

Software Design Specification

for

Food Bank Distribution App

Version <2.0>

Group No.: 15

Student Name	Student ID
Avinash Immanuel A/L Gana Raj Immanuel	1181103370
Kishen Kumar A/L Sivalingam	1191101423
Narvenesh A/L Gevaratnam	1181103203
SHEIKH NAIMULLAH BIN SH MOHD AMINLLAH	1201301338

Date : 18/10/2021



CONTENTS	2
REVISIONS (CHANGE LOG)	4
1 SYSTEM OVERVIEW	5
1.1 DESCRIPTION	5
1.2 ACTORS	6
1.3 ASSUMPTIONS AND DEPENDENCIES	8
1.4 USE CASE DIAGRAM	8
2 DATA DESIGN	10
2.1 DATA DICTIONARY	10
3 ARCHITECTURE DESIGN	12
3.1 SOFTWARE ARCHITECTURE	12
4 INTERFACE DESIGN	17
4.1 MAIN SCREENS	17
4.1A REGISTER PAGE (ADMIN, DISTRIBUTOR, DONOR, RECIPIENT)	17
4.1B LOGIN PAGE (ADMIN, DISTRIBUTOR, DONOR, RECIPIENT)	19
4.1C EDIT PROFILE PAGE (ADMIN, DISTRIBUTOR, DONOR, RECIPIENT)	22
4.1D HOME PAGE (DONOR)	24
4.2 DONOR	25
4.2A VIEW FOODS LOW ON STOCK (DONOR)	25
4.2B VIEW HISTORY OF DONATIONS (DONOR)	26
4.2C DONATE MONEY (DONOR)	27
4.2D DONATE FOOD (DONOR)	28
4.2E FIND FOOD BANK LOCATIONS (DONOR)	29
4.3 ADMIN	30
4.3A ADD USER (ADMIN)	30
4.3B FIND USER (ADMIN)	31
4.3C EDIT USER (ADMIN)	32
4.3D ADD FOOD BANK REPORT (ADMIN)	33
4.3E FIND FOOD BANK REPORT (ADMIN)	34
4.3F EDIT FOOD BANK REPORT (ADMIN)	35
4.3G ADD FOOD BANK (ADMIN)	36
4.3H FIND FOOD BANK (ADMIN)	37
4.3I EDIT FOOD BANK (ADMIN)	38
4.4 RECIPIENT	39

4.4A	VIEW FOOD MENU (RECIPIENT)	39
4.4B	RESERVE FOOD (RECIPIENT)	40
4.4C	CANCEL RESERVATION OF FOOD (RECIPIENT)	41
4.4D	VIEW COMMUNITY (RECIPIENT)	42
4.4E	VIEW NEWS (RECIPIENT)	43
4.4F	ACQUIRE ASSISTANCE (RECIPIENT)	44
4.5	DISTRIBUTOR	
4.5A	SEARCH FOR NEARBY FOODBANK LOCATIONS (DISTRIBUTOR)	45
4.5B	SELECT STATUS OF RESERVED FOOD (DISTRIBUTOR)	46
4.5C	VIEW DISTRIBUTION STATUS (DISTRIBUTOR)	47
5	COMPONENT DESIGN	48
5.1	RECIPIENT	49
5.2	DONOR	73
5.3	DISTRIBUTOR	90
5.4	ADMIN	106
6	DEPLOYMENT DESIGN	122
6.1	DEPLOYMENT DIAGRAM	122
7	UPDATED REQUIREMENTS	123
7.1	SEQUENCE DIAGRAM 1	123
7.2	SEQUENCE DIAGRAM 2	123



Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Ver 2.0	Avinash Immanuel A/L Gana Raj Immanuel Kishen Kumar A/L Sivalingam Narvenesh A/L Gevaratnam SHEIKH NAIMULLAH BIN SH MOHD AMINLLAH	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	18/10/21

1 System Overview

1.1 Description

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, will be effective.

TO DO: Describe the major processes to be performed by the system and the

.>

Donor

The system will allow the donor to login by keying in the donorID and password. If the login information matches, the donor is shown the home page and certain features which they can choose to access. If a profile is selected, the donor can check information related to the donor and choose to edit them. Moreover, by selecting the donate feature, the donor can donate food or money to the respective food bank directly from the home page. Also, if the donor decides to see the donations they made to the respective food bank they can choose the view history function. Furthermore, the donor can check for foods that are urgently needed through the view food low on stock feature. Find food bank location feature, allows donors to locate the nearest food bank.

Admin

Admin need to login with their AdminID and password in order to access the system with high privileges. Upon login, admin users are present into a main menu page to navigate said privileges. Privileges included in this system are adding, deleting, editing and finding the following data: Users (such as donor, recipients, and distributor), food bank database or reports (fields included are food category, distribution location, and amount), distribution location, and food categories. To control the above data is through their own respected pages with a simple user interface.

Recipient

The system will allow the recipient to login by entering the username and password. After login success, the recipient will have access to certain features which can be displayed in the home page of the app. One of them is viewing the profile in order to check his/her personal details which will be shown in a personal detail form. Then, the recipient also has the option to edit the profile if he/she desires to correct inaccurate profile details. Not only that, the recipient also can view all the foods and the information regarding it which are available at the specific period of time. After seeing all the food details, the recipient is able to reserve the food which he/she wants to eat. If the recipient does not want the food that he/she reserved, he/she can cancel the reservation. Furthermore, additional features were built up for the recipient such as 'community feed' function where the recipient will be notified regularly with the food waste, sharing tips and more and also the 'contact us' feature where the recipient is able to get help from the administration department of the app by receiving replies for the inquiry messages that the recipient sent.

Distributor

Distributors have to login with their *riderID* and password in order to access the application. If the login information matches, the distributor is shown the homepage and certain features which they can choose to access. When a profile is selected, the distributor can review and modify information about the distributor. Search for nearby food bank locations feature, allows the distributor to locate the nearest foodbank. After this process, the system will notify the distributor on the nearest food bank location. The distributor can access the check status of reserved food feature to identify the availability, type and quantity of food they are assigned to deliver. The distributor can then proceed to accept the delivery by selecting the pick up food feature. During the delivery process, the distributor can check the status of their delivery by selecting the view delivery status option. After the food has been successfully delivered to the recipient, the distributor has to select the drop off food function to complete their delivery task.

1.2 Actors

Actor	Use Cases
DONOR	View profile Edit profile Donate food Donate money View history of donations View foods low on stock Find food bank locations Sign Up / Log In
RECIPIENT	Sign Up Login View Profile Edit Profile View Food Menu Reserve food Cancel Reservation of Food

	View Community Feed
	Acquire Assistance

DISTRIBUTOR	Sign up as rider
	Login
	Edit profile
	View profile
	Search for nearby foodbank locations
	Check status of reserved food
	Pick up food
	View delivery status
	Drop-off food
ADMIN	Login
	Add Users
	Add Food bank report
	Add Distribution locations
	Add Food Categories
	Delete Users
	Delete Food bank report
	Delete Distribution locations
	Delete Food Categories
	Edit Users
	Edit Food bank report
	Edit Distribution locations

Edit Food Categories
Find Users
Find Food bank report
Find Distribution locations
Find Food Categories

1.3 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

TO DO: Provide a short list of some major assumptions that might significantly affect your design. For example, you can assume that your client will have 1, 2 or at most 50 Automated Banking Machines. Every number has a significant effect on the design of your system. >

1.4 Use Case Diagram

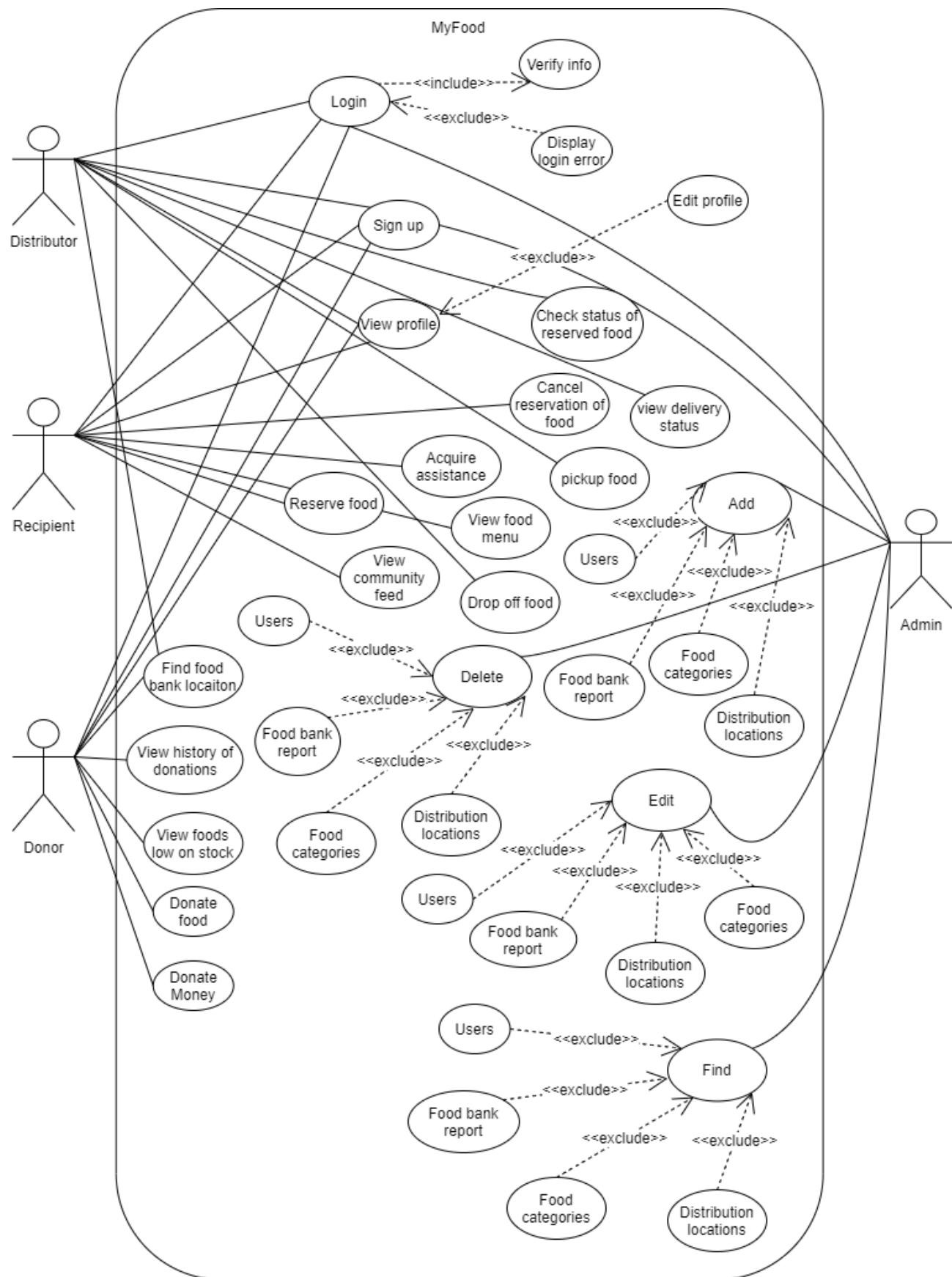


Figure 1.4 Use Case Diagram for all Users

2 Data Design

2.1 Data Dictionary

TABLE NAME	ATTRIBUTE NAME	CONTENTS	DATA TYPE	FIELD SIZE
User	- userID - password - fName - lName - telPhone - loginStatus - signUpStatus - registerDate	- User identity number - User account password - User first name - User last name - User telephone number - User login status - User sign up status - User account register date	- Char - String - Varchar - Varchar - Char - String - String - Date	- 10 - 20 - 20 - 20 - 10 - 10 - 10 - 10
Admin	- adminID	- Admin identity number	- Char	- 10
Recipient	- dateOfBirth - gender - nationality - address - email	- Date of birth - Gender - Nationality - Address - Email address	- Date - Char - String - Varchar - Varchar	- 10 - 5 - 20 - 100 - 30
Donor	- donorID - email - state - transaction - food	- Donor identity number - Email address - State - Transaction - Food	- Char - Varchar - String - Double - Varchar	- 10 - 30 - 10 - 10 - 20
Distributor	- riderID - dateOfBirth - gender - nationality - email - address - state	- Rider identity number - Date of birth - Gender - Nationality - Email address - Address - State	- Char - Date - Char - String - Varchar - Varchar - String	- 10 - 10 - 5 - 20 - 30 - 100 - 10
Foodbank	- name - state - donors	- Foodbank name - State - Donors	- Varchar - String - String	- 30 - 10 - 10

Menu	- foodName - dateReserved - numFoodAvailable - numFoodReserved - deliveryTime - foodType	- Food name - Date when food reserved - Number of food available - Number of food reserved - Delivery time - Food type	- Varchar - Date - Int - Int - Char - String	- 20 - 10 - 5 - 5 - 10 - 10
------	---	---	---	--

3 Architecture Design

3.1 Software Architecture

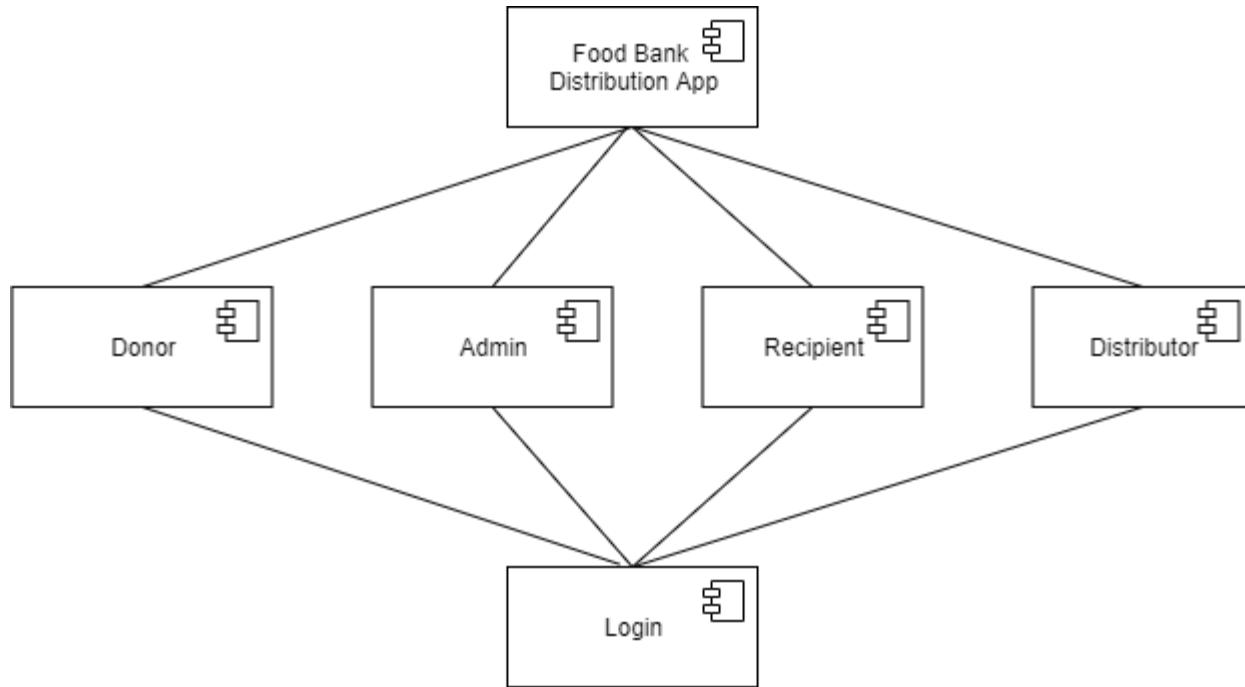


Figure 3.1 Food Bank Distribution Application's Software Architecture Diagram

3.1.1 Recipient subsystem

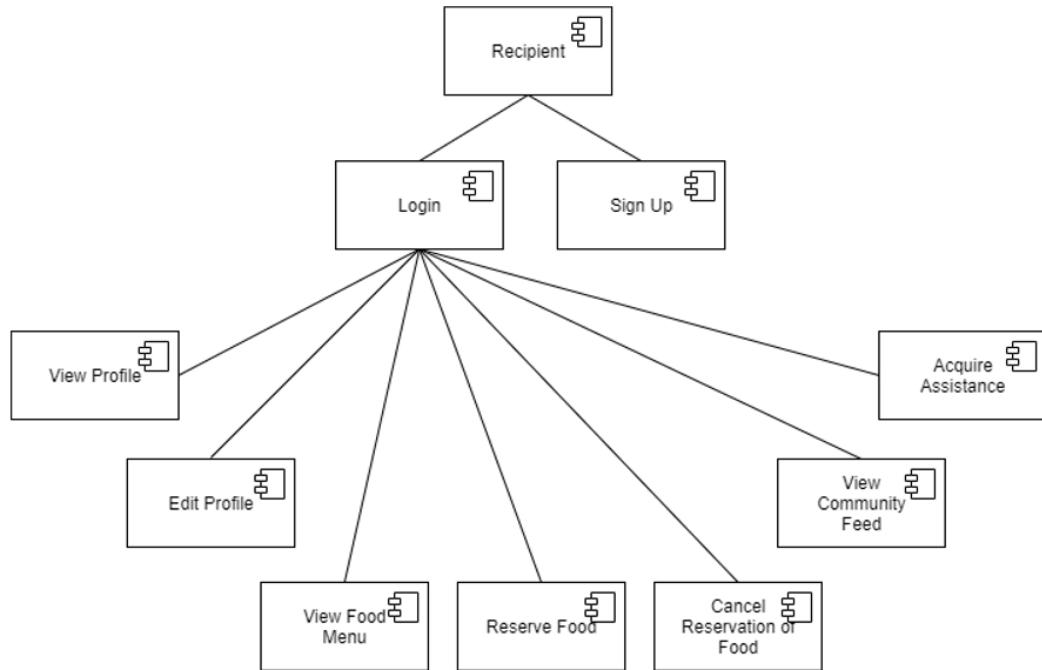


Figure 3.1.1 Recipient's Software Architecture Diagram

3.1.2 Distributor Subsystem

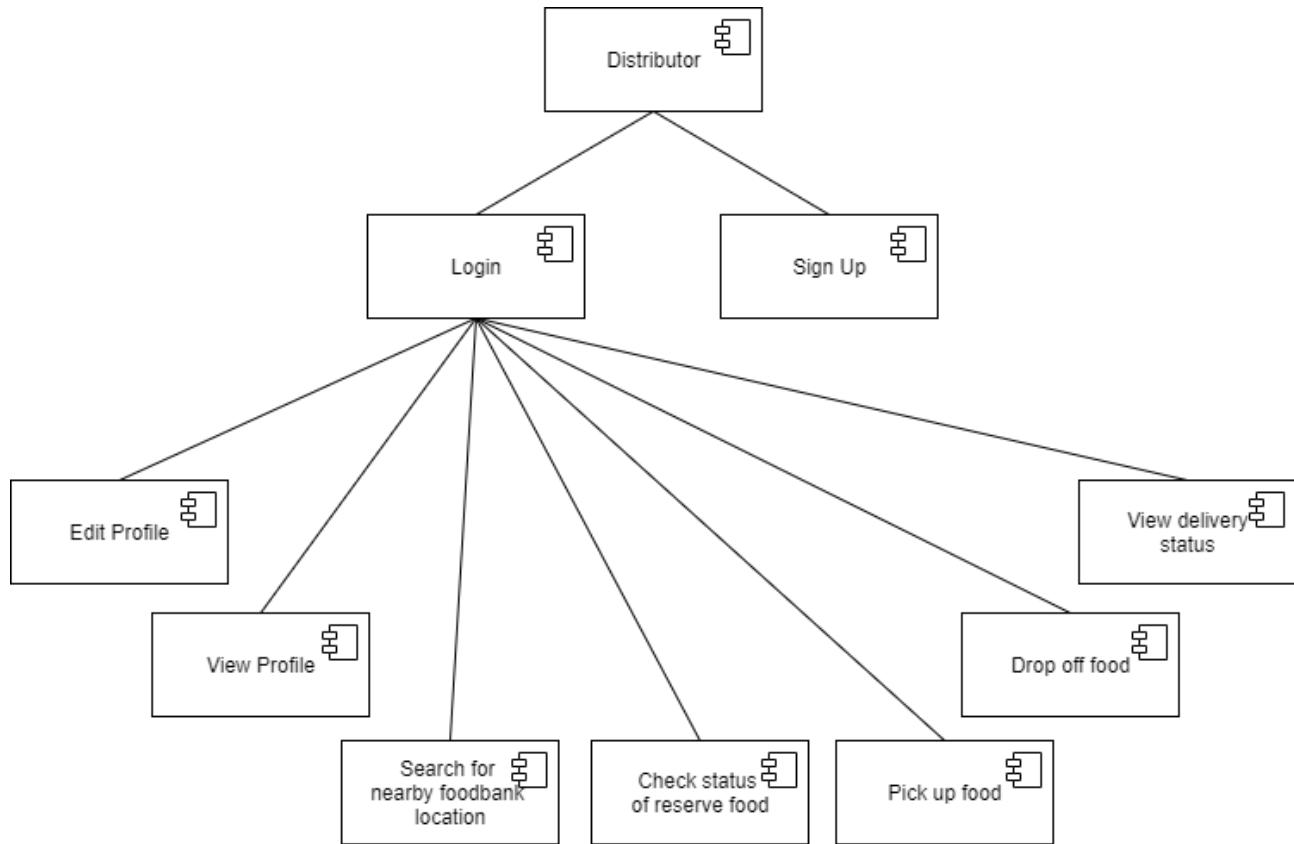


Figure 3.1.2 Distributor's Software Architecture Diagram

3.1.3 Donor Subsystem

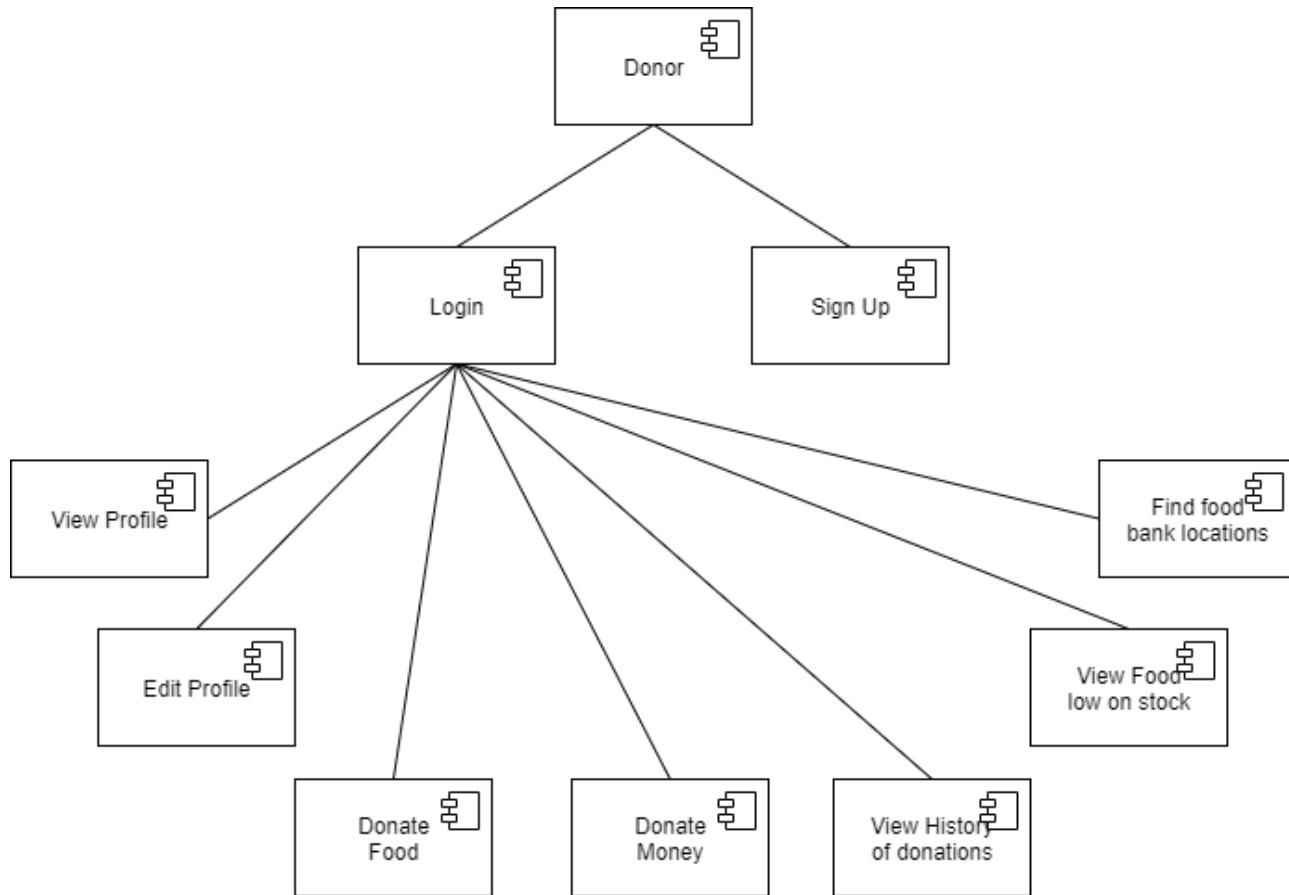


Figure 3.1.3 Donor's Software Architecture Diagram

3.1.4 Admin Subsystem

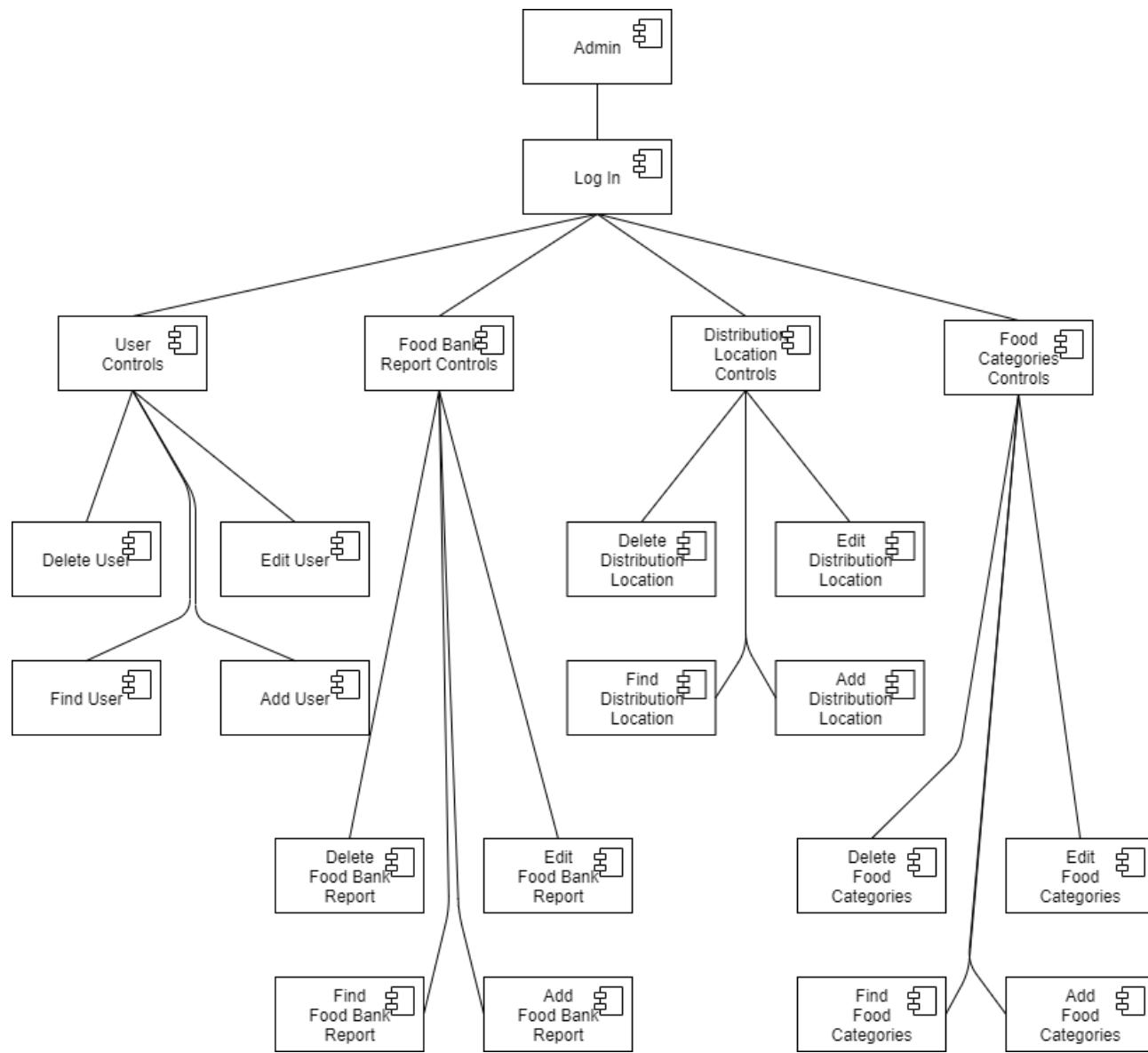


Figure 3.1.4 Admin's Software Architecture Diagram

4 Interface Design

4.1 Main Screens

4.1a Register Page (Admin, Distributor, Donor, Recipient)



REGISTER

Please enter the details below

First Name

Last Name

Mobile Number

UserID

Password

Birth Date

Gender

Male Female

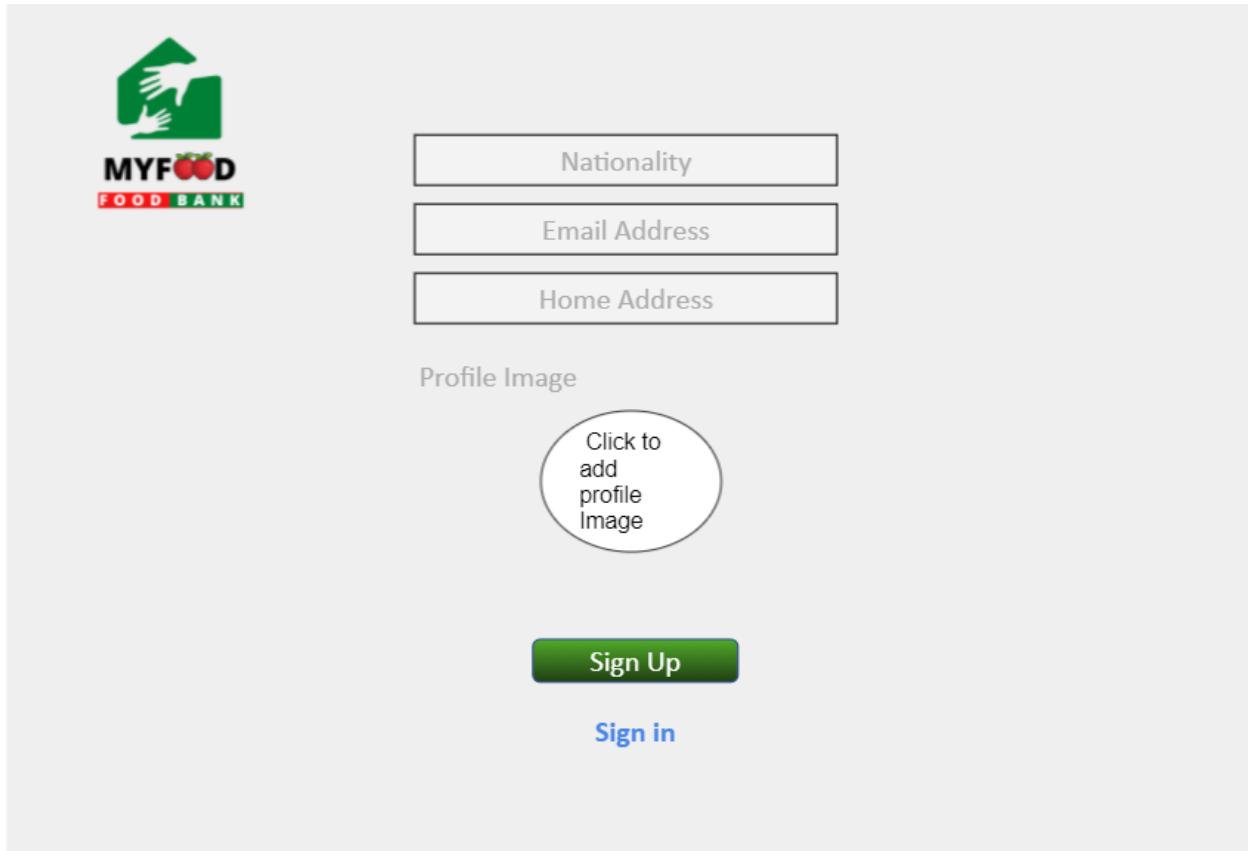


Figure 4.1a User's Register Page User Interface - This UI is used for registering their personal details which consist of Username, First name, Last name, Birth Date, Gender, Nationality, Email, Password, Mobile number, and Home Address.

