

SERVE THE DISHES

Project proposal

By

Chhavi Garg 300330385

Ekta Jain 300352096

Kirti Sethi 300352212

Blossom Blossom 300347123

Douglas College

Computing Studies and Information System (CSIS)

CSIS 3275– Software Engineering

Caesar Jude Clemente

06 August 2022

ABSTRACT

Canada is witnessing a rapid increase in the number of international students. Our project aims to build software that proves successful in tackling some common problems faced by international students when they arrive in a new country.

Getting a job, without any prior work experience, in a new country is a very difficult especially in your first few months of arrival. Hence, we are building this software that will provide students and other individuals, who are searching for jobs, a platform to deliver orders to nearby places. There are also many other exciting features in our software which are discussed further in details. We are excited to turn our ideas into real working prototype.

BUSINESS CASE

DESCRIPTION: -

The software uses the location search where the users can order from any restaurants with in few kilometers of their location it will also provide live location of servers using GPS technology. This software will provide the servers with a list of orders which are ready to be delivered and looking for servers to deliver them in that location. The servers could be students or any individual looking for jobs or a few hours to manage their finances and weekly hours.

One of the main features of this application would be that the servers not always need to use any mode of transportation such as a car or Trans link.

It also helps specially challenged people like people on wheelchair etc.

FEATURES: -

- I. No compulsory mode of transportation is required.
- II. Wheelchair friendly.
- III. Minimum one identification proof is required for servers.
- IV. Perks on good ratings.
- V. Provides hot leads in a local area.
- VI. Updates on every step of the order.
- VII. The details of the delivery associate are shared, and customers can also track the real-time location of their order.

MARKETING: -

People gets educated about a certain product through marketing which helps in boosting sales. For promoting our business, we would be taking a great help from all the social media platforms such as facebook, Instagram and linkedIn advertisements.

Improving website SEO and email marketing would be our other ways to attract more servers and restaurants.

PROBLEM STATEMENTS

Understanding The Problems: -

- I. Security of the application could be a major concern.
- II. Implementing all features of our prototype on time could be difficult.
- III. Performance could be an issue with increasing users.
- IV. Sharing live location of servers with the users could be challenging.

Planning A Solution: -

- I. We will try to put application on cloud to make it more secure.
- II. We will try to increase the coding hours of each developer.
- III. We will try to roll-up new updates on a regular basis to make our UI more user friendly.
- IV. We will try to make our maps clearer and more efficient for users.

Examine Result for Accuracy: -

- I. Dividing the software in-core functionality and validate each functionality individually to remove bugs from the software.
- II. Testing would be done for each environmental case possible to make sure any error input entered by the users could be resolved.

NON- FUNCTIONAL REQUIREMENTS

These are the quality attributes that describes the ways how our system should behave. The list of some Non-functional requirements are: -

1. Usability of the system.
2. Legal or Regulatory Requirements.
3. Reliability
4. Performance

Let us discuss these requirements one by one with respect to our proposed system (ServeTheDishes).

1. Usability of the system

It defines the ease with which a user can interact with our system to achieve required goals effectively and efficiently.

In our system (ServeTheDishes) , the user is that person who is interested in picking up the order within their 1 k.m. of the radius. So, to make the system efficient, we will try that the user can easily sign up, and if he or she forgets password then we will send an email on their gmail account. Moreover, if the user searches for the restaurant for picking up the order, then they can easily do it , so, system will track the location of the user, on basis of which the list of restaurants will be displayed, within their 1k.m. of the radius.

2. Legal or Regulatory Requirements

These are the requirements that describe the system adherence to laws, which our system will be designed to comply with.

For example, when we operate our system, then we will make sure that it should be licensed by the local food delivery system authority - Food System.

3. Reliability and Performance

Reliability is one of the most important non-functional requirements of the system. It is one of the factors that decides whether the system we developed is a failure or a success. To achieve high reliability, our team will eliminate all bugs that may influence the code safety and issues with system components, for example, debugger tools will be used with the IDE that we will use.

Performance is basically how our system will perform whenever user will interact with it. The focus of our team will be to improve the performance, as poor performance will decrease the customer's interest. For example, when the applicant searches for the nearby restaurant for order pickup, then our system will try to provide as much restaurants as possible, within the distance of 1 k.m.

REVENUE RECOGNITION

Revenue Model is a framework for generating financial income. To elaborate, total sum generated from the sales and services offered by the platform.

