Comsats University Islamabad

Computer Science Department

CLO-3
Assignment 3

Software Design Specification
for
(Save Food Stuff)

Submitted By:

Kainat Mudassar FA20-BCS-027

Maha Farooqi FA20-BCS-029

Supervisor

Tahseen Riaz Abassi

Submission Date: (April 29, 2022)

BSCS (2020-2024)

Table of Contents

1. Introduction

The main agenda of this document is brief description about the "Save food stuff application" which is primarily developed for the distribution of extra food from restaurants to different shelter homes in order to help poor and needy people and to avoid food wastage. In last research, a drastic increase can be seen in wastage of food. As per data given by WHO, 20% of the population faces extreme food shortages. Therefore, it is important to come up with a solution that can avoid food waste and can help feed the needy.
Furthermore, such systems have been tremendously helpful in spreading awareness of the food waste issue but they also have some downsides. In addition, it has its useful features as well for example admin can track the history of the whole system, NGOS that are assisting poor communities in their fights against hunger and can use this app to request meals from restaurants. After approval of the request they deliver that food to shelter homes. In this way this save food stuff application will help the restaurant to reduce food waste and will help in feeding the needy and poor people.
Moreover, it also keeps the shelter homes, NGOs and people updated about the food availability from time to time. It also provides an opportunity to the food receiver to give reviews about the restaurants.

2. Design methodology and Software Process Model

Different process models have been identified for designing software which includes Waterfall/Modified Waterfall model, V-Shaped model, Evolutionary process model as well. But here V-shaped Process Model will be adopted in our project because the testing and production of features will be going side by side.

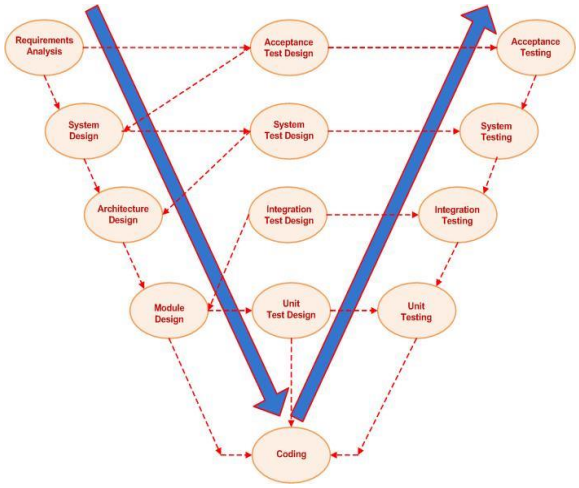


Figure-2.1: V-Shaped Process Model

\

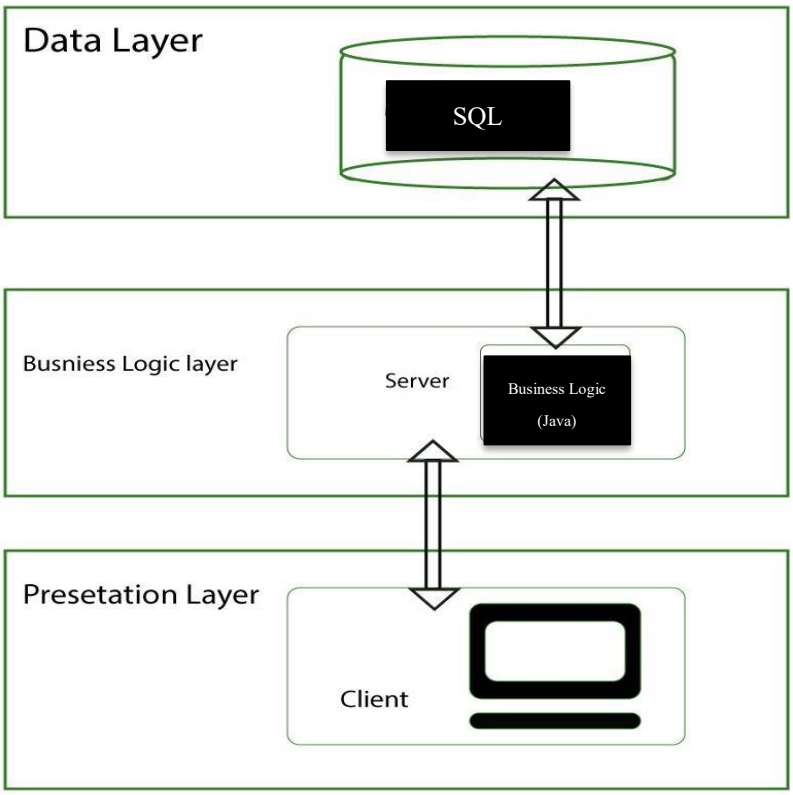
Design Methodology:

Object Oriented approach will be used as the Design Methodology. Changes can be easily done in the Object Oriented, However, there are some basic concepts which are useful to this methodology which includes Inheritance, Polymorphism, Composition, Aggregation and reusability of code. These are helpful during implementation of designed software.

3. System overview

Our system is a web application in which there is an admin, restaurant, donor and members. The restaurant gives away extra food to needy people for free or for very low price. The members can order this food and the donor can donate food to shelter homes and money to restaurants