4.1b Login Page (Admin, Distributor, Donor, Recipient)

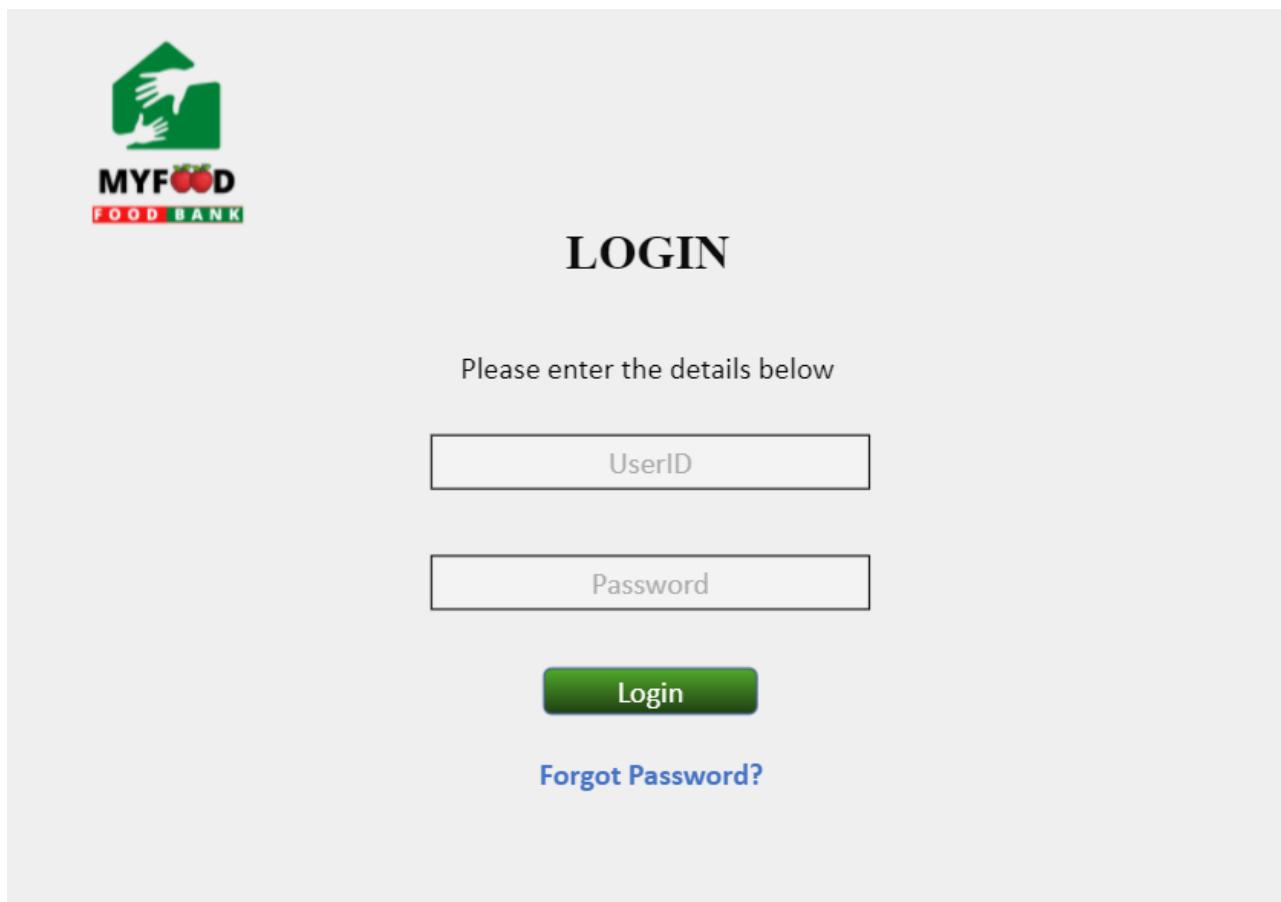
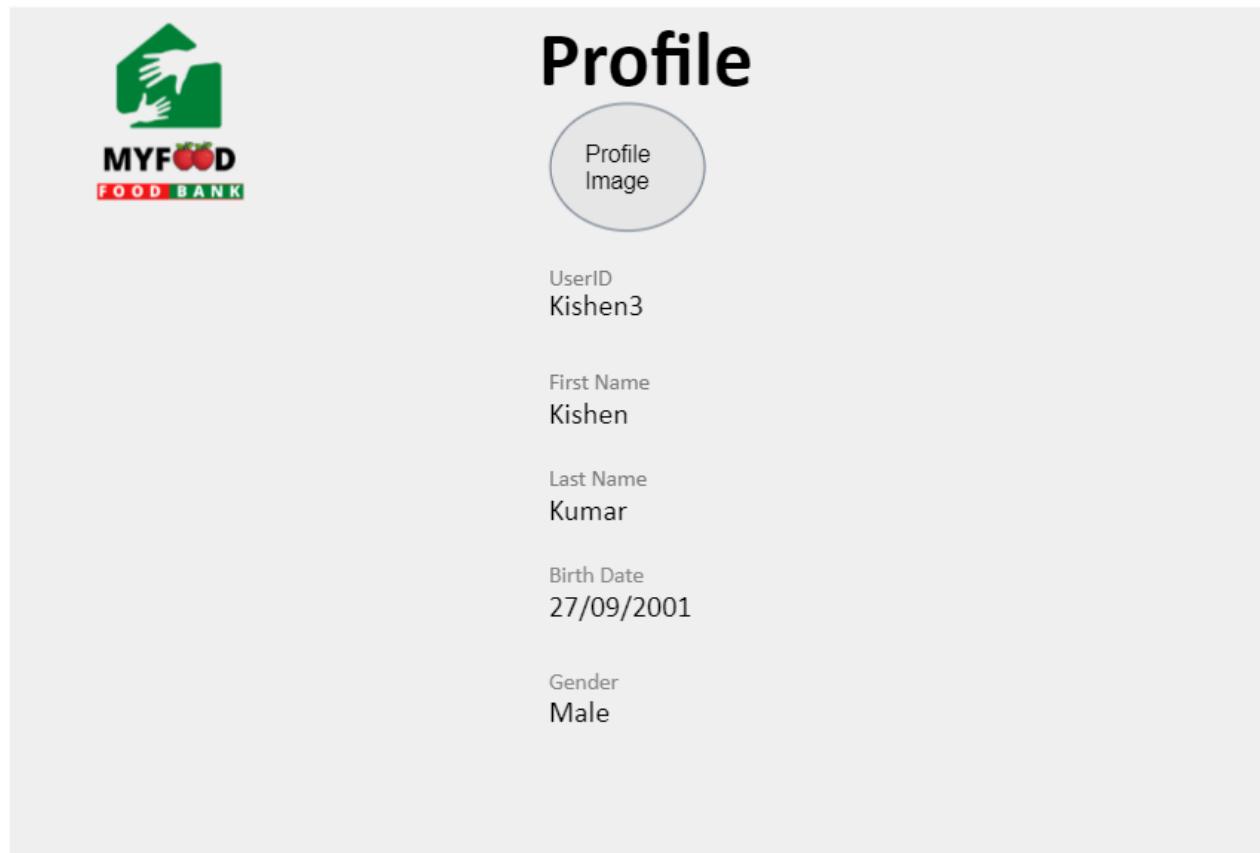


Figure 4.1b User's Login Page User Interface - This UI is used to login into the system



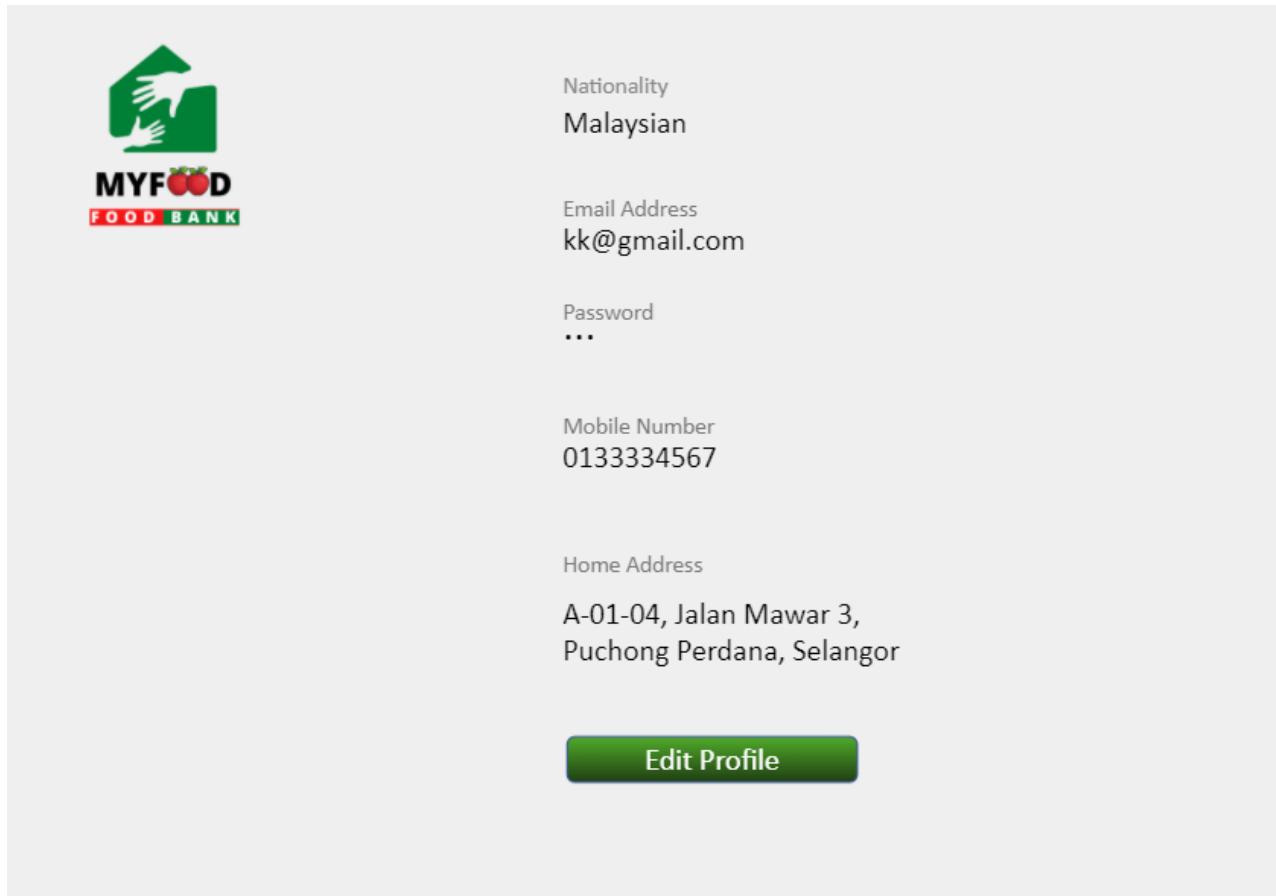


Figure 4.1c User's 'View Profile' User Interface - This UI is used for viewing their personal details which consist of Username, First name, Last name, Birth Date, Gender, Nationality, Email, Password, Mobile number, and Home Address.

4.1c Edit Profile Page (Admin, Distributor, Donor, Recipient)



Profile

Profile
Image

[Change photo](#)

[Upload photo](#)

User ID

First Name

Last Name

Birth Date

Gender

Male Female



Nationality

Password

Email Address

Mobile Number

Home Address

Figure 4.1d User's 'Edit Profile' User Interface - This UI is used when the user wishes to change his/her profile information

4.1d Home Page (Donor)



Figure 4.1.e Donor's 'Homepage' User Interface - This UI is used for viewing the home page of current events. From viewing the community feed to images on item stocks as well as pictures of ongoing charity events.

4.2a View foods low on stock (Donor)

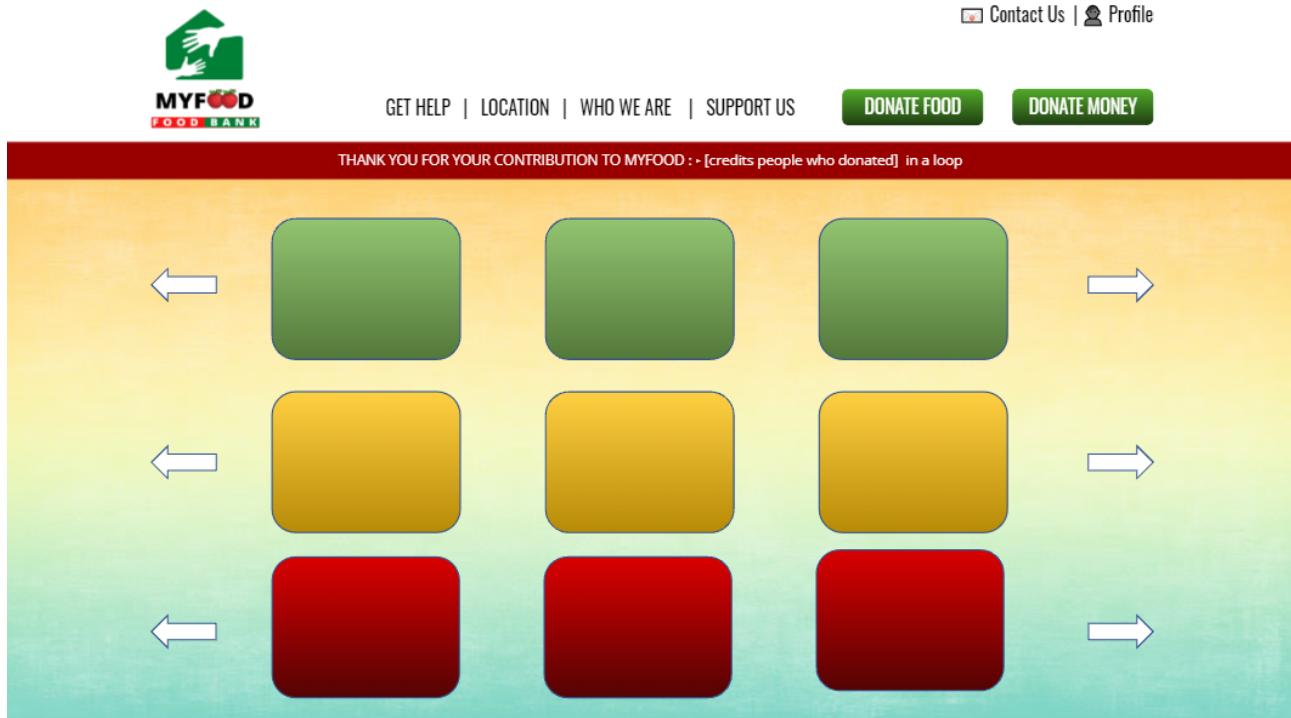


Figure 4.2.a Donor's foods low on stock User Interface - This UI is used for viewing the foods low on stock. They are highlighted in green, yellow and red. The first indicates that there is still an ample amount of food stock, the second indicates that those items are moderately low on stock while the red indicates that it is severely low on stock.

4.2b View history of donations (Donor)

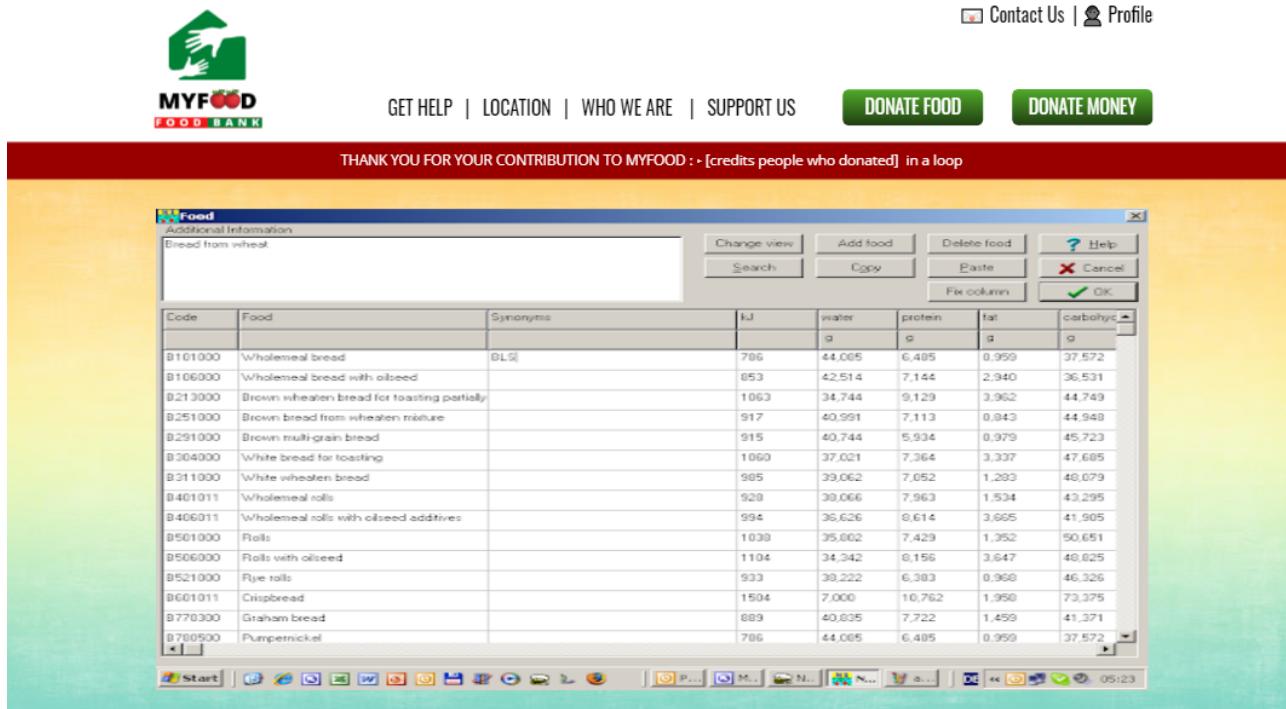


Figure 4.2.b Donor's history of donations User Interface - This UI is used for viewing the donations the donor has made to the food bank.

4.2c Donate money (Donor)

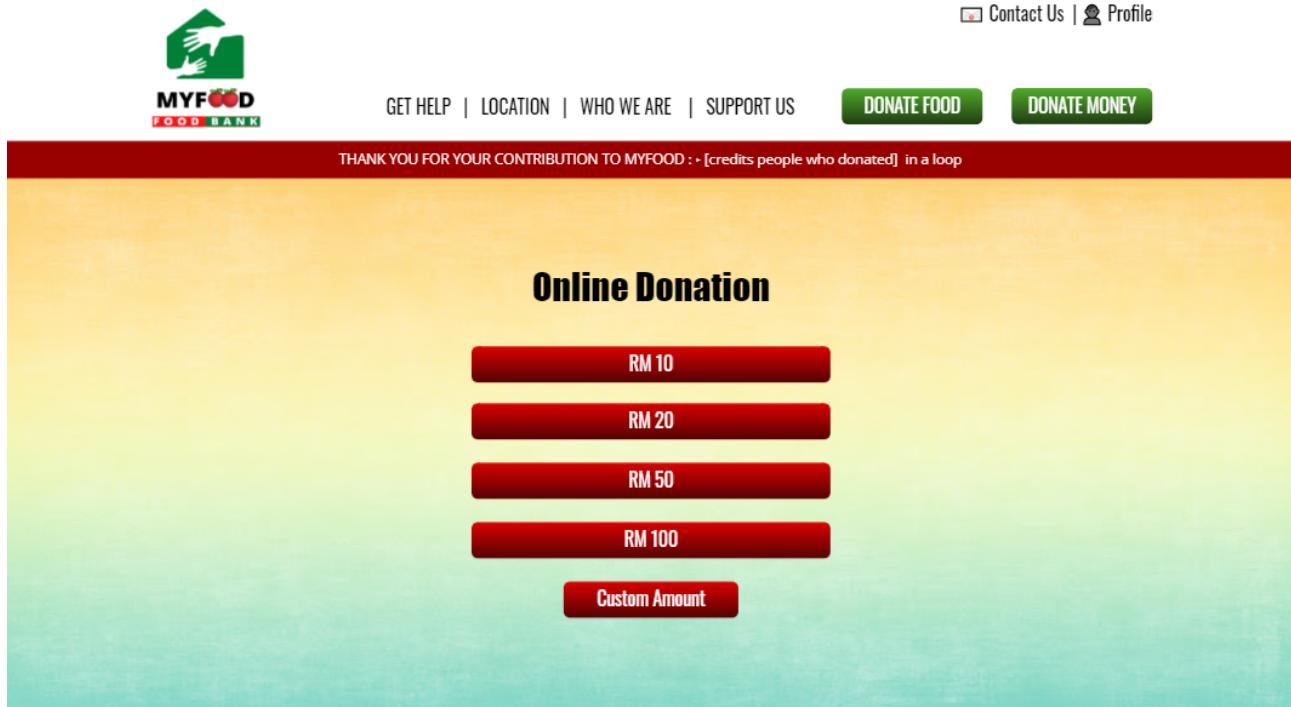


Figure 4.2.c Donor's donate money User Interface - This UI is used for making a donation in terms of cash to the food bank. There are 5 options to choose from. Four of them are amount presets while the other is a custom amount that the Donor can key in.

4.2d Donate food (Donor)



Figure 4.2.d Donor's donate food User Interface - This UI is used for making a donation in terms of food to the food bank. There are 3 options to choose from. Food Drive is a food collection drive. While the other two are informations on how the donor wishes to give away the foods his willing to donate

4.2e Find food bank locations (Donor)

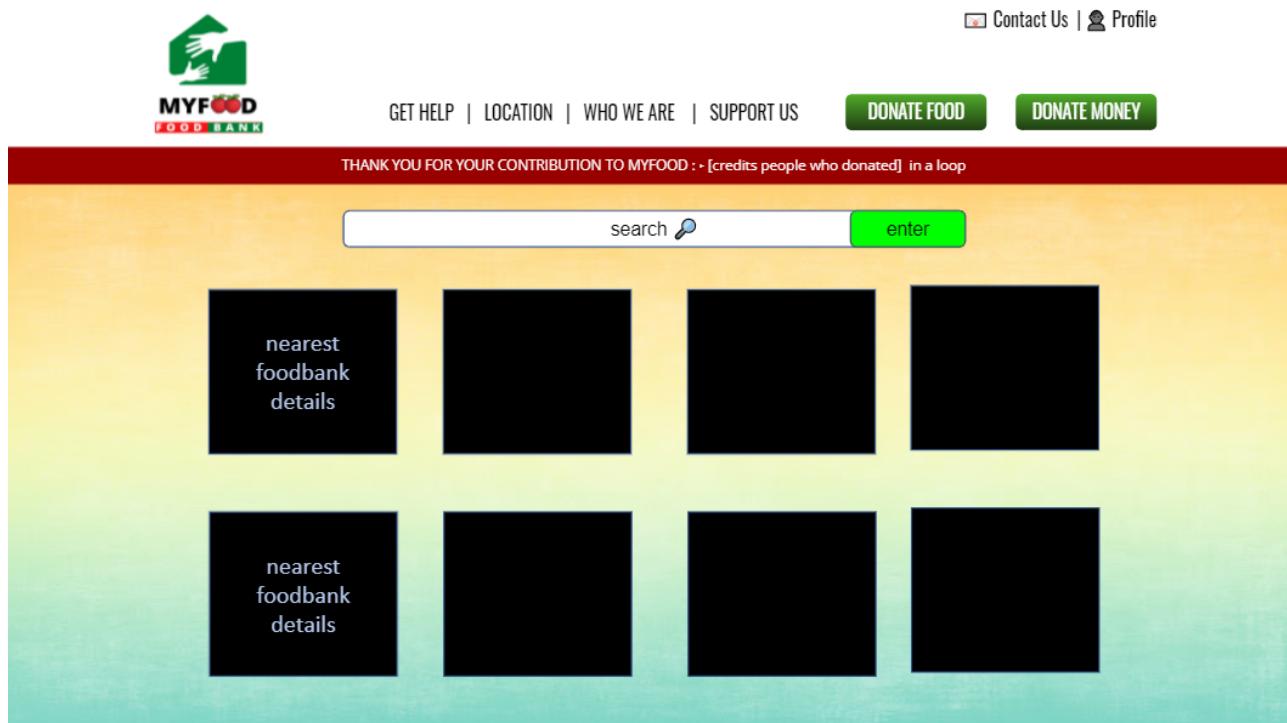


Figure 4.2.e Donor's find foodbank Interface - This UI is used to find the nearest food bank by comparing it with the donor's state. This shows the food bank in the same state as the Donor

4.3a Add User (Admin)

The screenshot shows the 'Add User' form within the 'USERS' section of the application. The interface includes a header with the logo 'MYFOOD FOOD BANK', navigation links ('GET HELP', 'LOCATION', 'WHO WE ARE', 'SUPPORT US'), and two green buttons ('DONATE FOOD', 'DONATE MONEY'). A red navigation bar at the top contains tabs for 'USERS', 'REPORT', 'LOCATIONS', and 'FOOD CATEGORIES'. The main form area is titled 'ADD USER' and contains nine input fields for 'Fname', 'Lname', 'telPhone', 'Gender', 'User_ID', 'Email', 'Username', 'Password', and 'Address'. Below the form are two red buttons: 'Submit' and 'Cancel'.

Figure 4.3.a Admin's Add User Interface - This UI is used to add users into the system

4.3b Find User (Admin)

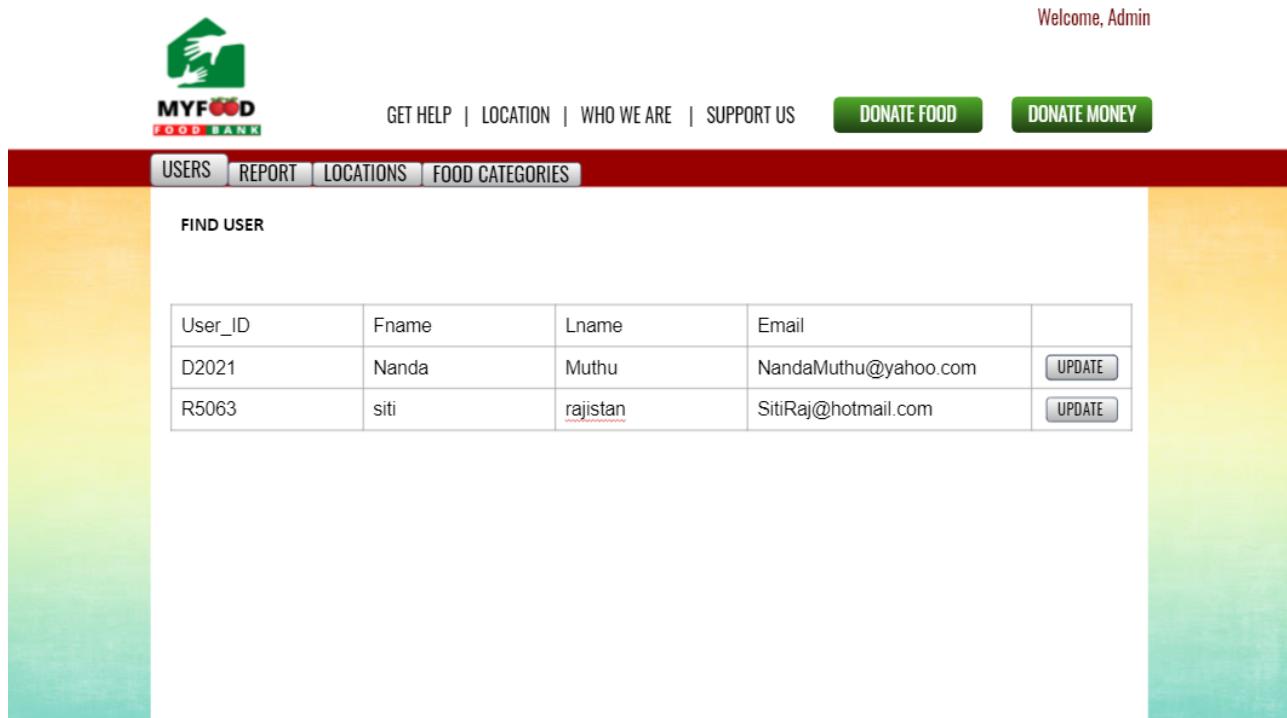


Figure 4.3.b Admin's Find User Interface - This UI is used to find users in the system

4.3c Edit User (Admin)

The screenshot shows the 'Edit User' interface for an administrator. At the top, there is a logo for 'MYFOOD FOOD BANK' featuring a hand holding a heart. To the right of the logo, the text 'Welcome, Admin' is displayed. Below the logo, there is a navigation bar with links: 'GET HELP | LOCATION | WHO WE ARE | SUPPORT US'. There are also two green buttons: 'DONATE FOOD' and 'DONATE MONEY'. The main menu below the navigation bar includes 'USERS', 'REPORT', 'LOCATIONS', and 'FOOD CATEGORIES', with 'USERS' being the active tab. The central area is titled 'EDIT USER' and contains a table with the following data:

Field	Value
Fname	siti
Lname	rajistan
telPhone	012-3334-212
Gender	Others
User_ID	R5063
Email	SitiRaj@hotmail.com
Username	City
Password	*****
Address	Jalan Desa Bakti 2, Taman Desa

At the bottom of the form, there are two red buttons: 'Submit' and 'Cancel'.

Figure 4.3.c Admin's Edit User Interface - This UI is used to edit users information in the system

4.3d Add Food Bank report (Admin)

Welcome, Admin

GET HELP | LOCATION | WHO WE ARE | SUPPORT US

DONATE FOOD DONATE MONEY

USERS REPORT LOCATIONS FOOD CATEGORIES

ADD FOOD REPORT

Food Category	<input type="text"/>
Distribution Location	<input type="text"/>
Quantity	<input type="text"/>

Submit Cancel

Figure 4.3.d Admin's Add Food Bank Report Interface - This UI is used to add a food report for distribution

4.3e Find Food Bank report (Admin)

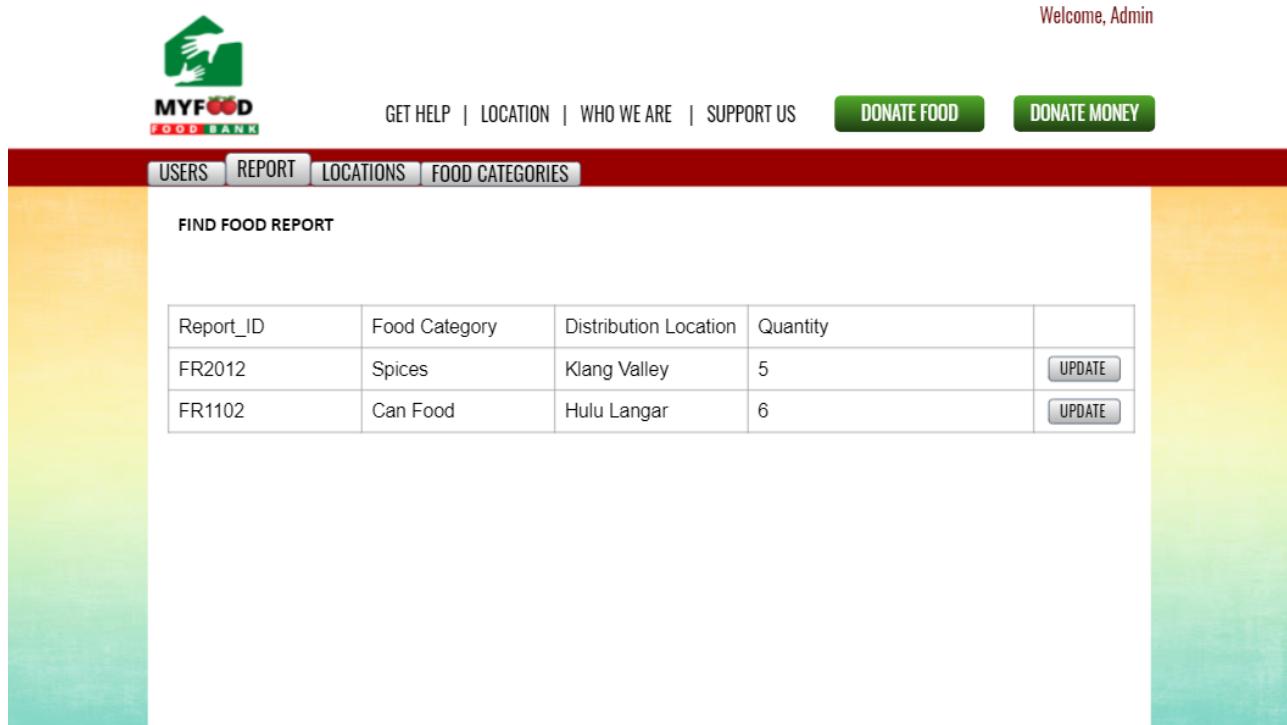


Figure 4.3.e Admin's Find Food Bank Report Interface - This UI is used to query a food report

4.3f Edit Food Bank Report (Admin)

Welcome, Admin

MYFOOD
FOOD BANK

GET HELP | LOCATION | WHO WE ARE | SUPPORT US

DONATE FOOD DONATE MONEY

USERS REPORT LOCATIONS FOOD CATEGORIES

EDIT FOOD REPORT

Food Category	Spices
Distribution Location	Klang Valley
Quantity	5

Submit Cancel

Figure 4.3.f Admin's Edit Food Bank Report Interface - This UI is used to edit a food bank report

4.3g Add Food Bank (Admin)

Welcome, Admin

GET HELP | LOCATION | WHO WE ARE | SUPPORT US

DONATE FOOD DONATE MONEY

USERS | REPORT | LOCATIONS | FOOD CATEGORIES

ADD FOOD BANK

FoodBank_ID	<input type="text"/>
Name	<input type="text"/>
Address	<input type="text"/>
Operating Hours	<input type="text"/>

Submit Cancel

Figure 4.3.g Admin's Add Food Bank Interface - This UI is used to add food banks into the system

4.3h Find Food Bank (Admin)

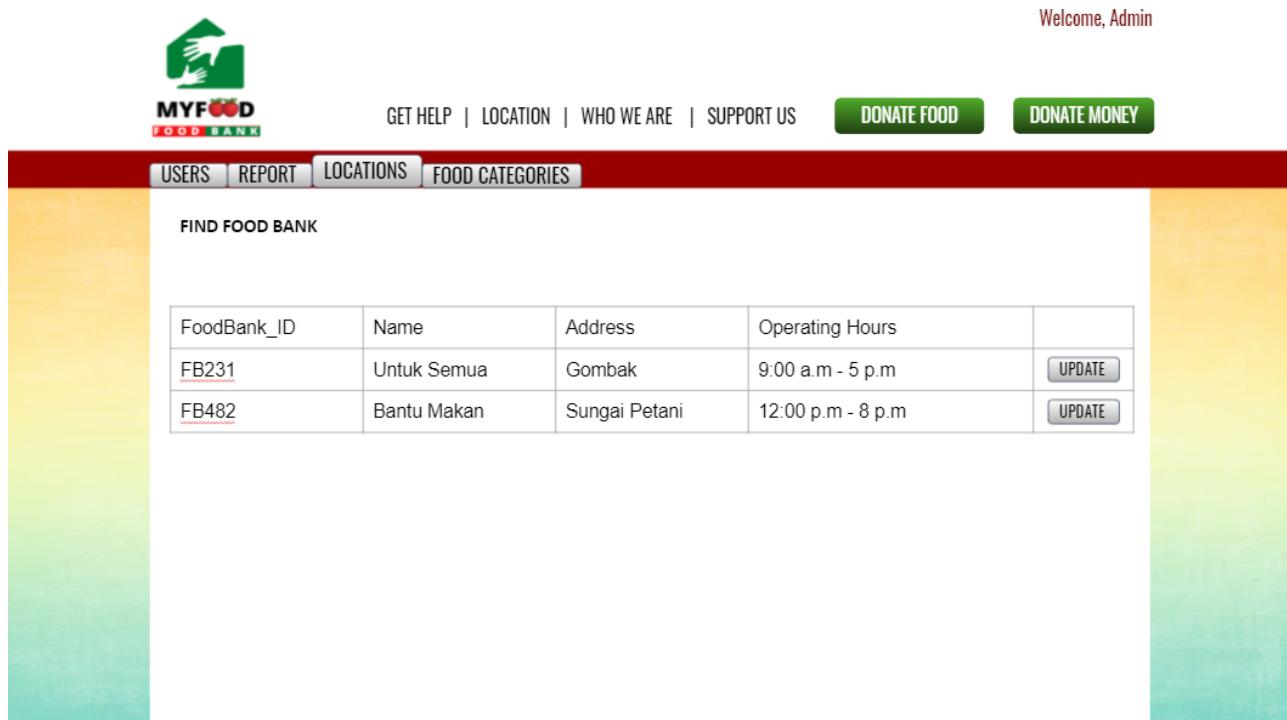


Figure 4.3.h Admin's Find Food Bank Interface - This UI is used to find food banks that are in the system. You also have an option to update it as well

4.3i Edit Food Bank (Admin)

Welcome, Admin

MYFOOD
FOOD BANK

GET HELP | LOCATION | WHO WE ARE | SUPPORT US

DONATE FOOD DONATE MONEY

USERS REPORT LOCATIONS FOOD CATEGORIES

EDIT FOOD BANK

FoodBank_ID	FB231
Name	Untuk Semua
Address	Gombak
Operating Hours	9:00 a.m - 5 p.m

Submit Cancel

Figure 4.3.i Admin's Edit Food Bank Interface - This UI is used to edit food banks that are in the system

4.4a View Food Menu (Recipient)



Figure 4.4a Recipient's 'View Food Menu' Interface - This Interface is used for viewing all the food that they offer and that is available for deliveries with all of the food information related transactions such as the food name, foodbank location, Date, Range of time for the food delivery, and remaining food available for reservation. It also allows them to select the food that they wanted to reserve from all the food that is available at the specific period of time.

