Epic Eats

An Online Food Ordering System with 3D Custom Food Builder

Department of Information and Communication Technology
Faculty of Technology
University of Sri Jayewardenepura
Sri Lanka

Project Title

Epic Eats: An Online Food Ordering System with 3D Custom Food Builder

Project Personnel

Academic Supervisor

Dr. Prabhani Liyanage
Senior Lecturer
Department of Information and Communication Technology,
Faculty of Technology
University of Sri Jayewardenepura,
prabani@sjp.ac.lk
+94 71 446 3920

Team Members

1. **Fathima Akeela**
Software Technology
Faculty of Technology
University of Sri Jayewardenepura
ICT/19/803
ict19803@fot.sjp.ac.lk
+94 76 605 3098

2. **Dhanushika Jeganadhan**
Software Technology
Faculty of Technology
University of Sri Jayewardenepura
ICT/19/817
ict19817@fot.sjp.ac.lk
+94 76 381 2831

3. **Yohan Sandun**
Software Technology
Faculty of Technology
University of Sri Jayewardenepura
ICT/19/838
ict19838@fot.sjp.ac.lk
+94 71 426 6758

4. **Pasindu Prabhath**
Software Technology
Faculty of Technology
University of Sri Jayewardenepura
ICT/19/803
ict19803@fot.sjp.ac.lk
+94 76 980 8220

5. **Nath Duminda**
Network Technology
Faculty of Technology
University of Sri Jayewardenepura
ICT/19/809
ict19809@fot.sjp.ac.lk
+94 76 549 2532

Table of Contents

1. Introduction and Background

Food ordering has risen rapidly in recent years, from ordering over the phone call to ordering through a mobile application. Nowadays, ordering food is as simple as a fingertip away, which saves a lot of time and money for customers. However, we would still choose food from the existing menu rather than customize it according to our preferences. Most of the customers are avoiding some kinds of foods because of their health concerns even though they like to have these foods and it's negatively affecting the revenue of the restaurants. As a result, there may be some customers who are dissatisfied with their orders. Thus, it's better to have a simple but comprehensive application to customize and order the food instead of directly ordering from the existing menu.

The proposed system has been designed to overcome the issue mentioned above. The main goal is to create a mobile application that allows users to personalize their food using accessible ingredients in an immersive 3D environment. The application will render changes that customers make in real-time and it will allow user to build their own food and order. With this system, customers can pick ingredients matching to their needs. Besides that, mobile app works same as other food ordering applications on the market.

The mobile application can be used by any user who is willing to order food from available restaurants. Once the user login to the application, it'll show a menu of the particular restaurant. Once the user clicks on a customizable food item, the Application will allow the user to add ingredients as he/she wants and the application will display in real-time how the food will look like. Once the user is satisfied with their recipe user can continue to check out either by using a credit/debit card or cash.

2. System Requirements

System requirements are divided into functional and non-functional requirements. Final product must meet below mentioned functional and non-functional requirements in order to full fill the user needs. The basic system behavior is defined by the functional requirements.

2.1. Functional Requirements

Functional Requirements are describing what the systems must do. The following Use Case diagram also depicts the functional requirements of the system.

- The user should be able to create an account if it does not exist already.
- The user should be able to login into the application.
- The user should be able to manage their account and personal information.
- The user should be able to customize food with selected ingredients using 3D interface.
- The user should be able to place an order.
- The user should be able to cancel an order.
- The user should be able to pick a delivery address.
- The user should be able to make a payment.
- The restaurant should be able to add food items to the menu.
- The restaurant should be able to accept or reject orders.
- The restaurant should be able to update the details of each food item.
- The system should be able to generate receipts.