There are multiple models available, and we will be using three of them: -

- Commission Model
- Advertising Model
- Subscription Model

Commission Model

Some basic income will be paid to the server (Independent contractor) for pick up and drop off services. 25% of the total earning will go to ServeTheDishes.

Full tip money given by the user will go the server.

Quest- Promotion strategies:

- Extra delivery offers between 12:30pm to 1:30pm.
- Extra \$10 dollars bonus for minimum 4 deliveries to the servers. Then \$3+ on every additional delivery.
- \$2 discount for the subscribed user on every order during the given hour.

Advertising Model

- 15% from each order that a restaurant receives through the advertised dishes/offers will go to ServeTheDishes.

Subscription Model

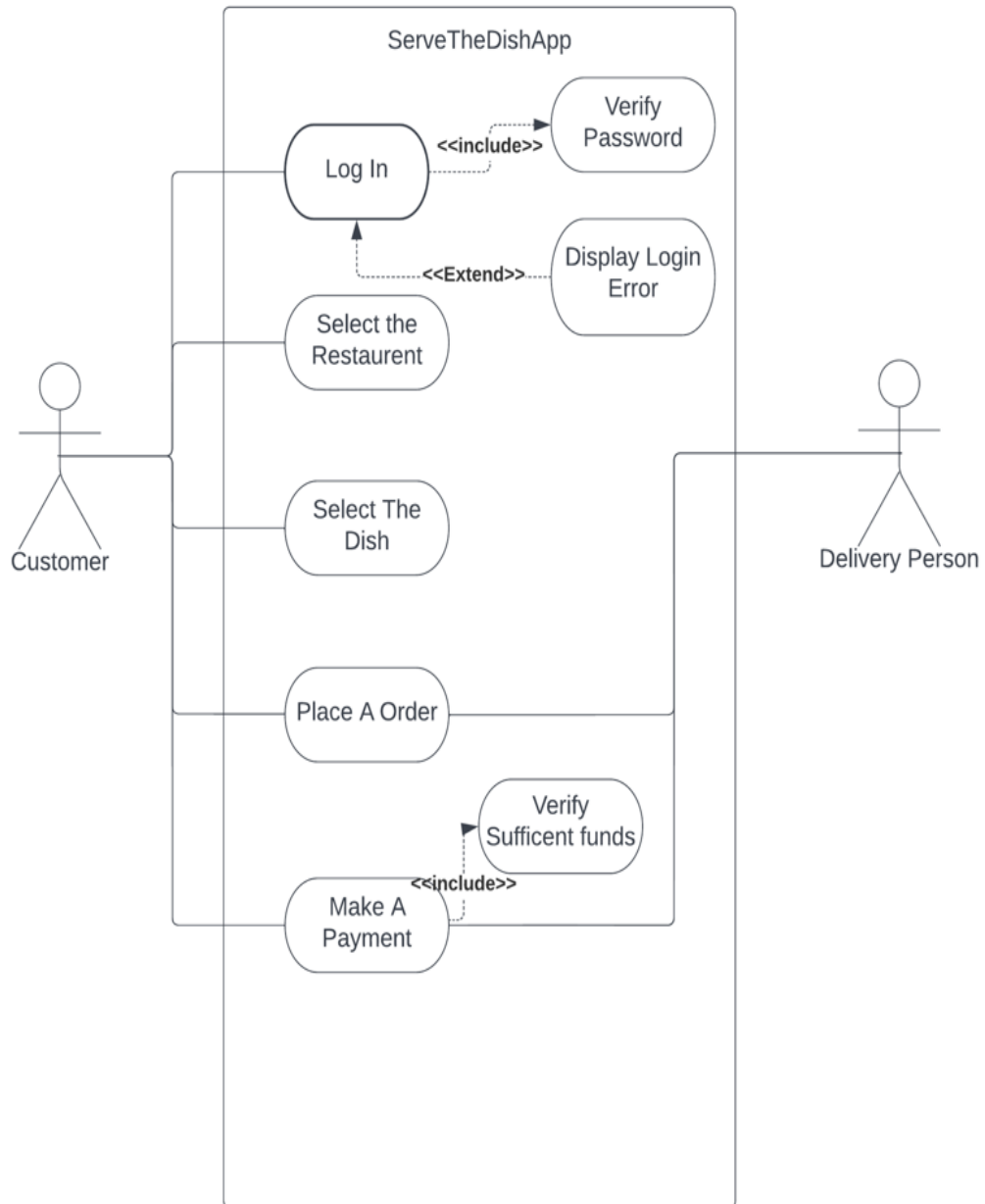
- \$8.90 charges for every subscriber within Canada
- Offers to entice subscription:
- 10% discount for the user on first three orders within the first 2 weeks of subscription.