4.4b Reserve Food (Recipient)

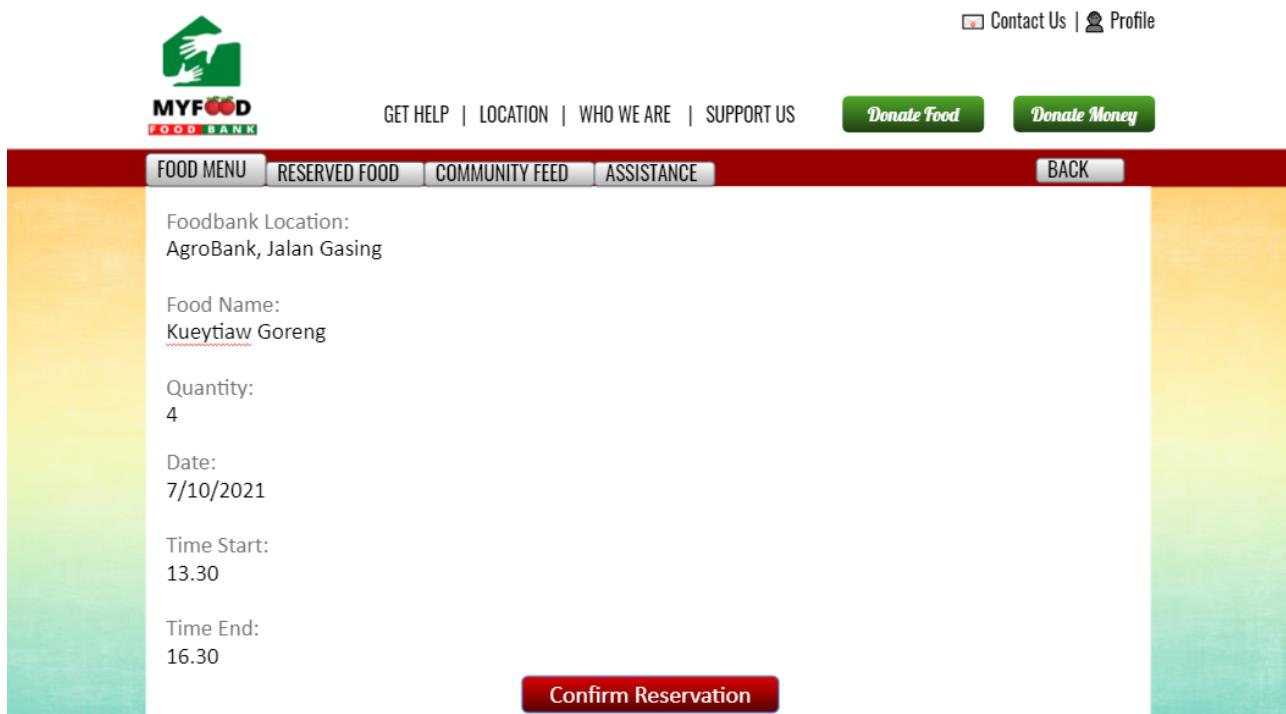


Figure 4.4b Recipient's 'Reserve Food' Interface - This Interface is used for displaying Food Bank Location, Food Name, Quantity, Date and range of time for the food delivery for the food that they selected before they confirm the reservation of the specified food.

4.4c Cancel Reservation of Food (Recipient)

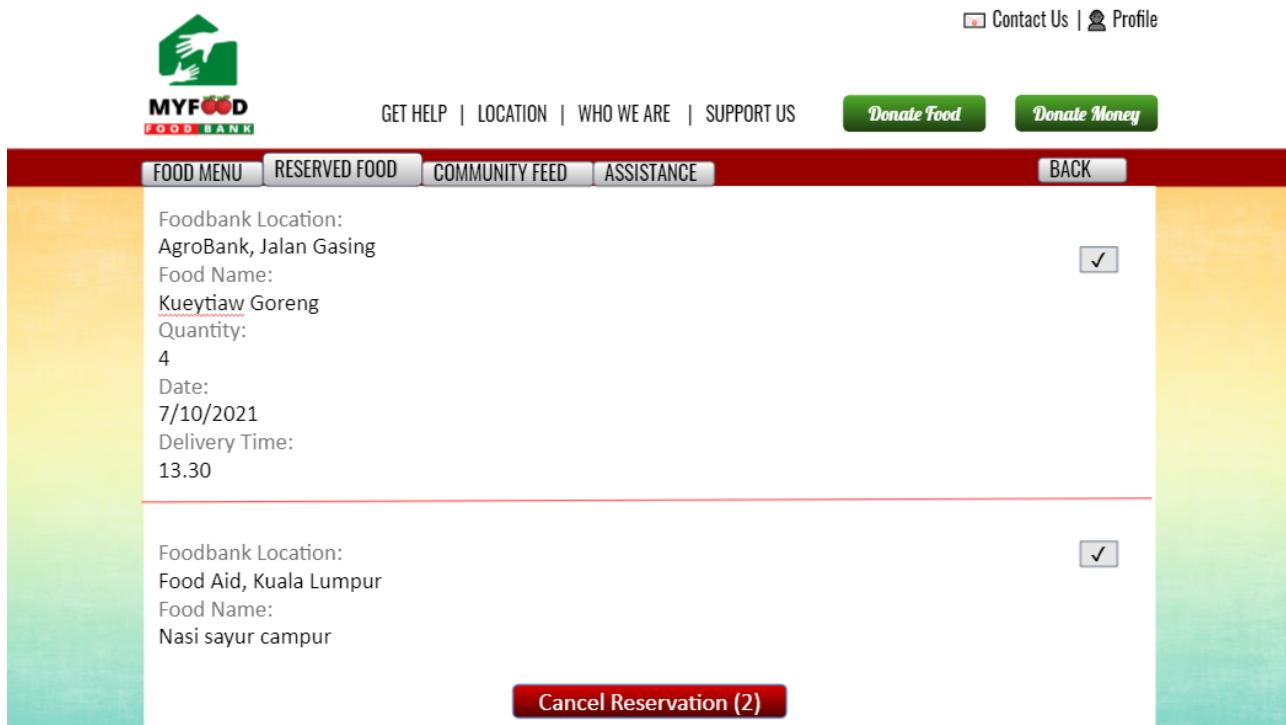


Figure 4.4c Recipient's 'Cancel Reservation of Food' Interface - This Interface is used for canceling the reservation back if the recipient does not want the food.

4.4d View Community (Recipient)

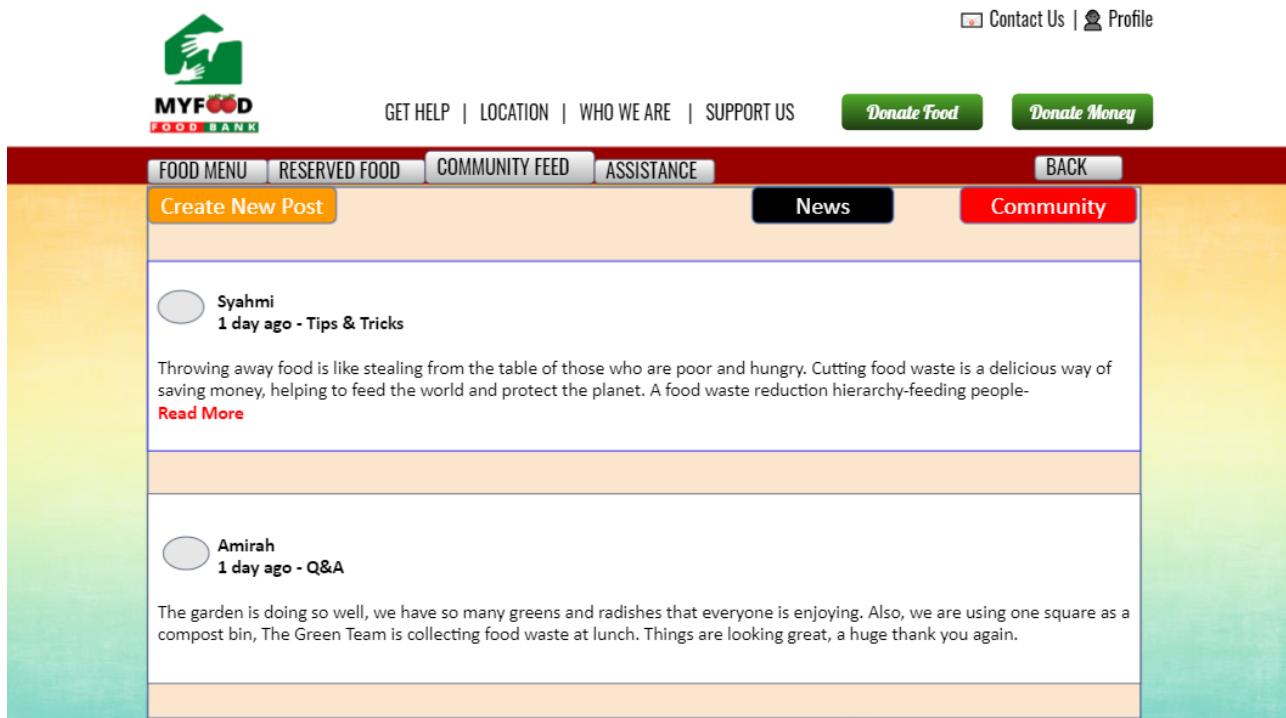
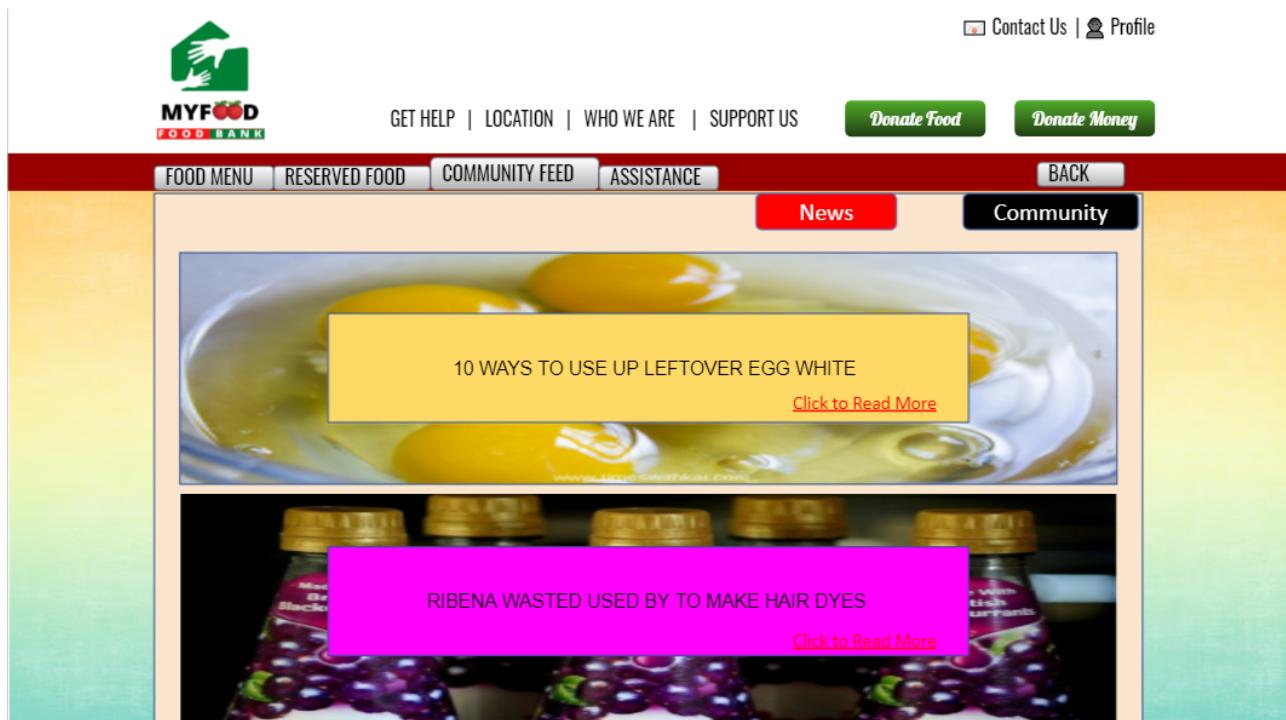


Figure 4.4d Recipient's 'View Community' User Interface - This UI is used for displaying daily posts about food waste, sharing tips and more from other users and it can be done by clicking the 'create a new post' button.

4.4e View News (Recipient)



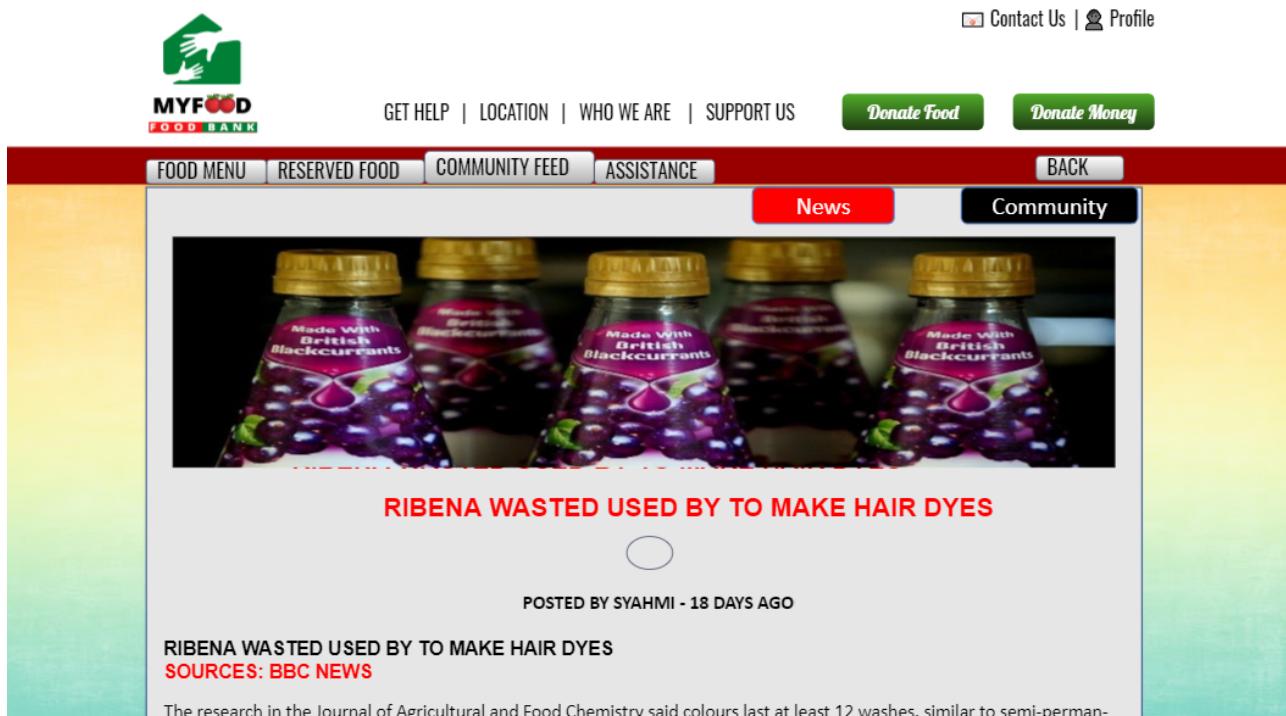


Figure 4.4e Recipient's 'View News' User Interface - This UI is used for displaying news and information about food which has been done by authorised users.

4.4f Acquire Assistance (Recipient)

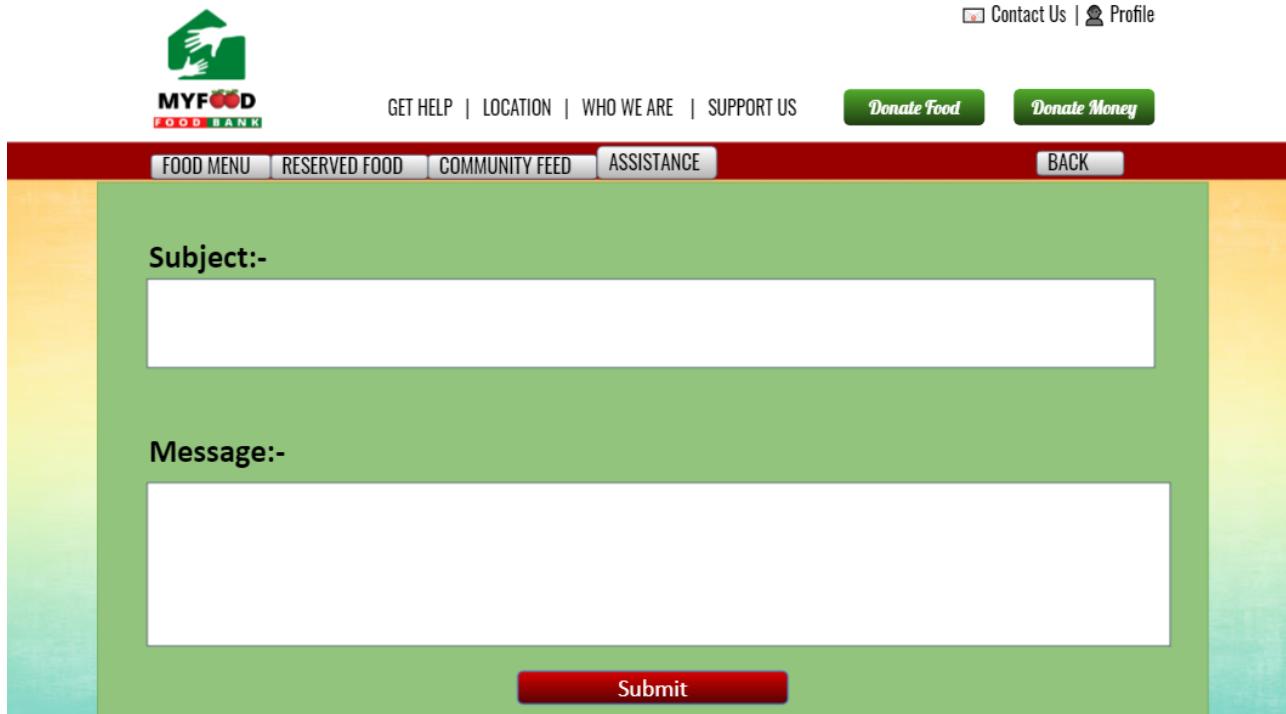


Figure 4.4f Recipient's 'Acquire Assistance' Interface - This interface is often the go-to for a new recipient on a mission. It's where they go when they have a question and truly want to speak to an individual at the organization so this interface is used to convert by clearly and effectively presenting the method(s) of getting in touch with the food bank organization as quickly as possible.

4.5a Search for nearby foodbank locations (Distributor)

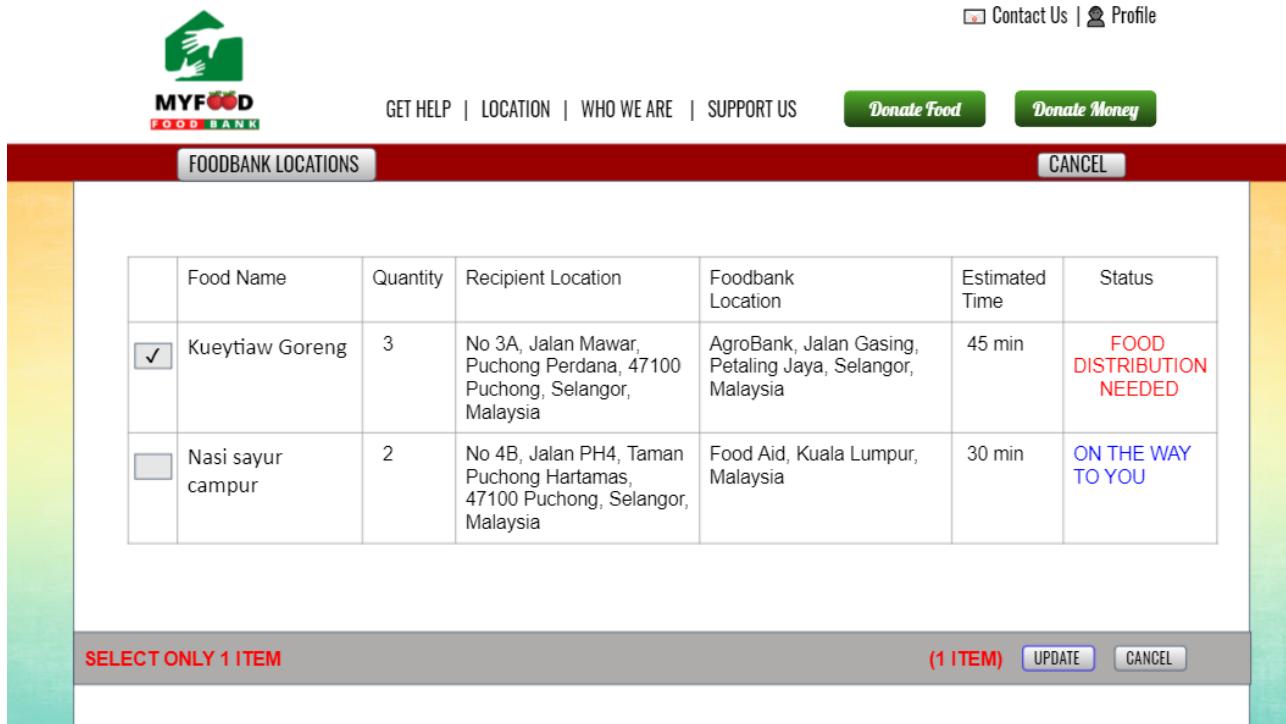


Figure 4.5a Distributor's 'Search for nearby food bank locations' Interface - This Interface allows distributors to search for nearby food bank locations by comparing between the distributor's and food bank's location so that they can decide whether they can accept the distribution of the specific food or not. This also allows them to check the status of the reserved food that is available at the specific food bank location and period of time.

4.5b Select status of reserved food (Distributor)



GET HELP | LOCATION | WHO WE ARE | SUPPORT US

RESERVED FOOD STATUS **TRACK RESERVED FOOD** **CANCEL**

Food Name	Status
Kueytiaw Goreng	SELECT STATUS <input checked="" type="checkbox"/> V
Quantity	ACCEPT THE DISTRIBUTION
3	CANCEL THE DISTRIBUTION
Recipient Location	PICKED UP FOOD
No 3A, Jalan Mawar, Puchong Perdana, 47100 Puchong, Selangor, Malaysia	ON THE WAY TO YOU
Foodbank Location	DROPPED OFF FOOD
AgroBank, Jalan Gasing, Petaling Jaya, Selangor, Malaysia	

Estimated Time
45 min

CONFIRM STATUS

Figure 4.5b Distributor's 'Select status of reserved food' Interface - This interface allows them to confirm to the system that they have chosen the status for the specified food. After they have chosen the status for it, the system will automatically update the status for the specified food in interface which is located in Figure 4.5a.

4.5c View distribution status (Distributor)

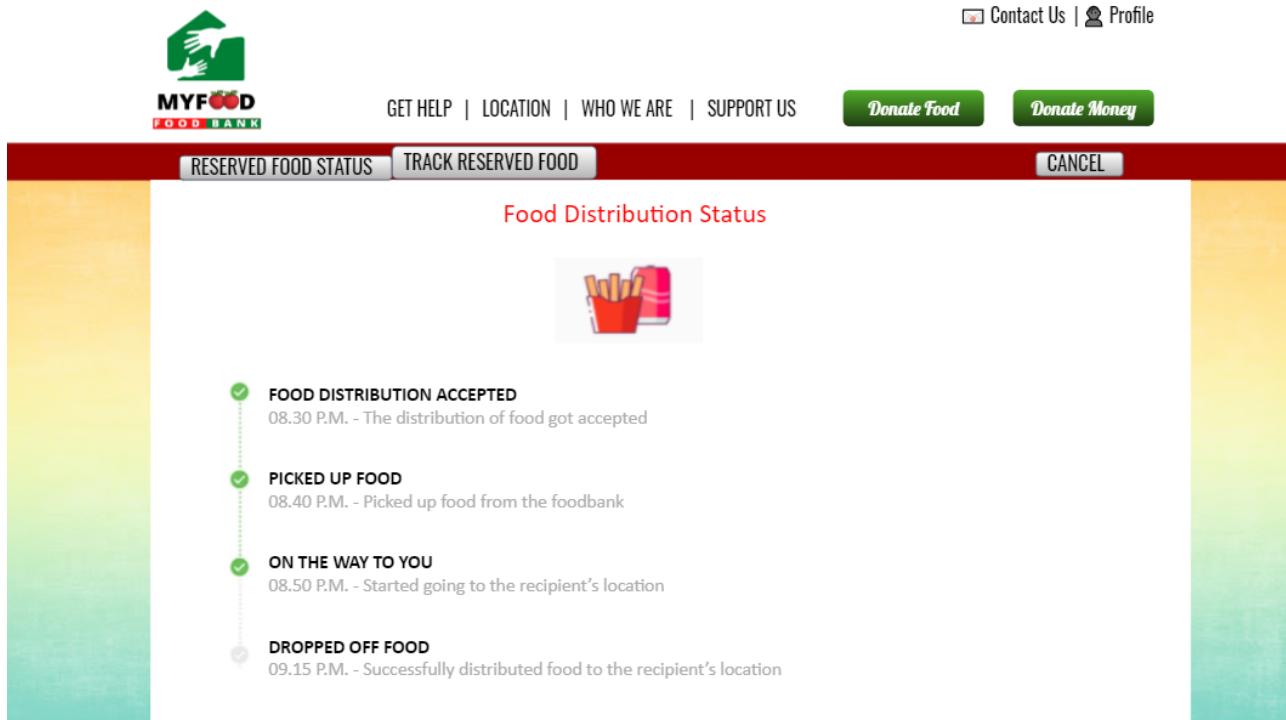


Figure 4,5c Distributor's 'View distribution status' Interface - This interface allows them to view their distribution status. The system will track the distributor's and recipient's location and update the distribution status from time to time to revise the estimated delivery time.