3.1 Architectural Design



Figur3-3.1: 3-tier Architecture Diagram

3.2 Process flow

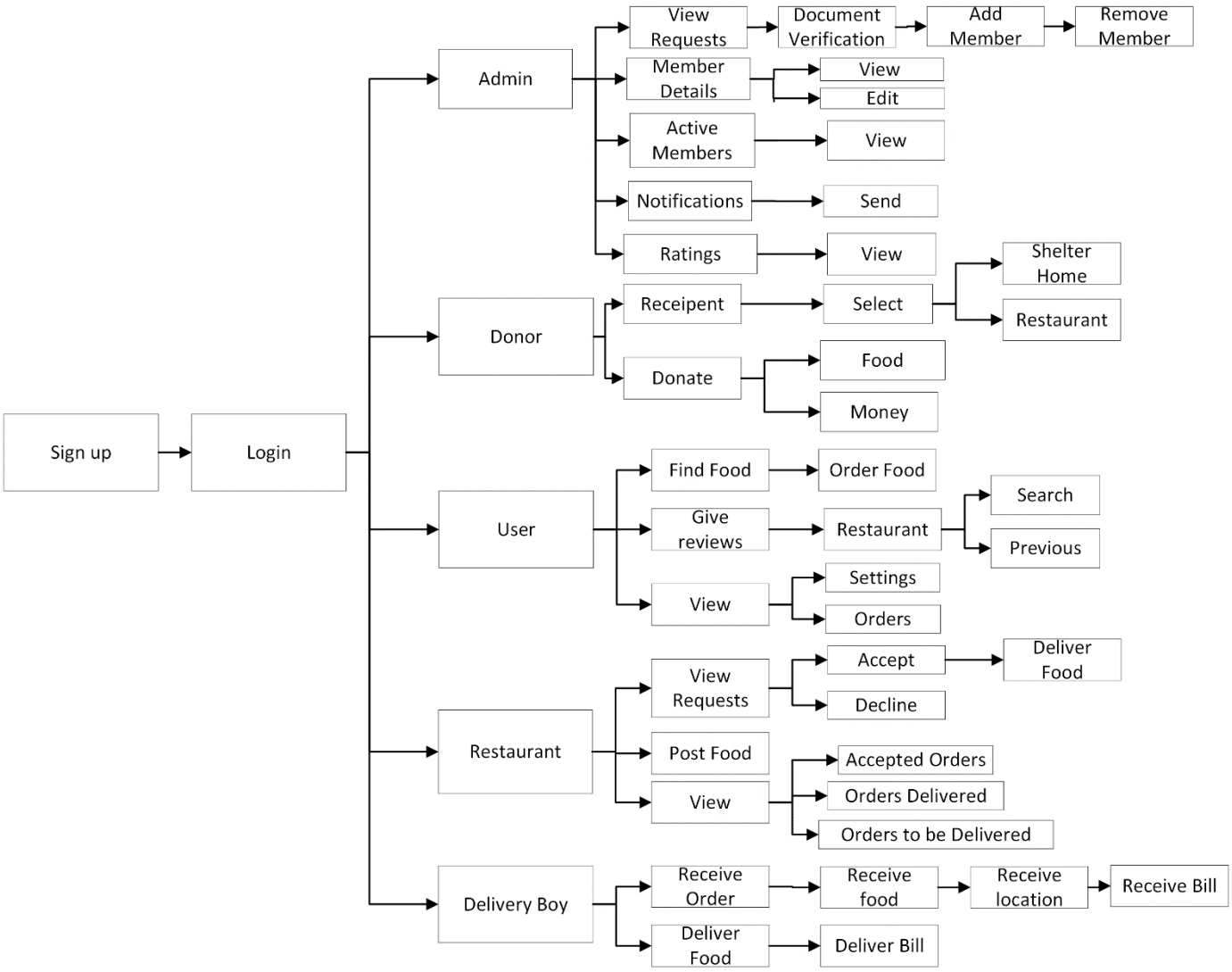


Figure-3.2: Process Flow Diagram

4. Design Models:

4.1: Activity Diagram

Following are some activity diagrams of major actions of our modules

1. Add Member

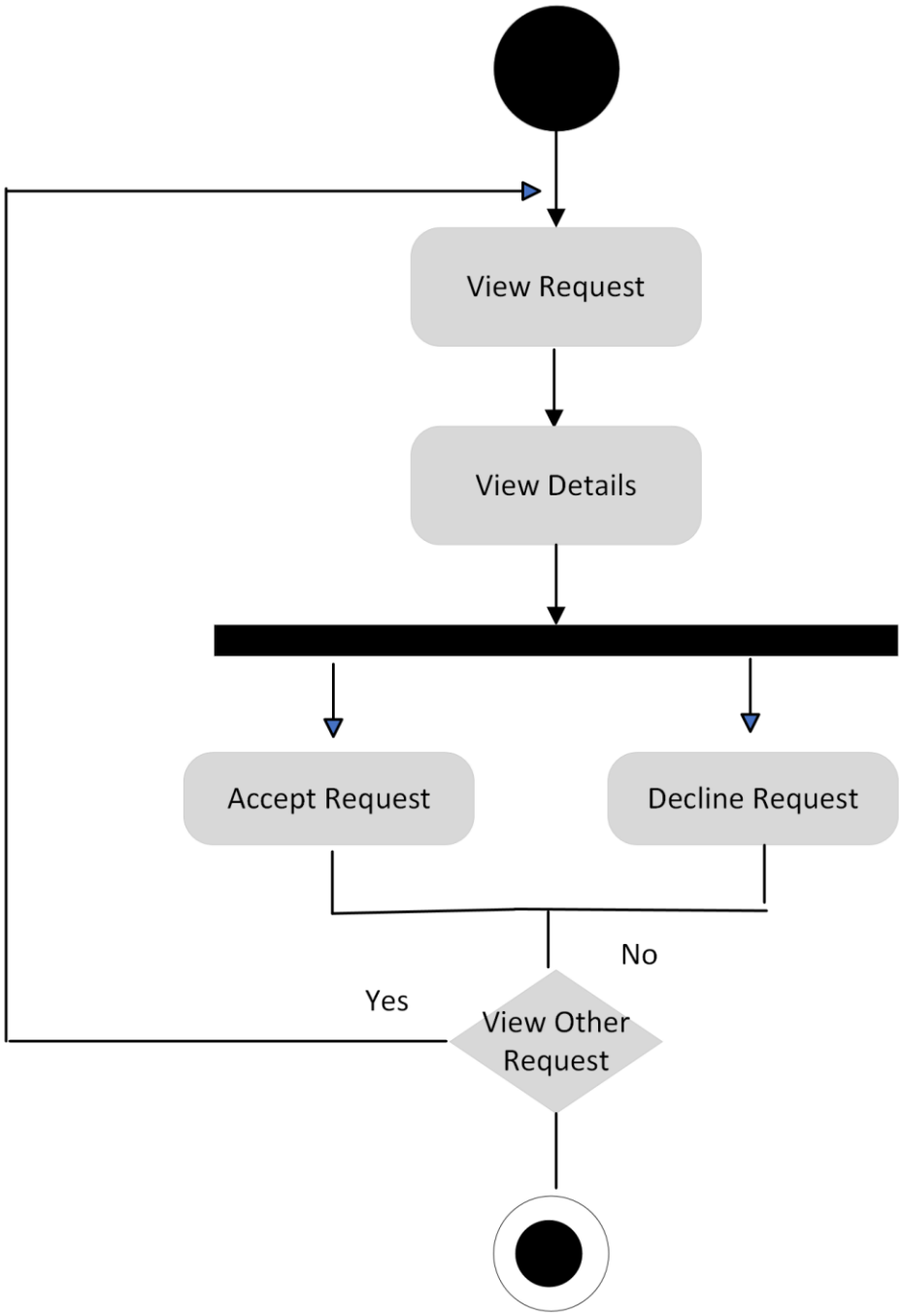


Figure-4.1.1: Add Member Activity Diagram

2. Delivery Boy

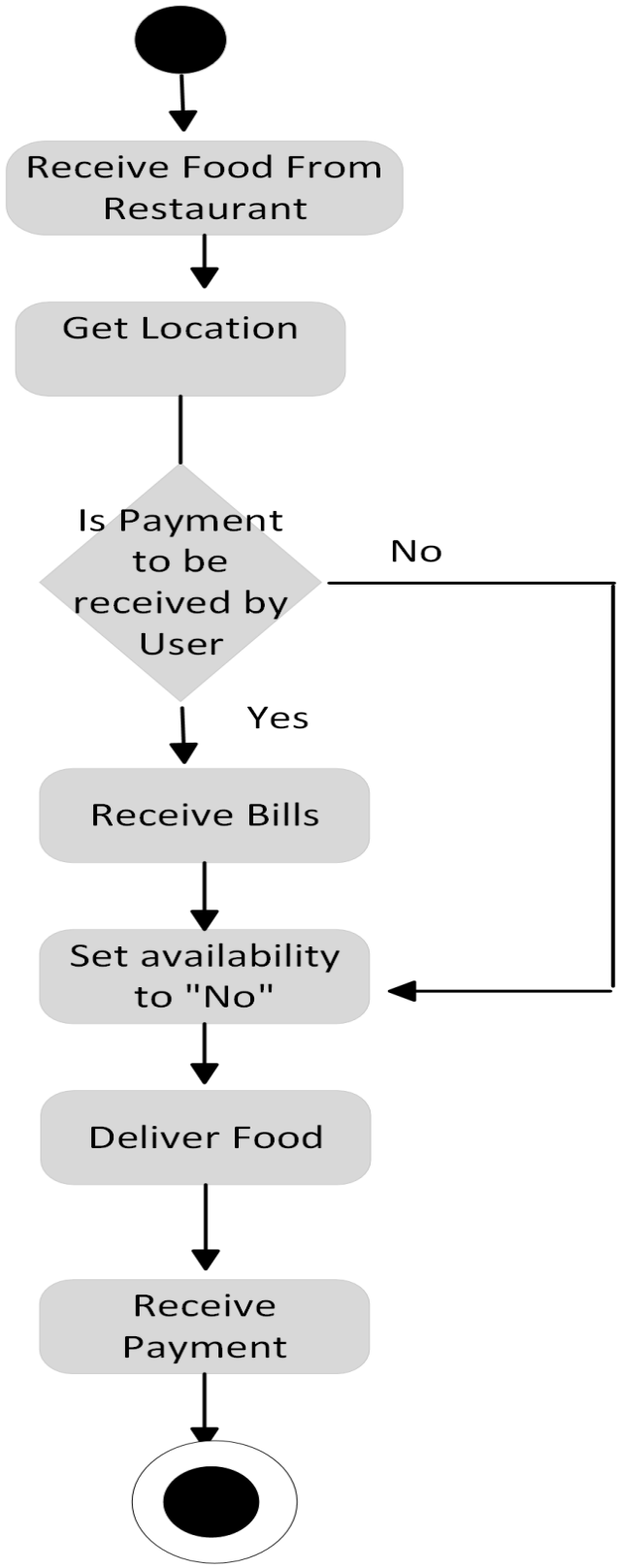


Figure-4.1.2: Delivery Boy Activity Diagram