FUNCTIONAL REQUIREMENTS

<u>USER</u>	<u>Server</u>
Customer should be able to view Restaurant's Menu.	Server must be able to see the tips given by the customer.
Changes into the menu can be made within first 5 minutes of the order. After that, order will be sent to the restaurant.	Further, the customer's entered destination should be visible and traceable.
Place maximum of three orders at the same time.	Quantity of orders from a restaurant will be displayed, and Server can decide the number of orders they are comfortable to deliver.
They should be easily able to store multiple card information and use it to order every time.	

USE CASE DIAGRAM

This diagram helped us to give a good picture of what we had in our minds in terms of different functionalities.



DESIGN GOALS

There are two kinds of users that our system should account for: Servers and Customers.

What we know about our system requirements is:

- Servers must be able to frequently notify the service regarding their current location and availability.
- User should be able to see all nearby servers in real-time.
- Users can request orders using a destination and pickup time.
- Nearby servers should be notified when a user orders from a restaurant.
- Once an order is accepted, both the server and user must see the other's current location for the entire duration of the trip.
- Once the order is complete, the server completes the order and should then be available for another customer.

System Design:

- We need to update data structures to reflect active server's reported locations every three seconds.
- We need a quick mechanism to propagate the current location of nearby server to customers in the area.

SYSTEM ARCHITECTURE

Application uses Spring MVC as a backend. Data storing tables will be made for restaurants their Menu Users and Delivery Associates (Servers). Specials components and models would be design for each table and with the help of controller would be displayed in View part of the Software.

Front End of the software would be design would be made by React. The functions to display things on Map and track user locations would be done with the help of JavaScript on frontend on React. All the Data to make these functions work would be taken from Backend.

CONFIGURATIONS

The two opposing elements that make up a computer system are computer hardware and computer software. Hardware without software is just an expensive piece of metal since software requires a hardware configuration to run on. Although they complement one another, the two are very distinct in their roles and skills. The hardware and software we are using are as follows: -

Hardware:

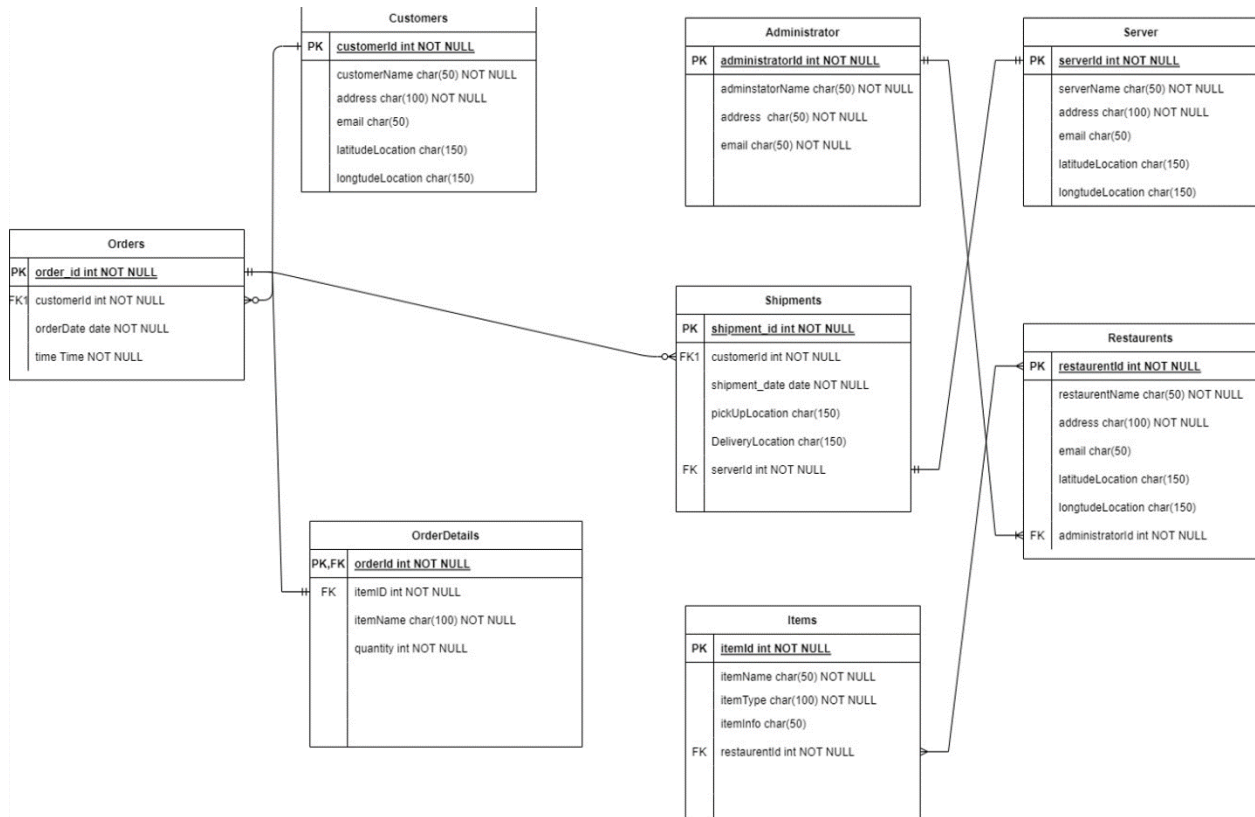
- Both IOS and Android compatible.
- Only one sim slot.
- Minimum Ram of 2 GB.
- Minimum of an 8 Core processor also referred to as Octa Core.
- Minimum processor speed of 2 GHZ.
- Device must have both A-GPS and GLONASS chips internally.
- Minimum of 2GB per of data a month.
- You will need airtime to occasionally call server.