5 Component Design

5.1 Main Components

5.1.1 Component 1

Recipient

Sign Up

Component Diagram

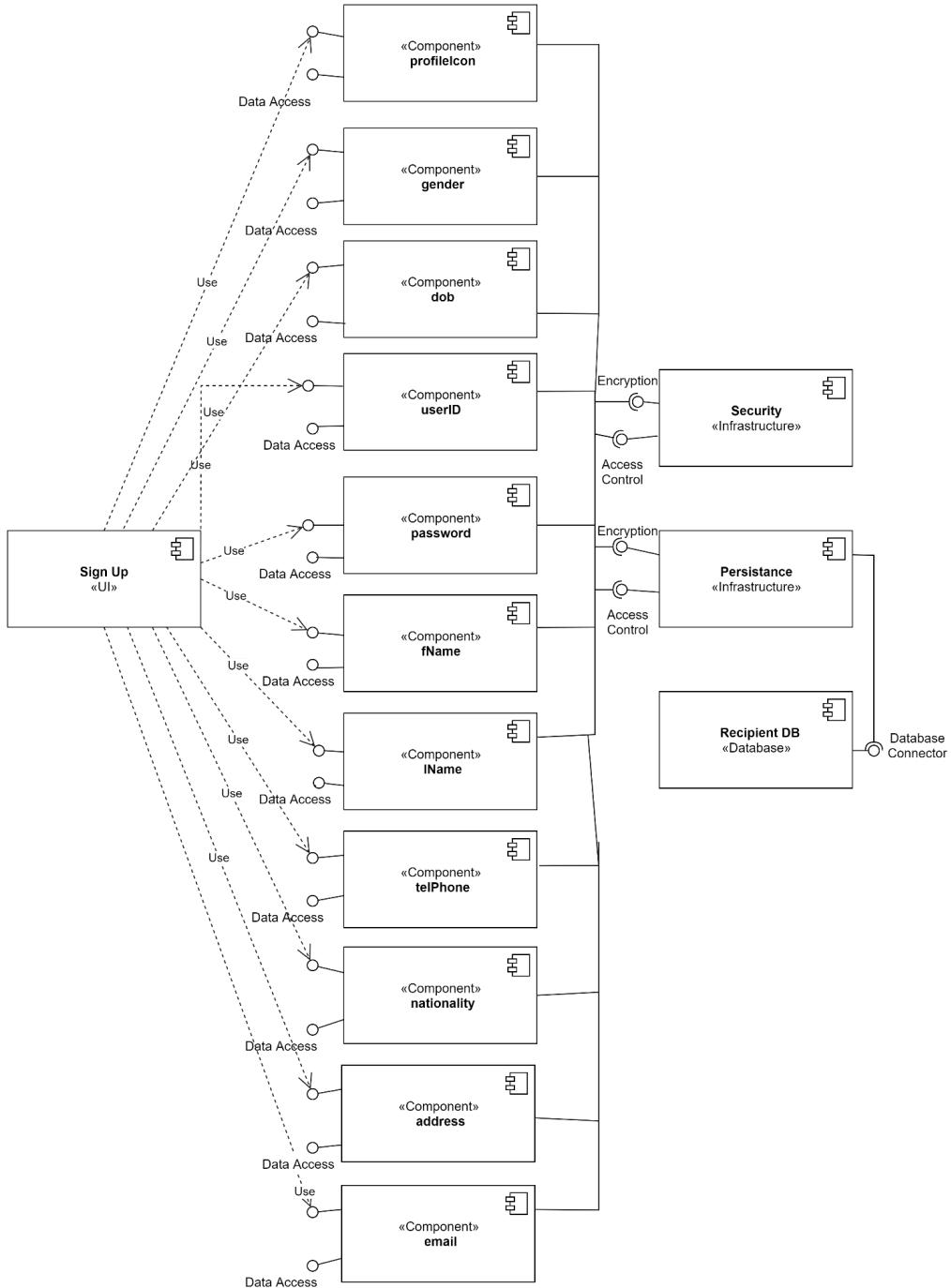


Figure 5.1a Recipient's 'Sign Up' Component Diagram

Activity Diagram

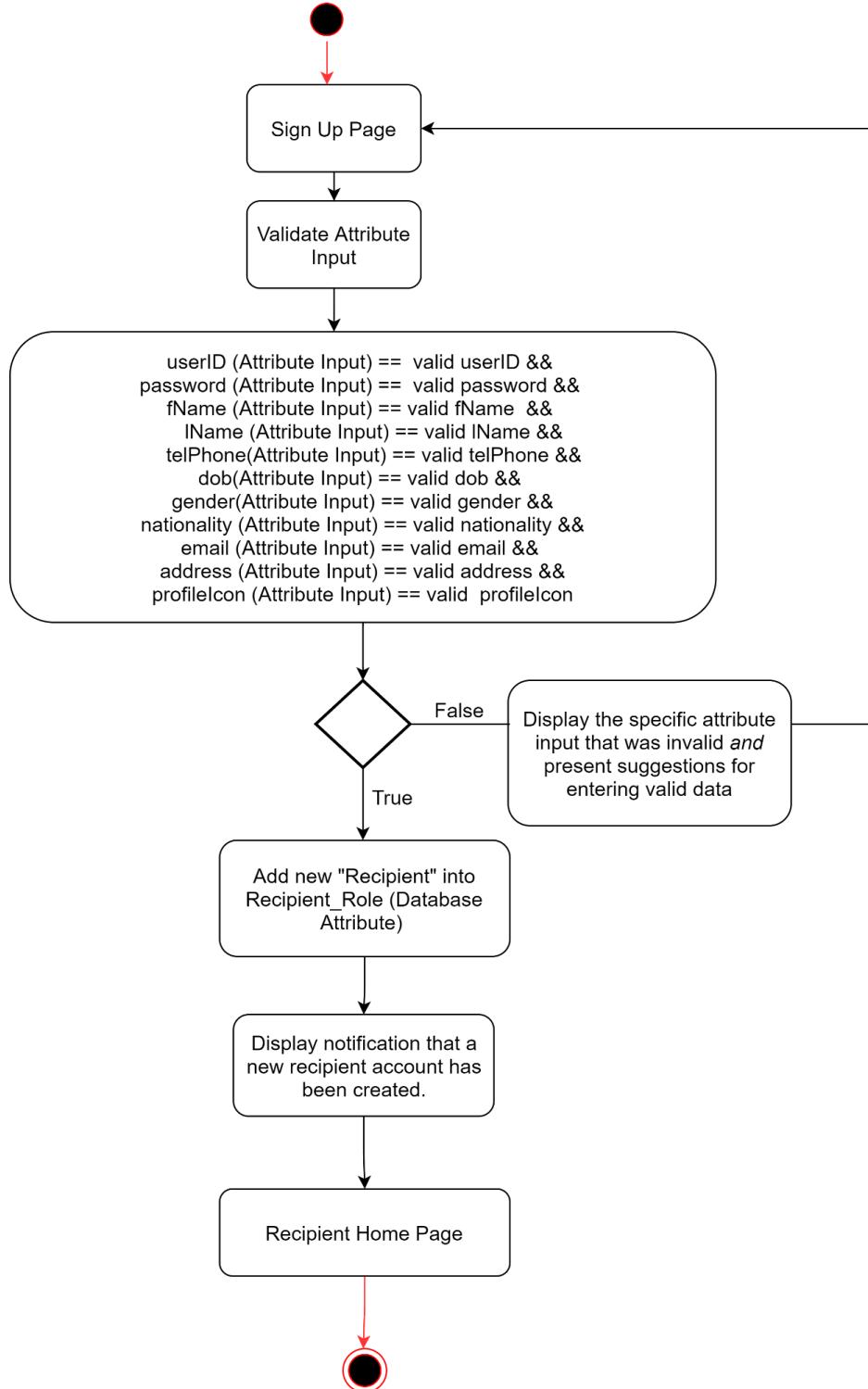


Figure 5.1b Recipient's 'Sign Up' Activity Diagram

Pseudocode

GET userID
GET password
GET fName
GET lName
GET telPhone
GET dob
GET gender
GET nationality
GET email
GET address
GET profileIcon

IF (Entered userID == valid userID && Entered password == valid password && Entered fName == valid fName && Entered lName == valid lName && Entered telPhone == valid telPhone && Entered dob == valid dob && Entered gender == valid gender && Entered nationality == valid nationality && Entered email == valid email && Entered address == valid address && Entered profileIcon == valid profileIcon) **THEN**

Save New Recipient's Details into the Recipient_Role(Database Attribute)

Display notification that a new recipient account has been created (Sign Up Successful)

Go to the Recipient Home Page

ELSE

Sign Up Failed

Display the specific attribute input that was invalid *and* present suggestions for entering valid data
Return to the Sign Up page

END IF

Log in

Component Diagram

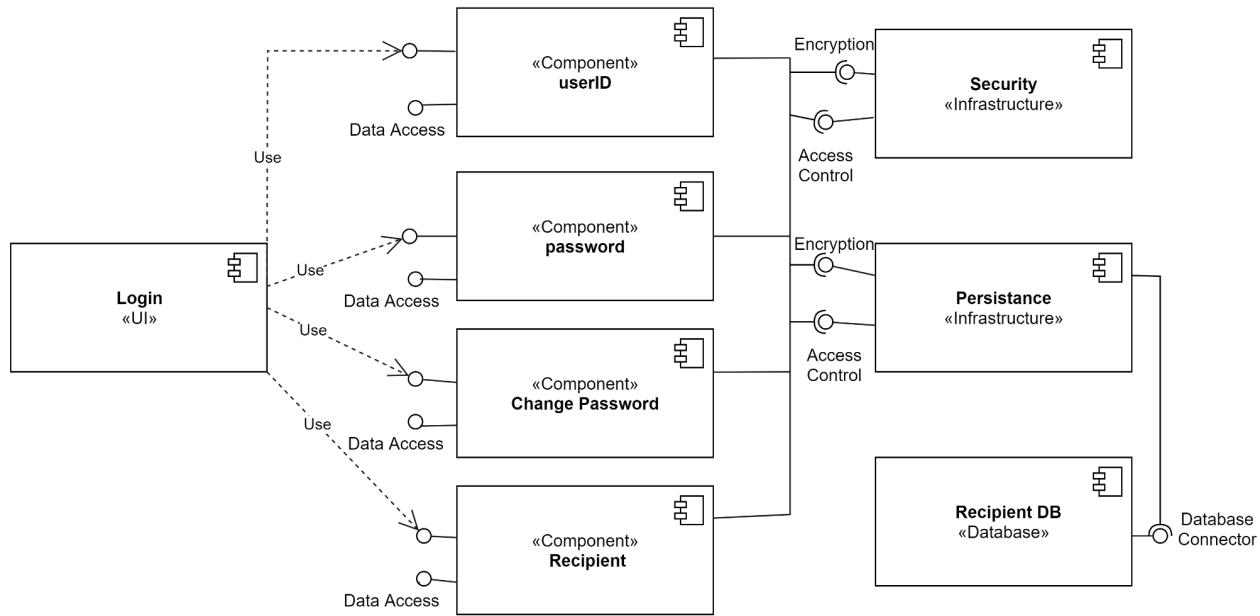


Figure 5.1c Recipient's 'Login' Component Diagram

Activity Diagram

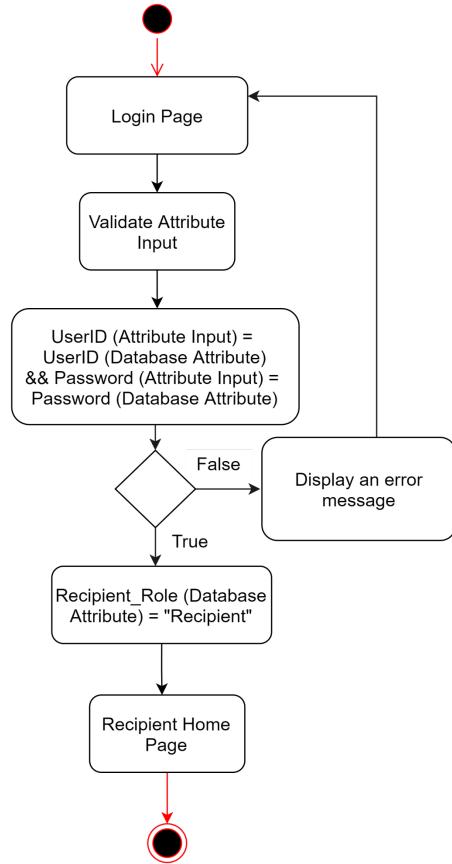


Figure 5.1d Recipient's 'Login' Activity Diagram

Pseudocode

```

GET userID
GET password

```

```

IF (userID == EnteredUserID && password == EnteredPassword) THEN
    Login Successful
    Go to the Recipient Home Page

```

```

ELSE
    Login Failed
    Return to the login page

```

```

END IF

```

View Profile

Component Diagram

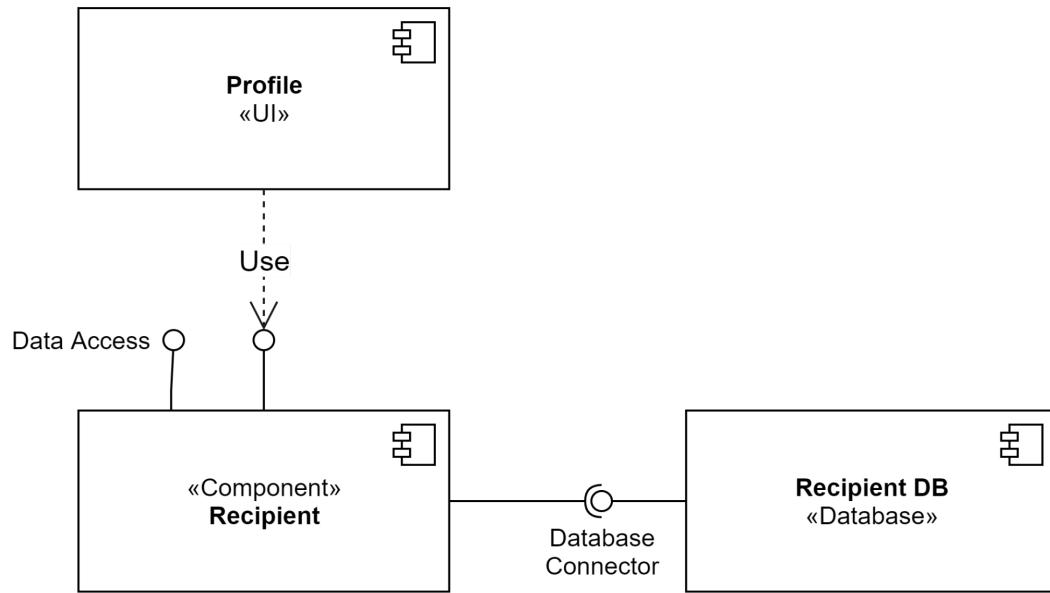


Figure 5.1e Recipient's 'View Profile' Component Diagram

Activity Diagram

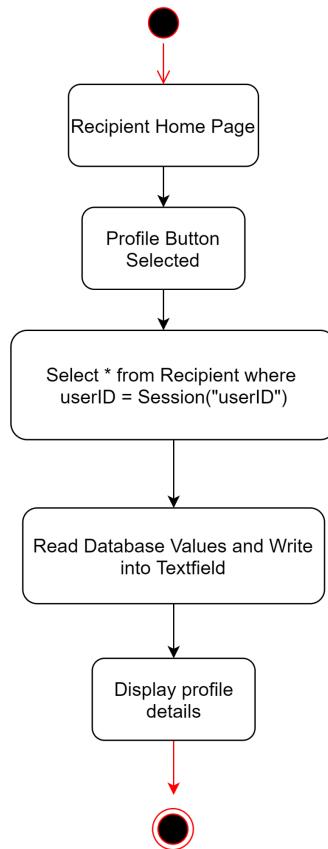


Figure 5.1f Recipient's 'View Profile' Activity Diagram

Pseudocode

Page Load

```
SELECT * FROM Recipient WHERE userID = Session("userID")
```

```

userID.Text = userID
password.Text = password
fName.Text = fName
lName.Text = lName
dob.Text = dob
gender.Text = gender
nationality.Text = nationality
address.Text = address
telPhone.Text = telPhone
email.Text = email
profile.Icon = profile

```

Edit Profile

Component Diagram

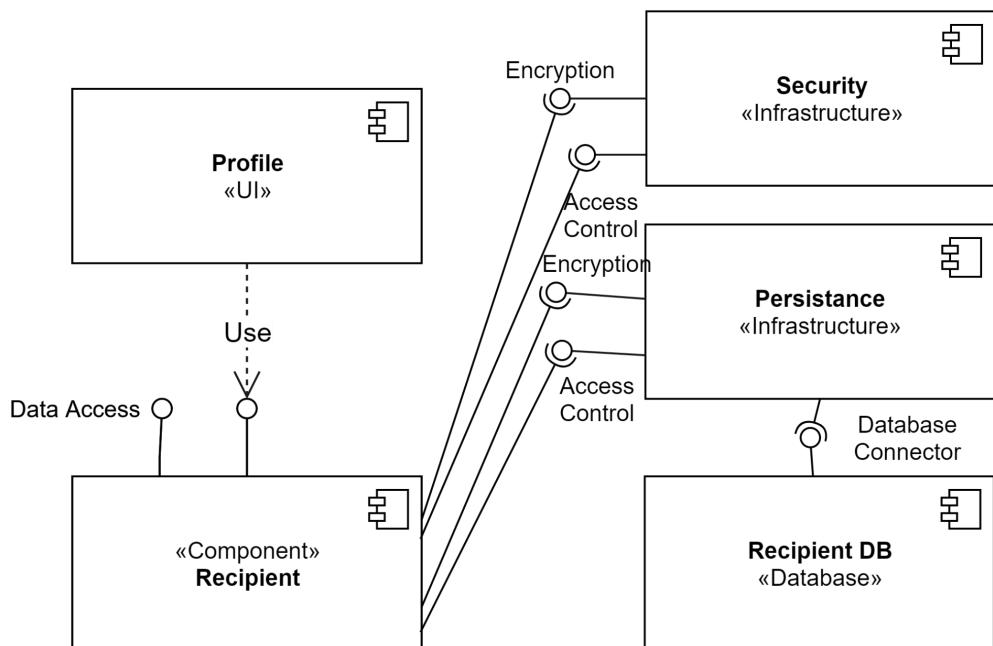


Figure 5.1g Recipient's 'Edit Profile' Component Diagram

Activity Diagram

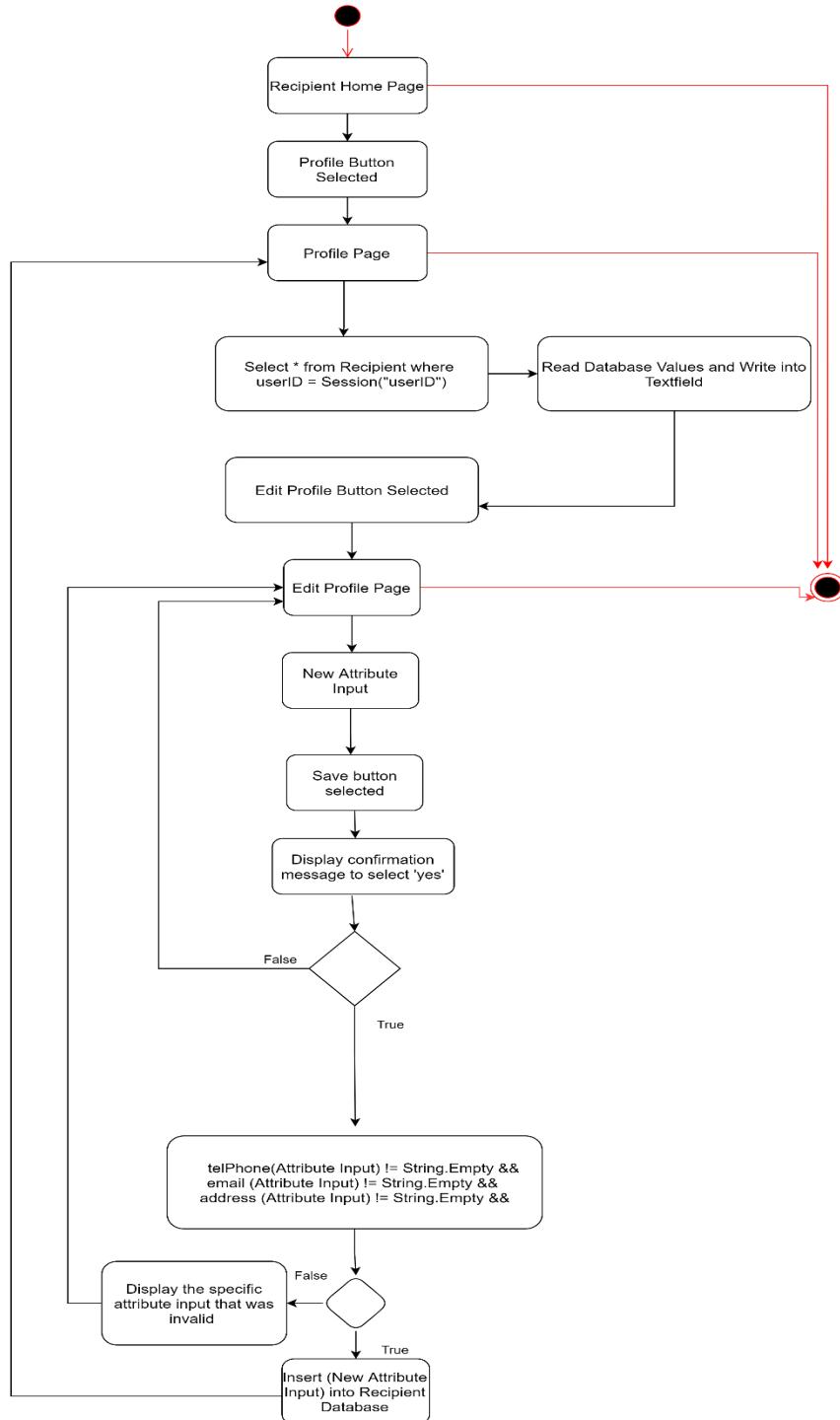


Figure 5.1h Recipient's 'Edit Profile' Activity Diagram

Pseudocode

Page Load

```
SELECT * FROM Recipient WHERE userID = Session("userID")
```

```
userID.Text = userID  
password.Text = password  
fName.Text = fName  
lName.Text = lName  
dob.Text = dob  
gender.Text = gender  
nationality.Text = nationality  
address.Text = address  
telPhone.Text = telPhone  
email.Text = email  
profile.Icon = profile
```

```
address.Text = "New address"  
telPhone.Text = "New telPhone"  
email.Text = "New email"
```

Save Button Selected

Displays Confirmation Message

IF ('Yes' Button Selected)

```
address.Text != String.Empty And  
telPhone.Text != String.Empty And  
email.Text != String.Empty
```

THEN

```
Update Recipient Address = "address.Text", telPhone = "telPhone.Text", email = "email.Text"  
Where userID = Session("userID")
```

ELSE

```
FailureText = "* Value Required or Invalid Value!"  
Return to Edit Profile Page
```

ELSE IF ('No' Button Selected)

[Return to Edit Profile Page](#)

END IF

View Food Menu

Component Diagram

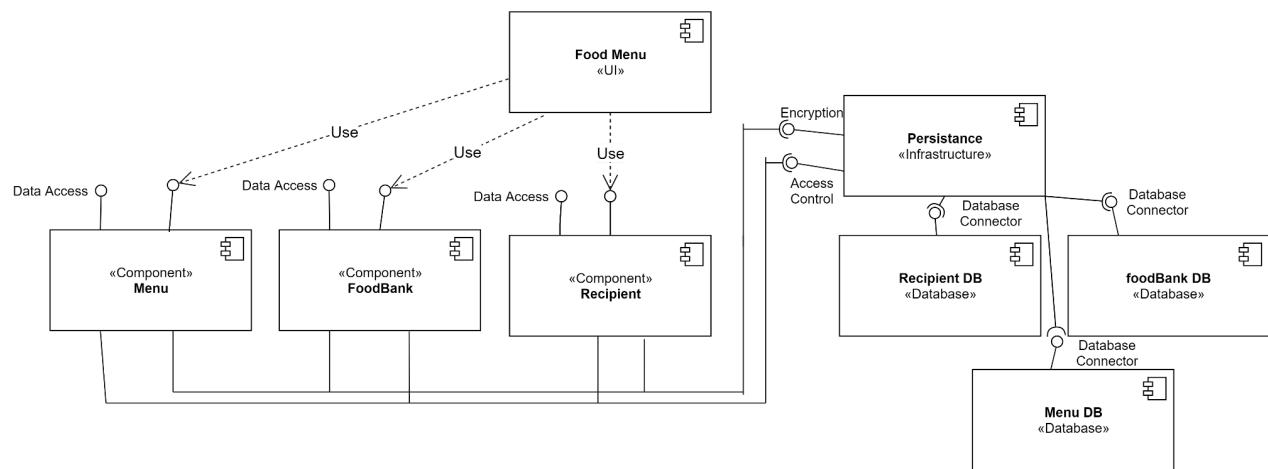


Figure 5.1i Recipient's 'View Food Menu' Component Diagram

Activity Diagram

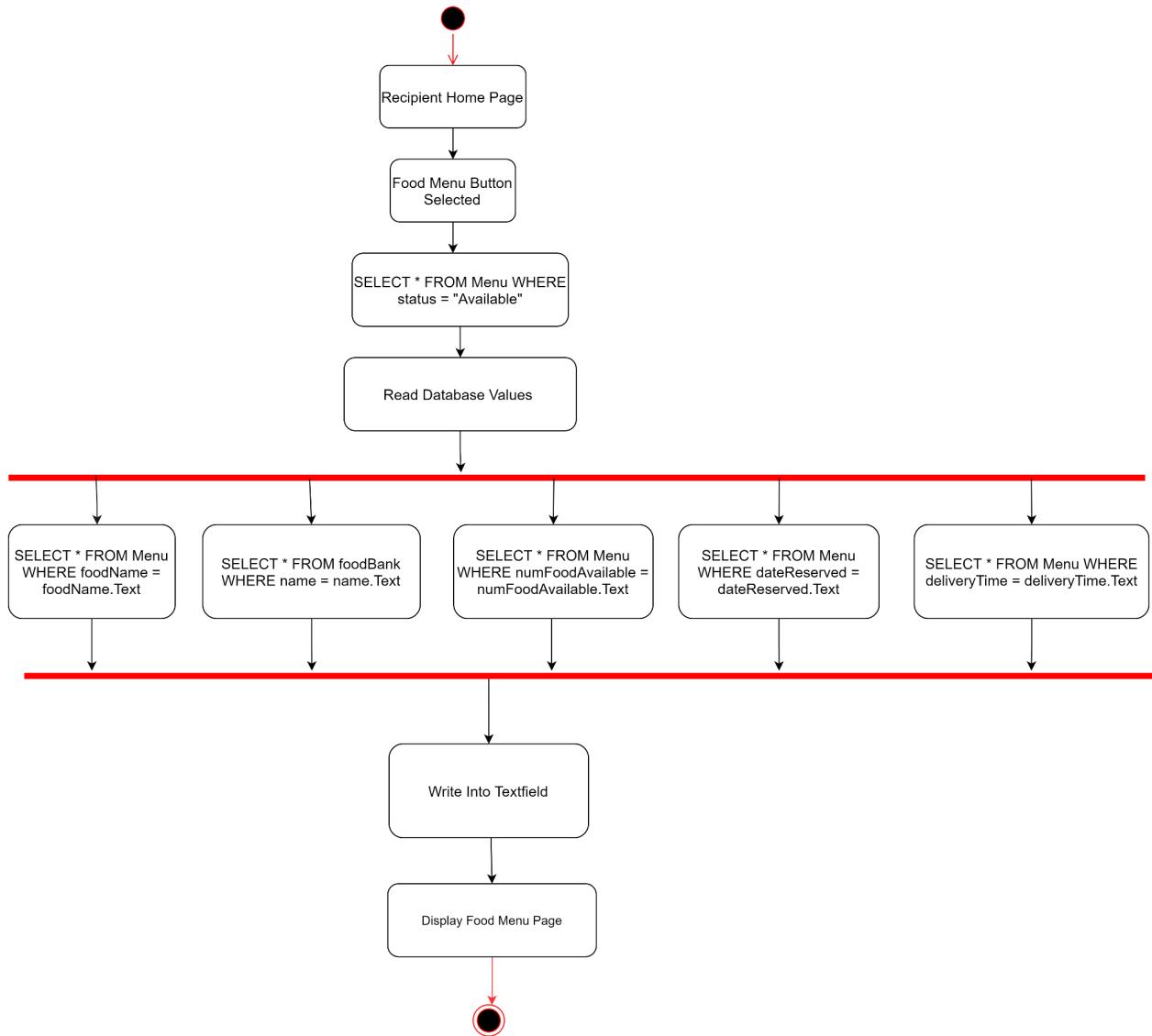


Figure 5.1j Recipient's 'View Food Menu' Activity Diagram

Pseudocode

Page Load

```

SELECT * FROM Menu WHERE status = Session("Available")
SELECT * FROM foodBank WHERE name = name.Text
SELECT * FROM Menu WHERE foodName = foodName.Text
SELECT * FROM Menu WHERE numFoodAvailable = numFoodAvailable.Text
SELECT * FROM Menu WHERE dateReserved = dateReserved.Text
SELECT * FROM Menu WHERE deliveryTime = deliveryTime.Text

```

```

foodBankName.Text = name
foodName.Text = foodName
numFoodAvailable.Text = numFoodAvailable
dateReserved.Text = dateReserved
deliveryTime .Text = deliveryTime

```

Reserve Food

Component Diagram

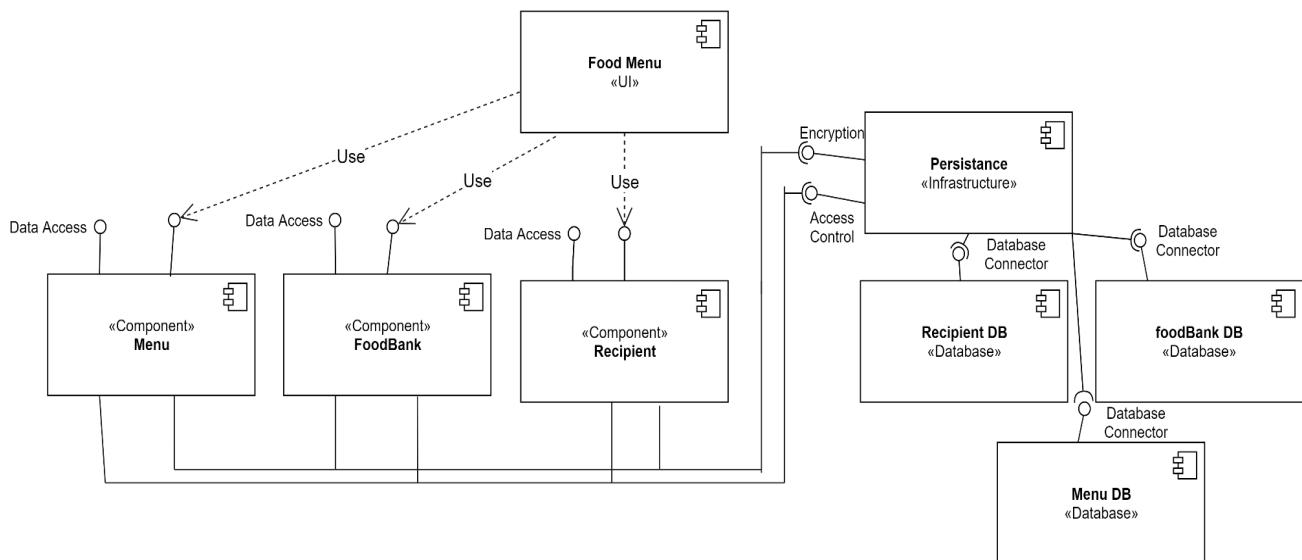


Figure 5.1k Recipient's 'Reserve Food' Component Diagram

Activity Diagram

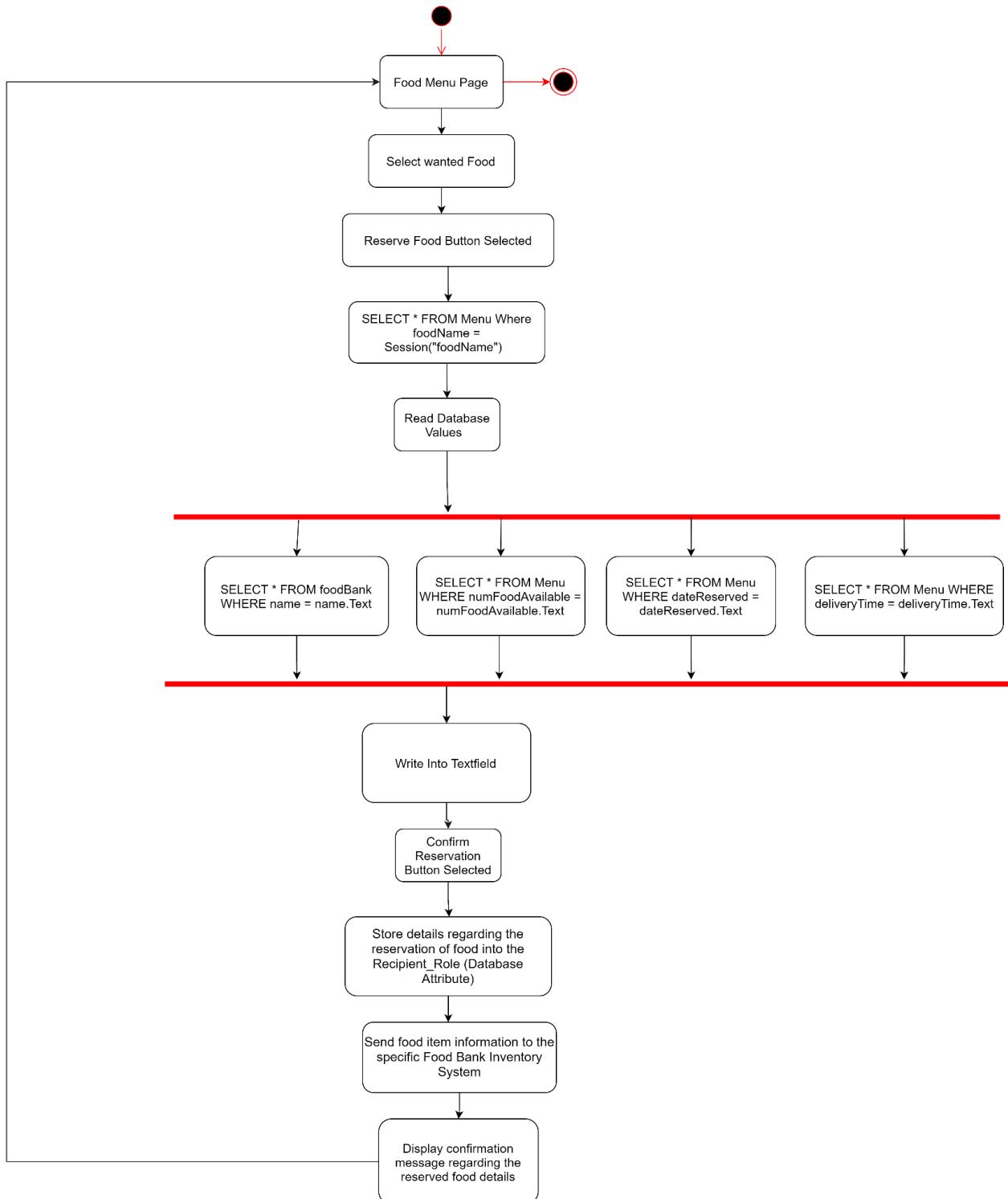


Figure 5.11 Recipient's 'Reserve Food' Activity Diagram

Pseudocode

Select wanted Food

Reserve Food Button Selected

```
SELECT * FROM Menu WHERE foodName = Session("foodName")
```

```
SELECT * FROM foodBank WHERE name = name.Text
```

```
SELECT * FROM Menu WHERE numFoodAvailable = numFoodAvailable.Text
```

```
SELECT * FROM Menu WHERE dateReserved = dateReserved.Text
```

```
SELECT * FROM Menu WHERE deliveryTime = deliveryTime.Text
```

foodBankName.Text = name

foodName.Text = foodName

numFoodAvailable.Text = numFoodAvailable

dateReserved.Text = dateReserved

deliveryTime.Text = deliveryTime

Confirm Reservation Button Selected

Save details regarding the reservation of food into the Recipient_Role(Database Attribute)

Send food item information to the specific Food Bank Inventory System

Displays confirmation message regarding the reserved food details (Food Reservation is successful)