3. Donation

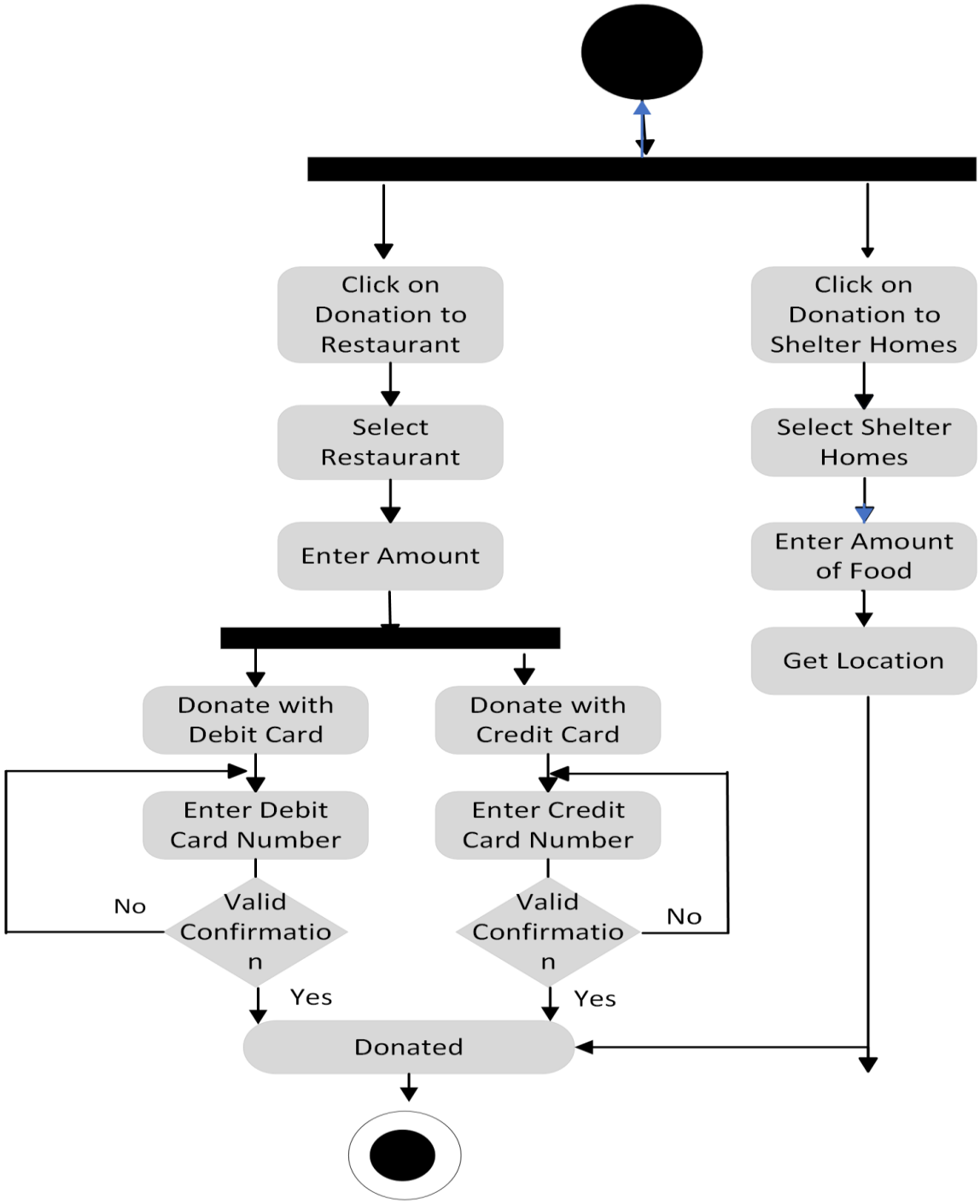


Figure-4.1.3: Donation Activity Diagram

4. Find Food

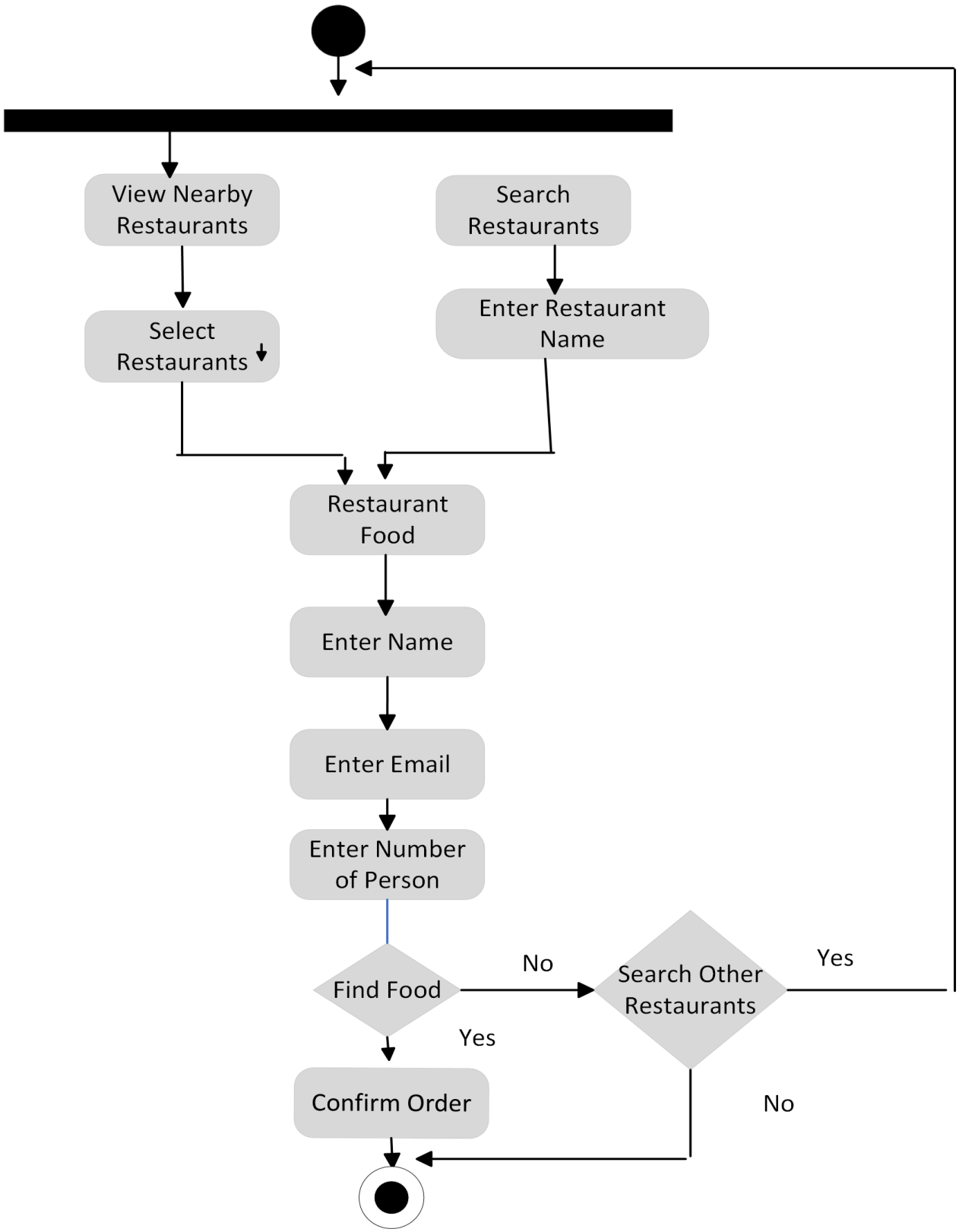


Figure-4.1.4: Find Food Activity Diagram

5. Post Food

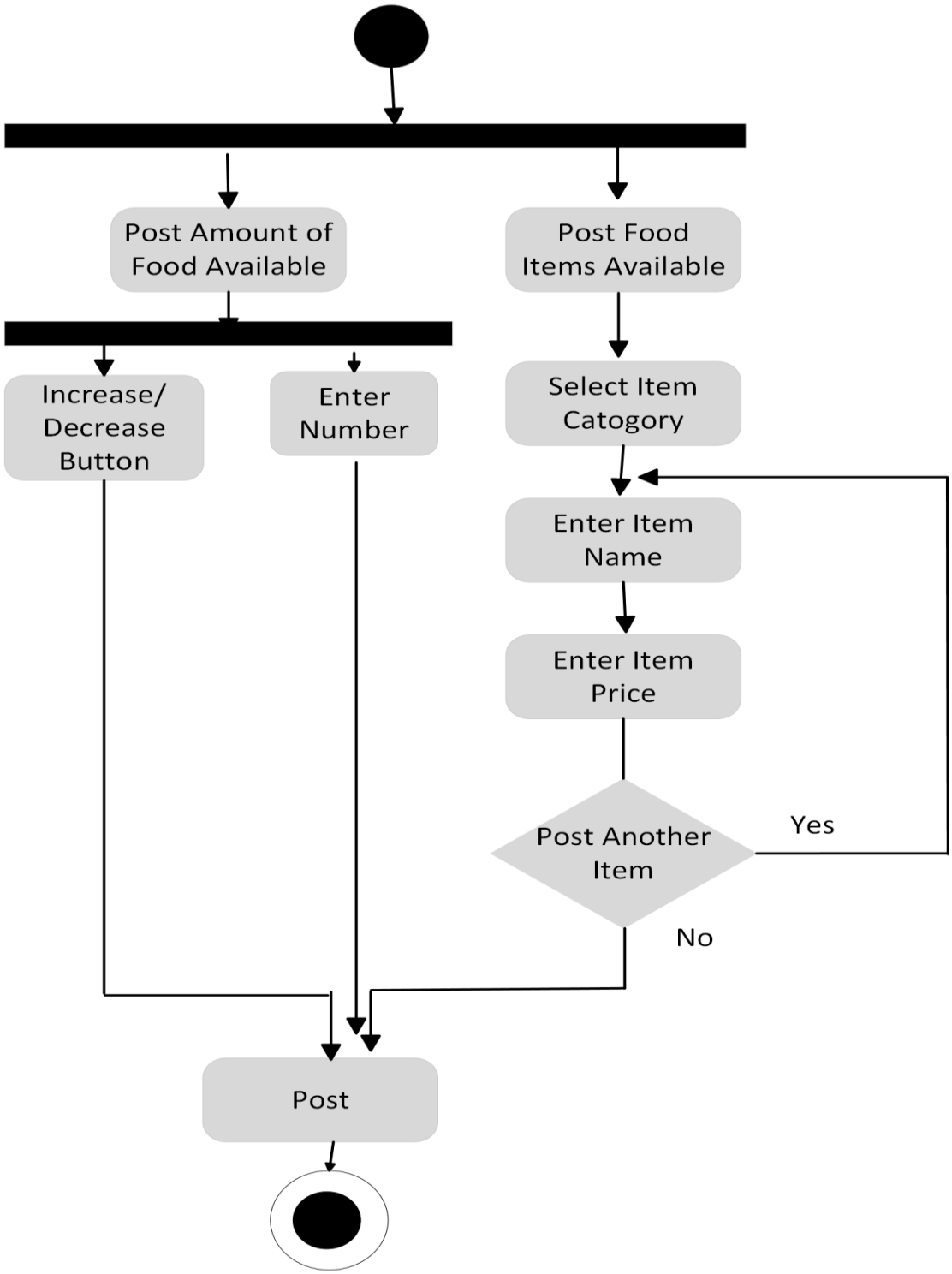


Figure-4.1.5: Post Food Activity Diagram

6. Review:

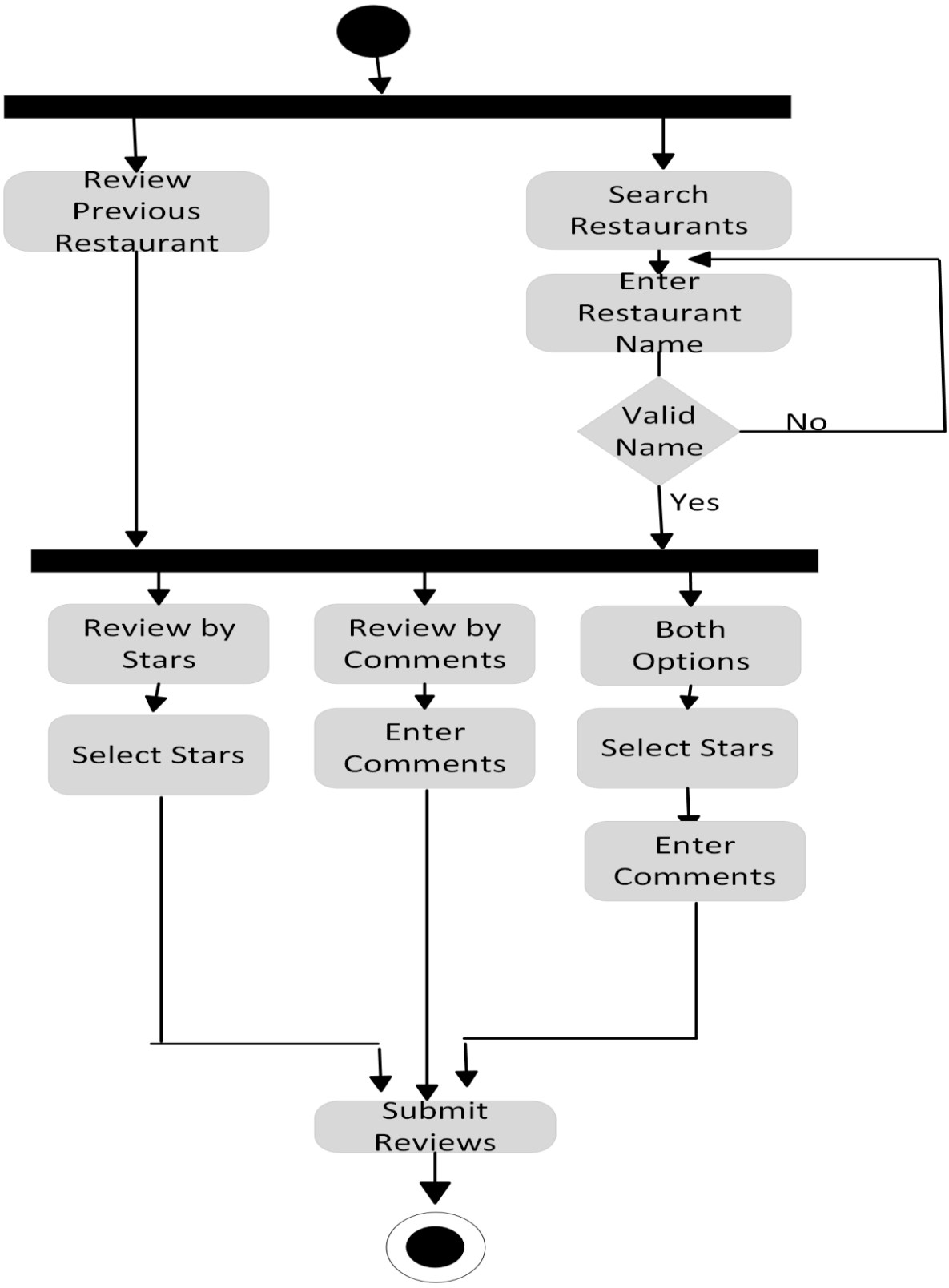


Figure-4.1.6: Reviews Activity Diagram

7. Profile Management:

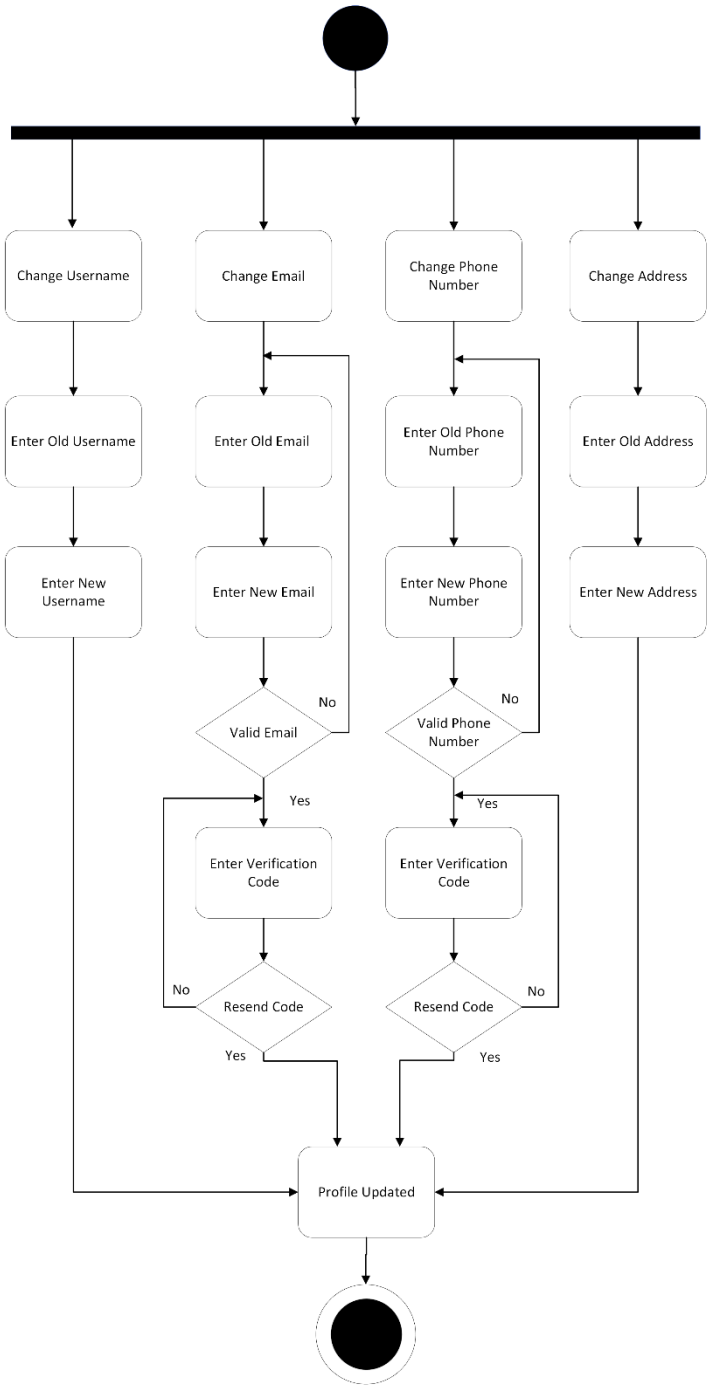