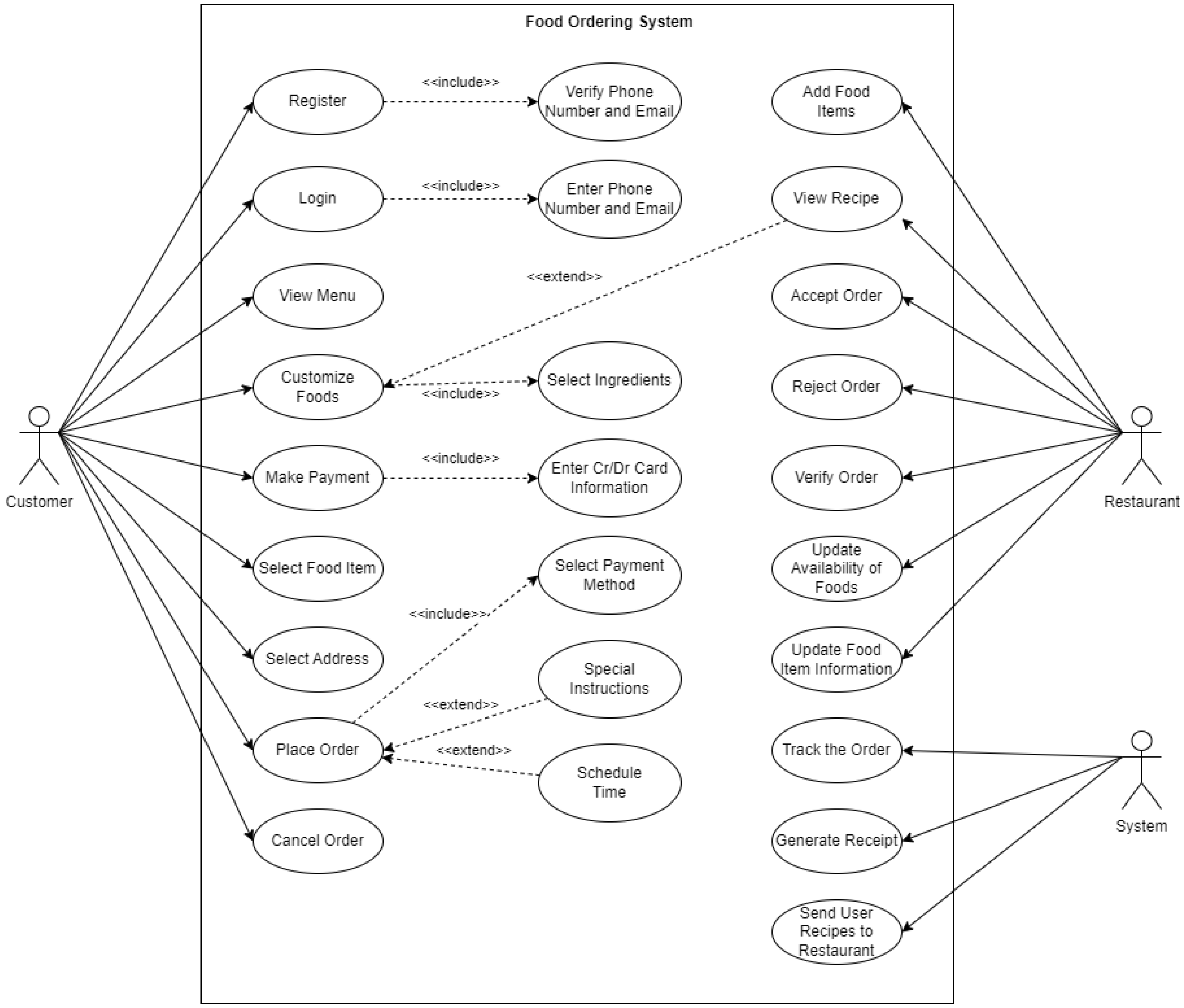

Figure 2.1.1 – Use case diagram of the proposed system

2.2. Non-Functional Requirements

- The system will be able to behave consistently when using.
- The system will be accessible at any time by the users.
- The system will be able to track the order in real time.
- The system will be able to check the distance between restaurant and user before placing an order.
- Usability – User interface should be easily understandable and easily usable by the users.
- Security – System should keep confidential data of the users safe like phone number and location.
- Performance – System is responsible for updating the 3D view in real-time.
- Customers should be able to schedule their orders.

3. System Design

The process of designing the elements of a system, such as architecture, modules, and components, as well as how they interact with each other and the data that flows through the system, is known as the system design. This section will describe the basic components of the system and their interactions. Methods like Wireframing, ER Diagrams, Class Diagrams, Activity Diagrams, CRC (Class Responsibility Collaboration) Cards and Sequence Diagrams will be used.

3.1. Main Functionalities

This section will describe the design of main functionalities of the system.

3.1.1. Access to the application

In order to access the application users must login to the system. If the user is registered, he/she can use his phone number which was provided at the time of registration to log in to the system. If the user is a new user, he/she must create a new account in the system by providing their phone number, name, and email address. Once the user enters their phone number, the system will send an OTP (One Time Password) to that number and prompt the user to enter the password to verify the phone number. Once the phone number is confirmed, the user will get redirected to the menu screen.

3.1.2. Customizing the Food

The system has two types of food items. Which are normal items and customizable items. To customize a food item, it must be the type of customizable item. Once the user clicks on a customizable food item, the user will redirect to a screen where he/she can add a different kind of ingredients and customize the food they want. Once the user adds or removes an ingredient system will render the 3D image of that food item according to the user’s recipe.

3.1.3. Placing an Order

In order to place an order, the user must have one or more items in the cart. A cart is a virtual container for the user to keep interesting items in a temporary manner. Users can add or remove items from the cart before placing the order. As well as a user must select a payment method before placing an order. The system offers two payment methods, which are Cash on Delivery and Credit/Debit card payment. If the user is willing to pay once the order is delivered, he/she can continue to place the order. Otherwise, users have to provide their credit or debit card details and verify the card before placing the order. Once the user confirms the order, it’ll send to the restaurant. Restaurant can either accept or reject the order.

3.1.4. Cancelling an Order

Users can cancel an order until the restaurant is not started to prepare it and user won’t get charged until that. Once the restaurant update that they have started to prepare the order, user is no longer be able to cancel the order. The money user paid is not refundable anymore.

3.1.5. Tracking the order

Users can track their order once they have placed one. There are five states for the orders. Which are as follows,

1. **Order received**

   The order is accepted by the restaurant and now it’s being prepared.
2. **Order Prepared**

   The order is prepared and now it’s waiting to be picked up by delivery personnel.
3. **Order Picked**

   The order has been taken by the delivery personnel and is now being delivered. Users can track the order in real time using the map provided by the application.
4. **Order Delivered**

   The order has been successfully delivered to the user.
5. **Order Cancelled**

   The order has been canceled either by the user or the restaurant.

These states are updated real time in the application interface.

3.2. Wireframe

In this section wireframes are used to illustrate the system's structural design. The sizes and placements of page elements and application functionality will be described.

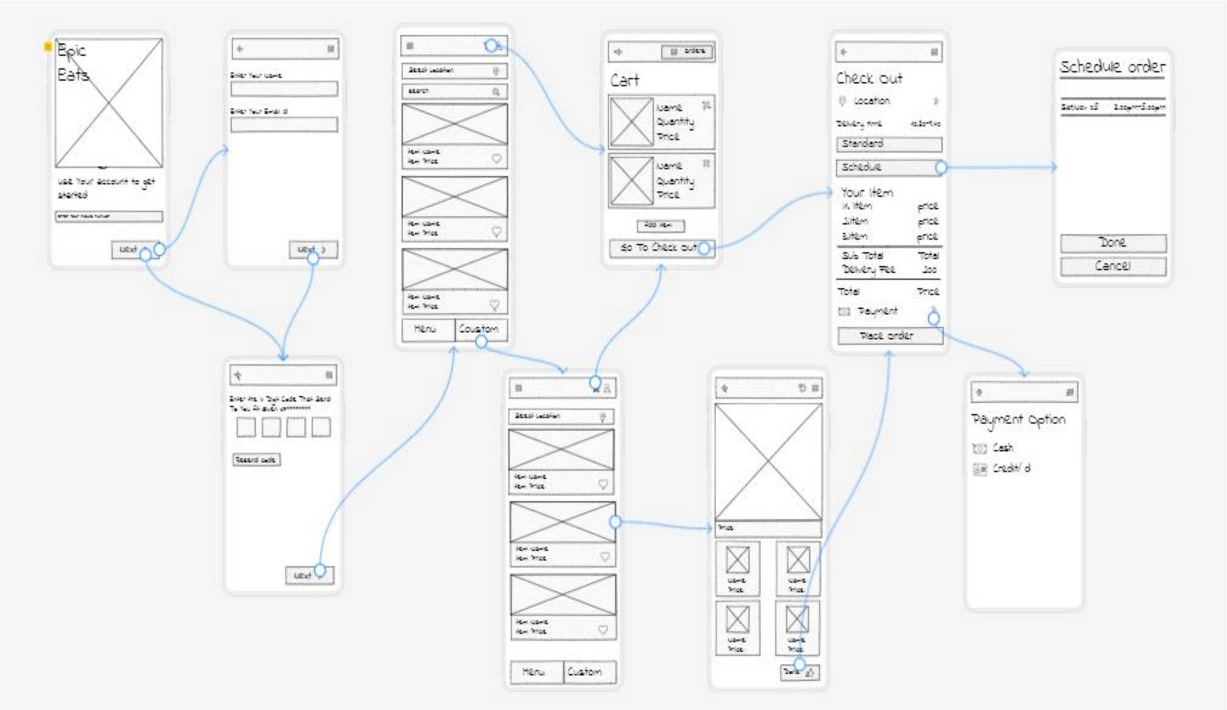

Figure 3.2.1 – Wireframe diagram illustrating the relationships between application screens.