Software:

- HTML
- CSS
- React JS
- Express
- NodeJS
- MongoDB

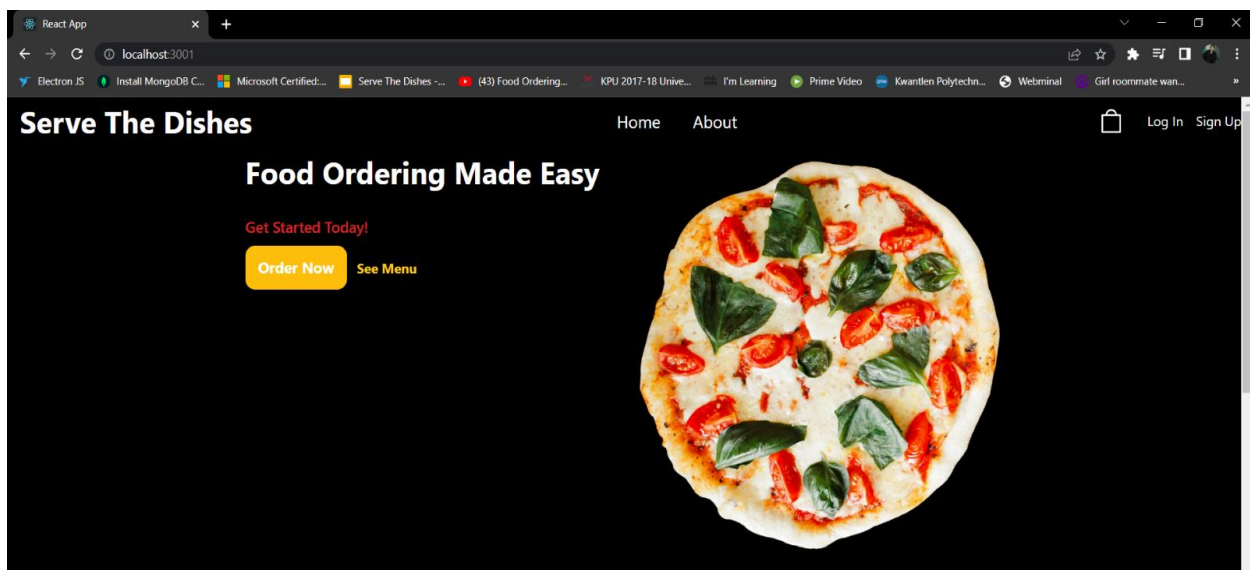
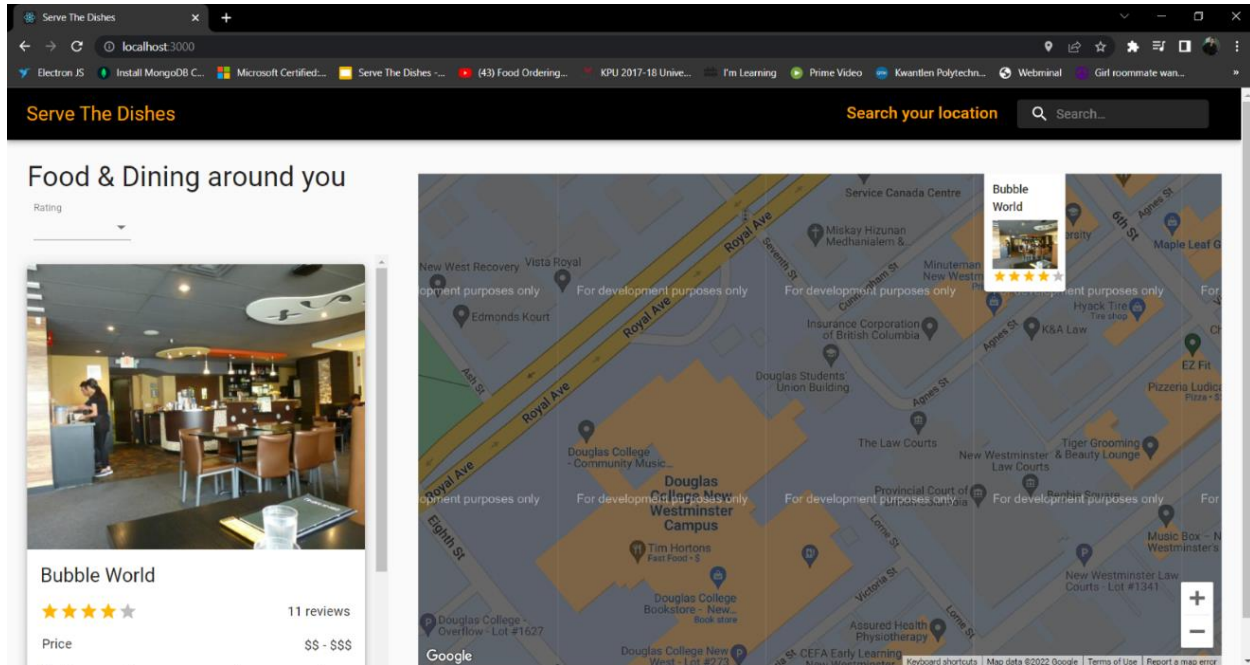
DATABASE DESIGN

