



# Hot Spot Pizza Mobile App

## Team :

Akanksha Avhad

[aavhad@uncc.edu](mailto:aavhad@uncc.edu)

Asasuddin Syed

[asyed8@uncc.edu](mailto:asyed8@uncc.edu)

Bhushan Deshpande

[bdeshpan@uncc.edu](mailto:bdeshpan@uncc.edu)

Jyoti Thakral

[jthakral@uncc.edu](mailto:jthakral@uncc.edu)

Kranthi Chinnakotla

[kchinnak@uncc.edu](mailto:kchinnak@uncc.edu)

# Contents

## [Introduction](#)

[Project Overview and Statement of Proposal](#)

[Project Scope and Objectives](#)

## [Non-Functional Requirements](#)

[2.1 Security Requirements:](#)

[2.2 Product Requirements:](#)

[2.4 External Requirements:](#)

## [Functional Requirements](#)

[E-R Diagram:](#)

[Class diagrams:](#)

[Architecture view of use-case model:](#)

## [Architectural Design](#)

[Subsystem Architecture](#)

[Deployment Model](#)

## [Use Case Realization Design](#)

[Use Case 1: Register](#)

[Use Case 2: Login](#)

[Use Case 3: Client Side QR Pay](#)

[Use Case 4: Merchant Side QR Pay](#)

[Use Case 5: Add to cart](#)

[Use Case 6: Place Order](#)

[Use Case 7: Logout](#)

[Database System:](#)

[Client and Merchant side pages](#)

## 1. Introduction

As the saying goes ‘You can't buy happiness but you can buy pizza and that's kind of same thing’. When it comes to the world of culinary achievements, pizza takes the cake or, in this case, the pie, every time. Pizza is probably one of the only foods that holds a special place in pretty much everyone’s hungry hearts. A mobile app for pizza store is always helpful to the user as it enables the user to browse various kinds of pizzas and order it online. This app extends an additional and attractive feature of QR pay which simplifies payments. QR pay increases profitability and helps maintaining cash flow by allowing their customers to pay instantly with payments cards. It allows customers to view and use the complete menu and content for the purpose of placing an order directly through the app. The purpose of this app is to continually strive to help customers order their food as quickly and seamlessly as possible. With the introduction of Hot spot Pizza app, customers will be able to place their favourite orders literally at the touch of a button and this smartphone app also enables the customers to make payment by scanning a QR code which means no need to carry your debit and credit cards all the time. In few years, ash will be made obsolete and everyone will pay for their meals via smartphone.

### 1.1 Project Overview and Statement of Proposal

Hot Spot pizza app enables diners to pay their bill by scanning a QR code that's either displayed on screen or printed on the bill. To do so, customer’s debit or credit card numbers are encrypted and stored safely on customer’s phone. It doesn't store card details on server somewhere where they can be hacked

User can register an account and login to the system to access the entire menu. He can browse the flavor menu available on the hot spot pizza app and build a cart with the selected dishes. Users can modify the cart by adding or removing meal before making the payment. The order can be reviewed before it is actually placed. Once the order is placed, receipt number is generated to track the order.

The primarily functionalities of the application are

- **User account and profile maintenance:** A new user can set up an account and then login to the system. Registered users can use the existing login information to access the system. Registered user also has access to modify their profile information.

- **Order Pizza using mobile app:** Customer can order pizza using this android platform supported app . He can select the pizza of his choice, customize it with different crust, flavor and toppings , add it to the cart and proceed for checkout. He has an option to modify or delete the cart, if needed.
- **Pay by scanning the QR code:** Customer generates QR Code, merchant scans it with QR code scanner and confirm the amount to be paid. Customer's payment card details are stored on his phone using secure encryption. Paying for pizza is Quick, Simple, Secure with this app.

**Statement of Proposal:** We propose to make a mobile-based application that will enable customers to order pizza online and pay via QR code scanning.