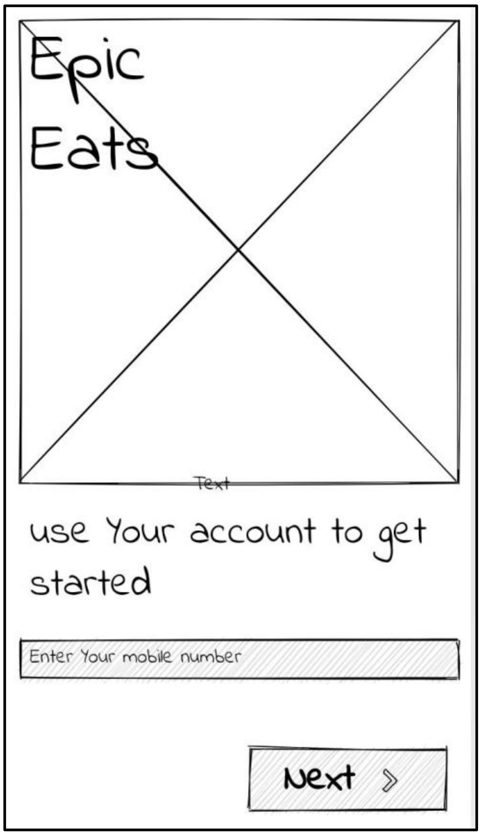

Figure 3.2.2 – Login screen

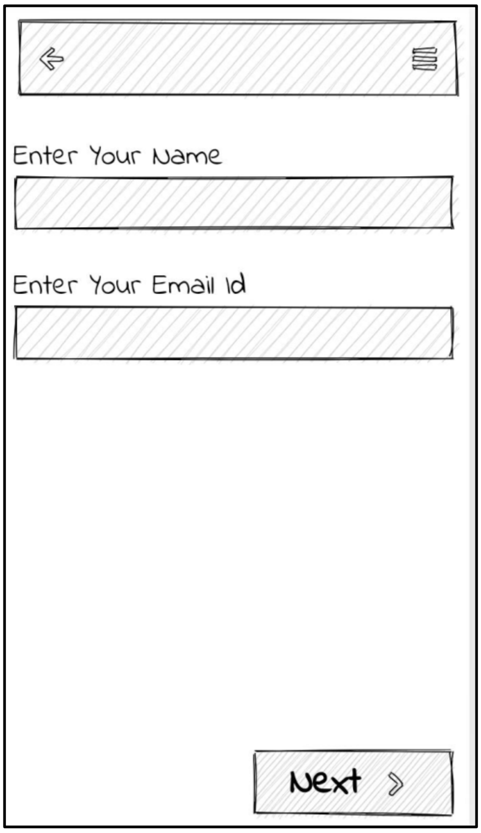

Figure 3.2.3 – Signup screen

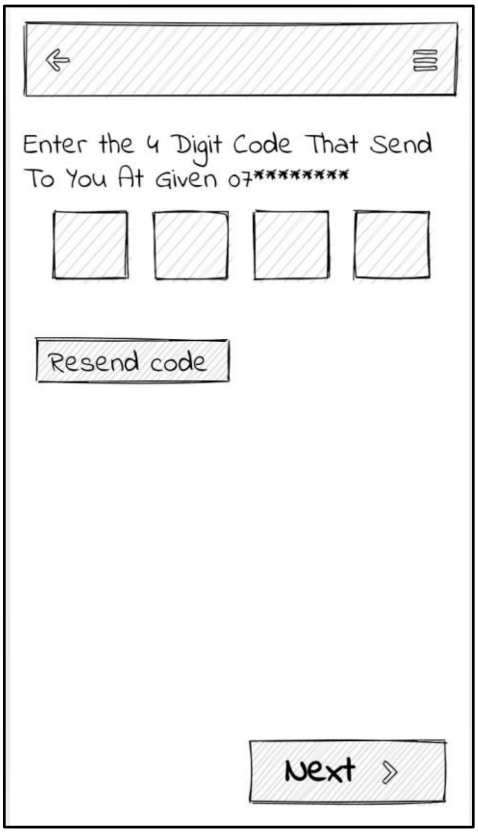
Enter the 4 Digit Code That Send To You At Given 07*********
Resend code
Next