Return to the Food Menu Page

Cancel Reservation of Food

Component Diagram

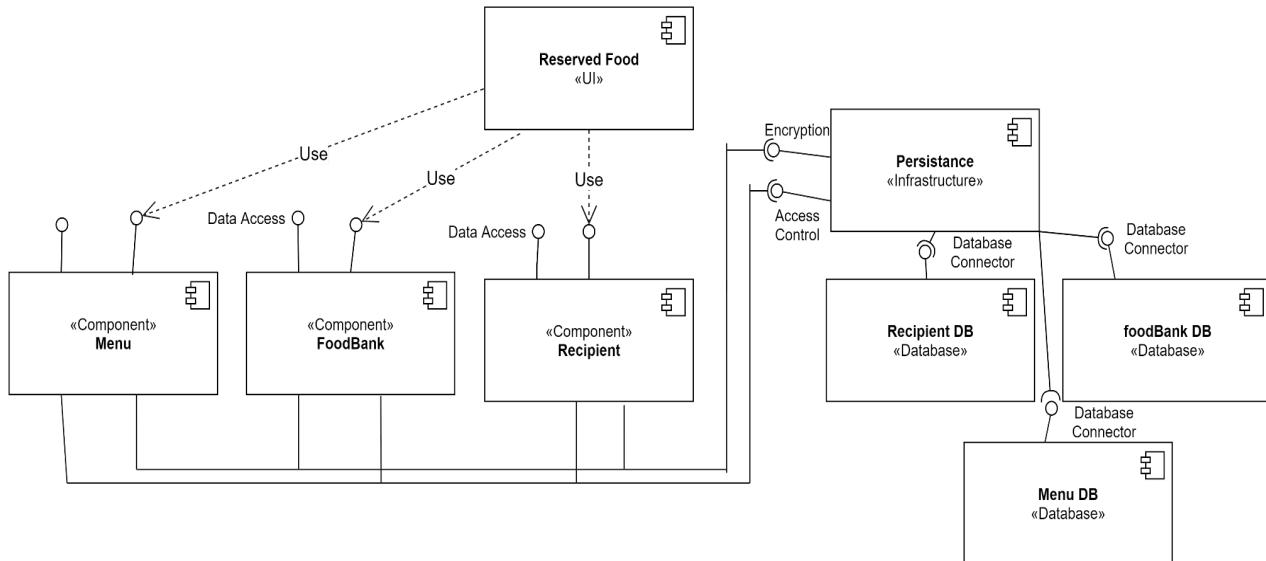


Figure 5.1m Recipient's 'Cancel Reservation of Food' Component Diagram

Activity Diagram

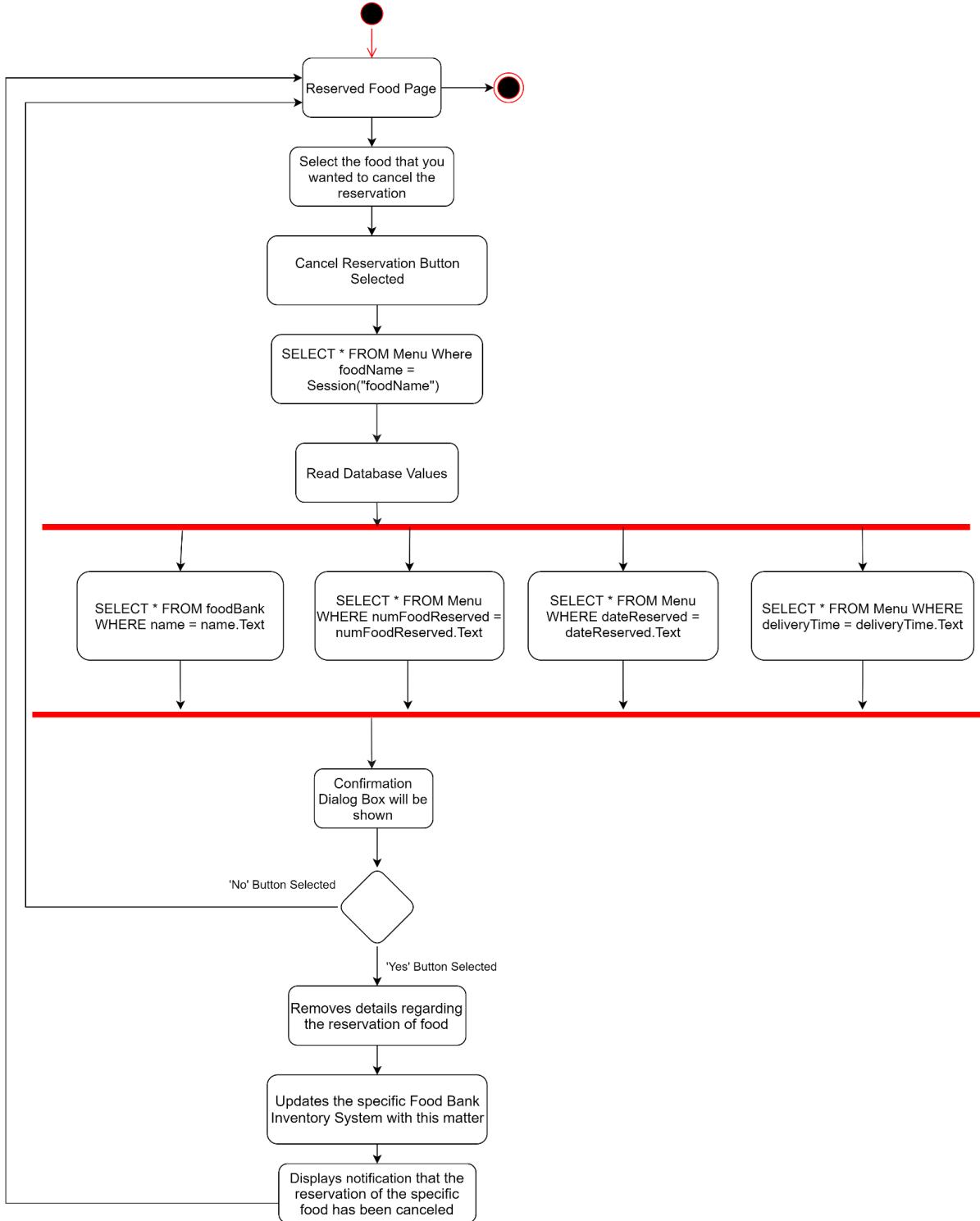


Figure 5.1n Recipient's 'Cancel Reservation of Food' Activity Diagram

Pseudocode

Select the food that you wanted to cancel the reservation
Cancel Reservation Button Selected

```
SELECT * FROM Menu WHERE foodName = Session("foodName")
SELECT * FROM foodBank WHERE name = name.Text
SELECT * FROM Menu WHERE numFoodReserved = numFoodReserved.Text
SELECT * FROM Menu WHERE dateReserved = dateReserved.Text
SELECT * FROM Menu WHERE deliveryTime = deliveryTime.Text
```

foodBankName.Text = name
foodName.Text = foodName
numFoodReserved.Text = numFoodReserved
dateReserved.Text = dateReserved
deliveryTime.Text = deliveryTime

Confirmation Dialog Box will be shown

IF ('Yes' Button Selected) THEN

Removes details regarding the reservation of food from the Recipient_Role(Database Attribute)
Updates the specific Food Bank Inventory System with this matter
Displays notification that the reservation of the specific food has been canceled (Cancel
Reservation of food is successful)
Return to the Reserved Food Page

ELSE IF ('No' Button Selected) THEN

Return to the Reserved Food Page

END IF

View Community Feed

Component Diagram

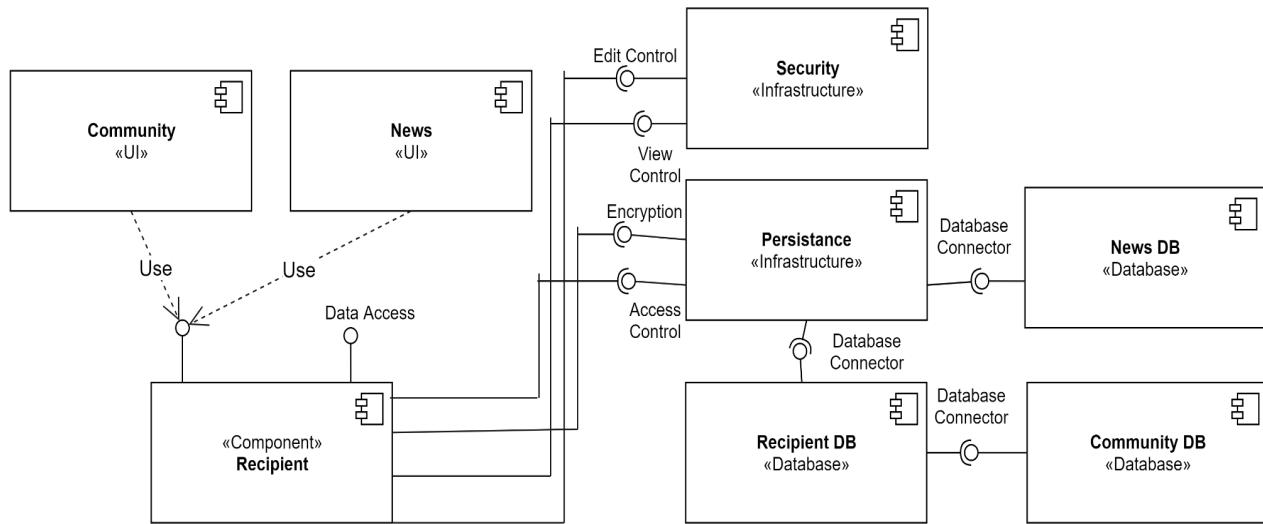


Figure 5.1o Recipient's 'View Community Feed' Component Diagram

Activity Diagram

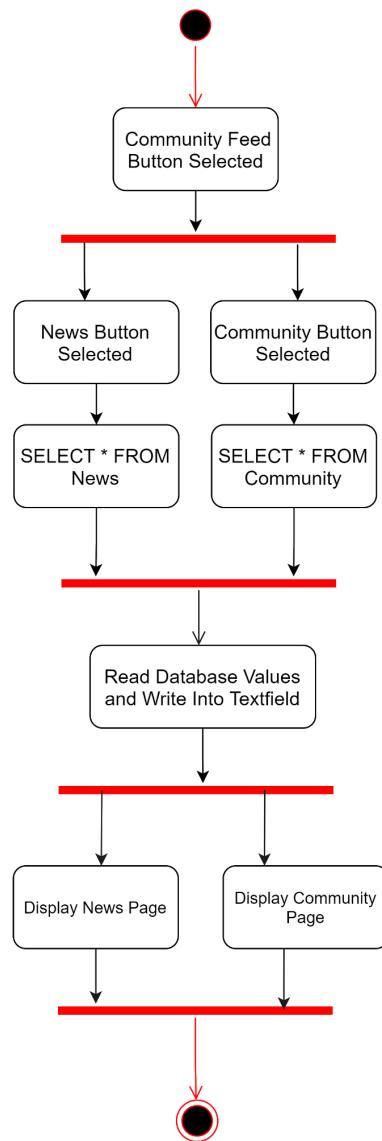


Figure 5.1p Recipient's 'View Community Feed' Activity Diagram

Pseudocode

Community Feed Button Selected

IF (News Button Selected) **THEN**

 SELECT * FROM News

 Display News Page

ELSE IF (Community Button Selected) **THEN**

 SELECT * FROM Community

 Display Community Page

END IF

Acquiring Assistance

Component Diagram

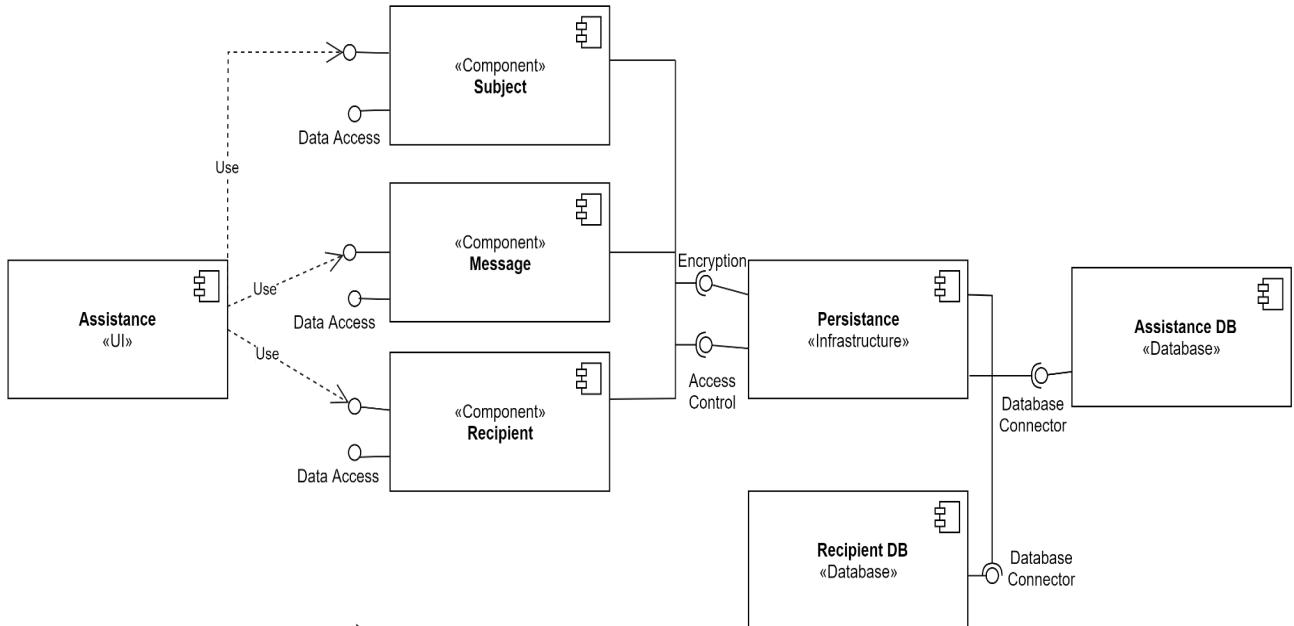


Figure 5.1q Recipient's 'Acquiring Assistance' Component Diagram

Activity Diagram

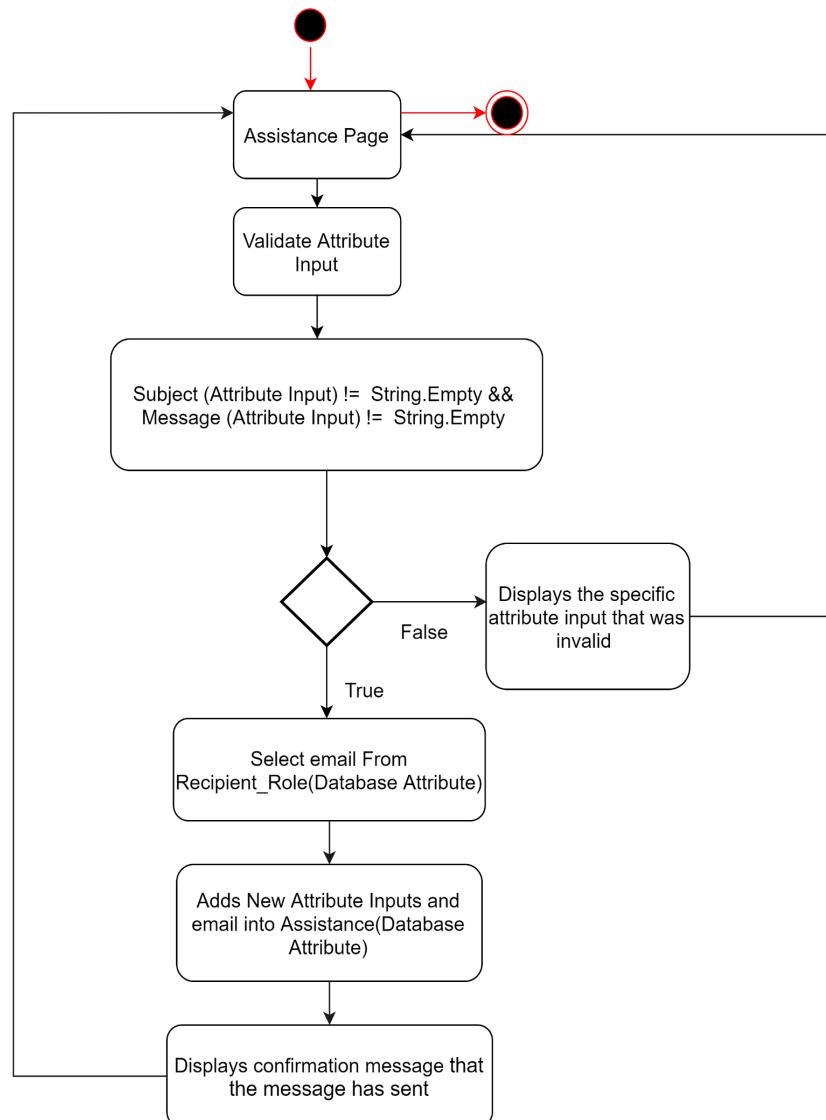


Figure 5.1r Recipient's 'Acquiring Assistance' Activity Diagram

Pseudocode

GET Subject
GET Message

IF (Entered Subject != String.Empty && Entered Message != String.Empty) **THEN**

Select email From Recipient_Role(Database Attribute)
Save New Attributes and email into the Assistance(Database Attribute)
Displays confirmation message that the message has sent (Message Sending Successful)
Return to the Assistance Page

ELSE

Message Sending Failed
Displays the specific attribute input that was invalid
Return to the Assistance page

END IF

Donor

Sign up

Component Diagram

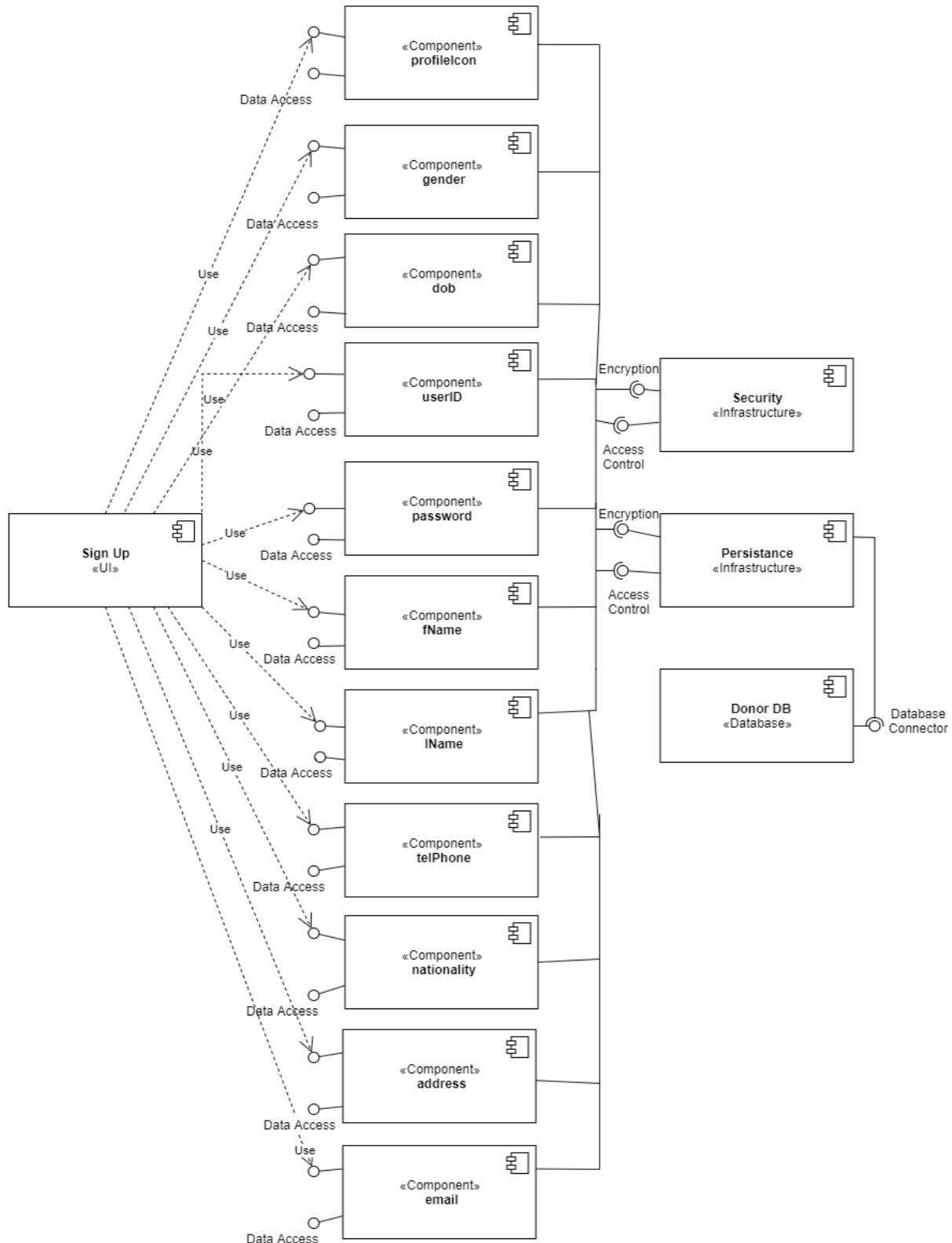


Figure 5.2a Donor's 'Sign Up' Component Diagram

Activity Diagram

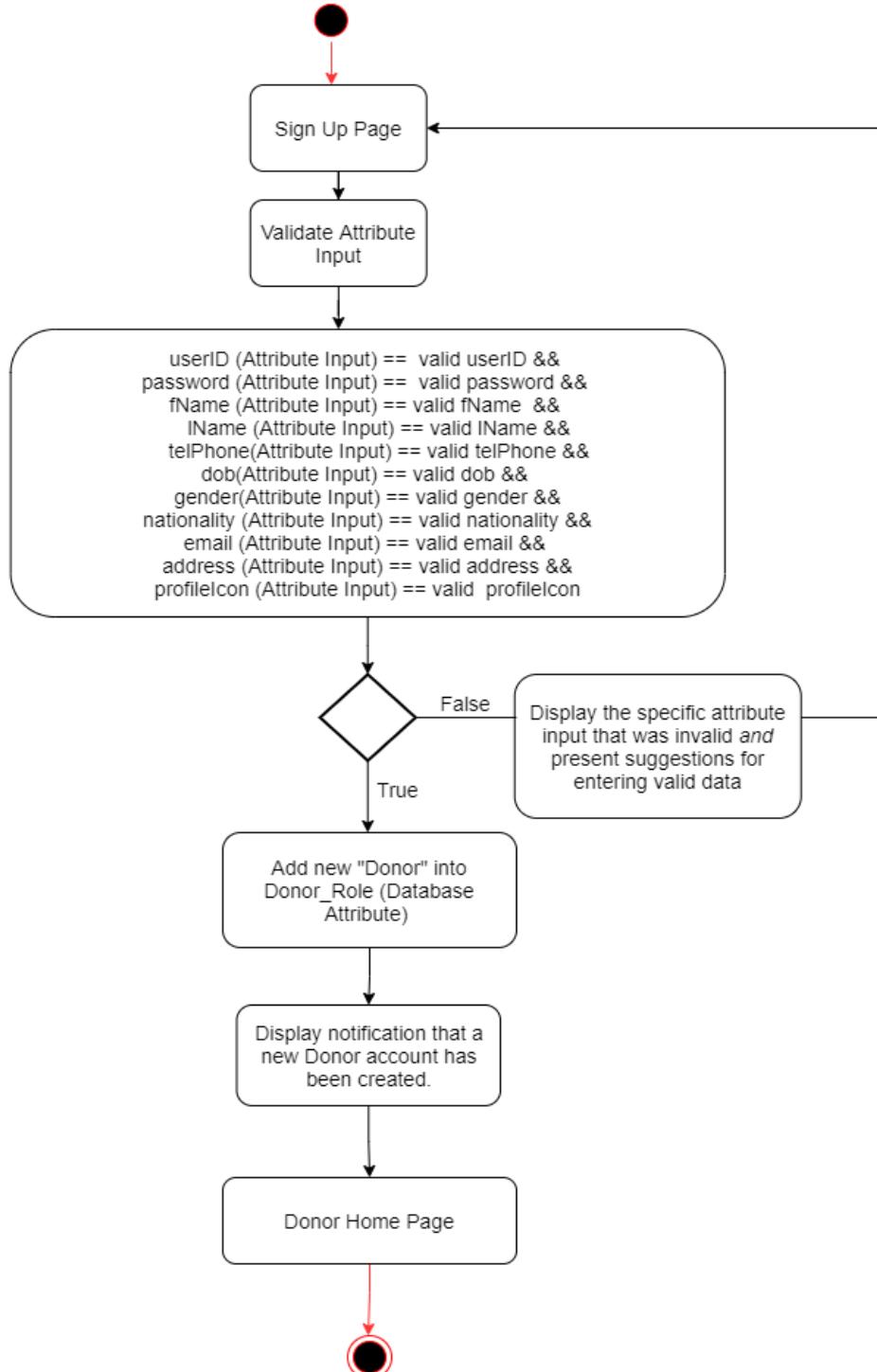


Figure 5.2b Donor's 'Sign Up' Activity Diagram

Pseudocode

GET userID
GET password
GET fName
GET lName
GET telPhone
GET dob
GET gender
GET nationality
GET email
GET address
GET profileIcon

IF (Entered userID == valid userID && Entered password == valid password && Entered fName == valid fName && Entered lName == valid lName && Entered telPhone == valid telPhone && Entered dob == valid dob && Entered gender == valid gender && Entered nationality == valid nationality && Entered email == valid email && Entered address == valid address && Entered profileIcon == valid profileIcon) **THEN**

Save New Donor's Details into the Donor_Role(Database Attribute)

Display notification that a new Donor account has been created (Sign Up Successful)

Go to the Donor Home Page

ELSE

Sign Up Failed

Display the specific attribute input that was invalid *and* present suggestions for entering valid data
Return to the Sign Up page

END IF

Login

Component Diagram

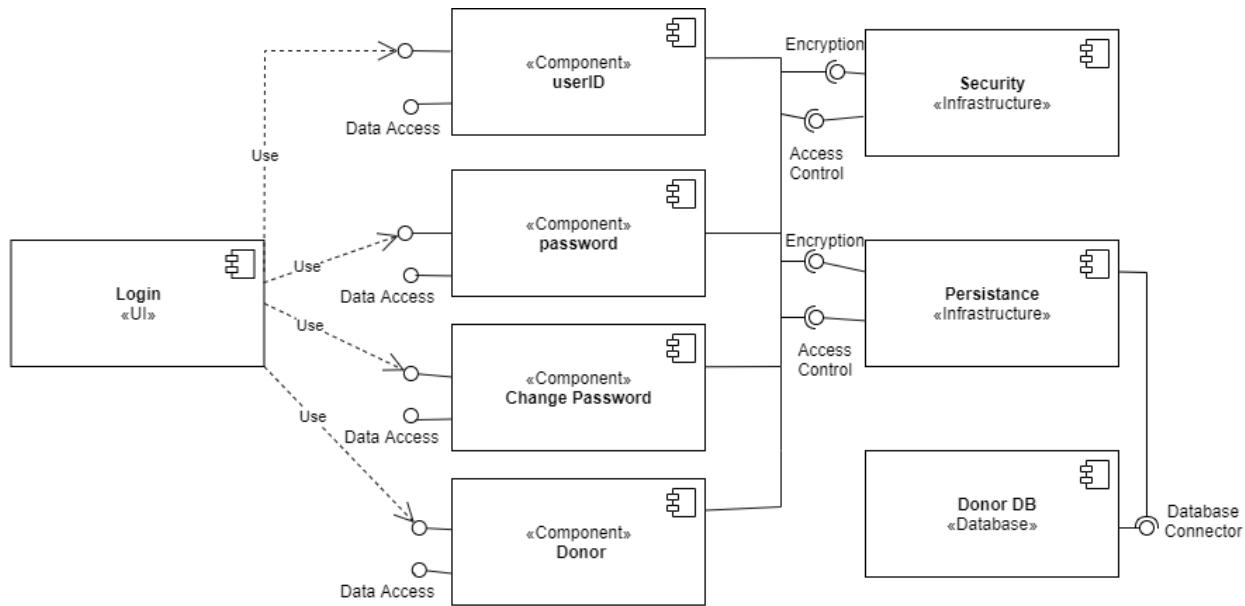


Figure 5.2c Donor's 'Login' Component Diagram

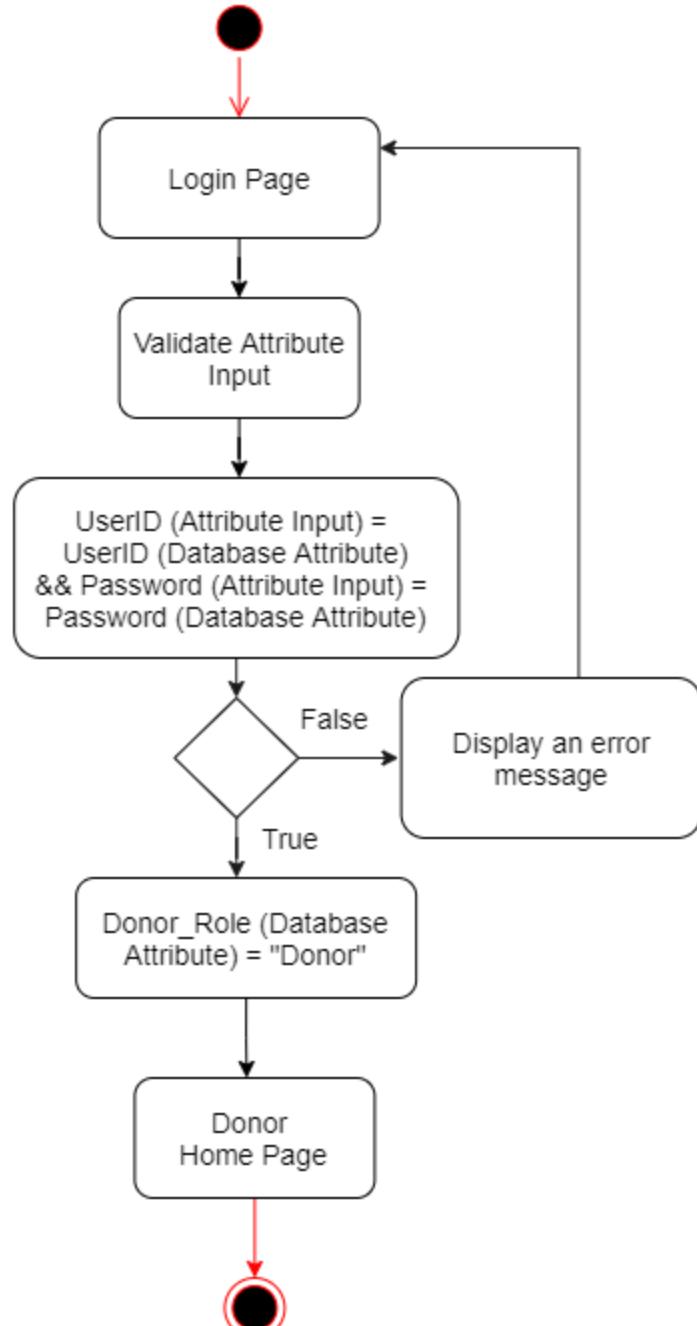
Activity Diagram

Figure 5.2d Donor's 'Login' Activity Diagram

Pseudocode

GET userID
GET password

IF (userID == EnteredUserID && password == EnteredPassword) **THEN**
 Login Successful
 Go to the Donor Home Page