Figure-4.1.7: Delivery Boy Activity Diagram

1. Add Member

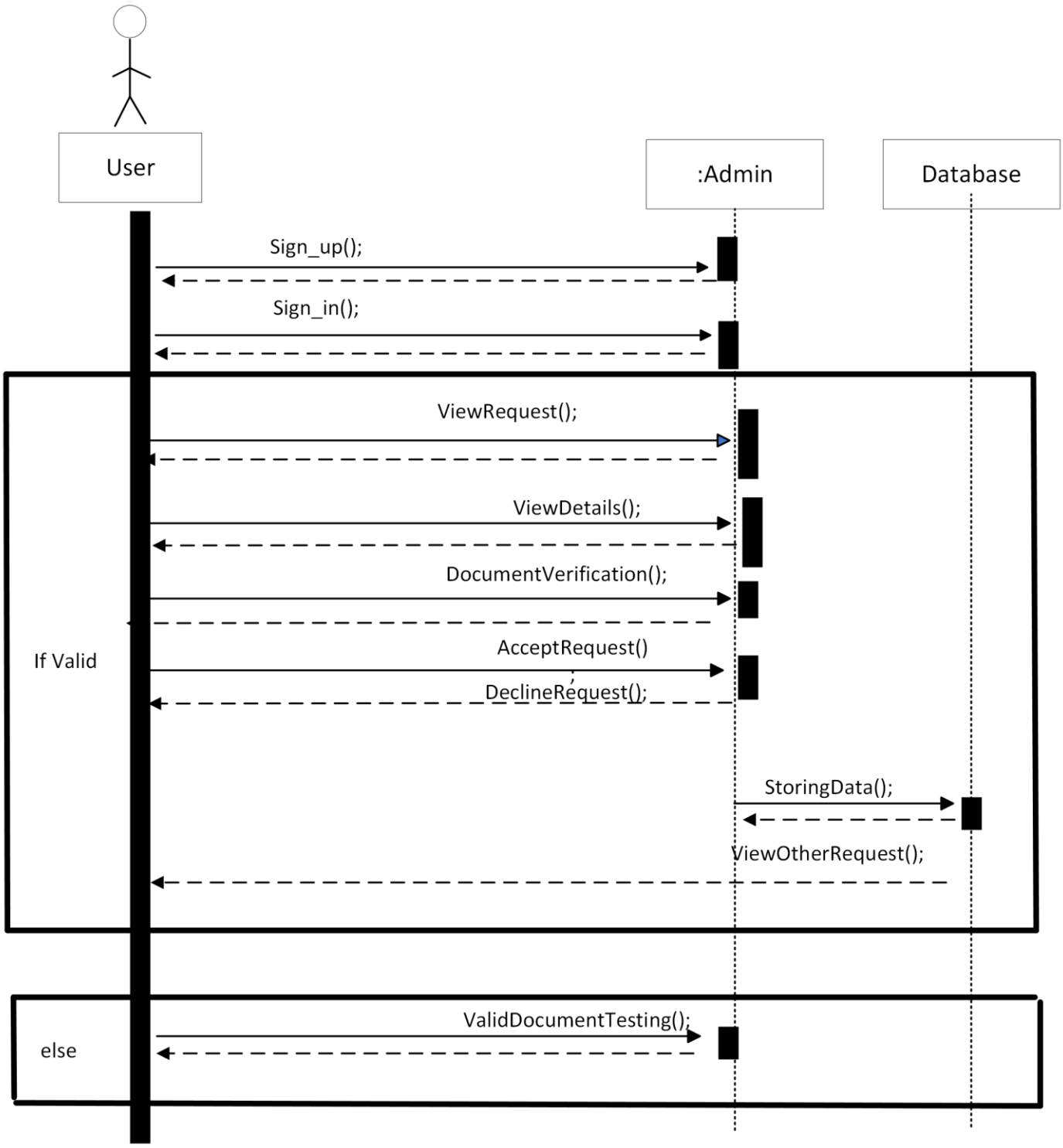


Figure-4.2.1: Add Member Sequence Diagram

2. Delivery

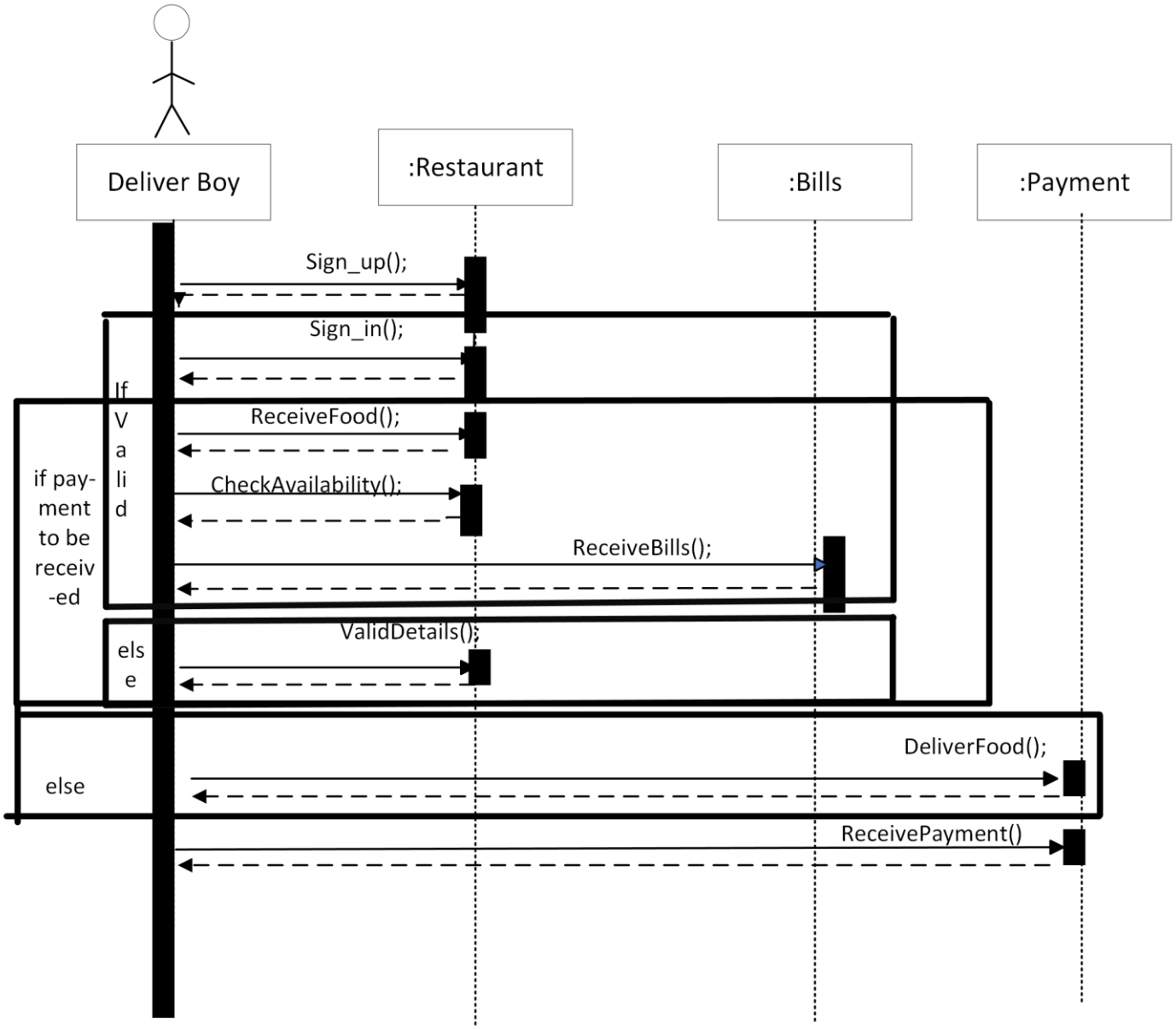


Figure-4.2.2: Delivery Sequence Diagram

3. Reviews:

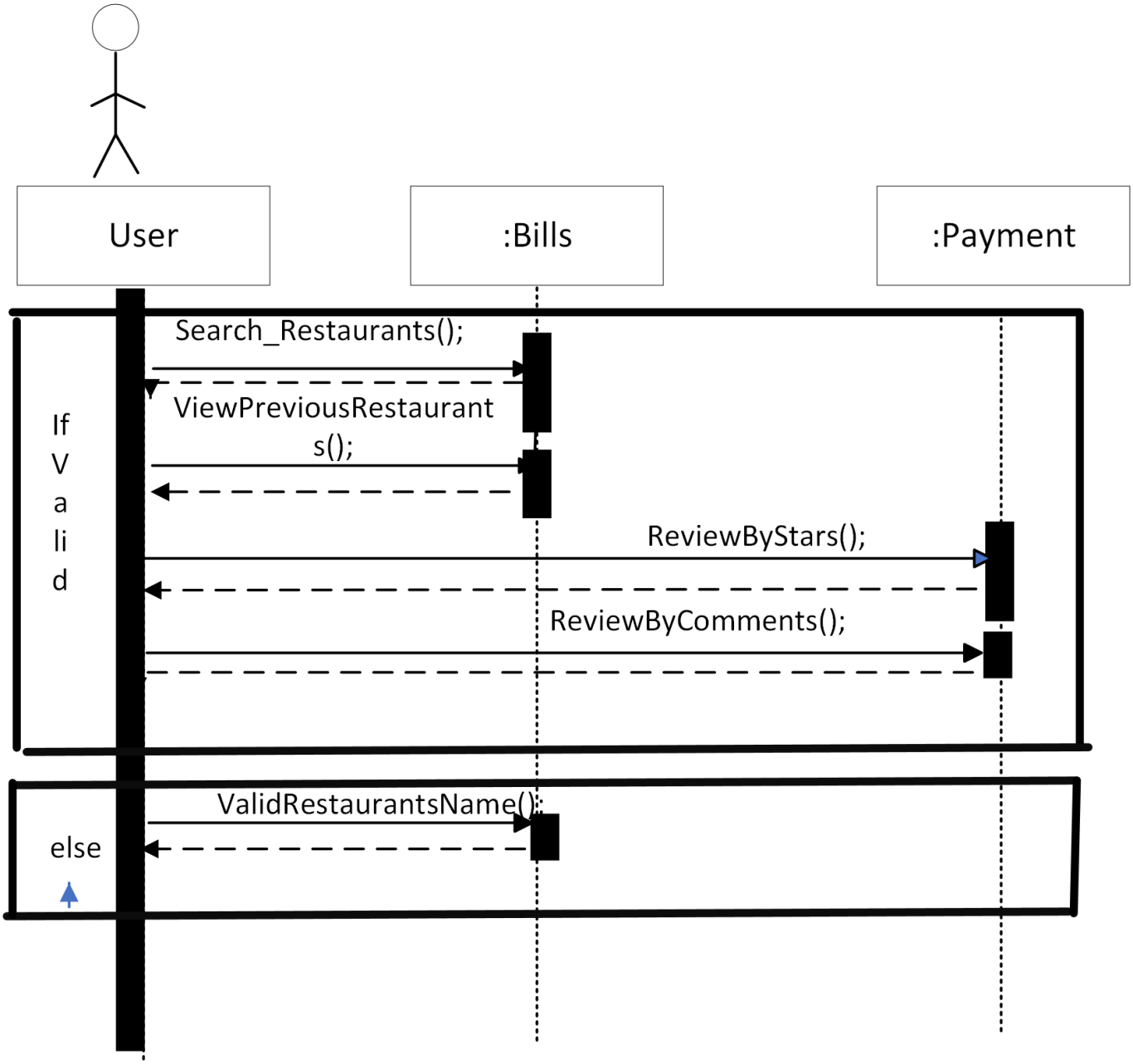


Figure-4.2.3: Reviews Sequence Diagram

4. Find Food:

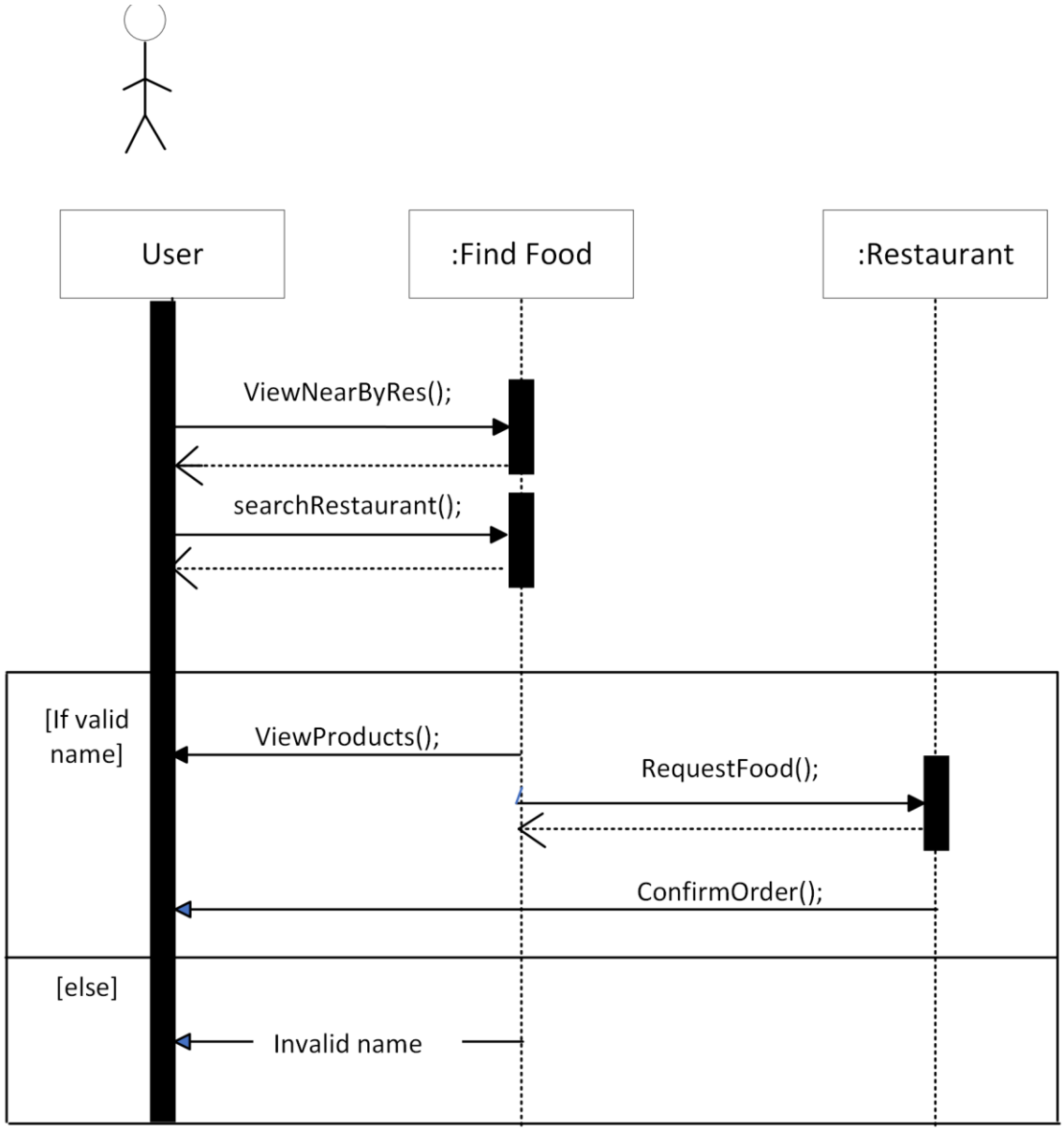


Figure-4.2.4: Find Food Sequence Diagram

5. Post Food

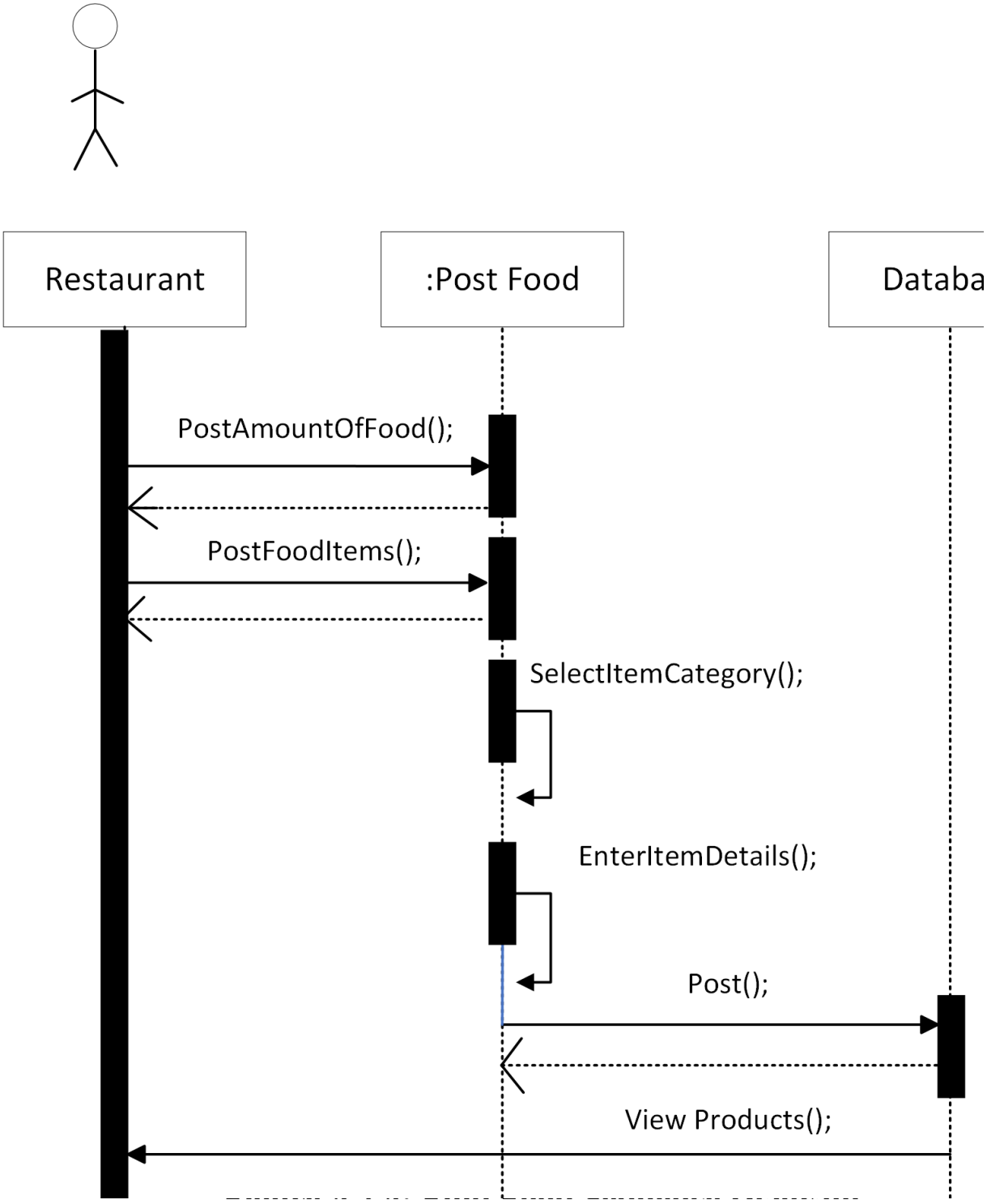