Figure 3.2.4 – Phone number verification screen

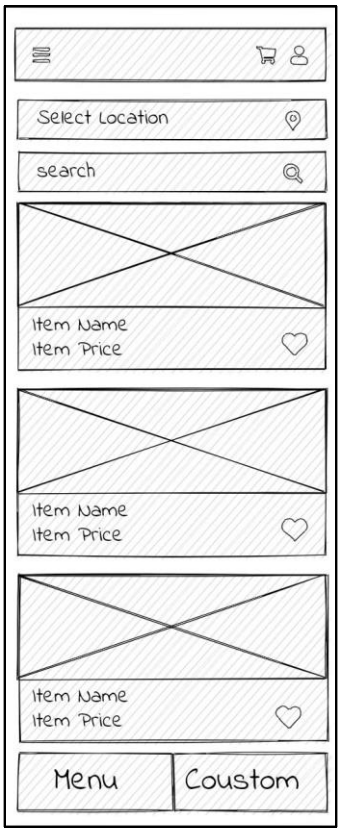

Figure 3.2.5 – Menu screen

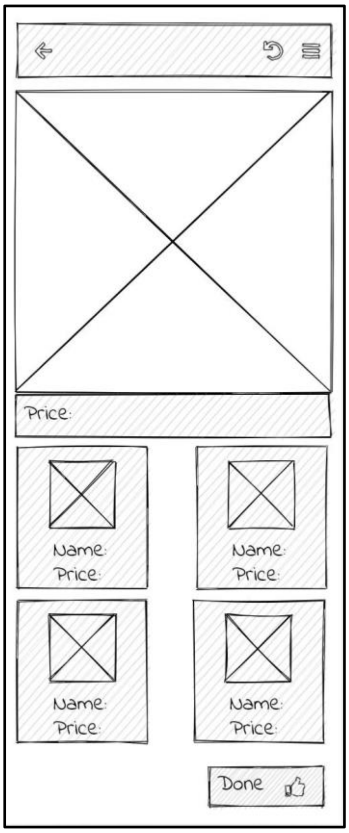

Figure 3.2.6 – Food item customizing screen

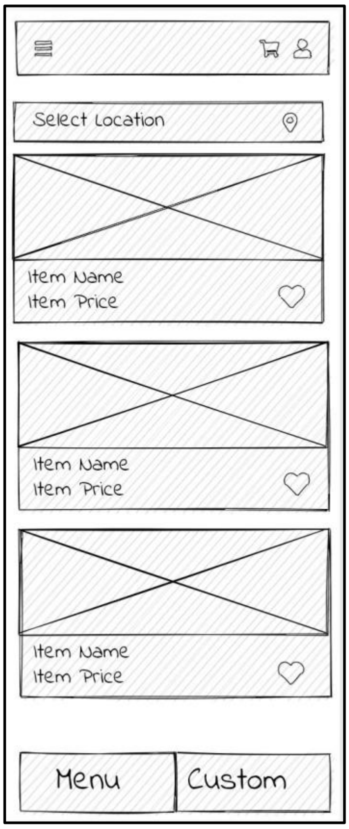

Figure 3.2.7 – Custom food items screen

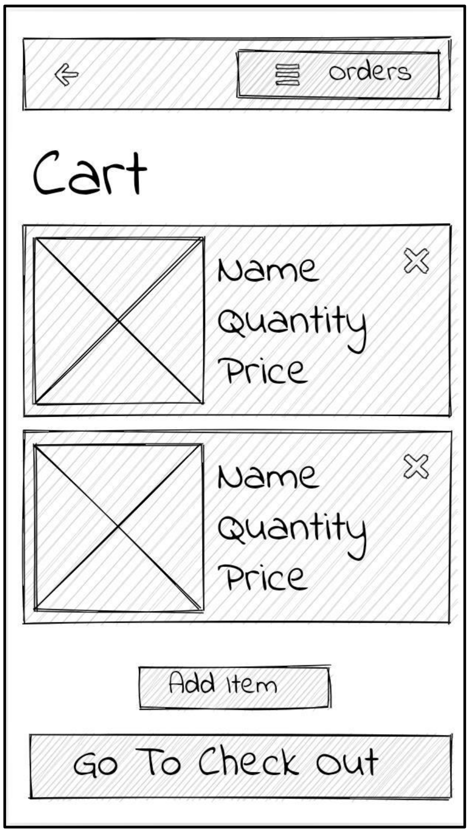

Figure 3.2.8 – Cart screen

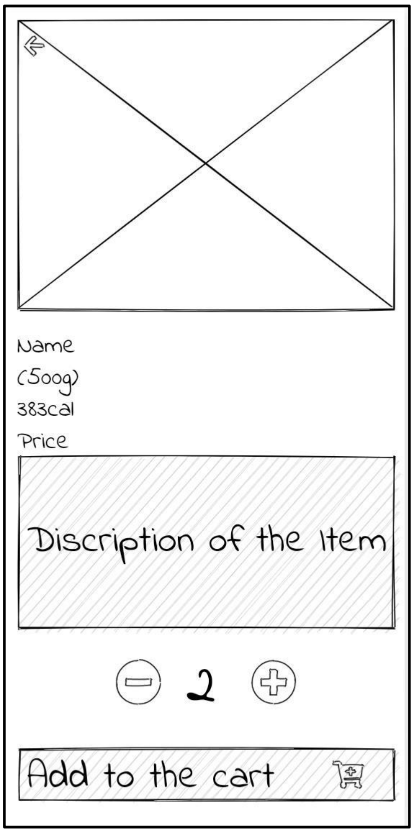

Figure 3.2.9 – Item description screen

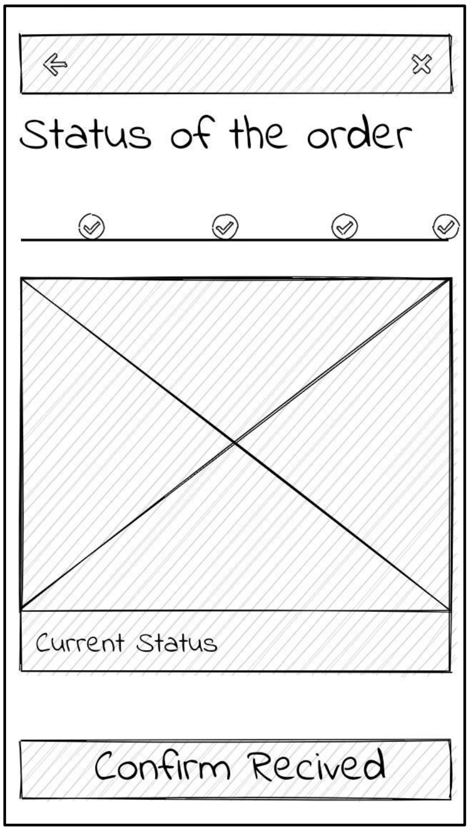

Figure 3.2.10 – Order status screen

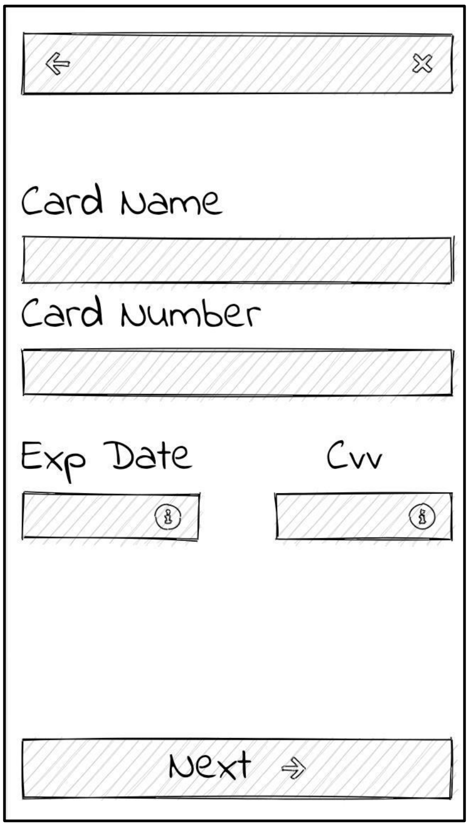

Figure 3.2.11 – Credit/Debit card screen

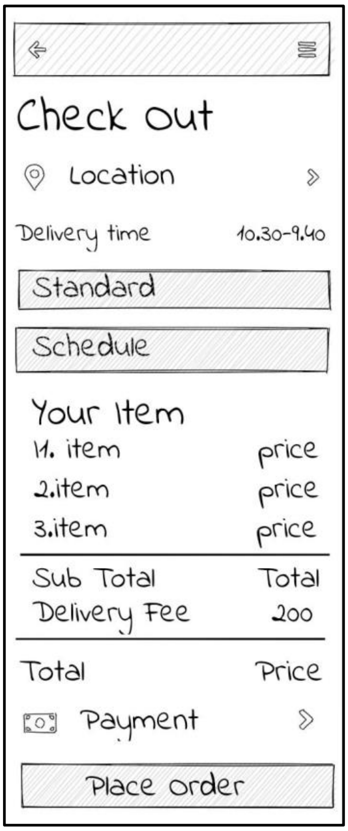

Figure 3.2.12 – Check out screen

