCOUNT()

Redirects to a page which specifies the list of verify requests

Display the application which is clicked
 IF("Accept" is clicked)
 The account is updated as verified
 Return to previous page
 ELSE IF("Reject" is clicked)
 The verification request is removed
 Return to previous page

6.VIEW_REQUESTEDFOODTAKEUPS()

Get the food take up requests from system database
 Display the list of requests to user

7.ADMIN/DONOR/RECIPIENT LOGIN()

Get the user privilege(Admin/Donor/Recipient),username and password from user.

IF user = "Admin" THEN

IF username exists in admin records and password matches THEN

 Display "Successfully logged in"

 Return to user's dashboard.

ELSE IF user = "Donor" THEN

IF username exists in donor records and password matches THEN

 Display "Successfully logged in"

 Return to user's dashboard.

ELSE IF user = "Recipient" THEN

IF username exists in recipient records and password matches THEN

 Display "Successfully logged in"

 Return to user's dashboard.

ELSE

Display "Wrong details !! try again".

Return to login page.

END IF

8.DONOR/RECIPIENT UPDATE_PROFILE()

Get current user details from the system database.

Display the current user details to the user.

Get the new user details from the user.

Update the user's record in the database.

IF update successful THEN

Display "Updated Successfully".

Return to user's dashboard.

ELSE

Display "Can't update details !! try again".

Return to edit profile page.

9.DONOR/RECIPIENT SIGN_UP()

Get the appropriate details from the user.

Check if the constraints for the details hold.

IF constraints satisfied THEN

Create a user(Donor/Recipient) record in the database.

Display "Successfully signed up"

Return to the donor/recipient login page.

ELSE

Display "re-enter the details".

Return to the donor/recipient sign up page.

10.LOGOUT()

Return to the home page.

11.VIEW DONOR/RECIPIENT LIST()

Ask the user(Admin) whether he/she wants to see the list of donors/recipients.

IF input = "Donor" THEN

 Get the list of registered donors from the system database.

 Display the list to the admin.

ELSE IF input = "Recipient" THEN

 Get the list of registered recipients from the system database.

 Display the list to the admin.

ELSE

 Display "Wrong choice !!".

 Return to user's dashboard.

END IF

12.DELETE DONOR/RECIPIENT()

Ask the user(Admin) whether he/she wants to delete the record of a donor/recipient.

IF input = "Donor" THEN

 Get the donor's username from admin.

 IF username exists in donor records THEN

 Delete the corresponding donor's record.

 Display "Record successfully deleted"

 ELSE

 Display "no donor record found"

 END IF

ELSE IF input = "recipient" THEN

 Get the donor's username from admin.

 IF username exists in donor records THEN

```
        Delete the corresponding donor's record.  
        Display "Record successfully deleted"  
    ELSE  
        Display "no recipient record found"  
    END IF  
ELSE  
    Display "wrong choice!"  
    Return to the user's dashboard.  
END IF
```

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The main users of the food waste management system will be administrator, donor and recipient. The system is designed in such a way that it is user friendly.