Figure-4.2.5: Post Food Sequence Diagram

6. Donor:

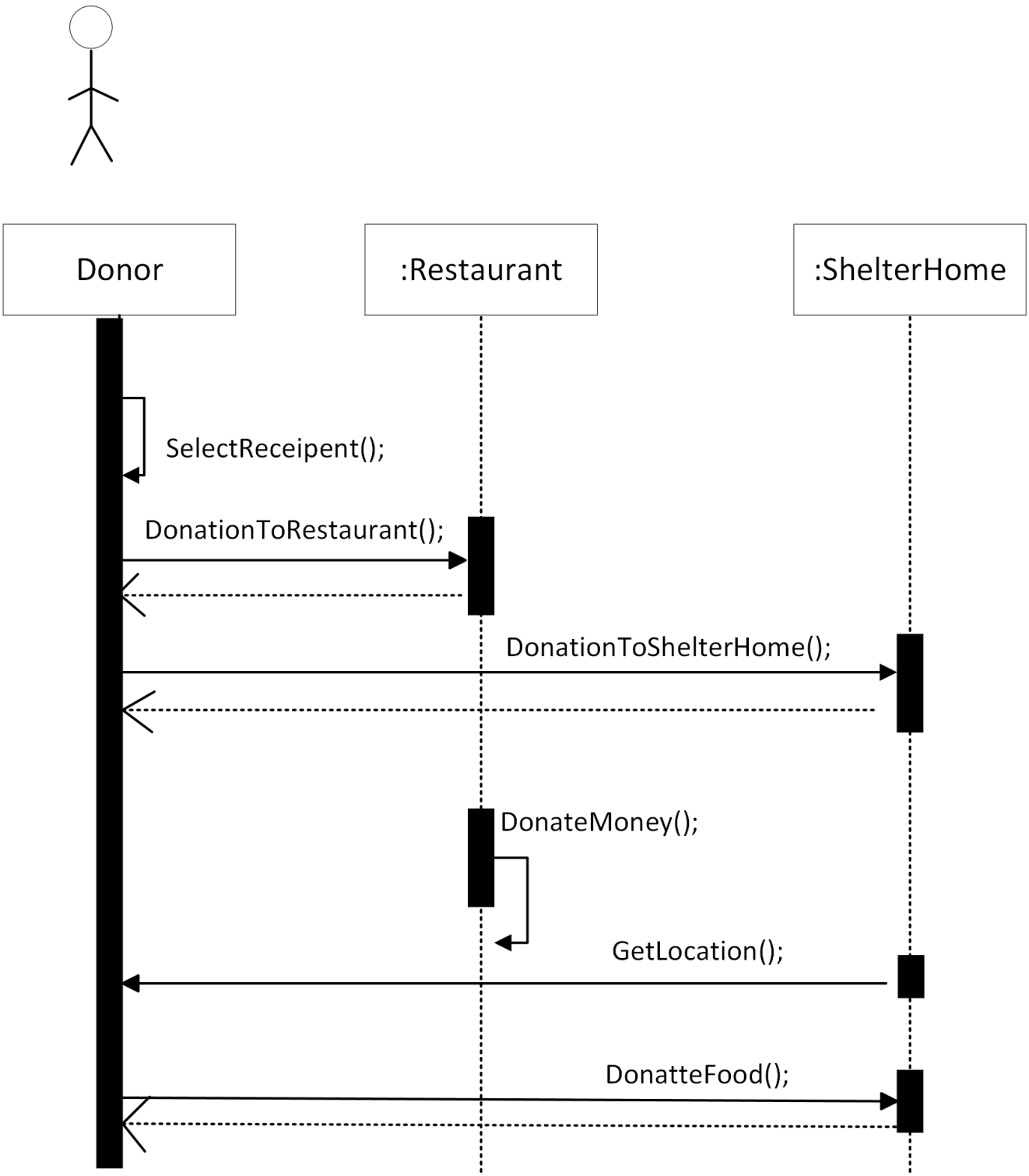


Figure-4.2.6: Donor Sequence Diagram

4.3 Class Diagram

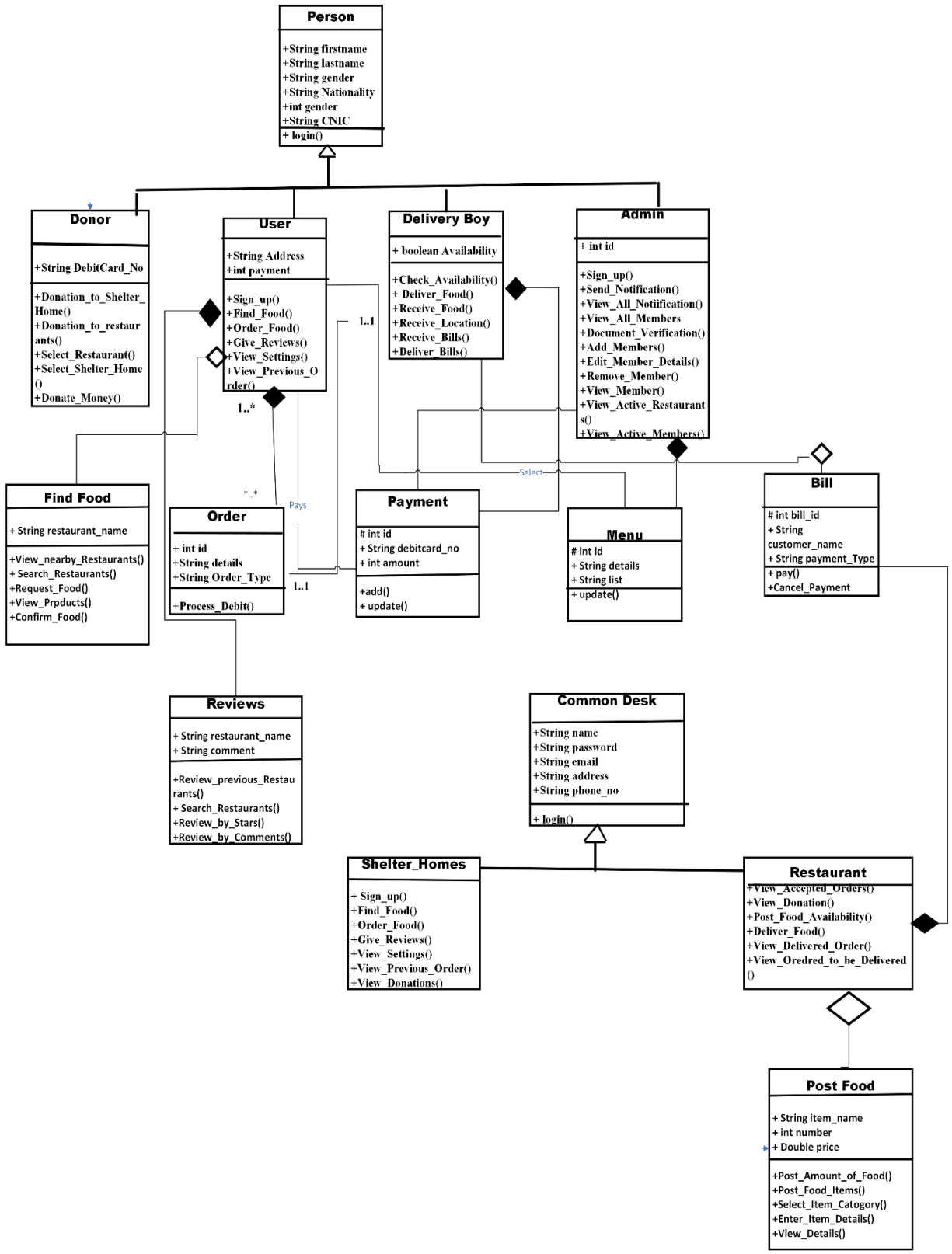


Figure-4.3: Save Class Diagram

5. Data design

5.1 Data design schema

User schema:

```
const monggose= require( 'mongoose' )
const UserSchema= monggose.Schema({
"Id":
{ description: "this is a unique id of a record",
type: "Number"
required: "true"
},
"Username"
{
"description": "A user's Username",
"type": "string",
required: "true"
},
"email"
{
"description": "user's email",
"type": "email",
required: "true"
},
"password"
{
"description": "User’s password",
"type": "password",
required: "true"
},
"Location"
{
"description": “user's location",
"type": "string",
required: "true"
}
“Mobile No”
{
"description": “user's mobile no",
"type": "string"
required: "true"
}

})
module. Exports= monggose.model ( "User " , UserSchema)
```