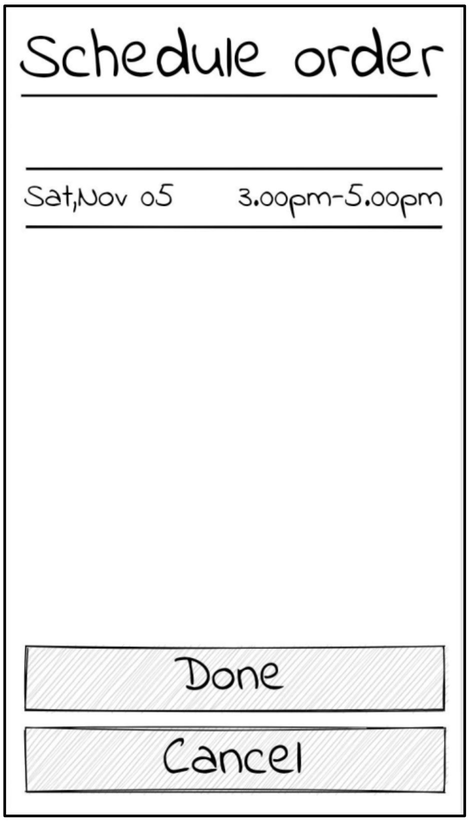

Figure 3.2.13 – Schedule order screen

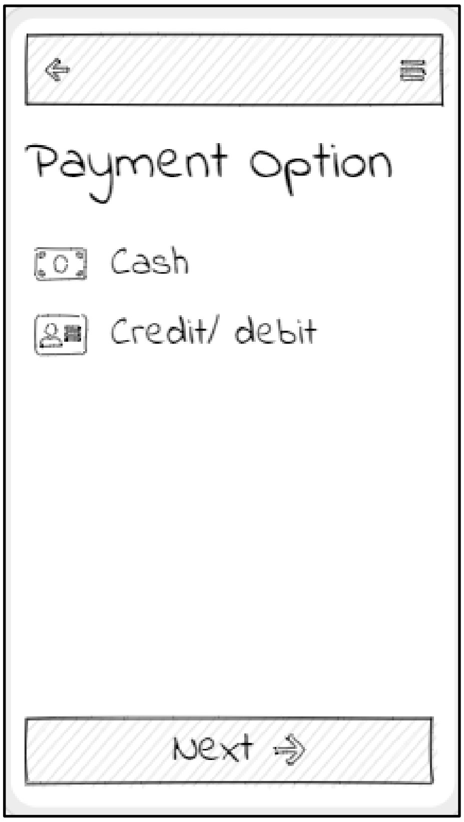

Figure 3.2.14 – Payment option screen

3.3. Entity Relationship (ER) Diagram

The following ER diagram describes the entities of the system and their relationships. This ERD will be used when creating the database. ER diagrams are playing a major role when producing high-quality databases.

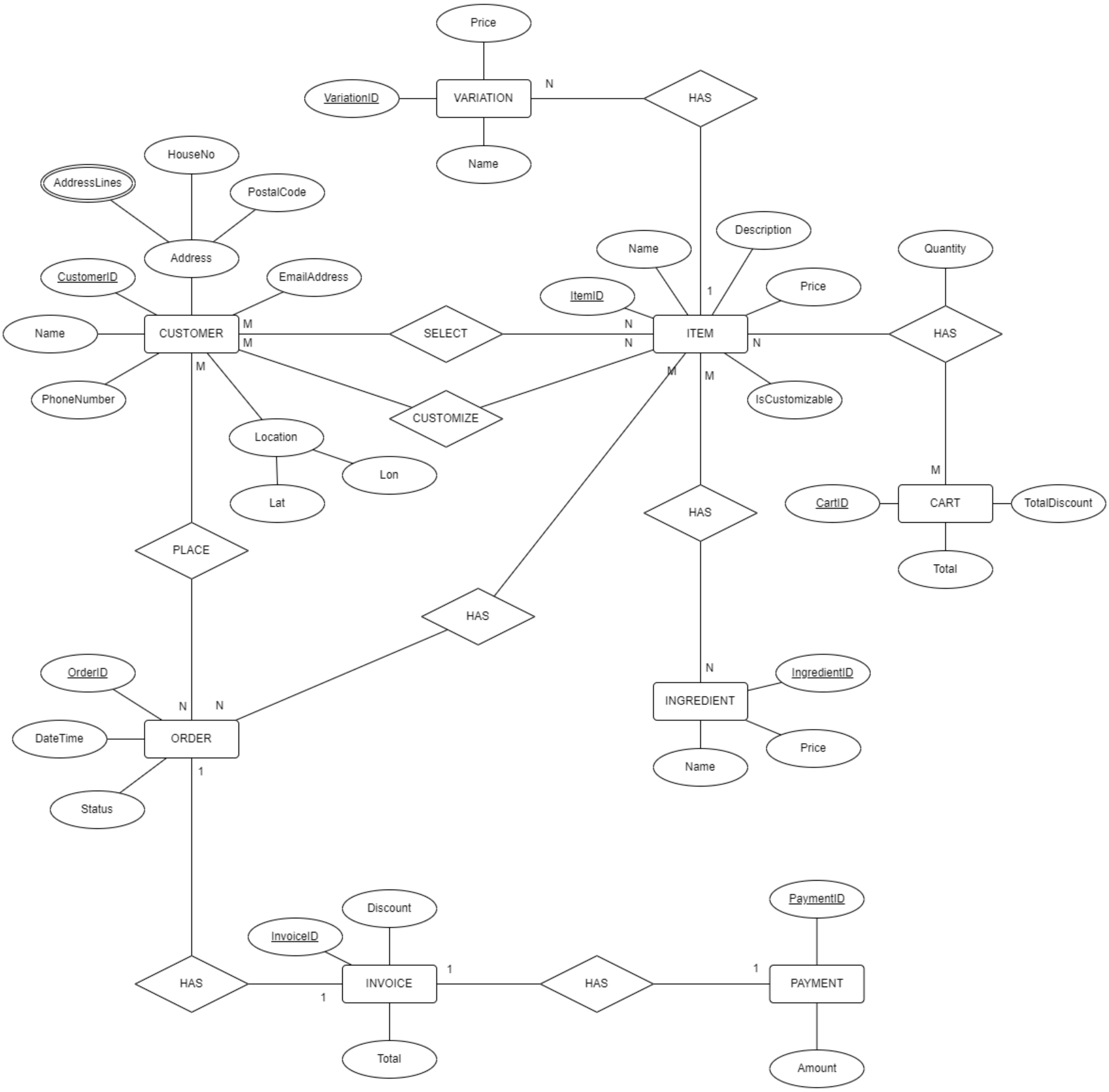

Figure 3.3.1 – Entity Relationship Diagram

3.4. Class Diagram

The following class diagram describes the objects in the system and their attributes, behaviors and relationships between them.