## 1.2 Project Scope and Objectives

The project will focus on the development of the hot spot pizza application with QR pay as an extended feature, also the app will be available publicly for downloads. The underlying architecture and database system will be built in. All features from browse to QR pay will be implemented and tested by the end of the project end date. The following are the objectives for the project:

1. Create a visually pleasing user interface.
2. New customers need to register first to get access to the system. After registration, the customer will be able to login to his account using his email address and password specified during registration. Each customer can only register one account and each account must belong to exactly one customer.
3. Customers can customise the pizza and can select from choice of crust, size and toppings.
4. Customers can choose to dine-in or carryout or delivery option. manage their shopping cart.
5. Each customer can place any number of orders. Customer has an option of paying through QR code generated for the total order amount. Customer can use debit or credit card to pay via QR scan.
6. Merchant shall be able to scan the QR code generated for payment. Upon scanning, the order amount shall display on merchant's screen and merchant shall be able to charge the customer for the amount shown in QR code.

## **2. Non-Functional Requirements**

### **2.1 Security Requirements:**

- For security purposes, every customer owns a separate account, only to be viewed by him and the administrator, to carry out customer transaction.
- All registered customers will have personal information and this will not be visible to other customers.
- The registered customers will also not be able to view the transactions of other customers.

### **2.2 Product Requirements:**

- A user friendly mobile application is designed with all the required functionalities for registered users.
- A survey is carried out for the customers, wherein a successful application should reach a minimum of 50% user satisfaction.
- A customer account is maintained for each registered user having all the information of the previous transactions and this can be retrieved any time by logging into the account.
- The application will provide fast paced data access to the customers with ease of loading. It is designed in a way that the interface should be loaded in less than 3 minutes and data queries should not take more than 2 minutes.

### **2.3 Organizational Requirements:**

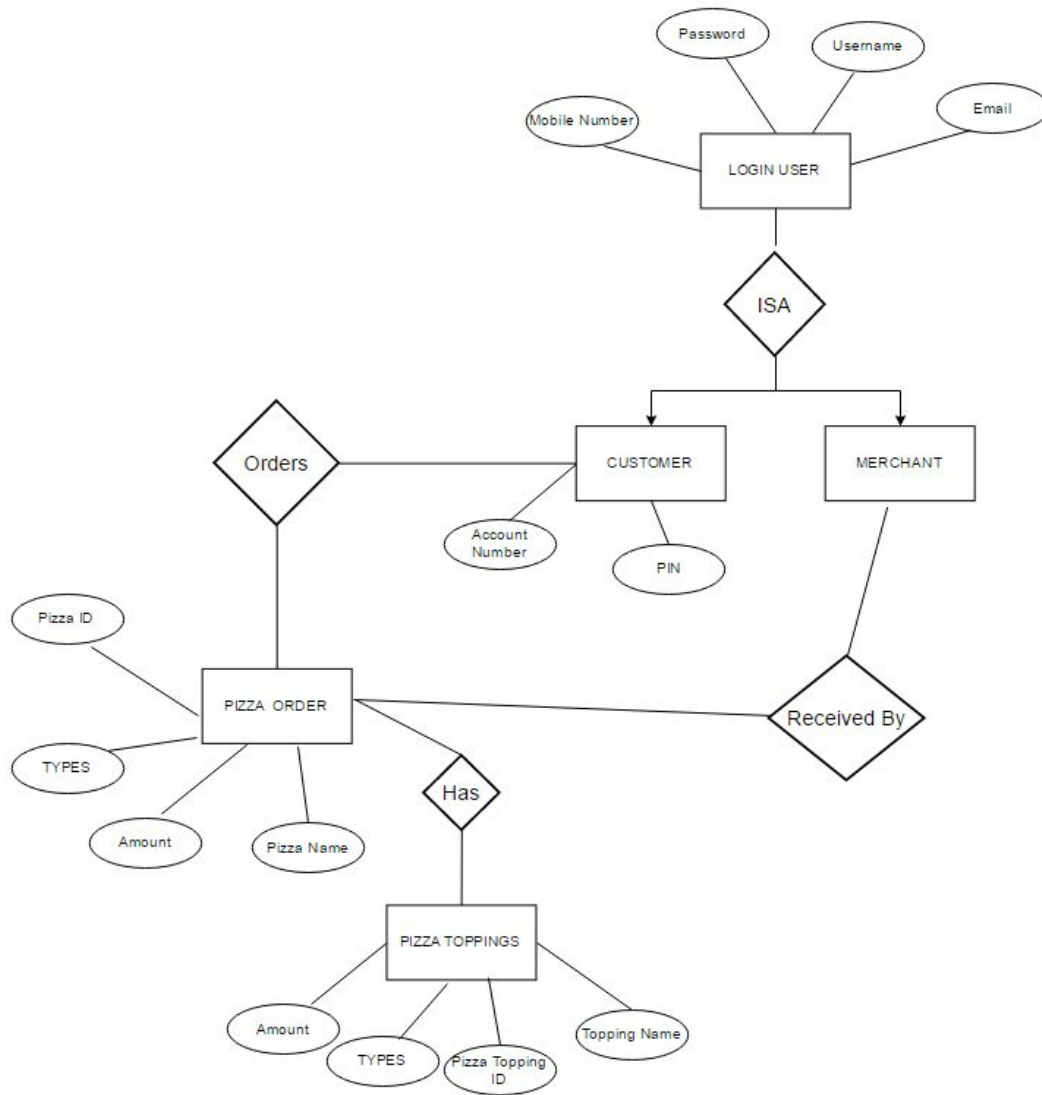
- In order to regulate traffic, the designed application will provide the functionality of serving all the customers through First Come First Serve policy.
- The application must run on the browser with all its functionalities efficiently through optimal methods.
- The displayed data should be clear and precise for application users.
- The MySQL database is to be used in the application.
- Application needs to have compatibility with Android platform. Future sprints will provide the hybrid functionality which will enable the same app to be installed on Iphone.