Admin Interfaces

- Log in
- Viewing Donor
- Viewing Recipient
- Accepting Verification Request

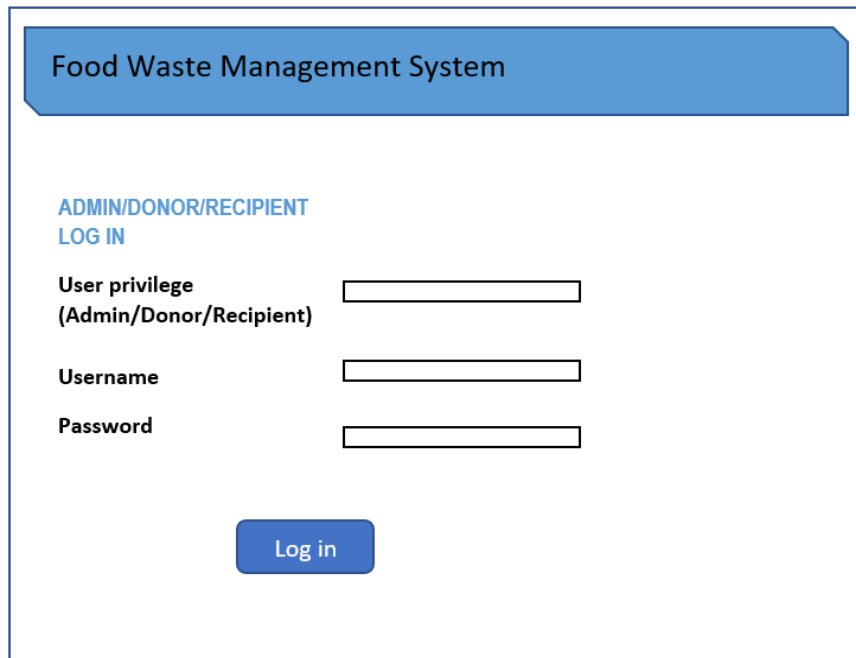
Donor Interfaces

- Sign up
- Log in
- Editing Profile
- Requesting food take-up

Recipient Interfaces

- Sign up
- Log in
- Editing Profile
- Accepting food take-up

6.2 Screen Images



Food Waste Management System

**ADMIN/DONOR/RECIPIENT
LOG IN**

User privilege (Admin/Donor/Recipient)

Username

Password

Log in

Figure 6.2.1: Login page



Food Waste Management System

ADMIN VIEWING DONOR TABLE

| Username | Donor Name | Contact No. | E-mail Id | Address |
|----------|------------|-------------|-------------------|---------------------------|
| Keerthi | Keerthiga | 9784356470 | keerthi@gmail.com | Rice mill road Erode |
| Sanju | Sanjith | 8765438021 | sanju@gmail.com | K.K street Salem |
| Hari | Haritha | 7645628901 | hari@gmail.com | Pillaiyar street Namakkal |

Figure 6.2.2: Admin viewing donors page

| Food Waste Management System | | | | |
|-------------------------------|----------------|-------------|-----------------|---------------------------|
| ADMIN VIEWING RECIPIENT TABLE | | | | |
| Username | Recipient Name | Contact No. | E-mail Id | Address |
| Krish | Krishiv | 674536890 | krish@gmail.com | Gandhi nagar,Karur |
| Pavi | Pavithra | 7899034523 | pavi@gmail.com | Saibaba colony,Coimbatore |
| Abi | Abishek | 7890123456 | abi@gmail.com | Peelamedu,Coimbatore |

Figure 6.2.3: Admin viewing recipients page

| Food Waste Management System | |
|--|----------------|
| ADMIN ACCEPTING VERIFICATION REQUEST | |
| Donor/Recipient Name | Pavithra |
| Username | Pavi |
| Contact No. | 7899034523 |
| E-mail Id | pavi@gmail.com |
| <input type="button" value="Accept"/> <input type="button" value="Decline"/> | |

Figure 6.2.4: Verification request view page

Food Waste Management System

**DONOR/RECIPIENT
SIGN UP**

User type
(Donor/Recipient)

Name

Username

Password

Address

Phone no

Email - id

Sign up/Register

Figure 6.2.5: Sign up page

Food Waste Management System

**DONOR REQUESTING
FOOD TAKE UP**

No of people to be
served

Food take up request
Address

Food validity (in hrs)

Items available

Request

Figure 6.2.6: Food take up requesting page

Food Waste Management System

**DONOR/RECIPIENT
EDITING PROFILE**

| | |
|-------------------|----------------------------------|
| Name | Siva kumar G.P |
| Password | |
| Address | Bharathi nagar, Chennai - 638001 |
| Phone no | 97564 12688 |
| Email - id | sivasiva@gmail.com |

Update Profile

Figure 6.2.7: Edit profile page

Food Waste Management System

RECIPIENT ACCEPTING FOOD TAKE UP

| | |
|--|--|
| No of people to be served | <input style="width: 90%;" type="text" value="20"/> |
| Food <u>take up</u> request Address | <input style="width: 90%;" type="text" value="MGR <u>nagar</u>,<u>Erode</u> - 638638."/> |
| <u>Food validity</u> (in hrs) | <input style="width: 90%;" type="text" value="4 hrs"/> |
| Items available | <input style="width: 90%;" type="text" value="Idly,sambar,rice."/> |
| Donor username | <input style="width: 90%;" type="text" value="Keerthi"/> |
| Donor name | <input style="width: 90%;" type="text" value="Keerthiga"/> |
| Phone no | <input style="width: 90%;" type="text" value="99543 78016"/> |
| Email - id | <input style="width: 90%;" type="text" value="keerthi021@gmail.com"/> |

Figure 6.2.8: Food take up request accepting page

6.3 Screen Objects and Actions

6.3.1 Log in

- **User privilege**

The user must enter whether he/she is an admin, donor or recipient.

- **Username**

- Username should contain 1 to 50 characters.
- It should be unique.
- It must start with a letter.

- **Password**

It must contain at least 8 characters. Passwords are allowed only when it contains an uppercase letter, a lowercase letter and a digit.