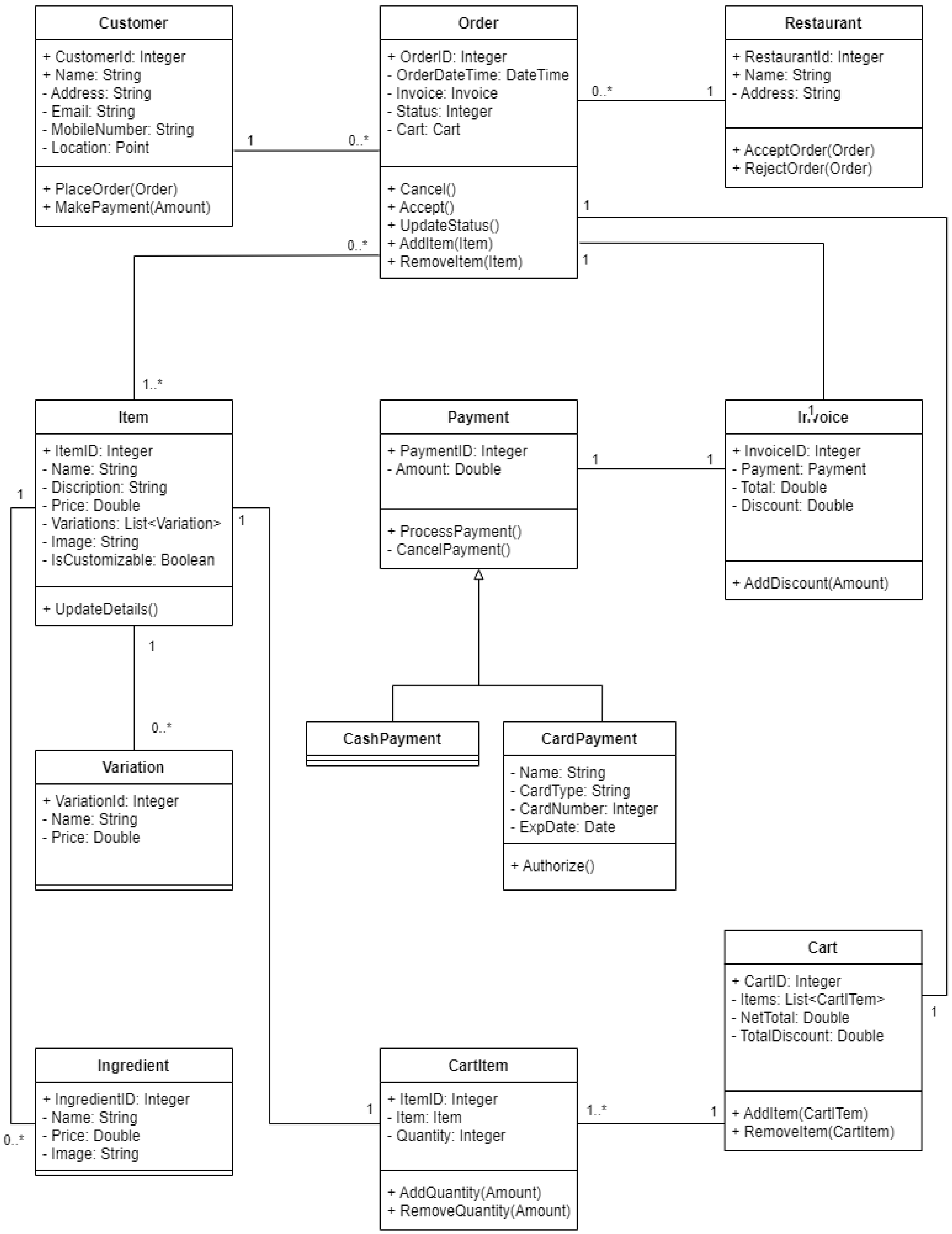

Figure 3.4.1 – Class Diagram

3.5. Activity Diagram

The following Activity diagram describes the main flow of control in the system. It states the series of actions related with basic functionality of the system.

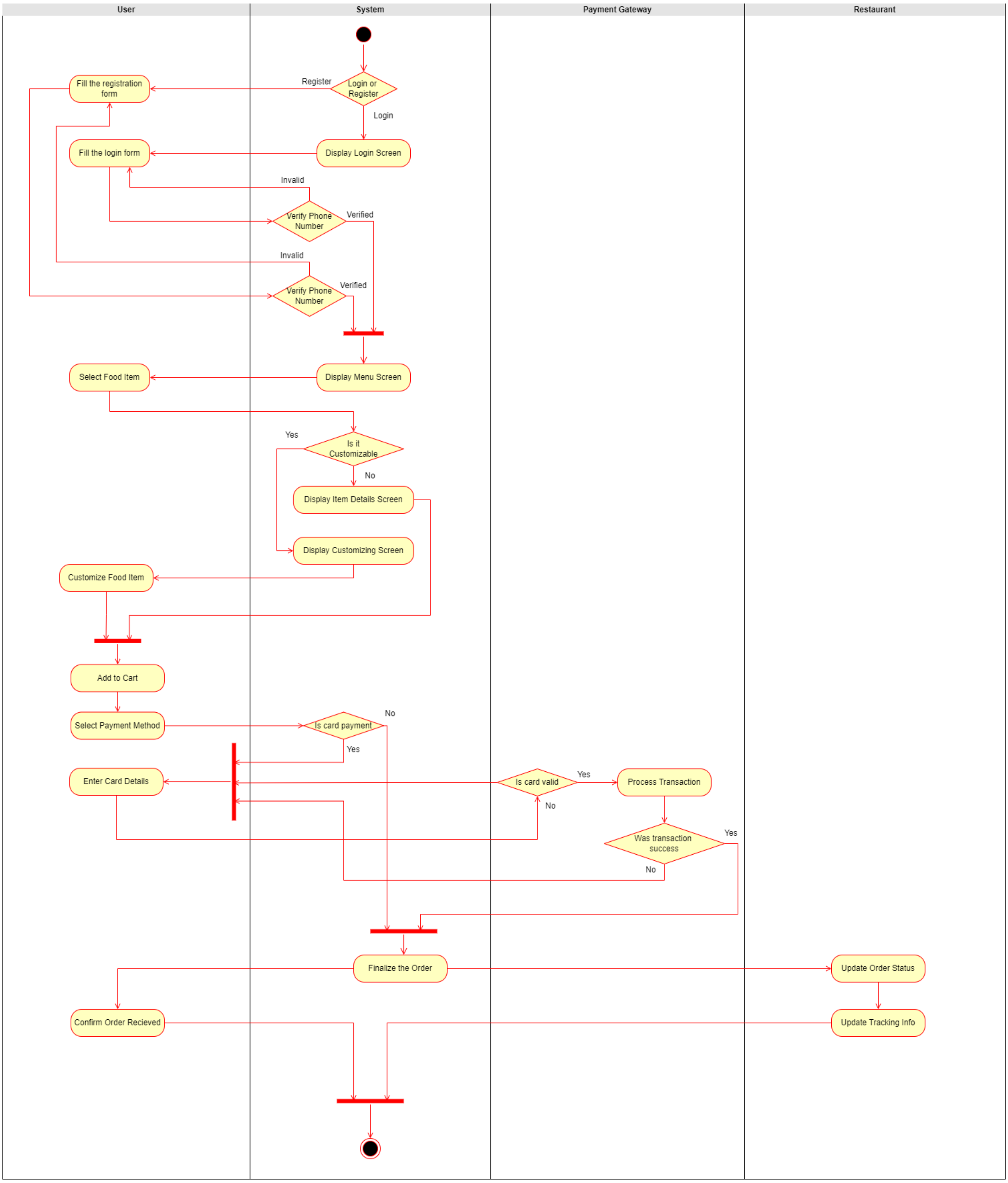

Figure 3.5.1 – Activity Diagram

3.6. Sequence Diagrams

Following sequence diagrams describe particular scenario’s message flows.