### **2.4 External Requirements:**

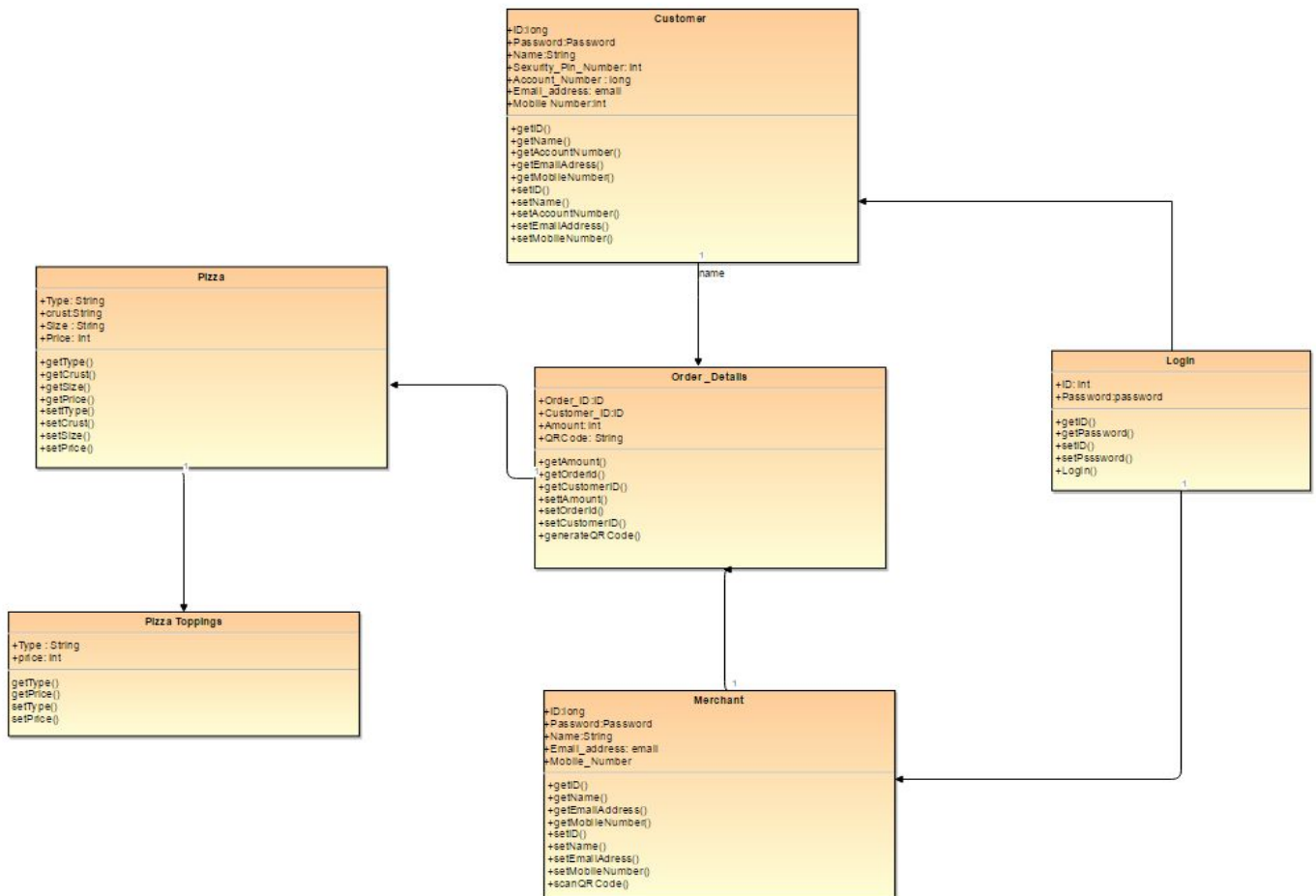
- The team should ensure that milestones set during the project plan are met timely during the project development of the project.
- Stakeholders need to be involved periodically i.e. weekly and monthly survey of their satisfaction needs to be captured and it should always be more than 50%.