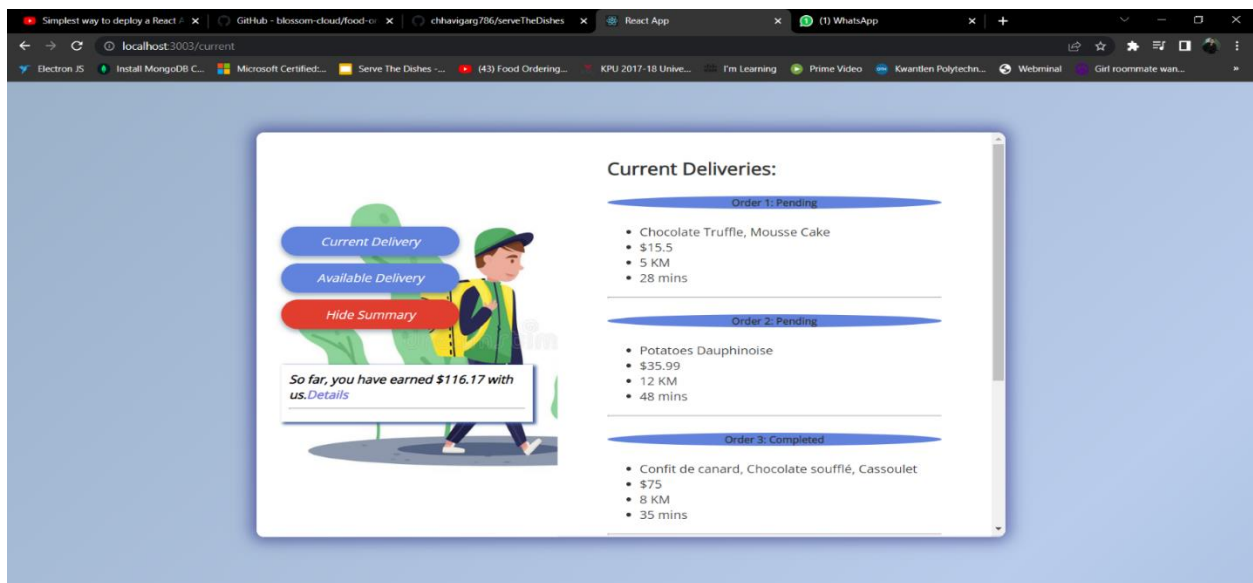
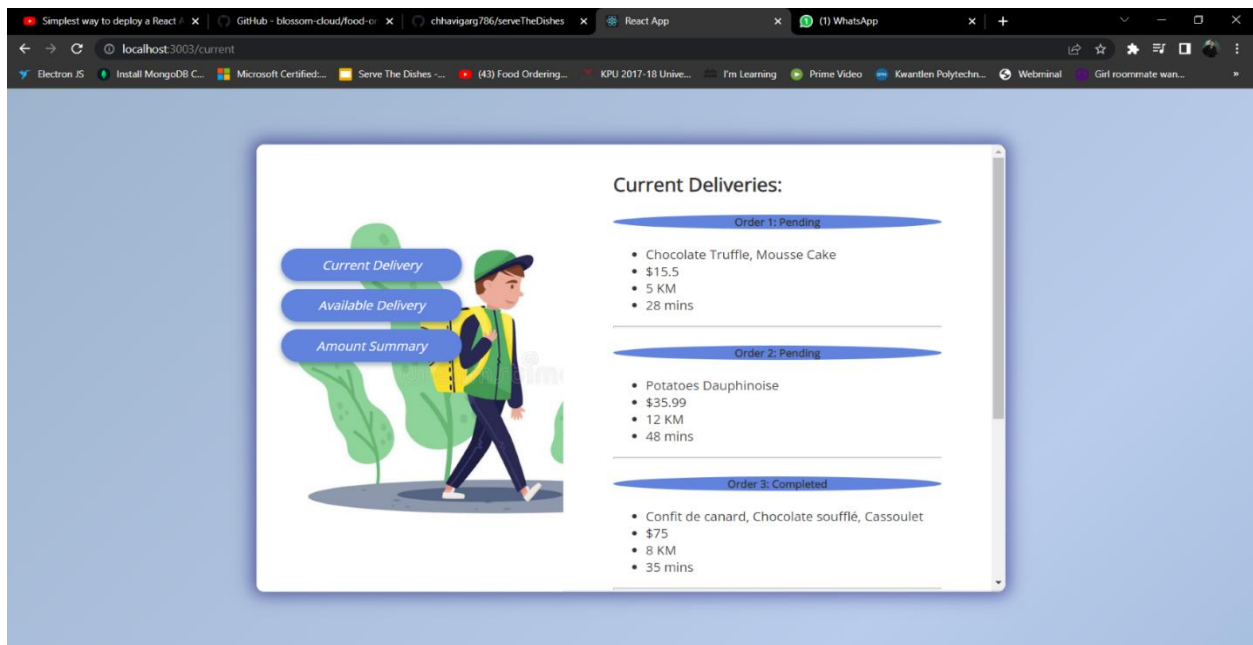
The database design for the application is shown as below: -



INTERFACE DESIGN

Currently, our application UI is design is in progress. The following are the screenshots: -





NON- FUNCTIONAL REQUIREMENTS

These are the quality attributes that describes the ways how our system should behave. The list of some non-functional requirements are:-