Restaurant Schema:

```
const monggose= require( 'mongoose' )
const restaurantSchema= monggose.Schema({
```

```
"Id":
{ description: "this is a unique id of a record",
type: "Number"
required: "true"
},
"name"
{
"description": "A restaurant name",
"type": "string",
required: "true"
},
"email"
{
"description": "restaurant's email",
"type": "email",
required: "true"
},
"password"
{
"description": "restaurant's password",
"type": "password",
required: "true"
},
"address"
{
"description": "restaurant's address ",
"type": "string",
required: "true"
}
"phoneNo"
{
"description": "restaurant's mobile no",
"type": "string"
required: "true"
}

})
module. Exports= monggose.model ( "Restaurant " , restaurantSchema)
```

Reviews Schema:

```
const monggose= require( 'mongoose' )
const reviewsSchema= monggose.Schema({
"restaurantId":
{ description: "this is a unique id of restaurant to whom reviews are given",
type: "Number"
required: "true"
},
"restaurantName"
{
"description": "A restaurant name",
"type": "string",
required: "true"
},
"comments"
{
```

```
"description": "comment reviews given to restaurant",
"type": "string",
required: "true"
}

})
module. Exports= monggose.model ( "Reviews " , reviewsSchema)
```

Payment Schema:

```
const monggose= require( 'mongoose' )
const paymentSchema= monggose.Schema({
"Id":
{ description: "this is a unique id of a record",
type: "Number"
required: "true"
},
"debitCard"
{
"description": "debit card number through which payment is done",
"type": "string",
required: "true"
},
"amount"
{
"description": "the amount that is to be paid",
"type": "double",
required: "true"
}

})
module. Exports= monggose.model ( "Payment " , paymentSchema)
```

6. Implementation and Algorithm:

- ## Sign-in algorithm pseudo code:

  Allows user to sign in to the system for further actions

  1. Enter username
  2. Enter password
  3. Check if username correct
  4. Check if password correct
  5. Display message “Signed in successfully”

- ## Donation Algorithm pseudo code:

  Allows user to donate to restaurants or shelter home

  1. Donation to Restaurant
     a. Select restaurant

   b. Enter amount
   c. Donate with credit card or debit card
   d. Enter card number
2. Donation to shelter home
   a. Select shelter home
   b. Enter amount of food
   c. Get location
3. Donate

- # Reviews Algorithm pseudo code:

Allows user to give reviews to restaurants

1. Review previous restaurant
2. Search restaurant
3. Enter restaurant name
4. Select stars to give review
5. Enter comment to give review
6. Submit review

- # Find Food Algorithm pseudo code:

Allows user to find and order food

1. View nearby restaurant
   a. Select restaurant
2. Search restaurant
   a. Enter restaurant name
3. Request food
4. Enter name
5. Enter email
6. Enter no. of persons
7. If food available
   a. Order for free
   b. Pay 10%
   c. Pay 15%
   d. Pay with debit card or COD
   e. Enter card number
8. Confirm order

- # Admin Algorithm pseudo code:

Allows admin to add new members.

1. Required Sign in
2. View all incoming requests
3. View the details of each request
4. Testing the documents for verification.
   a. Monthly Expenses Records
   b. Income Certificate

5. Accept user request if it is needy.
6. Decline request in case of wrong documents.
7. Store all the data in database using Sql/Mongo DB language.
8. Confirmation message send to user.

- # User Algorithm pseudo code:

Allow User to Order food from restaurants.

1. Required User Sign in.
2. To order food user press Find Food button.
3. All available nearby restaurants will display.
4. User can also search it by name.
5. Payment can be paid by hand or by debit card according to user choice.
6. Go to setting to modify any profile data.
7. Press pre Order Buttons to view previous orders.
8. On receiving ordered food bill will paid to delivery boy.

- # Delivery Algorithm pseudo code:

Allow Delivery boy to deliver food from restaurants.

1. On placing the order food request is reached delivery boy .
2. On requesting the check delivery boy's availability first.
3. If delivery boy is available he accepts the offer.
4. Food will be delivered on required location.
5. Bill will be received after food is received by the user.
6. Delivery charges will paid according to user's choice.
    a. No charges.
    b. 10% charges of total payment.
    c. 15% charges of total payment.

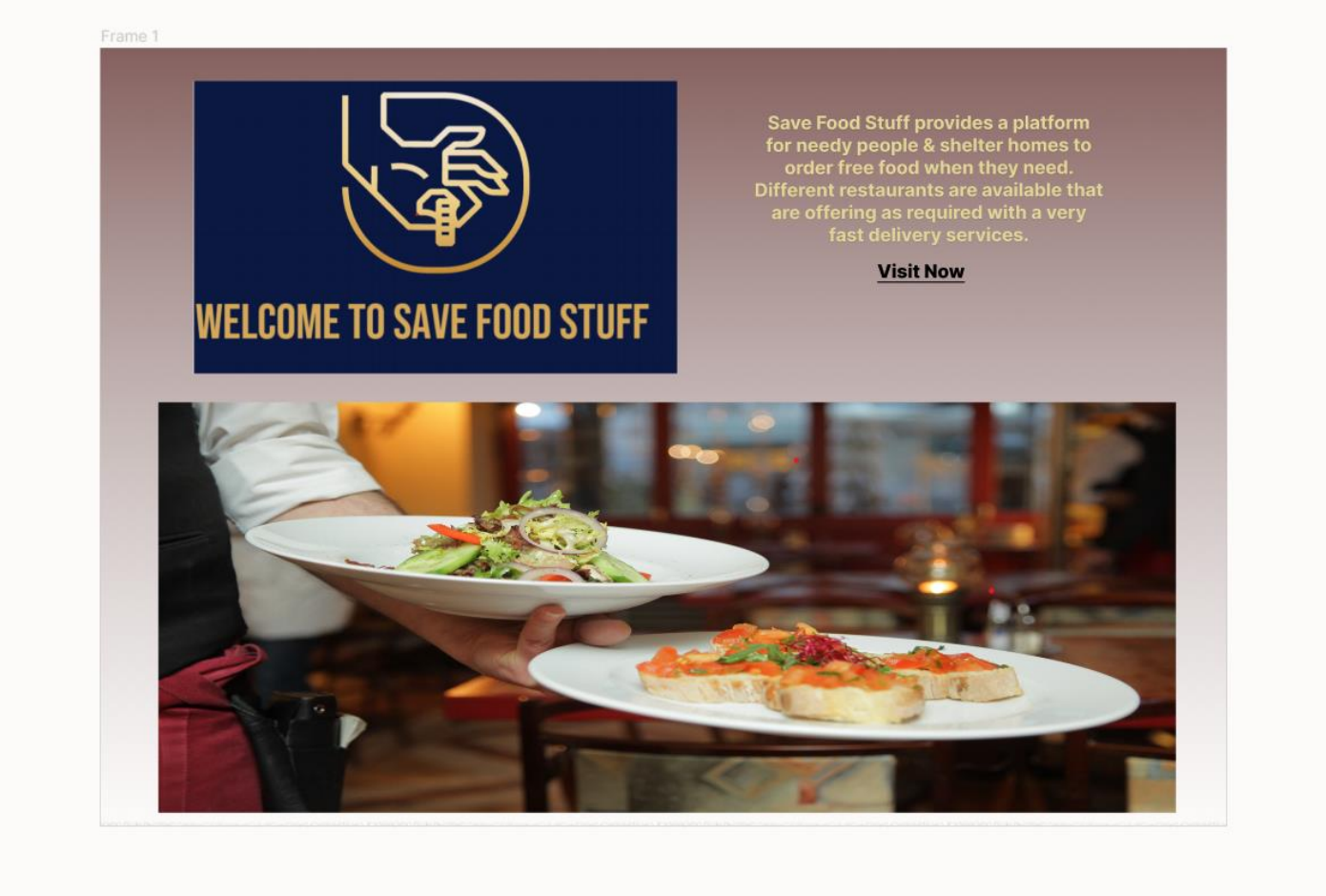
- # Restaurant Algorithm pseudo code:

Allow restaurants to transfer food to needy people.