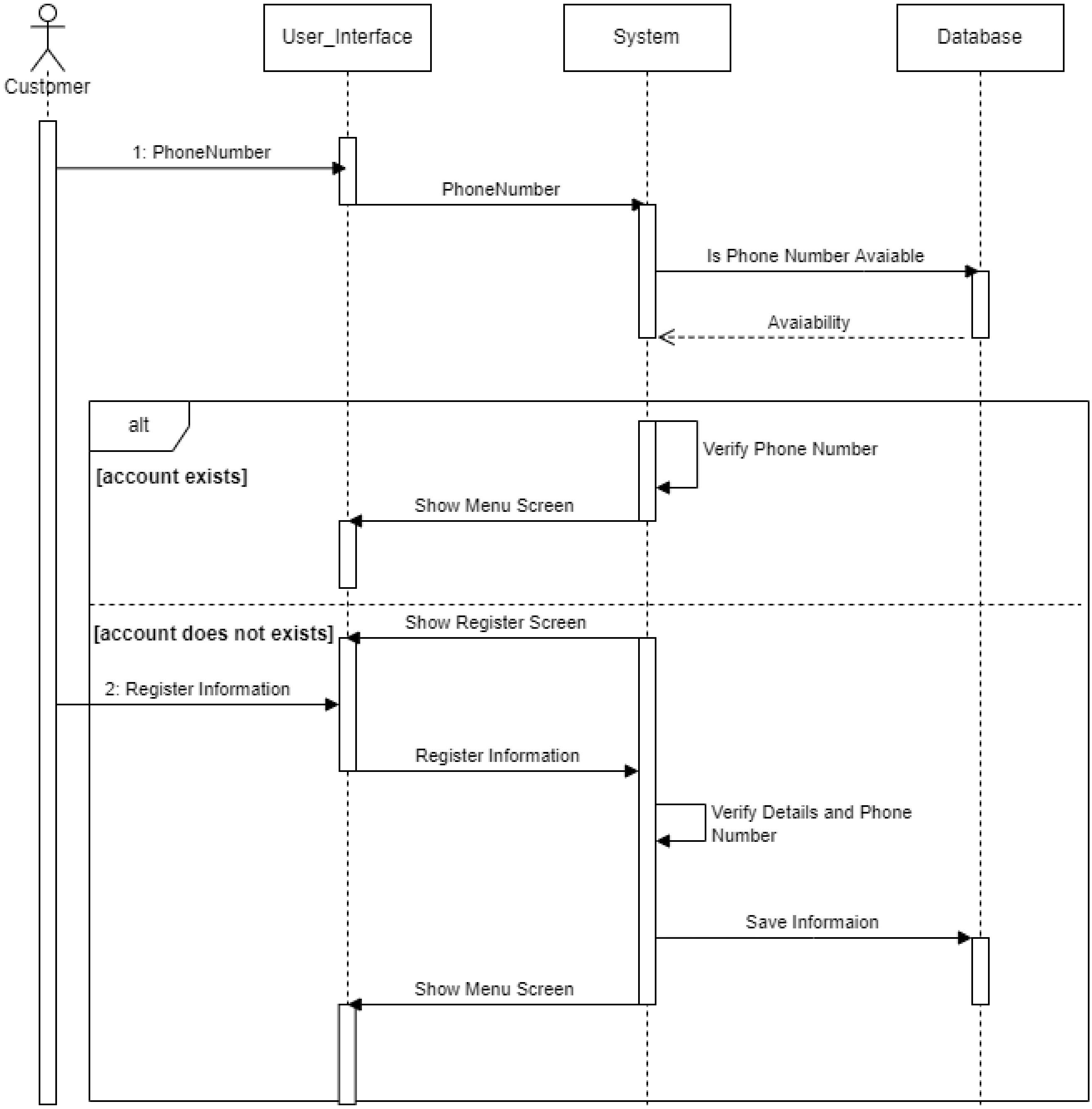

Figure 3.6.1 – Sequence diagram of Logn/Signup sequence

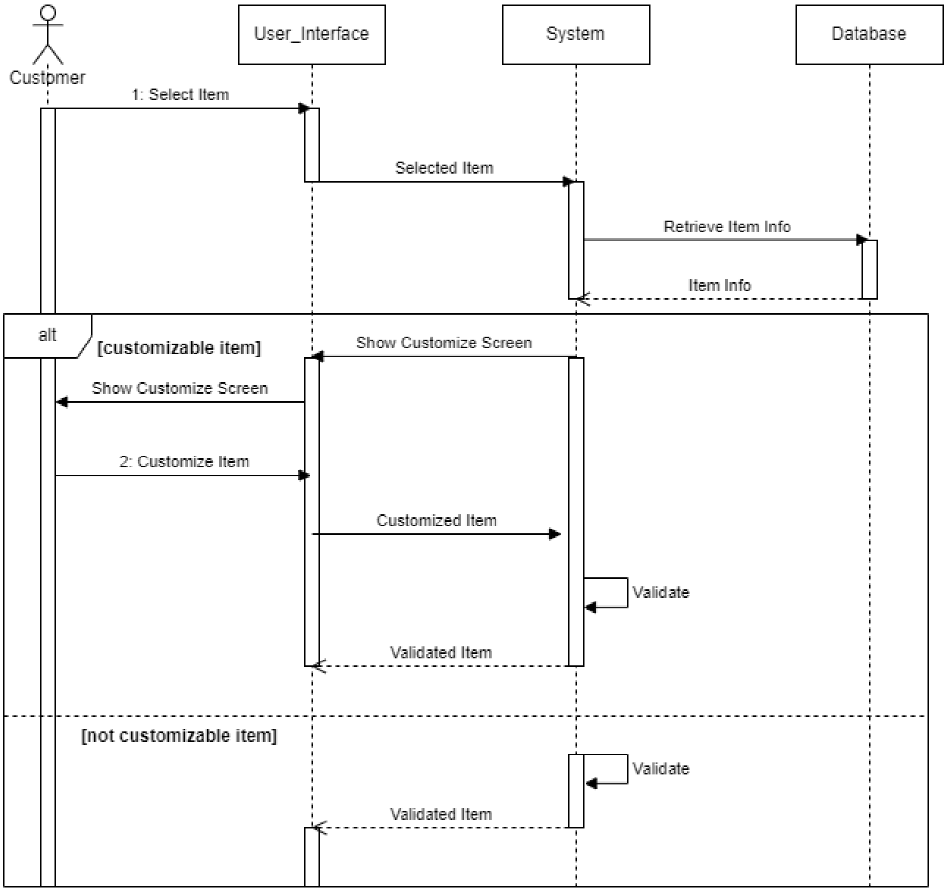

Figure 3.6.2 – Sequence diagram of food item customization sequence

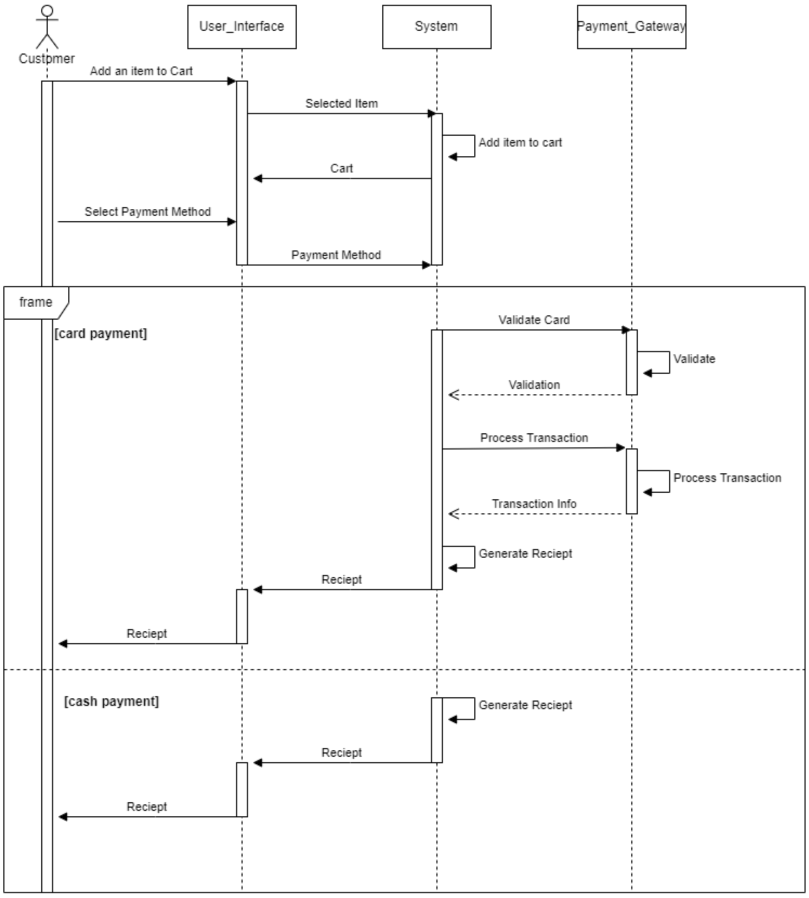

Figure 3.6.3 – Sequence diagram of place order sequence

4. System Development, Testing and Deployment Strategies

4.1. System Development

The proposed system will get developed using one of the main types of software development life cycle (SDLC) called Agile development. The reason behind going with the Agile development methodology is to keep the product's quality and Agile development will allow everybody to collaborate on the project iteratively. As well as with agile development and proper testing would help to minimize the risks.

4.1.1. Agile Development

The systematic sequence of phases that a product goes through as it develops from start to finish is known as the agile software development life cycle. After each milestone, the software goes through a testing phase in order to validate. The basic Agile methodology has five phases known as Analyze user requirements, Design the program, Implement the program, Document and test the system, and maintain the system.

4.1.2. Development Technologies and Tools

This section describes the technology stack which will be using when developing the proposed system.

The mobile application will get developed using Kotlin language. Kotlin is a powerful new language developed by Google which supports huge number of features. User interfaces will get designed using XML markups which Android development using by default.

The database will get developed using Google's Firebase Firestore database. Which is a document-based MongoDB like database. In order to connect application and database, Firebase SDK will be used. Since the database supports real-time callbacks it's possible to update the tracking information of orders in real-time without a delay.

The 3D interface which allows users to customize their food, will get developed using SceneView library which is built on top of Google's Filament library. It's an optimized version for using on mobile devices. Therefore, it'll give seamless experience to the end-user.

In order to keep track of account details and verification purposes, Firebase Authentication will be used. Firebase auth allows to send OTP (One Time Password) codes to phone numbers.

To allow collaborative development and source control we are planning to use GitHub as version controlling system.

4.2. Testing

System testing is an integral part in software development. It helps to identify whether the system meets user requirements and to confirm that it's bug-free. The following testing types will be used to test the proposed system during the development and after the development.

i. **Unit Tests**