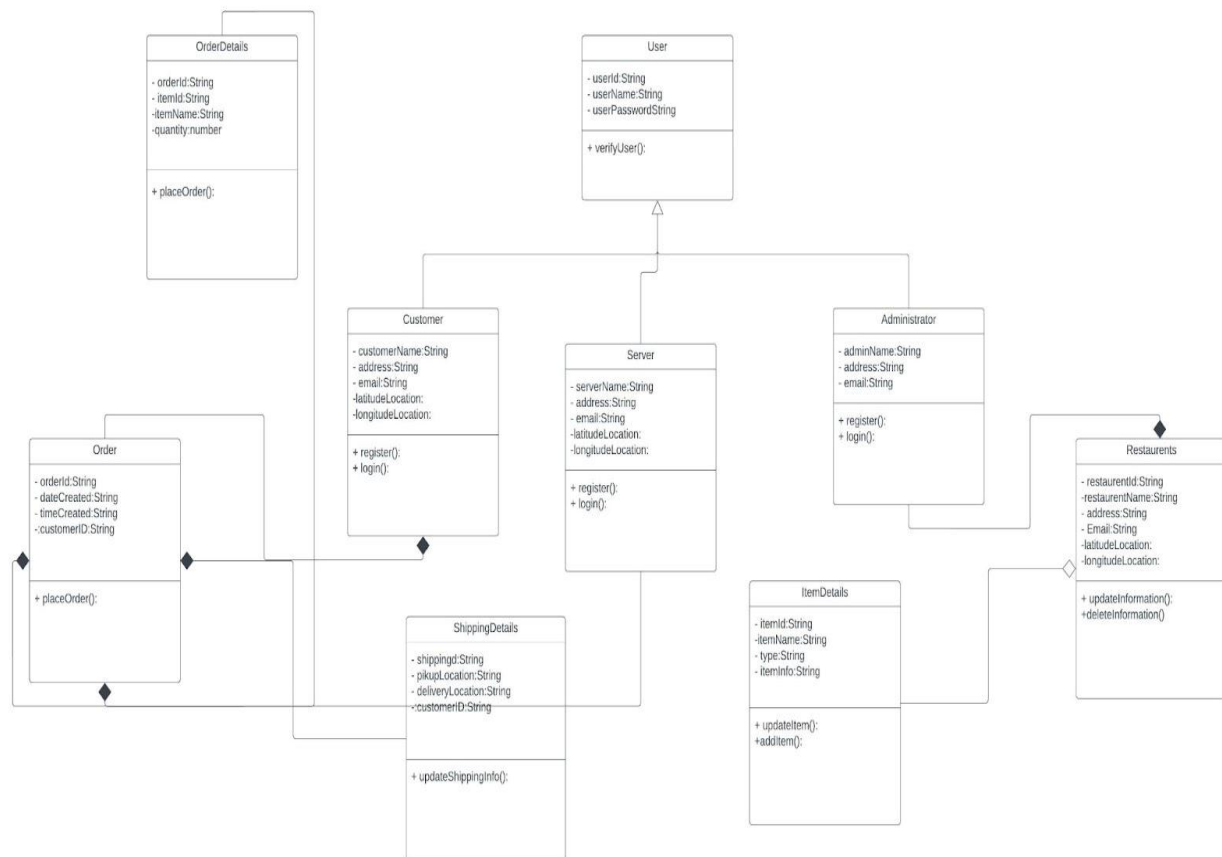
- Usability of the system.
- Legal or Regulatory Requirements.
- Reliability
- Performance

To achieve these requirements, the application would follow the mentioned approaches: -

- Use a defined classification
- Engage with the development team
- Use 'Invented Wheels'
- Use automated testing tools