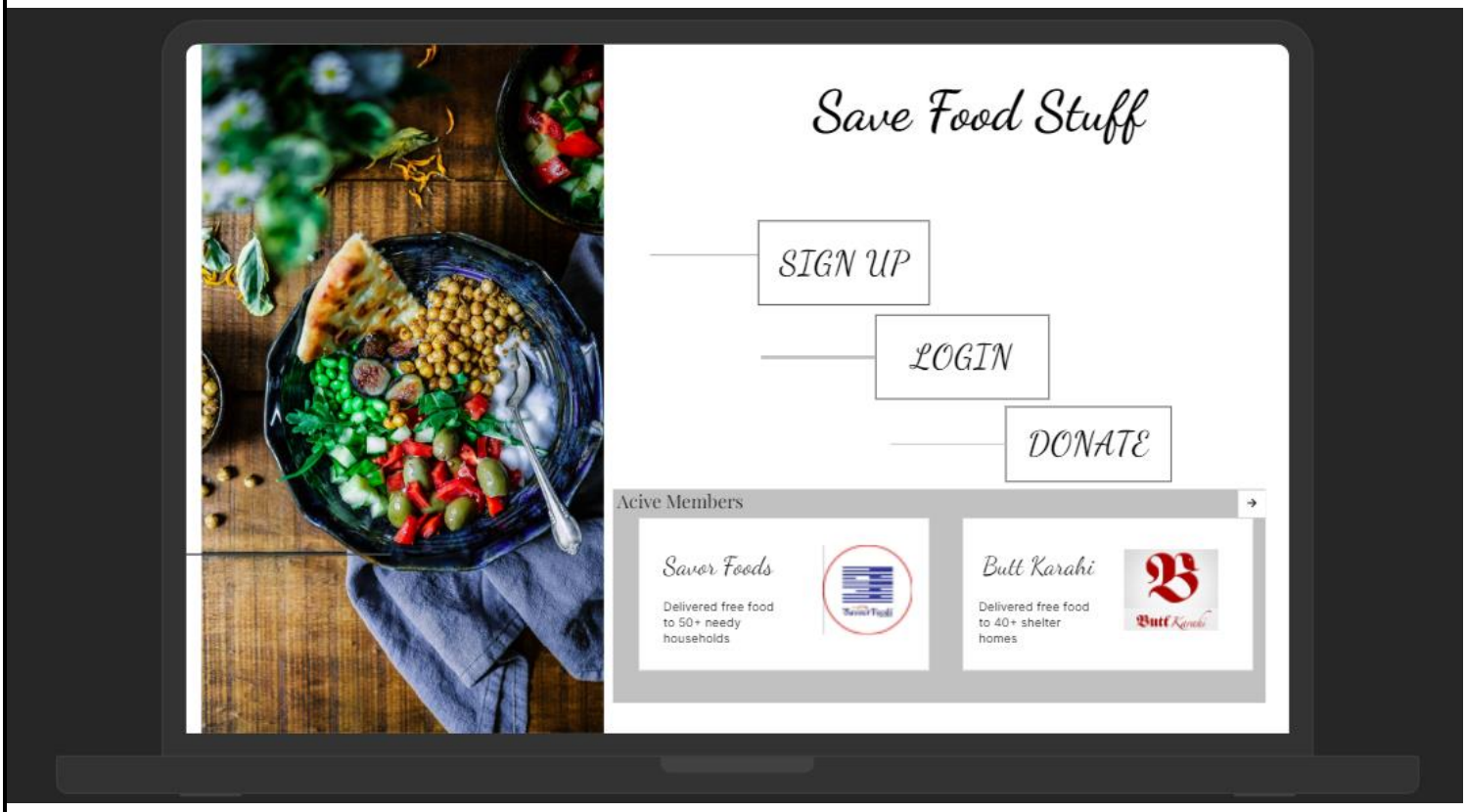
1. On this site restaurants add specific amount of food per person.
2. By visiting restaurants user can easily viewed their profile.
3. It will serve Food on FIFO operation.
4. User have to register him first.
5. Go to find food Button.
6. List of restaurants appeared that offering food.
7. Ordered food will be delivered on required location in minimum time.

8. User Interface Design:

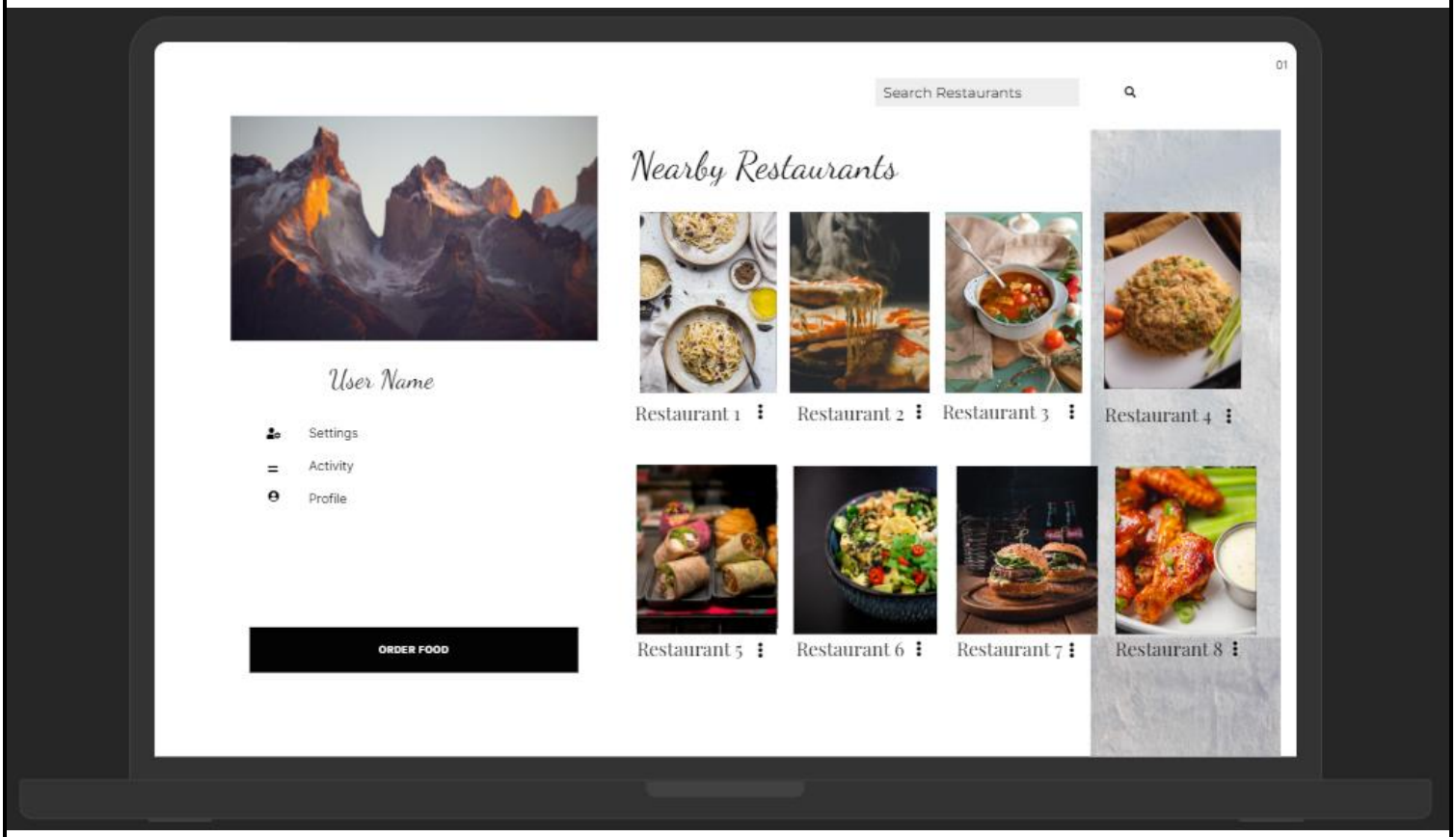
ASSIGNMENT#3
SOFTARE ENGINEERING CONCEPTS (CSC-291)



ASSIGNMENT#3
SOFTARE ENGINEERING CONCEPTS (CSC-291)



ASSIGNMENT#3
SOFTARE ENGINEERING CONCEPTS (CSC-291)



Conclusion

This document specifies the design, architecture and user interface of our project.

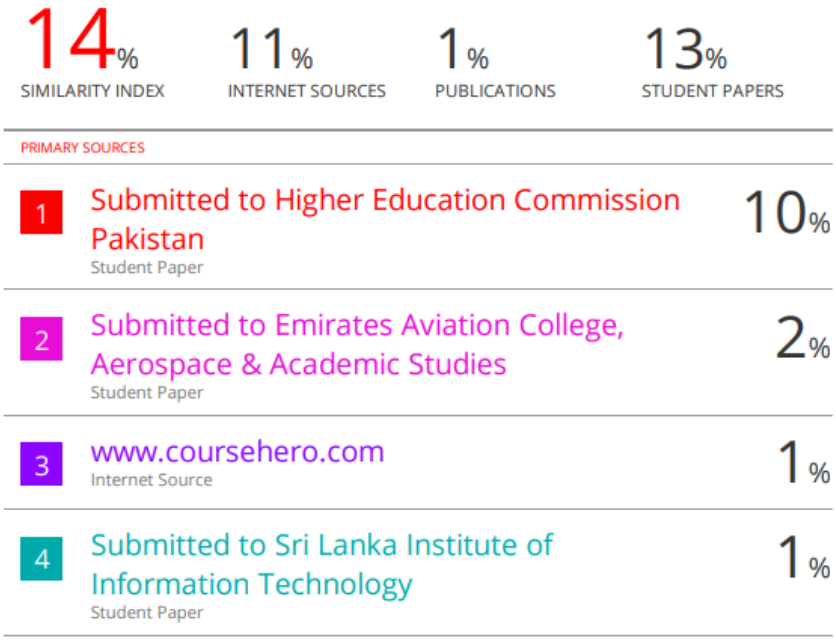
7. References

https://app.uizard.io/prototypes/rgLp11mKjnuVKgg9zBj5

https://pern-my.sharepoint.com/:u:/r/personal/fa20-bcs-029_isbstudent_comsats_edu_pk/_layouts/15/Doc.aspx?sourcedoc=%7B02241a2e-caa7-41b4-a398-39e8c9f346f1%7D&action=edit&or=PrevEdit&cid=c241ae5f-c708-4f2a-9154-c206132fea4e

https://pern-my.sharepoint.com/:u:/r/personal/fa20-bcs-029_isbstudent_comsats_edu_pk/_layouts/15/Doc.aspx?sourcedoc=%7B65c509d0-a61b-4f5e-b57f-034f8870ca51%7D&action=edit&or=PrevEdit&cid=2be4cc84-c4f8-49c1-85a5-ed656f546433

8. Plagiarism Report



Work Division:

ASSIGNMENT#3
SOFTARE ENGINEERING CONCEPTS (CSC-291)

FA20-BCS-027: Introduction, Design Methodology and Software Process Model, Activity Diagrams, Sequence Diagram (Delivery, Reviews. Admin), Class Diagram, Algorithms & Implementation (Sign- in, Donation, Find Food, Reviews), Mockup (Interface 1), Entity Relationship Diagram

FA20-BCS-029: Overview, Activity Diagrams, Sequence Diagram (Find Food, Post Food, Donor), Process Flow Diagram, Data Design Schema, Algorithms & Implementation (Sign- in, Donation, Find Food, Reviews), Mockups (Interface 2,3)