ELSE
 Login Failed
 Return to the login page

END IF

Donate Food

Component Diagram

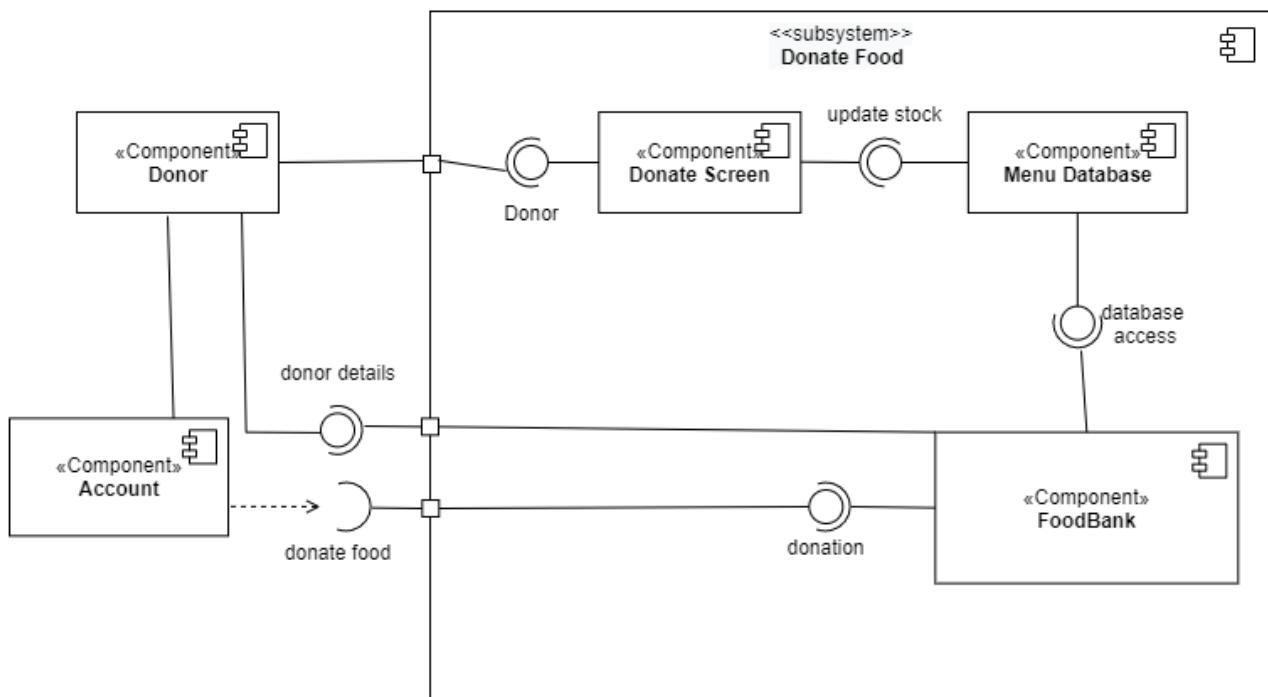


Figure 5.2e Donor's 'Donate' Component Diagram

Activity Diagram

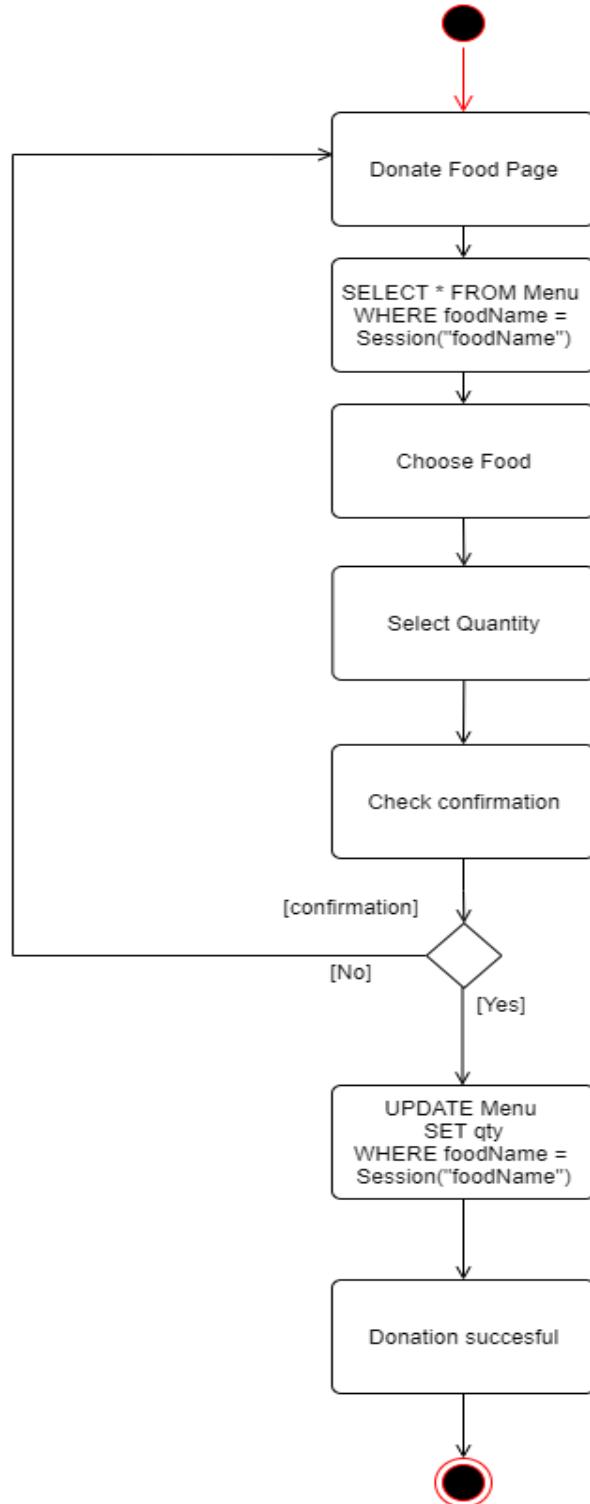


Figure 5.2f Donor's 'Donate Food' Activity Diagram

Pseudocode

BEGIN

Select food to donate

Donate food Button Selected

```
SELECT * FROM Menu WHERE foodName = Session("foodName")
```

foodBankName.Text = name

Confirm donation Button Selected

Save details regarding the donation of food into the Donor_Role(Database Attribute)

Update food item information to the specific Food Bank Inventory System

Display successful message

Donate Money

Component Diagram

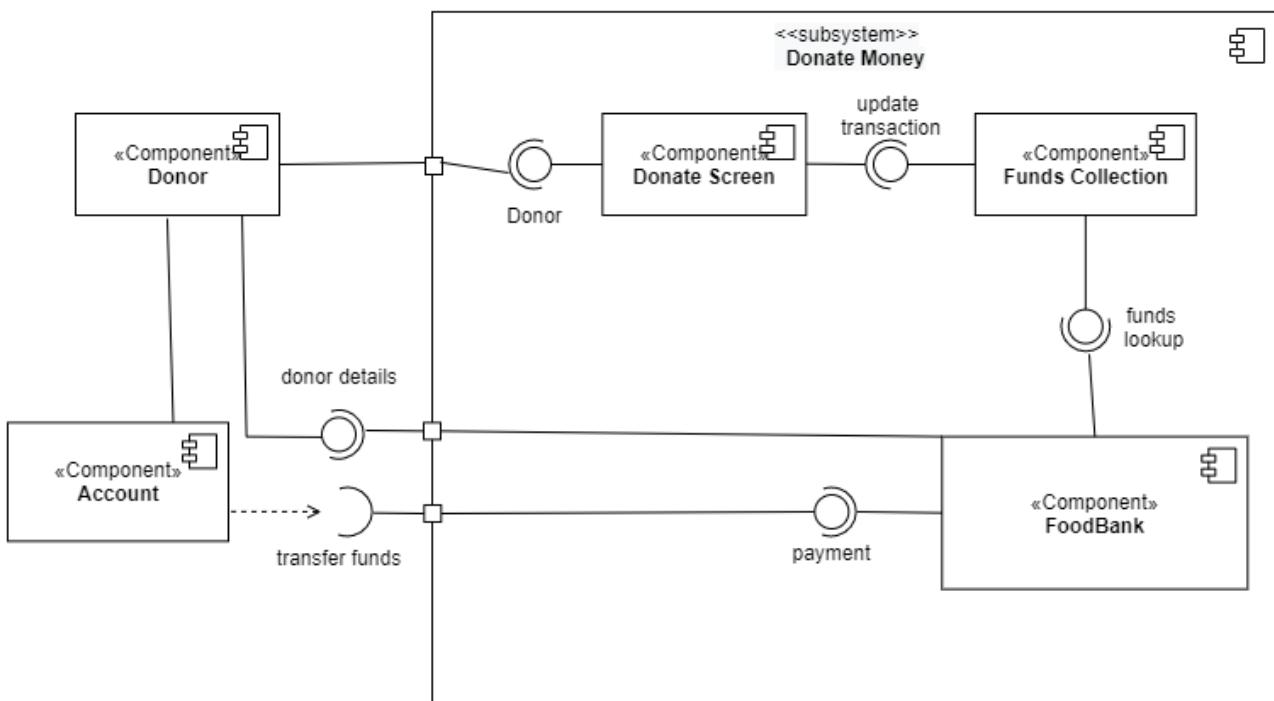


Figure 5.2g Donor's 'Donate Money' Component Diagram

Activity Diagram

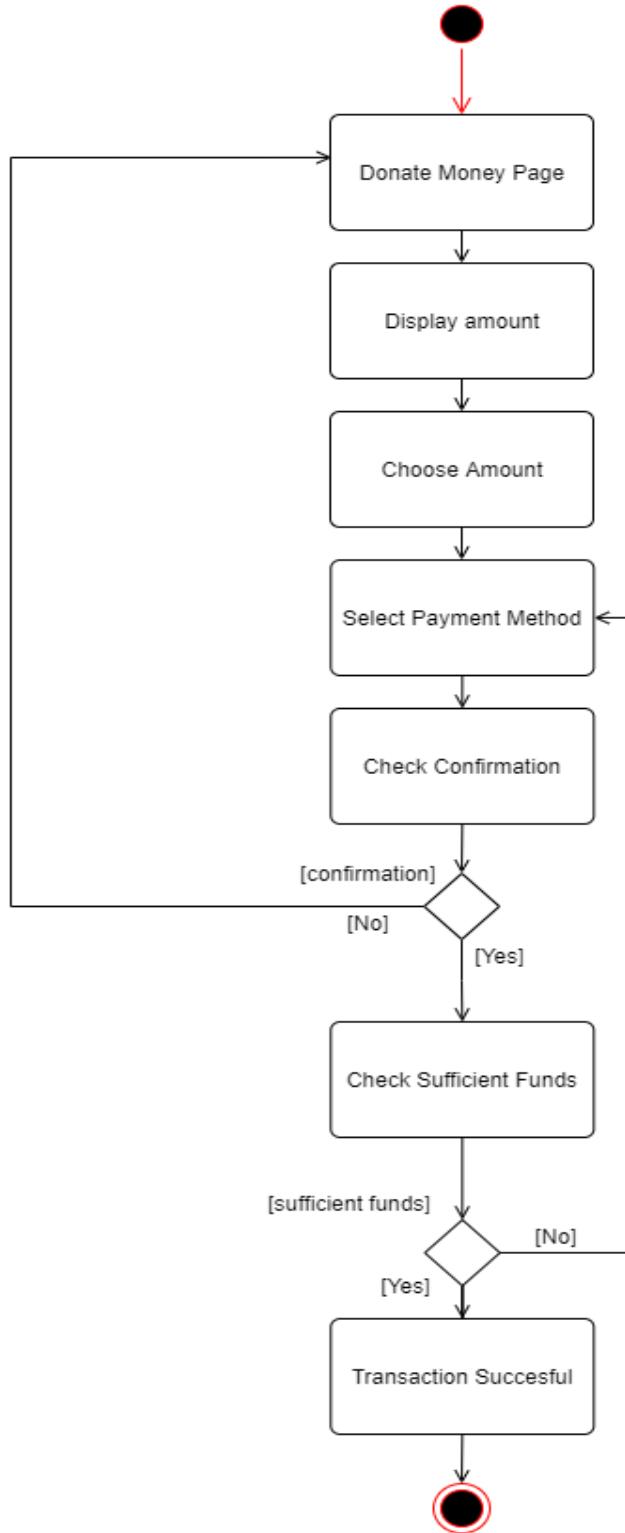


Figure 5.2h Donor's 'Donate Food' Activity Diagram

Pseudocode

```
BEGIN  
  
GET Choice  
GET Balance  
IF (Choice == Amount) THEN  
    IF(Balance < Amount) THEN  
        PRINT "Insufficient Balance"  
    ELSE  
        totalFunds ← totalFunds + Amount  
  
    END IF  
  
ELSE  
  
    PRINT "Incorrect funds"  
    Return to start  
  
END IF  
  
END
```

View history of donations

Component Diagram

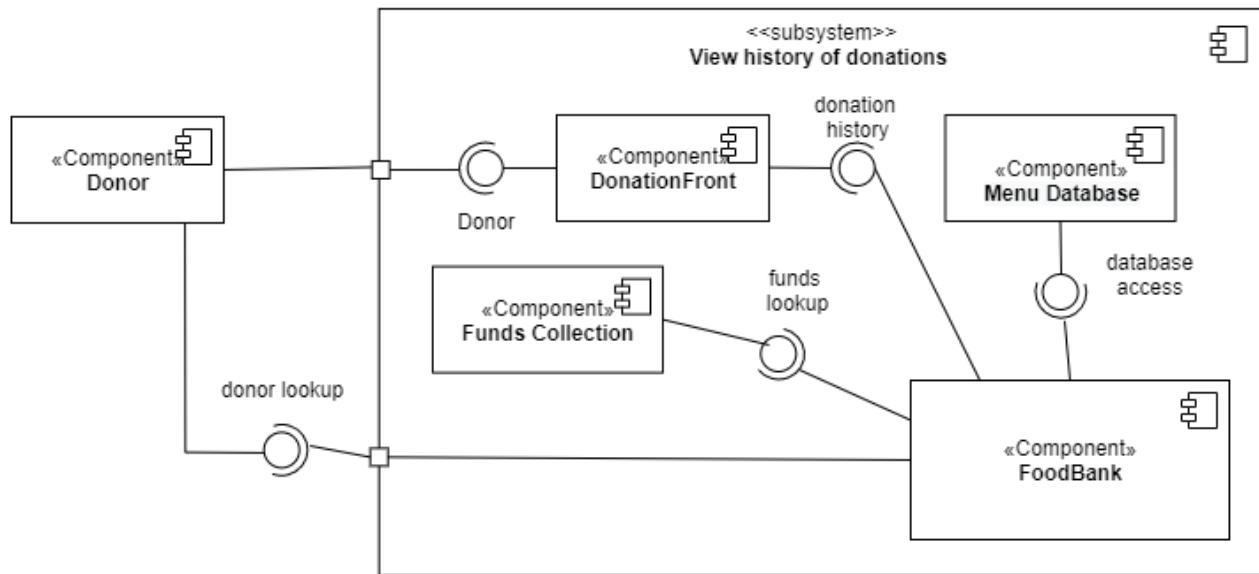


Figure 5.2i Donor's 'History of Donation' Component Diagram

Activity Diagram

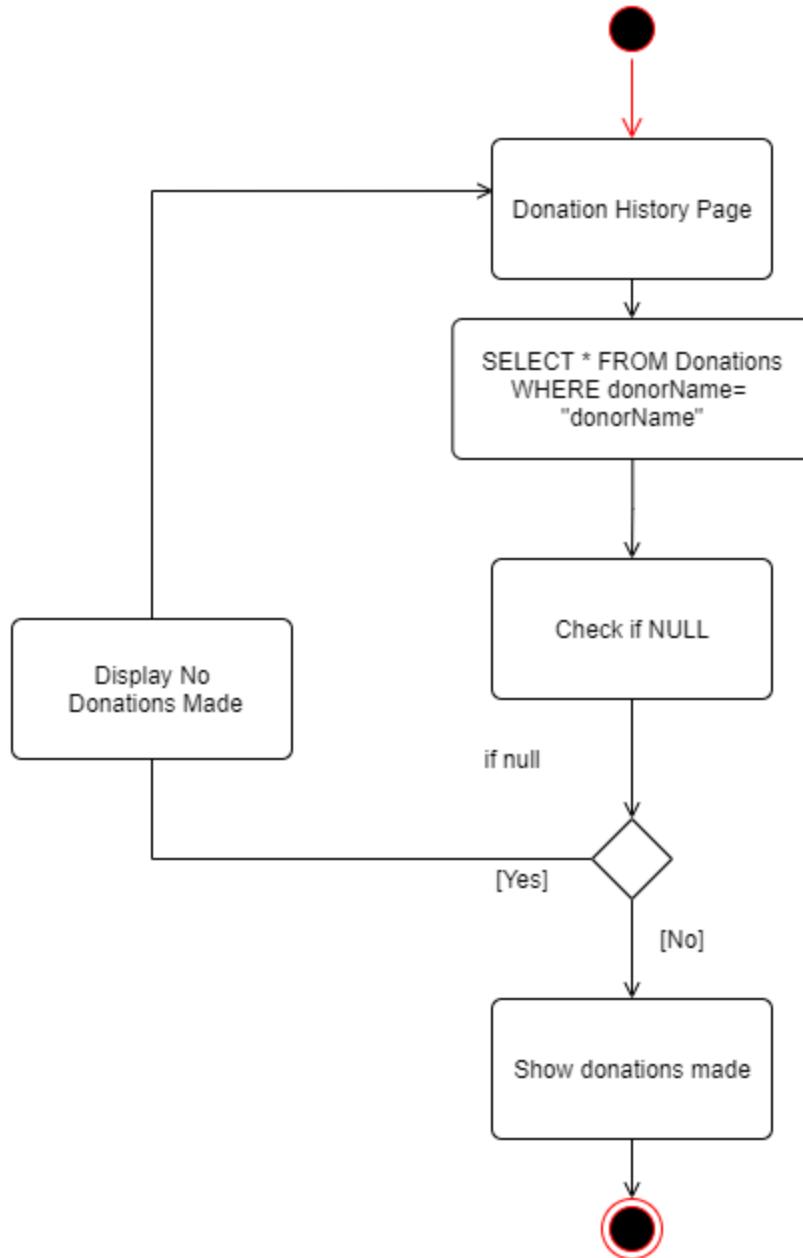


Figure 5.2j Donor's 'History of Donation' Activity Diagram

Pseudocode

BEGIN

IF (View history button selected) THEN
WHILE(SELECT*FROM Donations != NULL)
 PRINT Donations Made
 Display Community Page

END IF

END

View foods low on stock

Component Diagram

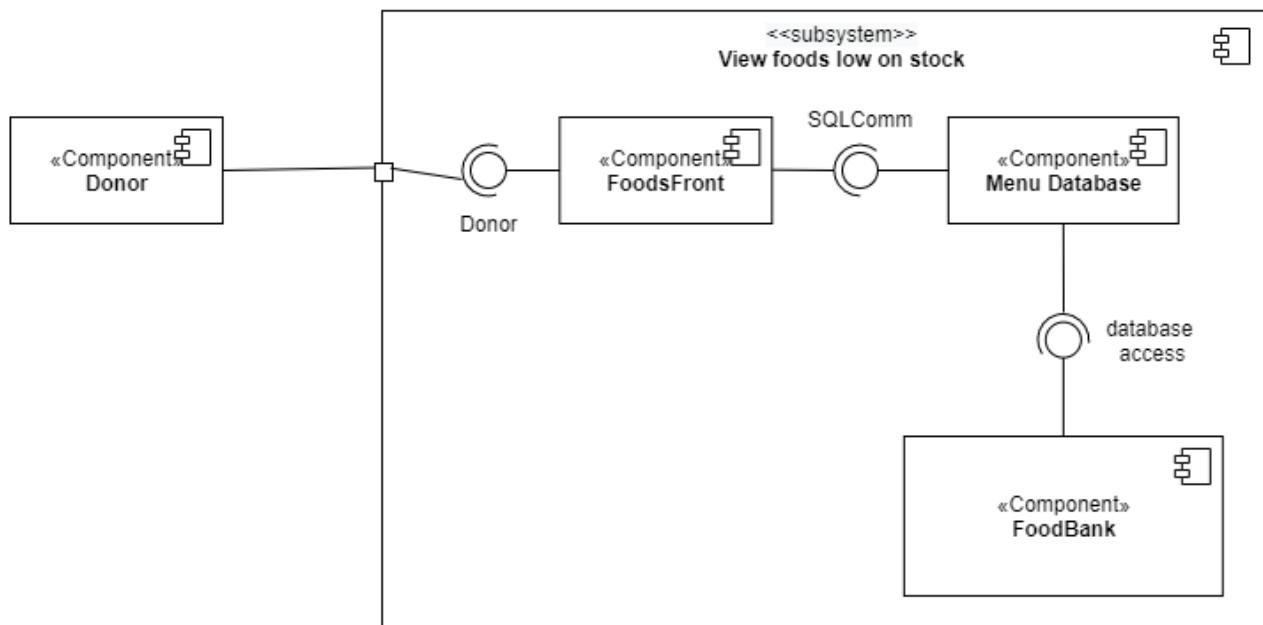


Figure 5.2k Donor's 'Foods low on stock' Component Diagram

Activity Diagram

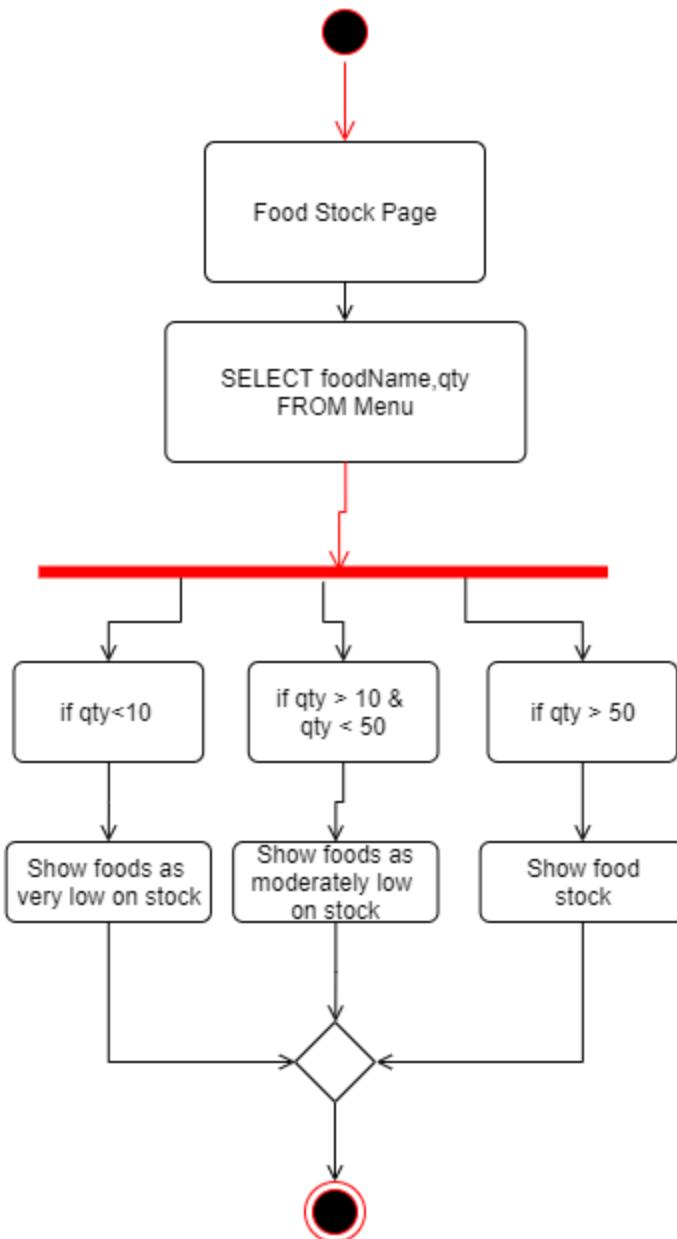


Figure 5.21 Donor's 'Foods low on stock' Activity Diagram

Pseudocode

```

BEGIN

IF (Show Food Stock page selected) THEN
    SELECT foodName, qty
    FROM Menu
    IF (qty< 10) THEN
        PRINT foods very low on stock
    ELSE IF (qty >10 & qty < 50 ) THEN
        PRINT foods moderately low on stock
    ELSE
        PRINT food stock
    END IF

END IF

END

```

Find food bank locations

Component Diagram

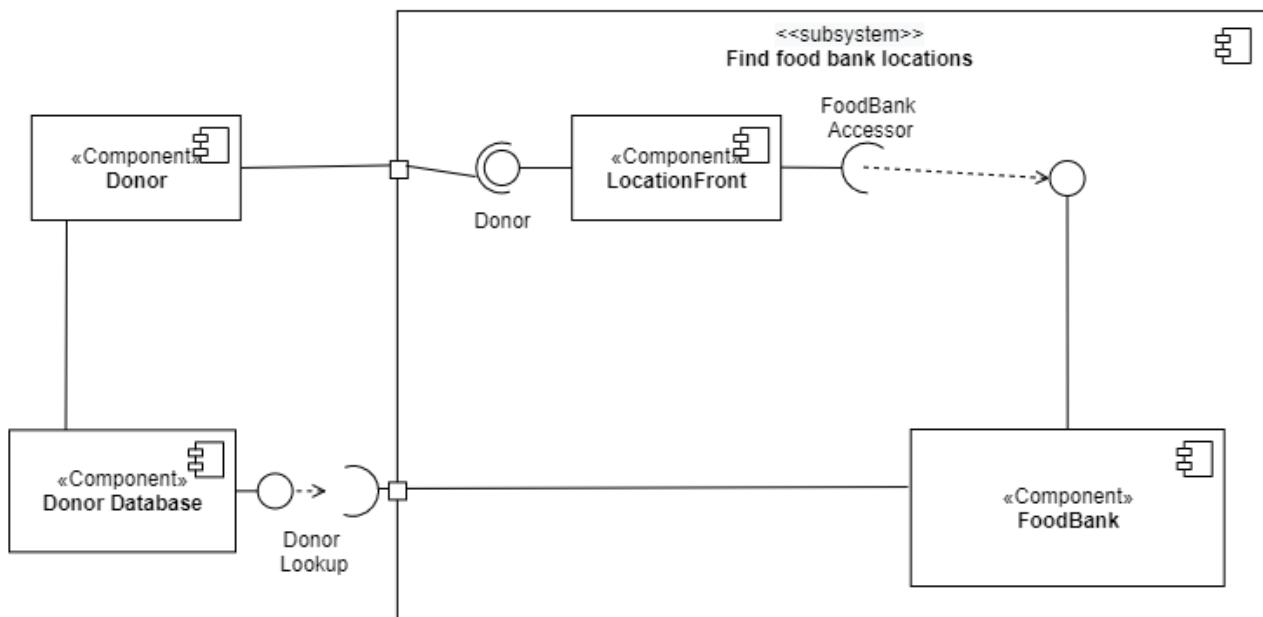


Figure 5.2m Donor's 'Find Food Bank' Component Diagram

Activity Diagram

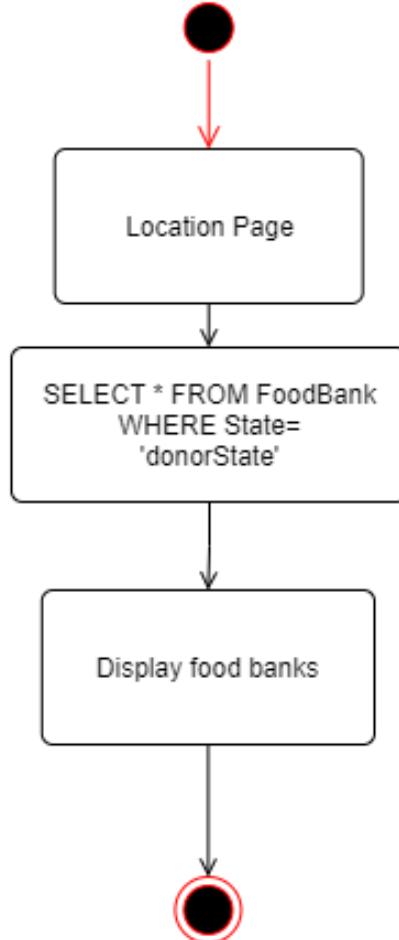


Figure 5.2n Donor's 'Find Food Bank' Activity Diagram

Pseudocode

```
BEGIN
  IF( Location Page selected) THEN
    SELECT * FROM FoodBank WHERE State = "donorState"
    PRINT Food Banks
  END IF
END
```

Distributor

Signup

Component Diagram

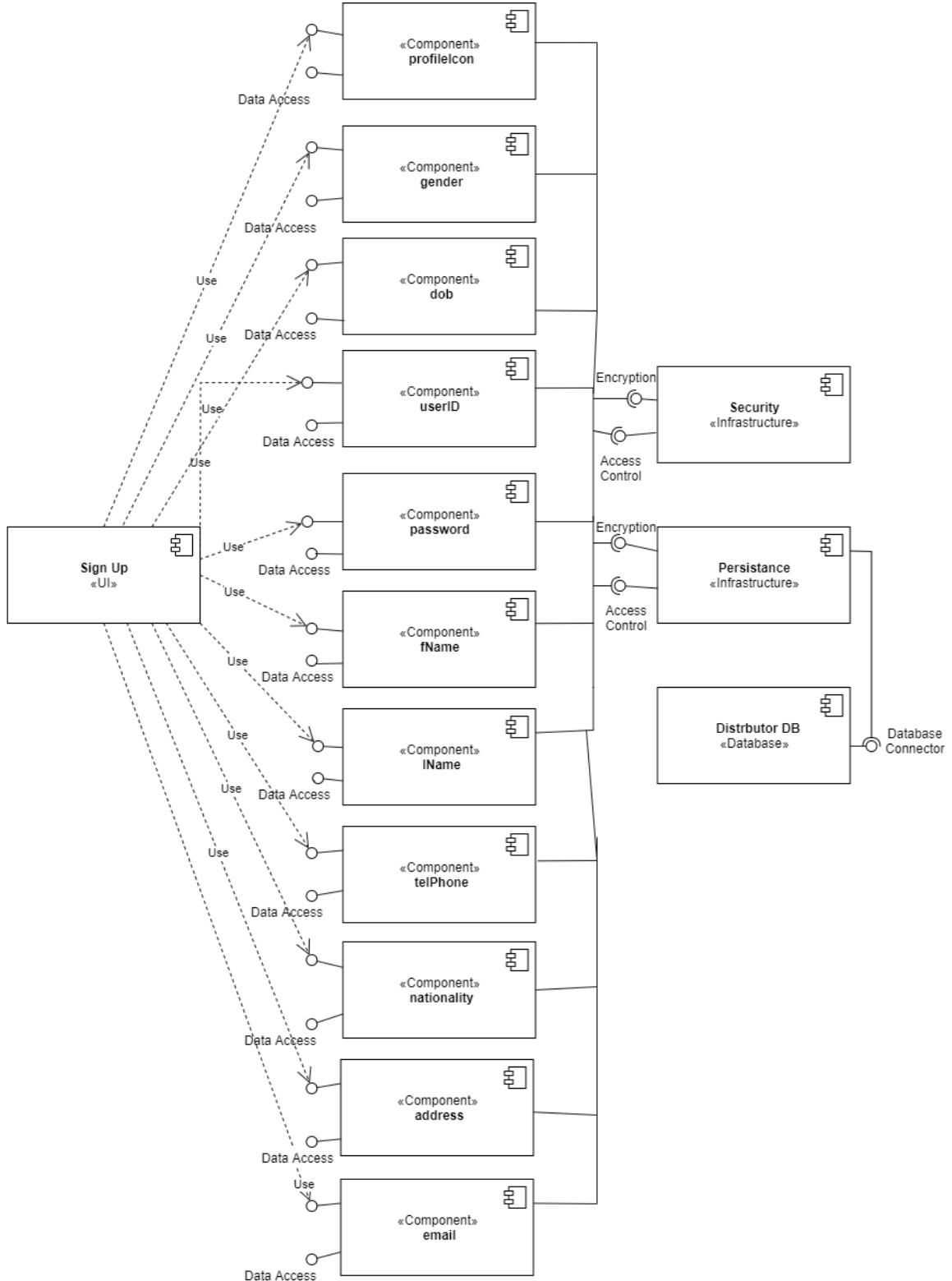


Figure 5.3a Distributor's "Signup" component diagram

Activity Diagram

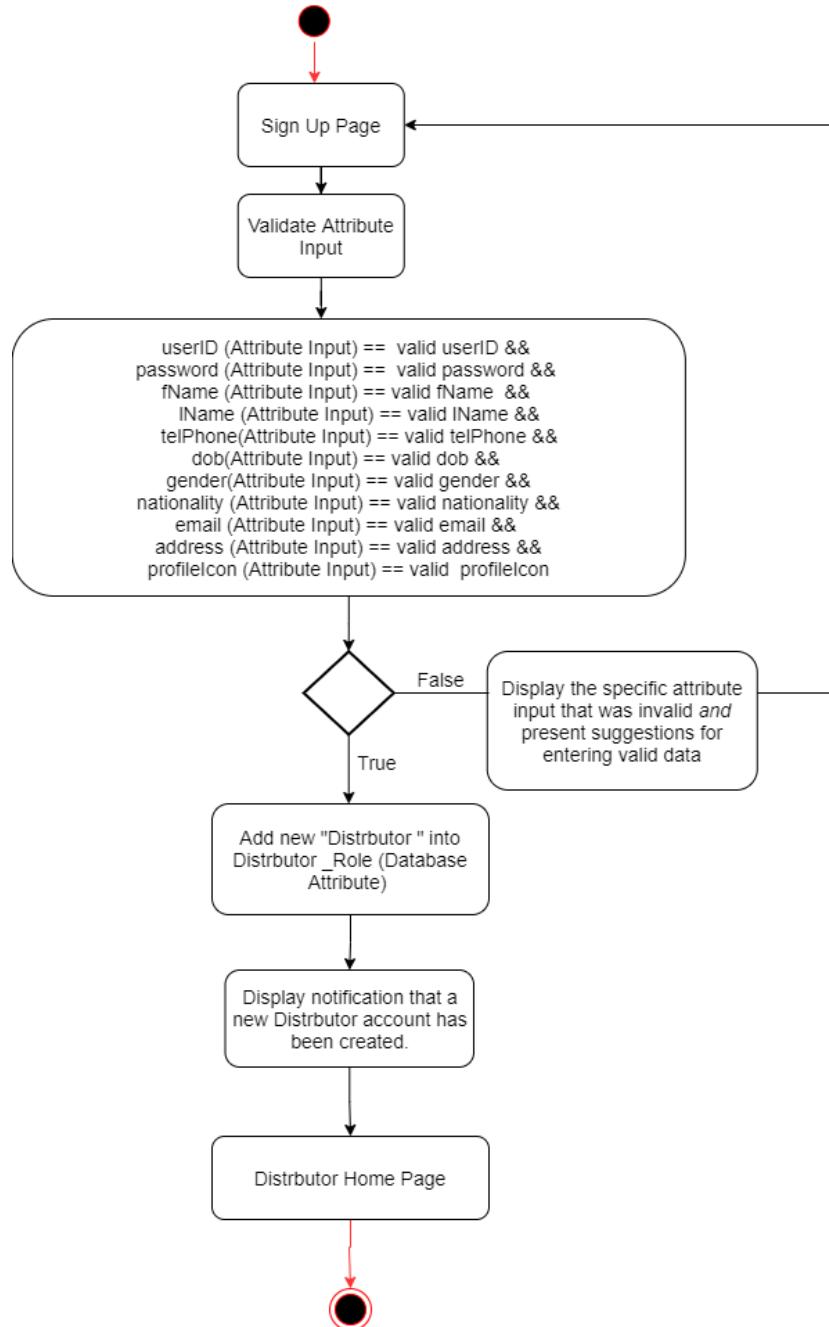


Figure 5.3b Distributor's "Signup" activity diagram

Psuedocode

GET userID
GET password
GET fName
GET lName
GET telPhone
GET dob
GET gender
GET nationality
GET email
GET address
GET profileIcon