6.3.2 Admin viewing donor table

In this screen admin can view the donor table which contains the details of donor such as,

- Username
- Donor name
- Contact No.
- E-mail Id
- Address

6.3.3 Admin viewing recipient table

In this screen admin can view the recipient table which contains the details of recipient such as,

- Username
- Recipient name
- Contact No.
- E-mail Id
- Address

6.3.4 Admin Accepting Verification Request

In this screen admin can accept the verification request send by donors and recipients.

Here the admin can see details of each user(donor/recipient). The details are,

- Donor/Recipient name
- Username
- Contact No.
- E-mail Id

After seeing all these details of the donor/recipient, the admin can accept the request by clicking the 'accept' button or reject by clicking 'decline' button.

6.3.5 Donor/Recipient Sign up

Sign up is only for the donor and recipient. In this screen the user have to fill details such as,

- **User type**
 - User must enter whether he/she is a donor or recipient

- **Name**
 - It should be 1 to 50 characters
 - User full name.
- **Password**
 - At least 8 characters
 - It must contain an uppercase letter, a lowercase letter and a digit
- **Address**
 - User address
- **Phone no**
 - User contact number
- **E-mail Id**
 - User e-mail id

After filling all this details the user must click ‘Sign up/Register’ button in the screen

6.3.6 Donor Requesting Food Take up

In this screen the donors request the recipients to take up the food through this system.

The details donors have to fill are,

- **No of people to be served**
 - Enter how many people can be served
- **Food take up request Address**
 - The location where food is available
- **Food Validity (in hrs)**
 - Donor must specify the time, how long the food will be available from the time of requesting
- **Items available**
 - Food items available to donate

At last the donor must click the 'Request' button in the screen

6.3.7 Donor/Recipient editing Profile

In this screen donor/recipient can edit their profile details.

The details that can be edited are,

- Name
- Password
- Address
- Phone no
- E-mail id

After editing the new changes the user must click 'Update Profile' button in the screen

6.3.8 Recipient Accepting Food Take up

In this screen the recipients can accept the food take up request by donors.

The details that are available in this screen for recipient are,

- **No of people can be served**
 - how many people to be served
- **Food take up request Address**
 - The location where food is available
- **Food Validity (in hrs)**
 - From the time of sending request how long the food is available
- **Items available**
 - Food items available

After seeing all this details if the recipient wants to accept the request, then he/she must click 'accept' button

7. REQUIREMENTS MATRIX

| S.No | Function Name | Design Reference | Applicable Roles | Description |
|------|-----------------------------|------------------|------------------|---|
| 1 | FOOD_TAKE-UP_REQUEST() | 4.1 | Donor | The registered donors can post the availability of food in their place with the help of this feature. |
| 2 | ACCEPT_FOOD_REQUEST() | 4.2 | Receiver | The receivers can log in into their account and accept the food take-up request by the donors |
| 3 | NOTIFY_FESTIVE_SEASON() | 4.3 | Admin | The receivers are notified during the festival season that there may be an increase in food take-up requests. |
| 4 | REQUEST_TO_VERIFY_ACCOUNT() | 4.4 | Donor/Receiver | The donors and receivers can request the admin to get a verified account |
| 5 | VERIFY_ACCOUNT() | 4.5 | Admin | The admin can view and verify the donor and receiver application requesting for verified account |

| | | | | |
|---|-------------------------------------|-----|----------------|---|
| 6 | VIEW_REQUESTED FOODTAKEUPS() | 4.6 | Receiver | The food recipient(NGOs) after successfully logging in, can view the list of food take-ups requested by the donors(restraunts).The recipients can accept any request from the list displayed. |
| 7 | ADMIN/DONOR/ RECIPIENT LOGIN() | 4.7 | Donor/Receiver | The registered donor/recipient visiting the site can log in to the system to get access to some user specific system features. |
| 8 | DONOR/RECIPIENT UPDATE_PROFILE() | 4.8 | Donor/Receiver | The donors and recipients can update their profile. |
| 9 | DONOR/RECIPIENT SIGN_UP() | 4.9 | Donor/Receiver | For the donor and recipient to have the privileges of a registered user,the donor and recipient must be able to sign up to the system. |

| | | | | |
|----|--|------|----------------|---|
| 10 | VIEW_DONOR/ RECIPIENT LIST(), .DELETE DONOR/RECIPIENT() | 4.10 | Admin | The admin of the system can view the list of donors and recipients of the system and can remove any registered user from the system |
| 11 | LOGOUT() | 4.11 | Donor/Receiver | After using the intended features,the users (Admin/Donor/Recipient) can logout from the system. |