This is the lowest level of testing. Unit testing is concerned with testing individual components such as methods, functions, modules, etc. This test will carry out to identify bugs in the code.

ii. **Functional Test**

Functional tests are done to identify whether the application full fills the user requirements. It focuses on the business requirements of the system. This test is only testing the output of a particular action not the way it produced that output.

iii. **Usability Testing**

This will be used to check whether the system is usable by the end-users. The testing will be done by the end-users to check whether it's easy to learn, easy to navigate and it satisfies the user. If any issues are found those will be addressed to fix in future updates.

iv. **Interface Testing**

This test is used to check the connection between system interfaces. It verifies that communication between interfaces is done correctly. Database versions, hardware aspects, and library compatibilities will be addressed.

v. Performance Testing

Since the proposed system is a mobile application, performance will be a major factor determining the product quality. The application must work smoothly with low number of resources. 3D rendering of the application must be done without lagging. Factors such as Responsiveness, Stability and Speed will get addressed.

vi. Security Testing

This test is done to make sure that the system is safe. It makes sure that the software system and application remain secure and unaffected by any risks or threats that could result in harm.

4.3. Deployment

Any restaurant can purchase the system. Once they purchased it, the mobile application with a custom name and icon will get published on Google Play so customers can download it into their mobile phones. Normal goods can be added by the interface provided to the restaurant. In order to add customizable items, the restaurant has to send the details to the development team through the app, because adding customizable goods are concerned with creating a 3D representation of the item and its ingredients which might be out of the client's capability.

5. Project Milestones and Timeline

The project is planned to finish within a span of 10 weeks. Since the development is using Agile development methodology, Testing and Validating will happen iteratively along with the development. Front-end and Back-end development will take a maximumly of up to 5 weeks. The rest of the activities are spread over 5 weeks.

Activity	November				December				January			
	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
System Requirement Specification		🟦										
System Design		🟩	🟩									
Front-end Development				🟧	🟧	🟧	🟧	🟧				
Back-end Development				🟧	🟧	🟧	🟧	🟧				
Testing					🟨	🟨	🟨	🟨				
System Integration								🟩	🟩			
System Testing										🟨		
Documenting										▨	▨	
Project Report											🟦	

Figure 5.1 – Project milestones and Timeline