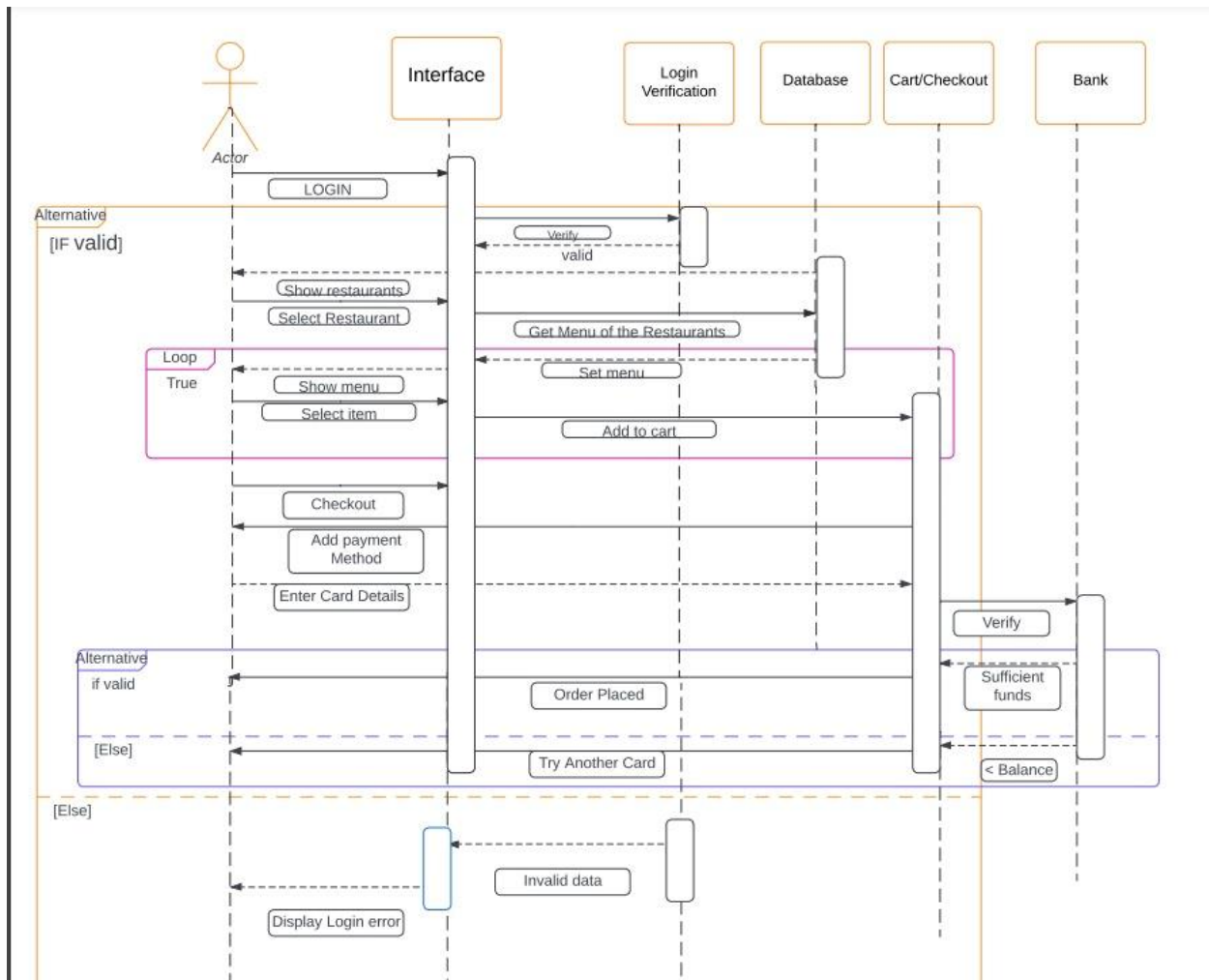
CLASS DIAGRAMS

This diagram helped us to give a good picture of what we had in our minds in terms of different functionalities.



SEQUENCE DIAGRAM

The sequence diagram is a useful diagram to use when laying out a system's design and describing its requirements.



IMPLEMENTATION

Scrum is a framework for team collaboration. Scrum enables teams to learn from experiences, self-organize while working on a problem, and reflect on their victories and losses to continuously improve.

Approach used for this project was scrum so that the whole team works together to achieve the goal. We divided our work in many small phases: -

- Creating backlogs
- Planning
- Conducting regular sprints
- Implementation
- Testing

We are using react technology with that we can integrate different react apps together. To complete our project, we also took some ready to go components to make it look more professional and efficient as stated previously in our reports.

SPECIFICATIONS: -

- All the available restaurants near our live location are displayed.
- A user gets two options: -
 - I. Order directly from the restaurant website.
 - II. Order can redirect to our app.
- Ratings of the restaurants available as a filter.
- Works in every country.
- Easy login and signups for the user.
- Provides hot leads in a local area.
- Works on different screen sizes.

TESTING

Testing of the code is done to make sure that the code works correctly. This helps to detect and protect against bugs in the future. Sometimes developers write unit tests first, then write the code.

The following are the testing done for this app: -

- Bugs were removed while coding.
- Added filters to remove the restaurants which does not have any ratings or any name which are taken from the API to provided user a clear view with their choices.
- Approach for testing was black box testing, as it helps to find the gaps in functionality, usability, and other features.

FUTURE SCOPE

The following are the future scopes for the application: -

- Connect to more restaurants and expand the business.
- Except than food, we can add parcels and other packages too in our system.
- More career options in the future for students in the professional sector to provide them with the best experience.