IF (Entered userID == valid userID && Entered password == valid password && Entered fName == valid fName && Entered lName == valid lName && Entered telPhone == valid telPhone && Entered dob == valid dob && Entered gender == valid gender && Entered nationality == valid nationality && Entered email == valid email && Entered address == valid address && Entered profileIcon == valid profileIcon) **THEN**

Save New Distributor's Details into the Distributor_Role(Database Attribute)
Display notification that a new Distributor account has been created (Sign Up Successful)
Go to the Distributor Home Page

ELSE

Sign Up Failed
Display the specific attribute input that was invalid *and* present suggestions for entering valid data
Return to the Sign Up page

END IF

Login

Component Diagram

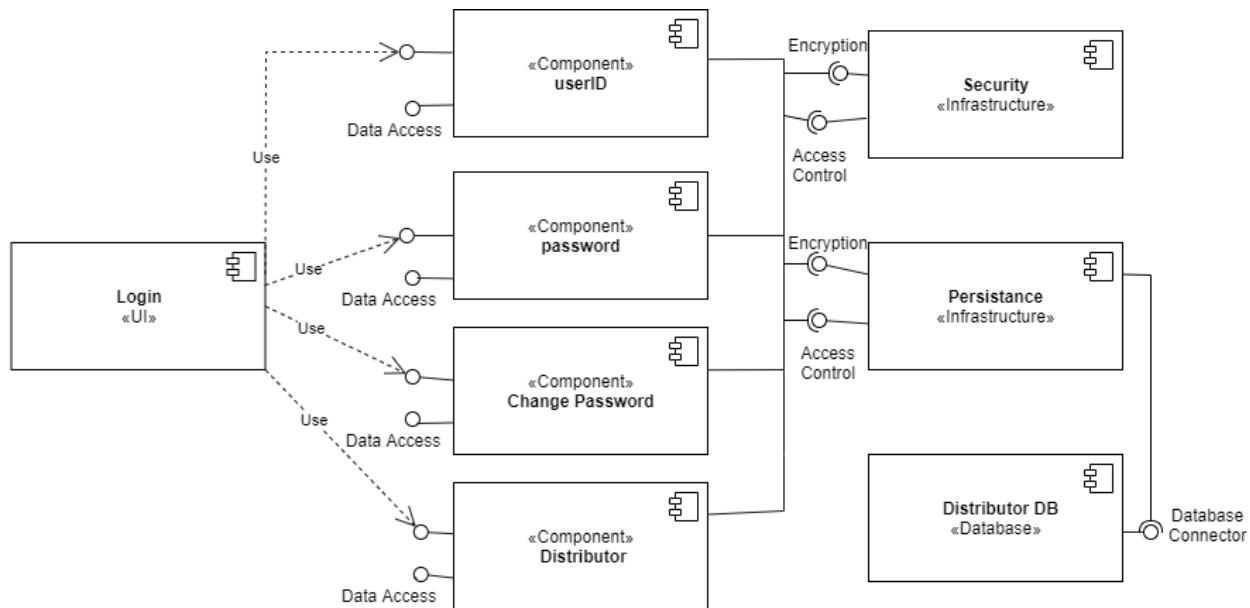


Figure 5.3c Distributor's "Login" component diagram

Activity Diagram

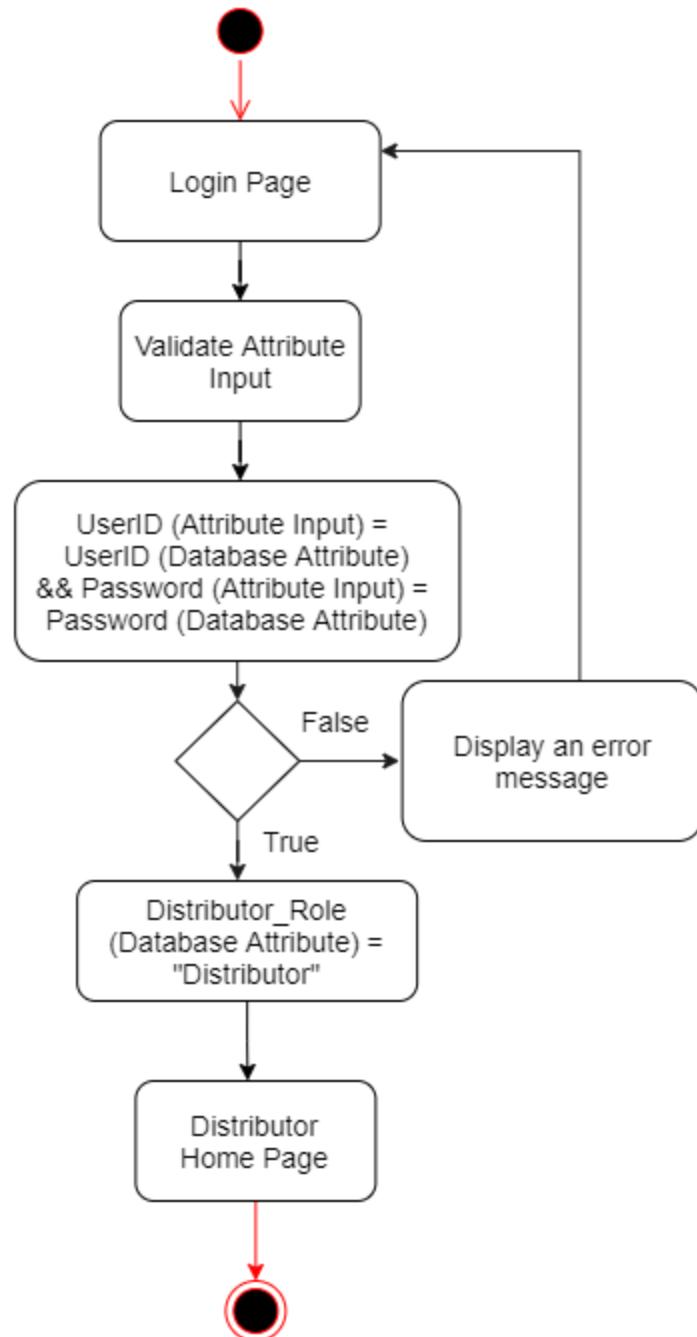


Figure 5.3d Distributor's "Login" activity diagram

Pseudocode

GET userID
GET password

IF (userID == EnteredUserID && password == EnteredPassword) **THEN**

 Login Successful

 Go to the Distributor Home Page

ELSE

 Login Failed

 Return to the login page

END IF

Search for nearby foodbank locations

Component Diagram

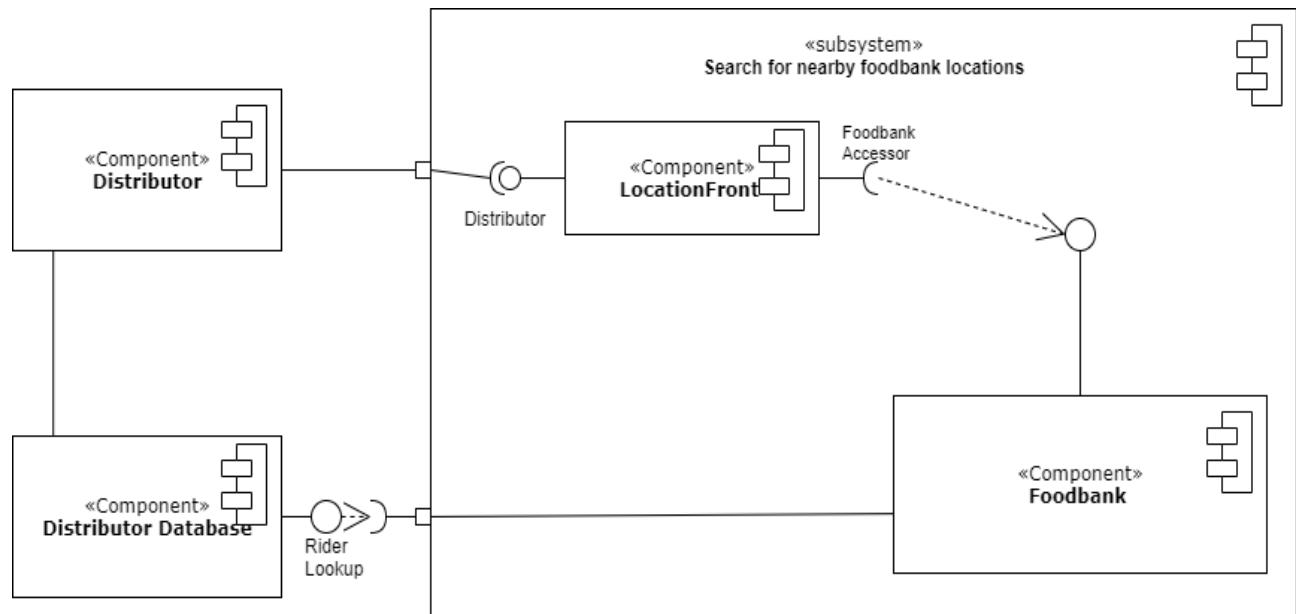


Figure 5.3e Distributor's 'Search for nearby foodbank locations' component diagram

Activity Diagram

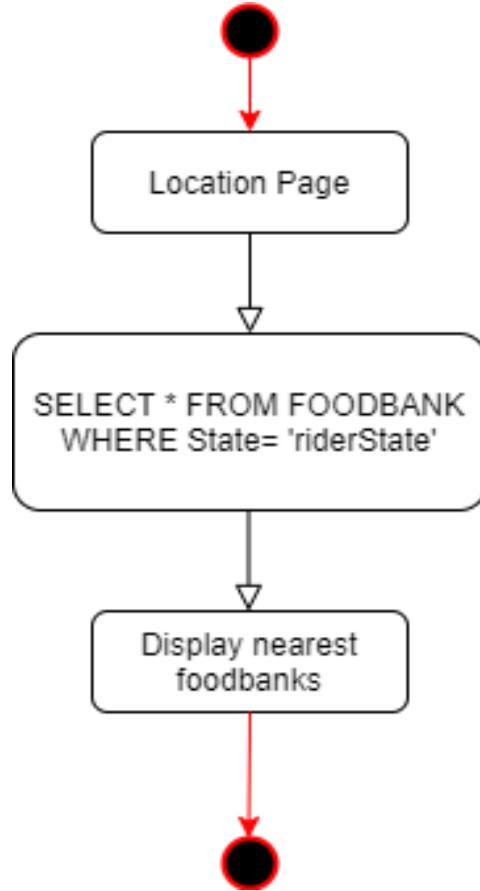


Figure 5.3f Distributor's 'Search for nearby foodbank locations' activity diagram

Pseudocode

```
BEGIN
  IF( Location Page selected) THEN
    SELECT * FROM FoodBank WHERE State = "riderState"
    PRINT Food Banks
  END IF
END
```

Check status of reserved food

Component Diagram

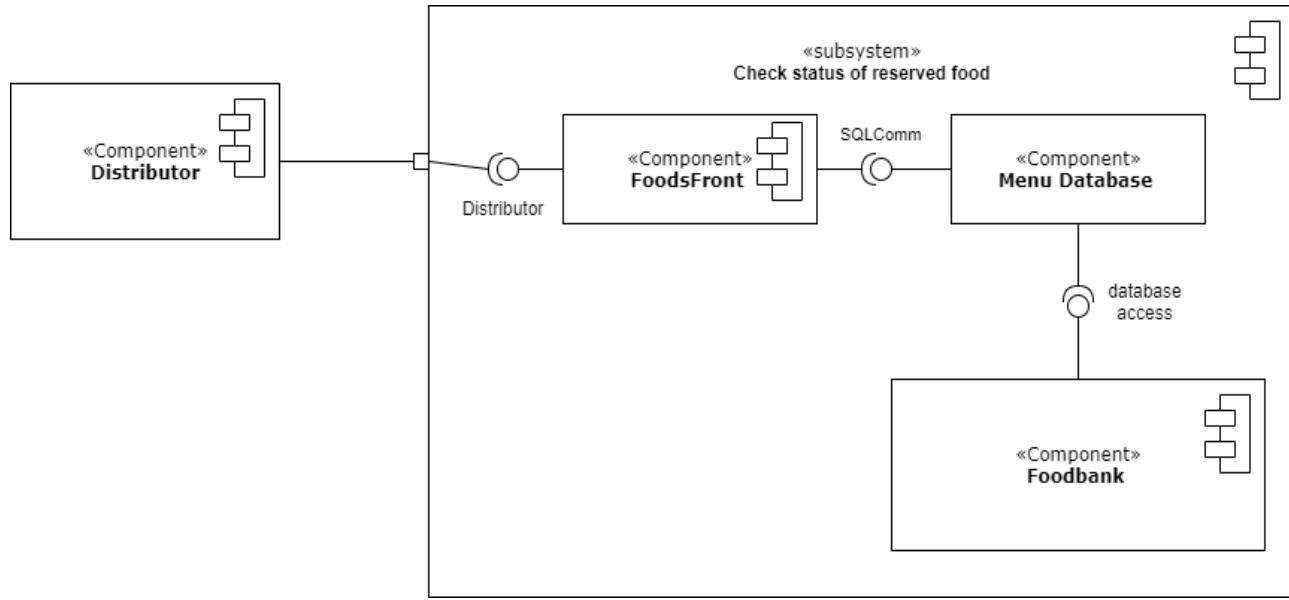


Figure 5.3g Distributor's 'Check status of reserved food' component diagram

Activity Diagram

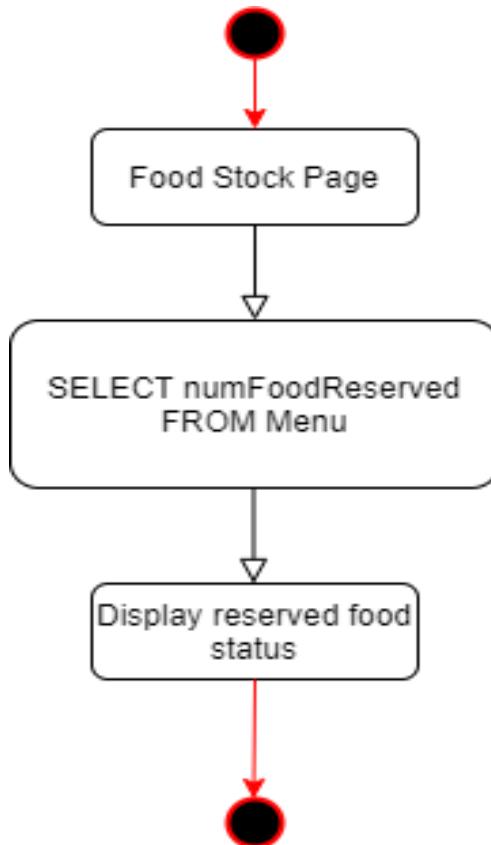


Figure 5.3h Distributor's 'Check status of reserved food' activity diagram

Pseudocode

```
BEGIN  
  IF (Show Food Stock page selected) THEN  
    SELECT numFoodReserved  
    FROM Menu  
    PRINT reserved food status  
  END IF  
  
END
```

Pick up food

Component Diagram

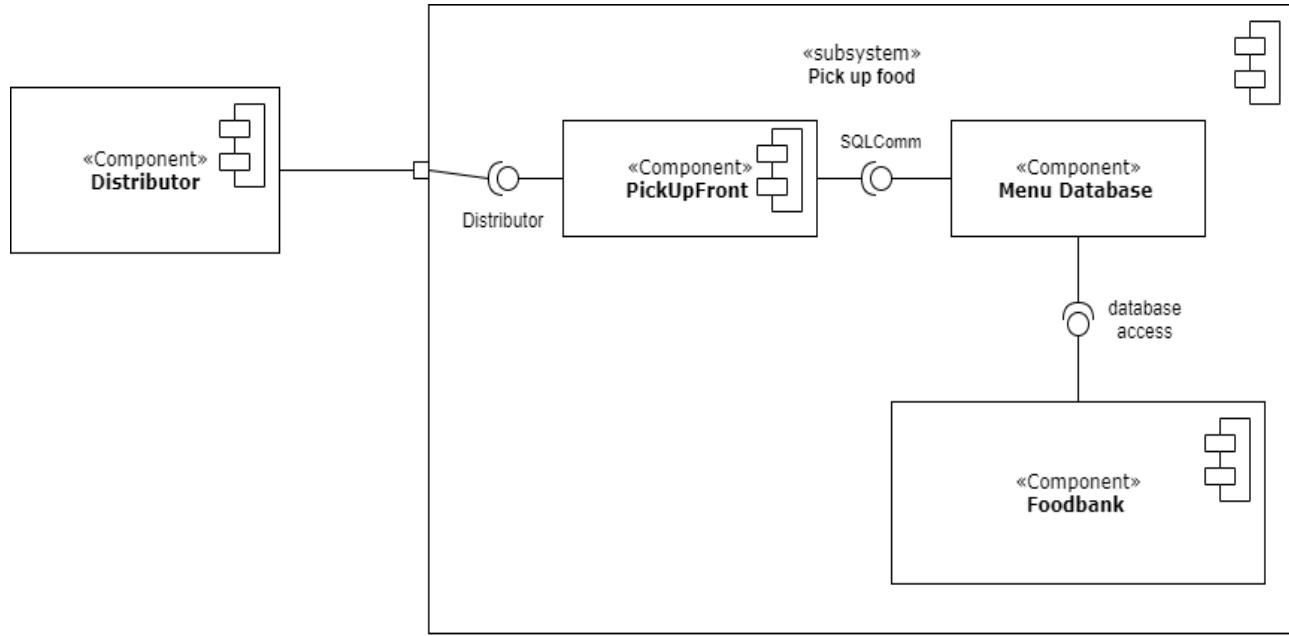


Figure 5.3i Distributor's 'Pick up Food' component diagram

Activity Diagram

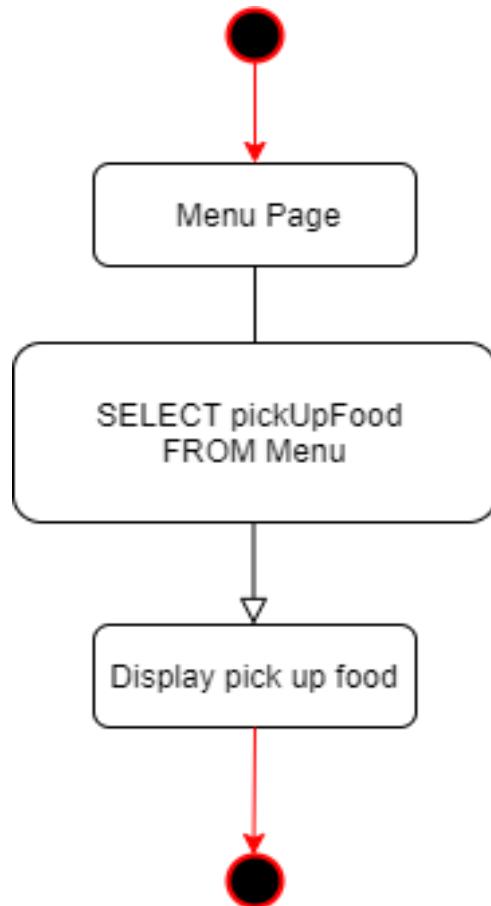


Figure 5.3j Distributor's 'Pick up Food' activity diagram

Pseudocode

BEGIN

IF (Show Menu page selected) THEN
 SELECT pickUpFood
 FROM Menu
 PRINT Pick up food
END IF

END

View delivery status

Component Diagram

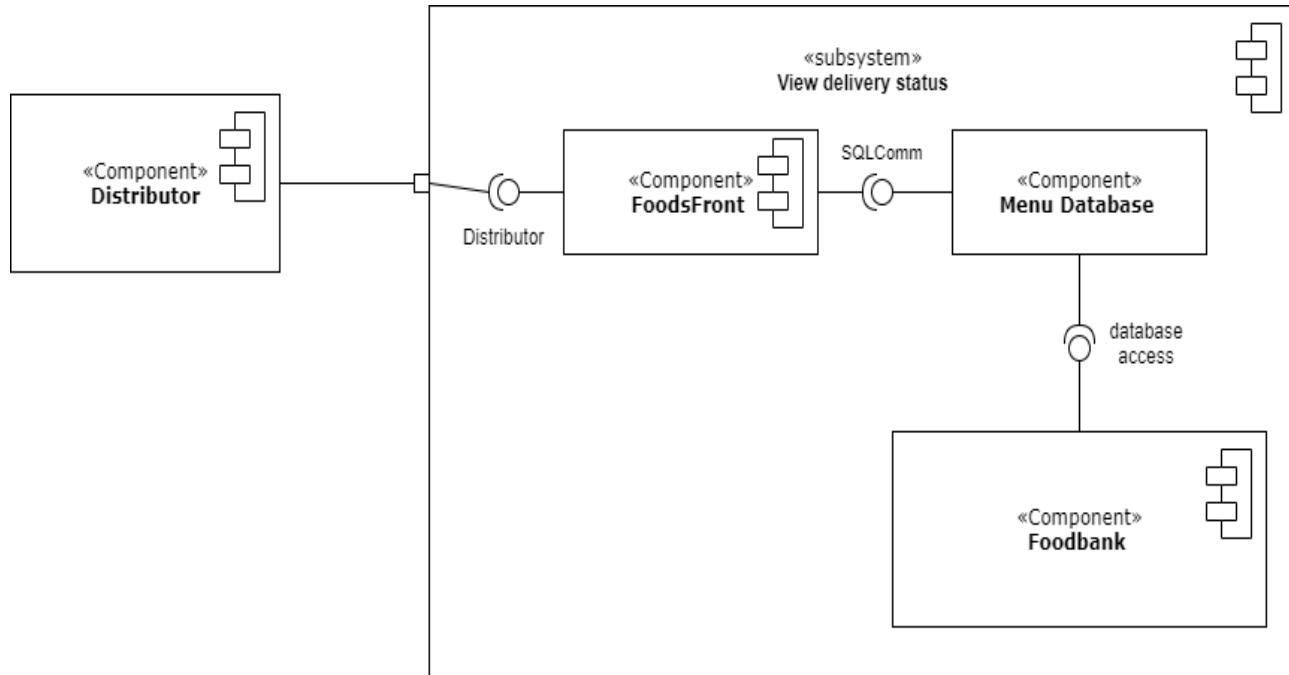


Figure 5.3k Distributor's 'View delivery status' component diagram

Activity Diagram

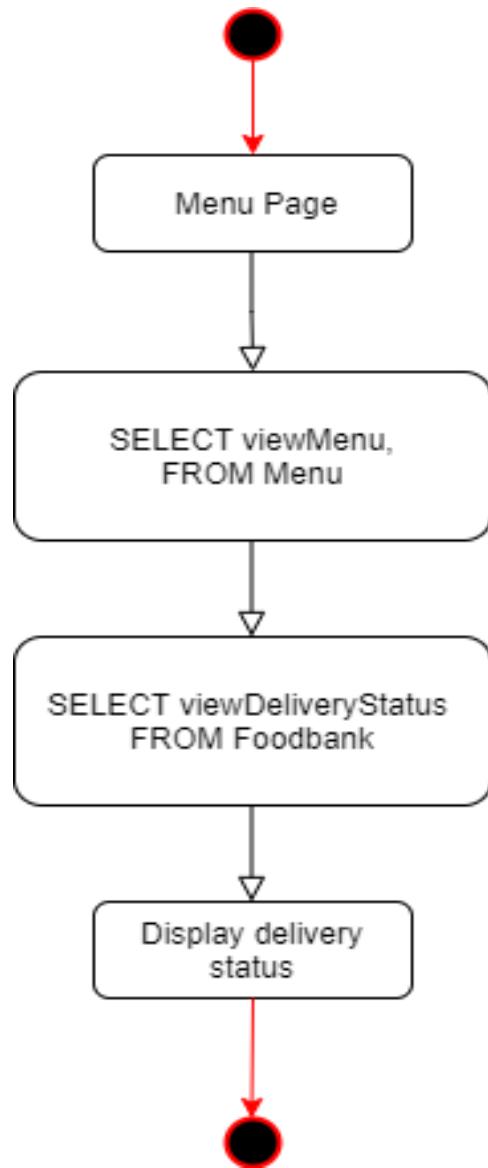


Figure 5.3I Distributor's 'View delivery status' activity diagram

Pseudocode

BEGIN

IF (Show Menu page selected) THEN
SELECT viewMenu

```
FROM Menu
IF (Show Foodbank page selected) THEN
    SELECT viewDeliveryStatus
    FROM Foodbank
END IF

PRINT delivery status

END IF

END
```

Drop off food

Component Diagram

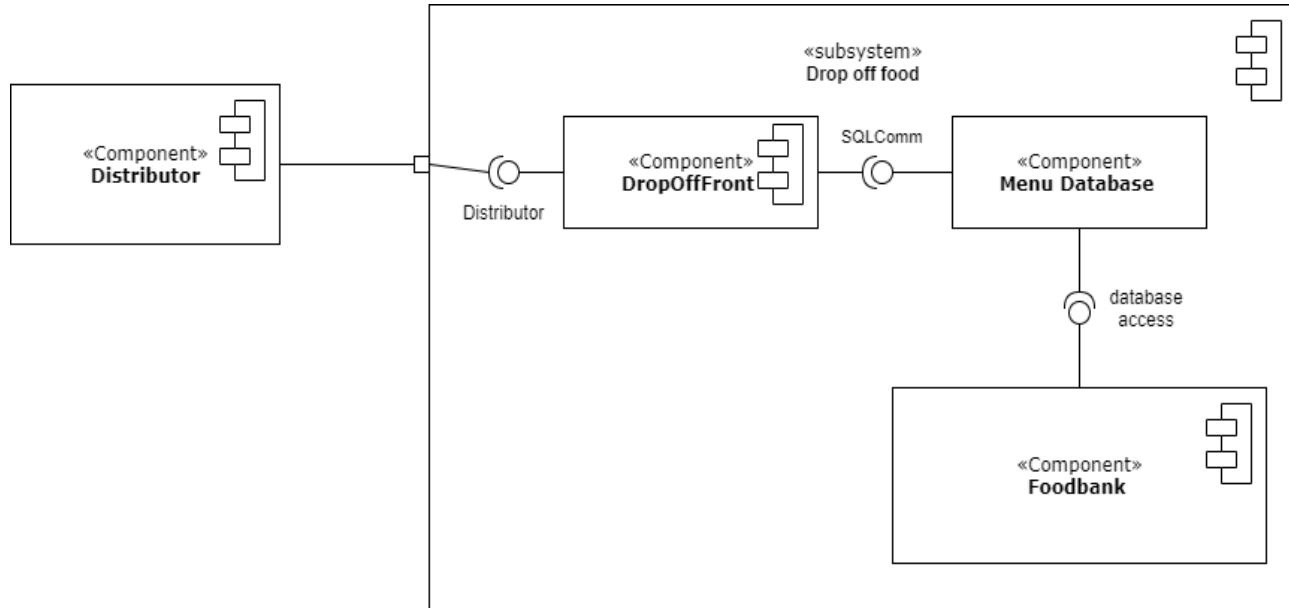


Figure 5.3m Distributor's 'Drop off Food' component diagram

Activity Diagram

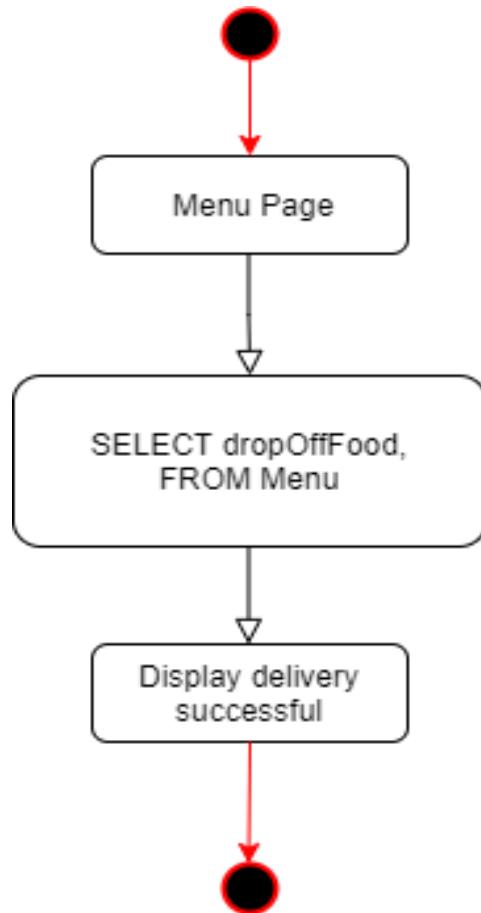


Figure 5.3n Distributor's 'Drop off Food' activity diagram

Pseudocode

```
IF (Show Menu page selected) THEN
    SELECT dropOffFood
    FROM Menu
    PRINT delivery successful
END IF

END
```

Admin

Login

Component Diagram

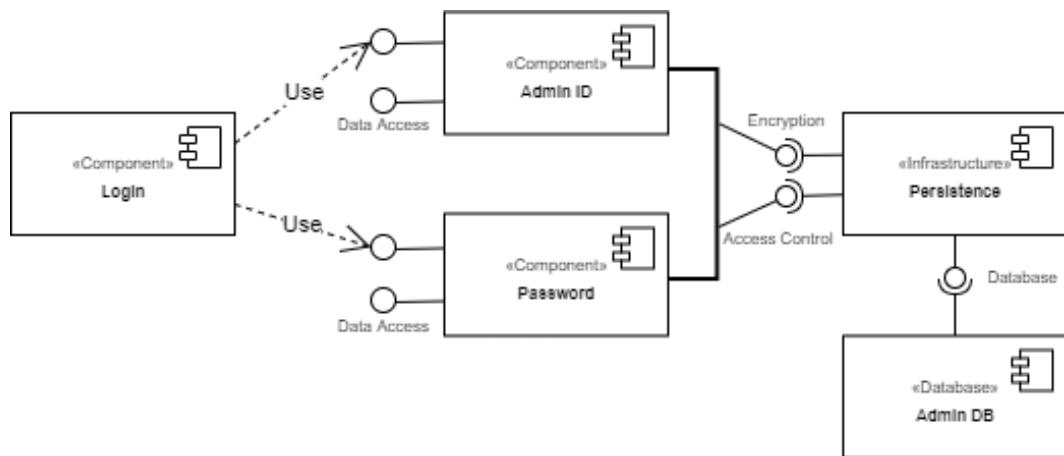


Figure 5.4a Admin's Login Component Diagram

Activity Diagram

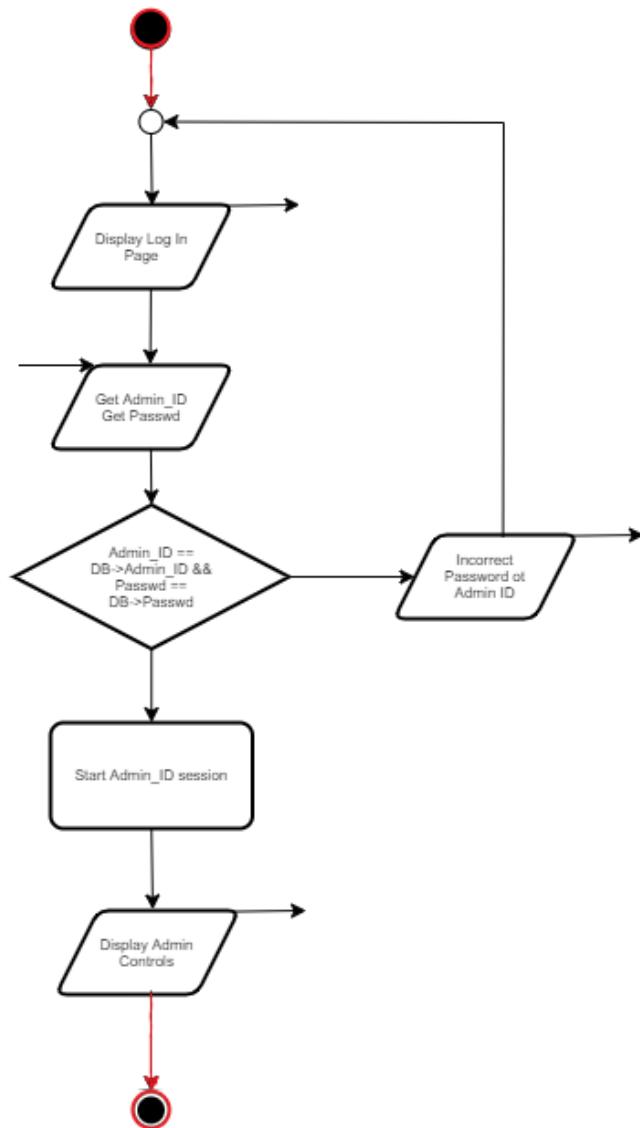


Figure 5.4b Admin's Login Activity Diagram

Pseudocode

```
DISPLAY Login_Screen
GET Admin_ID
GET Passwd

if(DB->Admin_ID == Admin_ID && DB->Passwd == Passwd)
    SET current_Session = Admin_ID
    DISPLAY Admin_Controls
```

User Controls

Component Diagram

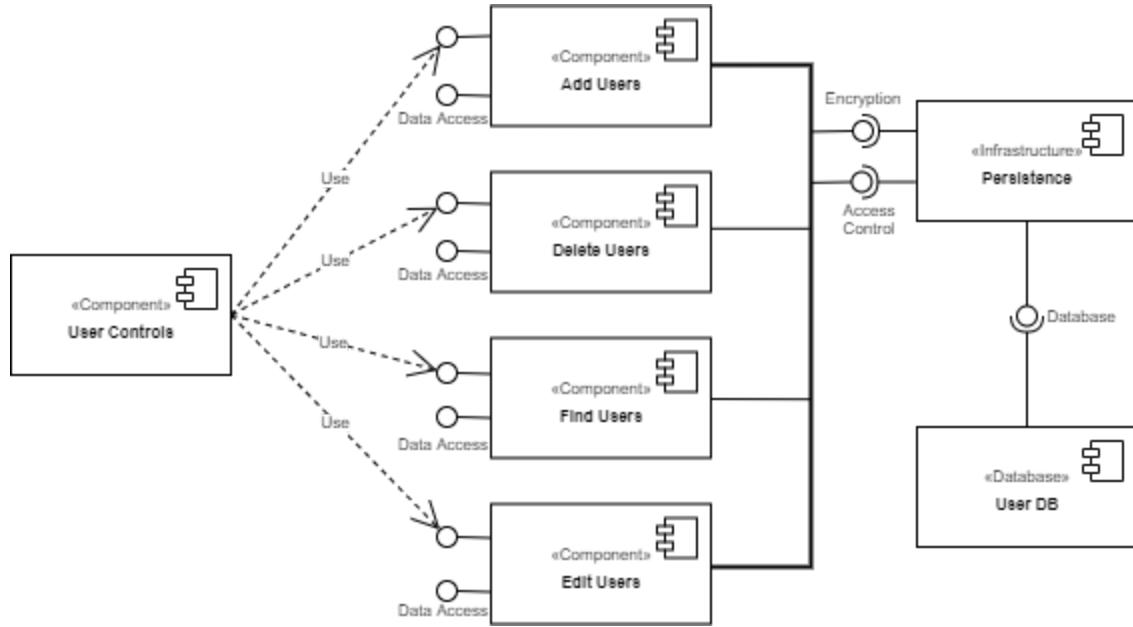


Figure 5.4c Admin's User Controls Component Diagram

Activity Diagram

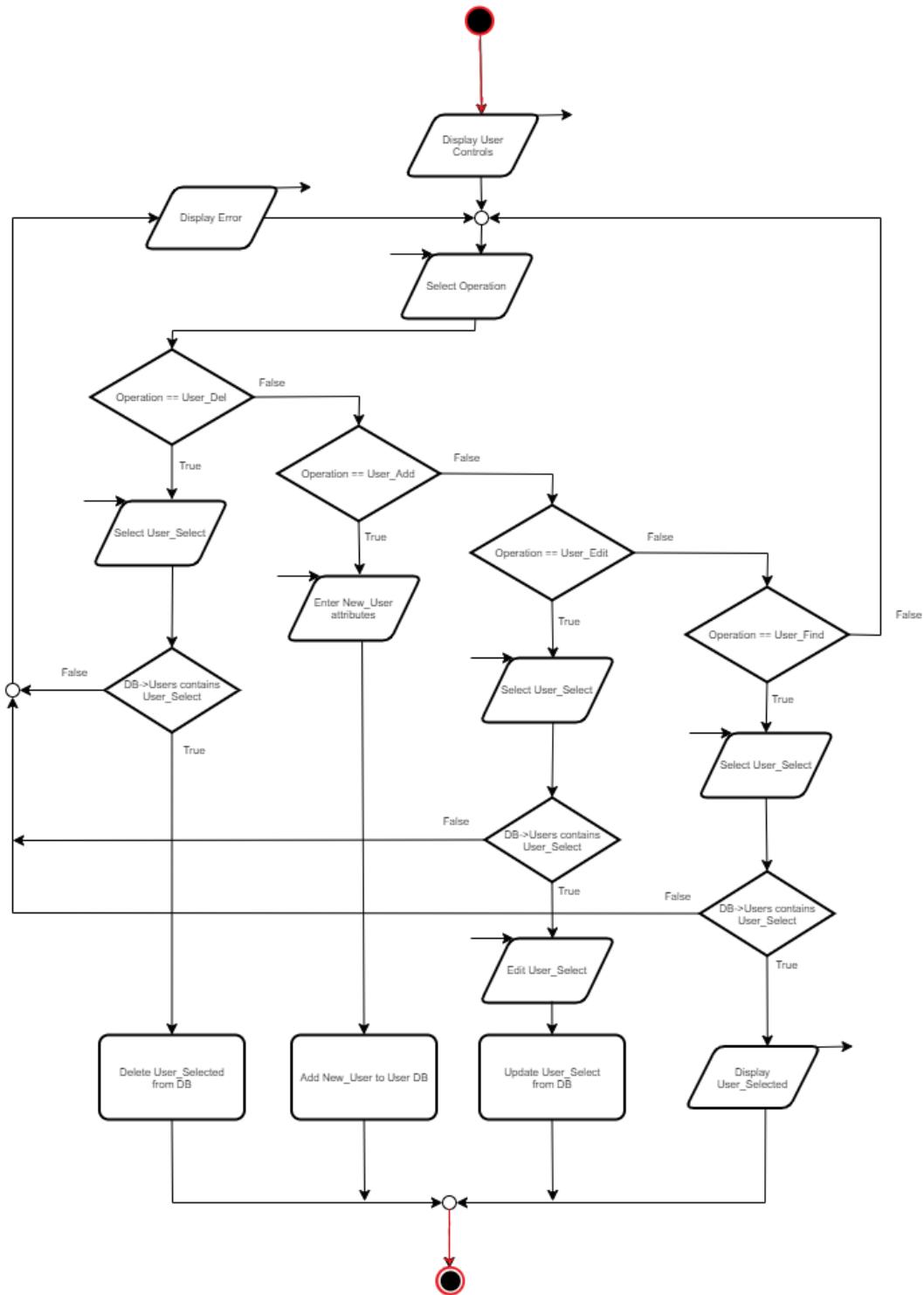


Figure 5.4d Admin's User Controls Activity Diagram

Pseudocode

DISPLAY User_Controls

GET Operation

CASE Operation

User_Delete:

 GET User_Select
 IF(DB->Users == User_Select)
 DELETE User_Select
 ELSE
 DISPLAY Error

User_Add:

 GET New_User(Attributes)
 ADD New_User to DB->Users

User_Edit:

 GET User_Select
 IF(DB->Users == User_Select)
 GET User_Select(Attributes)
 UPDATE DB->Users(User_Select)
 ELSE
 DISPLAY Error

User_Find:

 GET User_Select
 IF(DB->Users == User_Select)
 DISPLAY User_Select from DB->Users
 ELSE
 DISPLAY Error

Food Bank Report Controls

Component Diagram

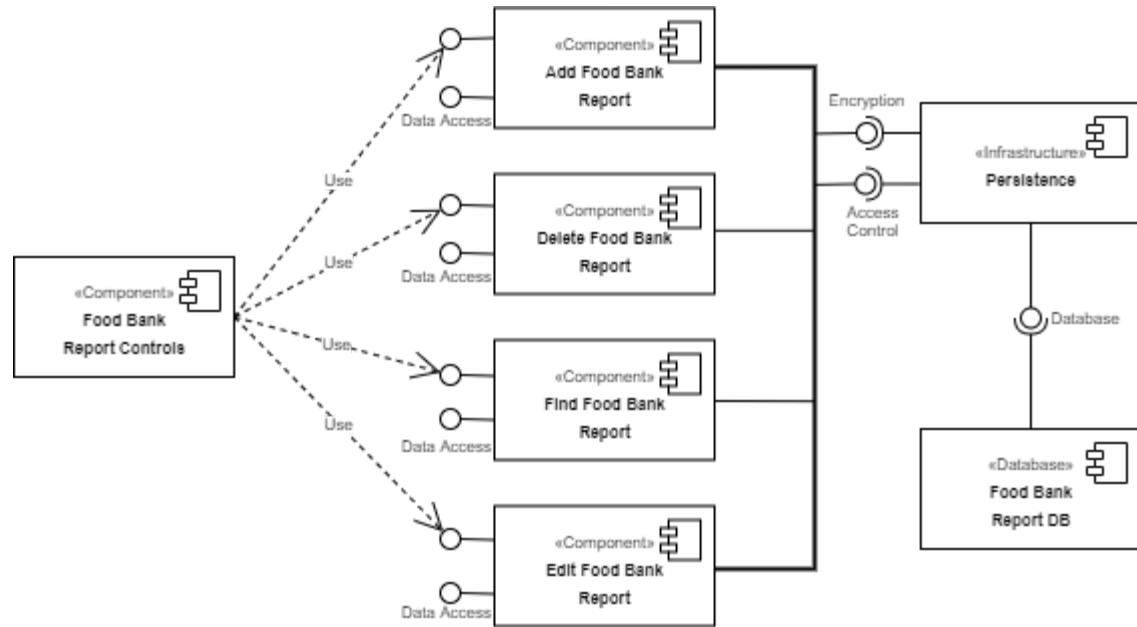


Figure 5.4e Admin's Food Bank Report Controls Component Diagram

Activity Diagram

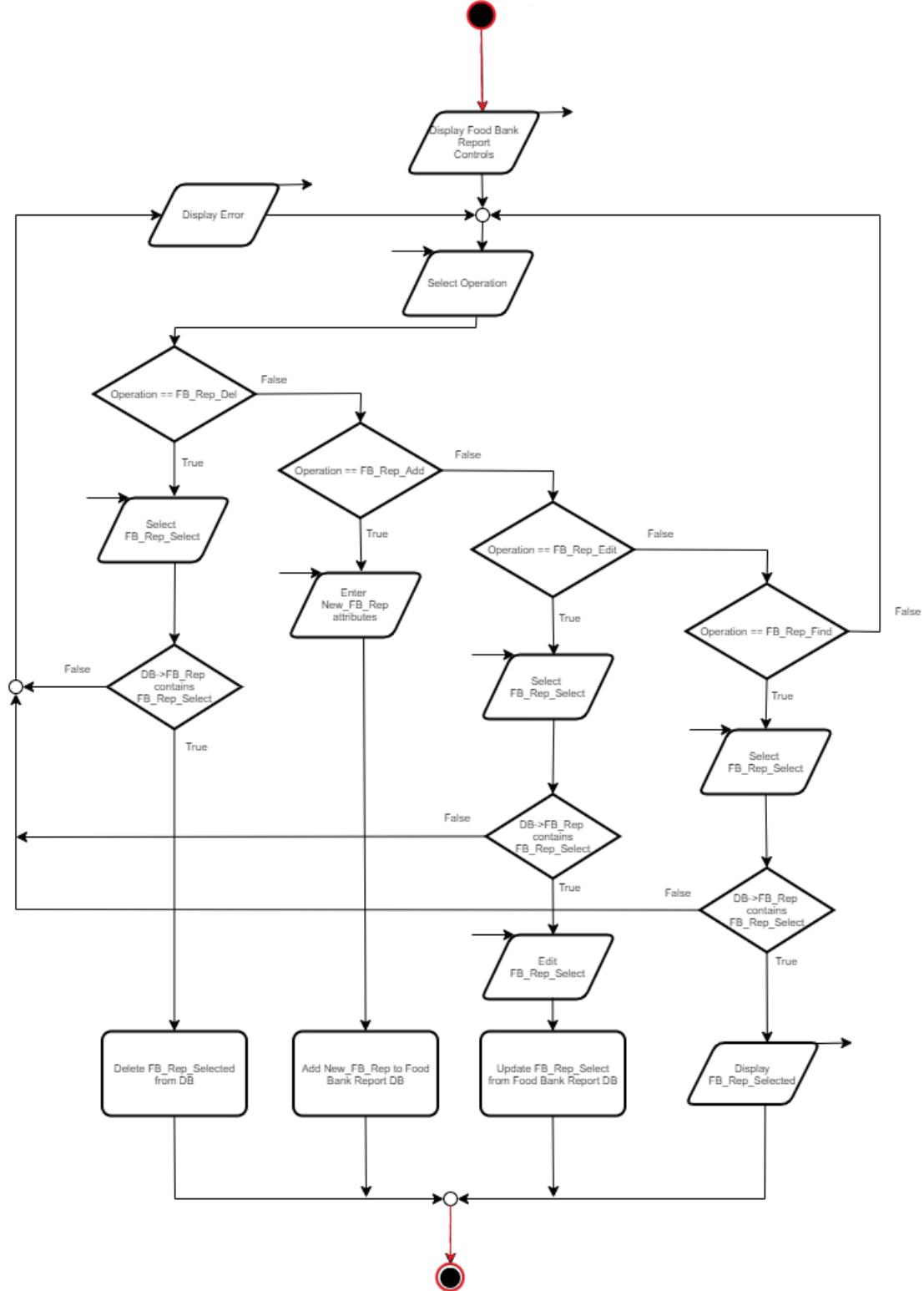


Figure 5.4f Admin's Food Bank Report Controls Activity Diagram

Pseudocode

DISPLAY FB_Rep_Controls

GET Operation

CASE Operation

FB_Rep_Delete:

 GET FB_Rep_Select
 IF(DB->FB_Rep == FB_Rep_Select)
 DELETE FB_Rep_Select
 ELSE
 DISPLAY Error

FB_Rep_Add:

 GET New_FB_Rep(Attributes)
 ADD New_FB_Rep to DB->FB_Rep

FB_Rep_Edit:

 GET FB_Rep_Select
 IF(DB->FB_Rep== FB_Rep_Select)
 GET FB_Rep_Select(Attributes)
 UPDATE DB->FB_Rep(FB_Rep_Select)
 ELSE
 DISPLAY Error

FB_Rep_Find:

 GET FB_Rep_Select
 IF(DB->FB_Rep == FB_Rep_Select)
 DISPLAY FB_Rep_Select from DB->FB_Rep
 ELSE
 DISPLAY Error

Distribution Location Controls

Component Diagram

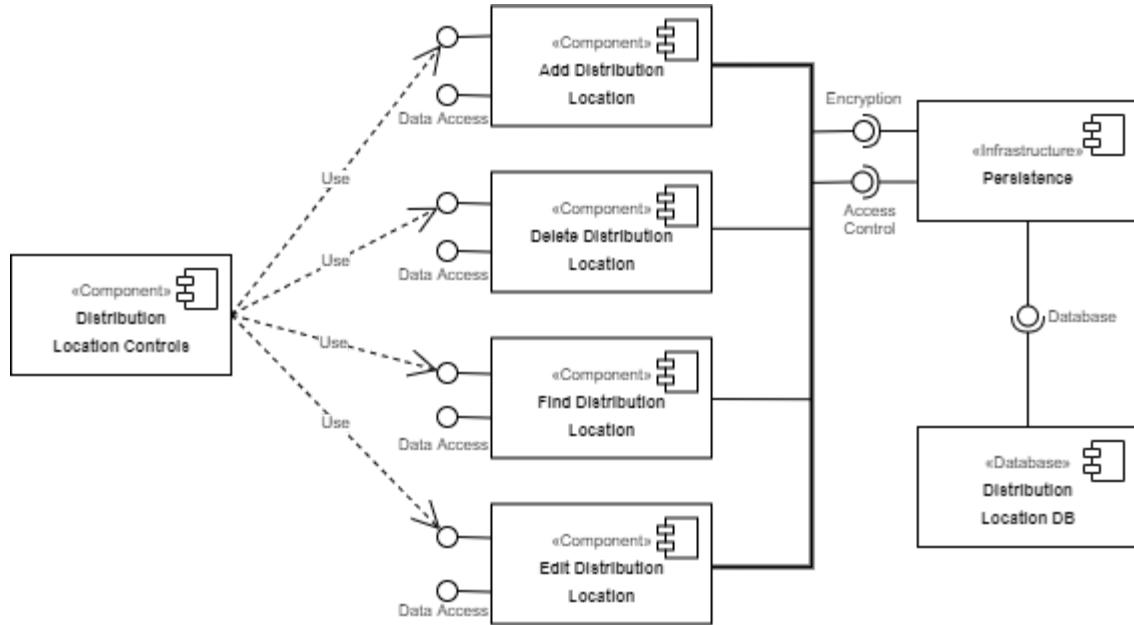


Figure 5.4g Admin's Distribution Location Controls Component Diagram

Activity Diagram

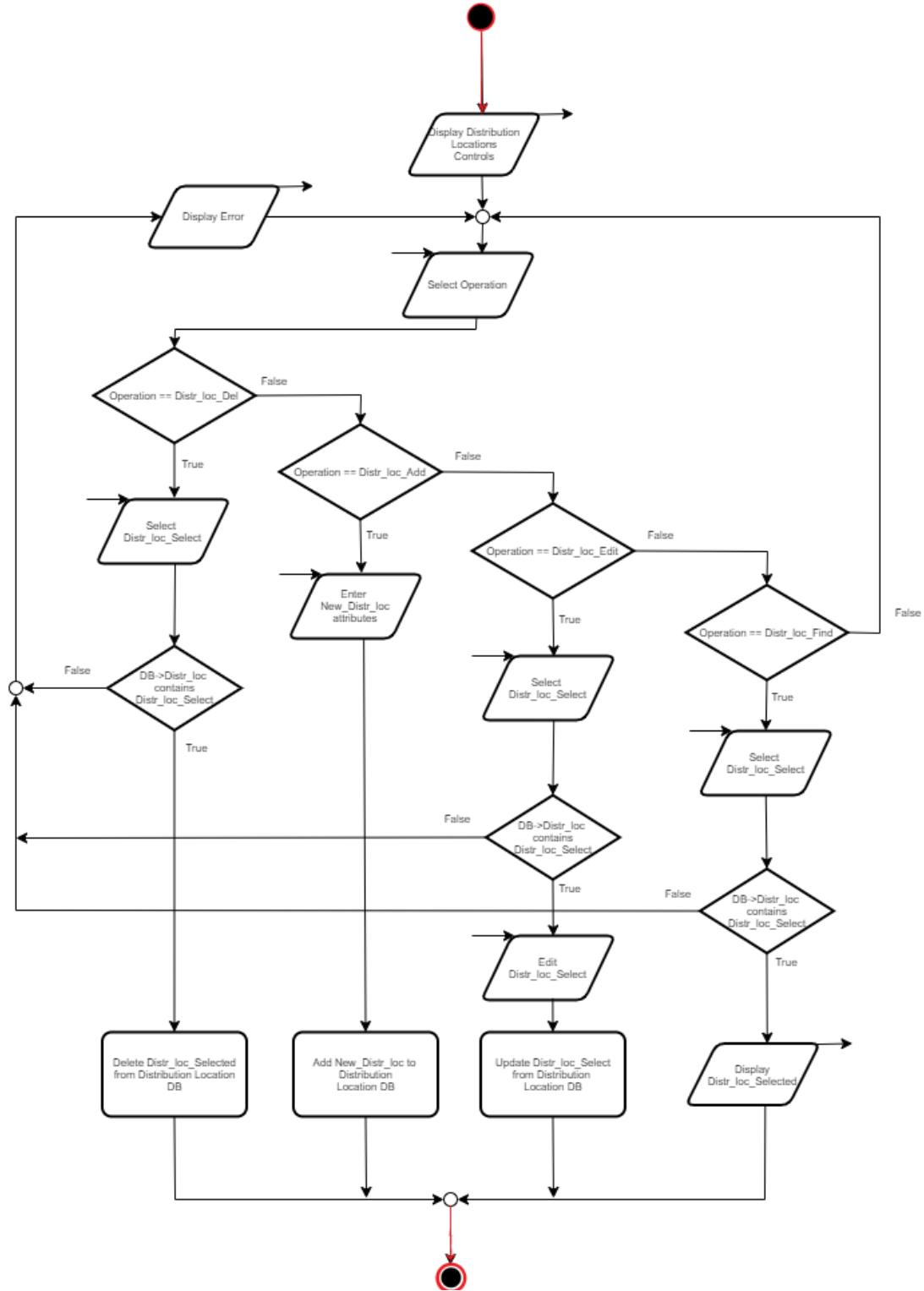


Figure 5.4h Admin's Distribution Location Controls Activity Diagram

Pseudocode

DISPLAY Distr_Loc_Controls

GET Operation

CASE Operation

Distr_Loc_Delete:

 GET Distr_Loc_Select
 IF(DB->Distr_Loc == Distr_Loc_Select)
 DELETE Distr_Loc_Select
 ELSE
 DISPLAY Error

Distr_Loc_Add:

 GET New_Distr_Loc(Attributes)
 ADD New_Distr_Loc to DB->Distr_Loc

Distr_Loc_Edit:

 GET Distr_Loc_Select
 IF(DB->Distr_Loc == Distr_Loc_Select)
 GET Distr_Loc_Select(Attributes)
 UPDATE DB->Distr_Loc(Distr_Loc_Select)
 ELSE
 DISPLAY Error

Distr_Loc_Find:

 GET Distr_Loc_Select
 IF(DB->Distr_Loc == Distr_Loc_Select)
 DISPLAY Distr_Loc_Select from DB->Distr_Loc
 ELSE
 DISPLAY Error

Food Categories Controls

Component Diagram

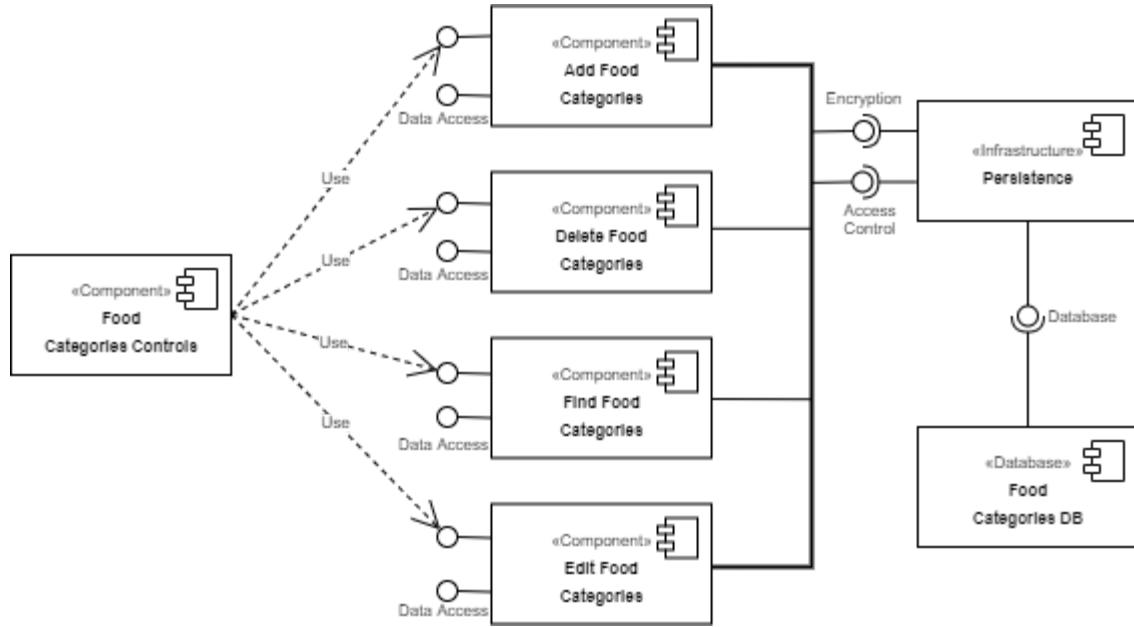


Figure 5.4i Admin's Food Categories Controls Component Diagram

Activity Diagram

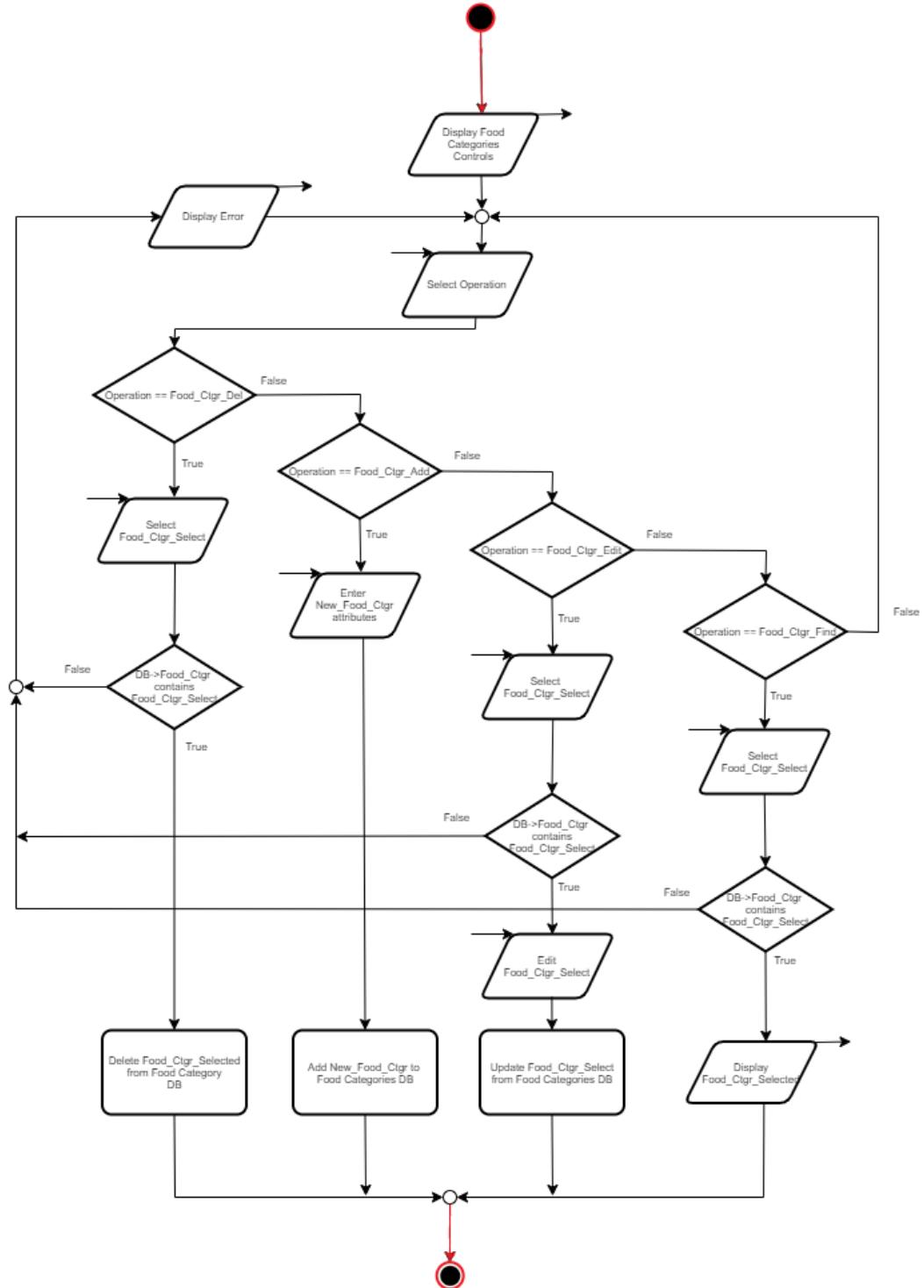


Figure 5.4j Admin's Food Categories Controls Activity Diagram

Pseudocode

DISPLAY Food_Ctgr_Controls

GET Operation

CASE Operation

Food_Ctgr_Delete:

GET Food_Ctgr_Select

IF(DB->Food_Ctgr == Food_Ctgr_Select)

DELETE Food_Ctgr_Select

ELSE

DISPLAY Error

Food_Ctgr_Add:

GET New_Food_Ctgr(Attributes)

ADD New_Food_Ctgr to DB->Food_Ctgr

Food_Ctgr_Edit:

GET Food_Ctgr_Select

IF(DB->Food_Ctgr == Food_Ctgr_Select)

GET Food_Ctgr_Select(Attributes)

UPDATE DB->Food_Ctgr(Food_Ctgr_Select)

ELSE

DISPLAY Error

Food_Ctgr_Find:

GET Food_Ctgr_Select

IF(DB->Food_Ctgr == Food_Ctgr_Select)

DISPLAY Food_Ctgr_Select from DB->Food_Ctgr

ELSE

DISPLAY Error

6 Deployment Design

6.1 Deployment Diagram

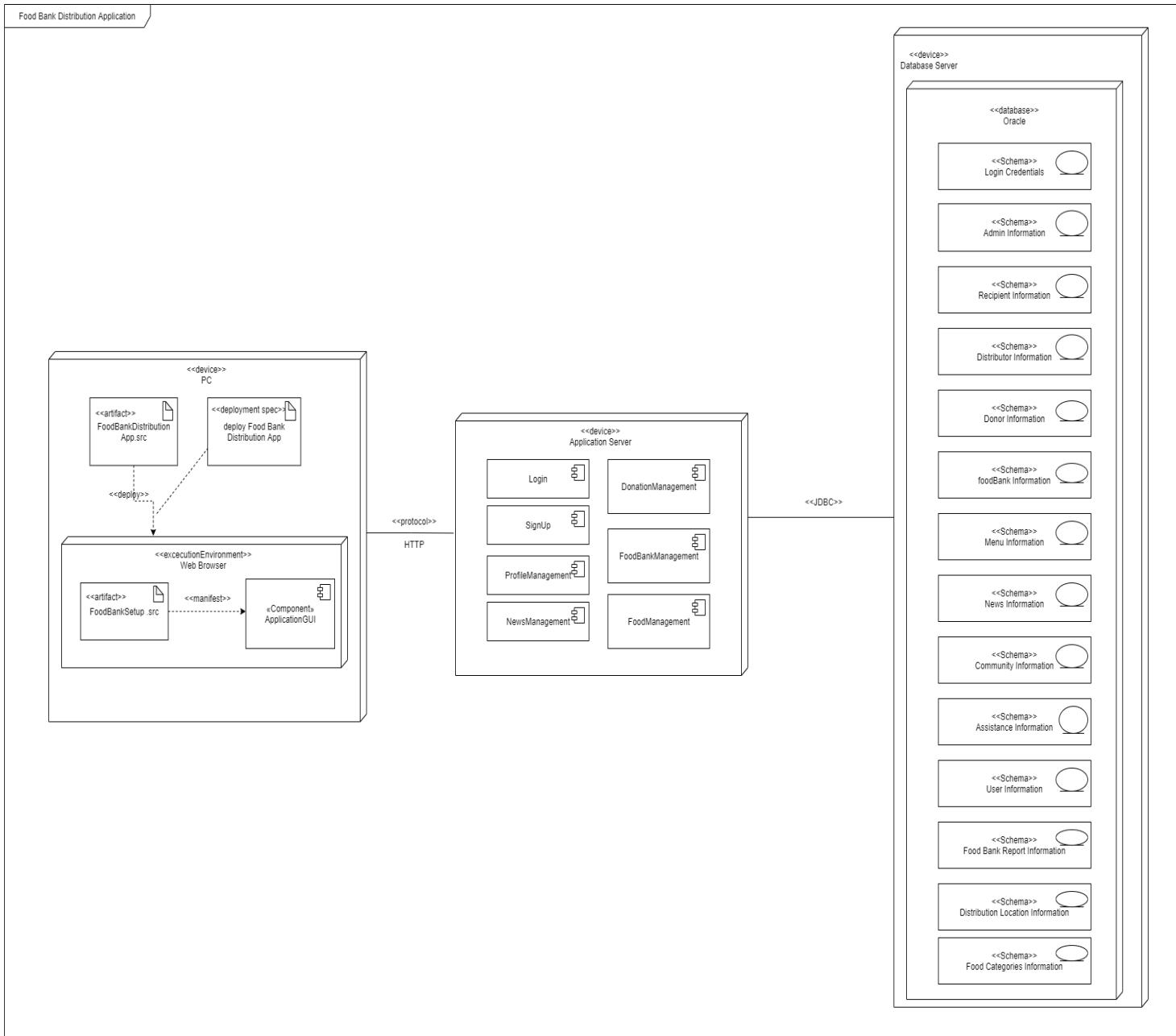


Figure 6.1 Food Bank Distribution Application's Deployment Diagram

For clearer view of Food Bank Distribution Application's Deployment Diagram, Copy and paste this link into Google:

https://drive.google.com/file/d/174TogGBYlTuIV_O2PcP50UM1cC9kNmcW/view?usp=sharing

7 Updated Requirements

<This section is Optional. Place the requirements that have been updated from the SRS, particularly those that would have impact on the software design.>

7.1 Sequence Diagram