### 3. Functional Requirements

#### 3.1 E-R Diagram:



### 3.2 Class diagrams:



### 4. Architecture view of use-case model:

#### Use-case: I

1. Name: Sign Up
2. Actor: New Customer
3. Goal : New customer to the pizza app should be able to order pizza with mandating him to Sign up for the Pizza store.
4. Narration: A new should always be able to order pizza without a need for sign up. Sign up is completely optional.

5. Pre-post conditions:

1. Pre-conditions: There is no requirement that customer should have signed up before.
2. Postconditions: Customer orders a pizza without signing up at the pizza store.

6. Primary and Alternative flow of Events: Primary flow is customer orders a pizza, the alternative flow would be, sign up the customer at pizza store and then customer orders a pizza.

#### **Use-case: II**

1. Name: Log In

2. Actors: New Customer /Returning customer

3. Narration: Providing benefits for the customers who had signed up or registered at the pizza store through the app, the benefits like discount coupons and free delivery of pizza.

4. Pre-Post Conditions:

1. Pre-Conditions: There is no requirement that customer should have signed up before.
2. Postconditions: Customer is provided with benefits after signing up at the pizza store, benefits like discount coupons and free delivery of pizza.

3. Primary and Alternative Flow of Events:

1. Primary: The customer signs up at the pizza store and then login using his set credentials for sign up.
2. Alternative: A returning customer no need to sign up, he can directly login through the login page of pizza app.

#### **Use-case: III**

1. Name: Customize Pizza

2. Actors: Customer (New and returning)

3. Narration: The customer logs in to the pizza app and customizes his own pizza, with various toppings and the type of bread he prefers.

4. Pre-Post Conditions:

- a. Pre-Conditions: The customer has logged in to the pizza app, either as guest or registered customer.
- b. Postconditions: Customer is able to add all the required toppings and select the bread type he prefers.

5. Primary and Alternative Flow of Events:

- a. Primary: Customer logs in and customizes the pizza to order.
- b. Alternative: No alternative flow.

#### **Use-case: IV**

1. Name: Customize Pizza

2. Actors: New Customer/Return customer



3. Narration: Customer is not allowed more than a certain predefined number of toppings.
4. Pre-Post Conditions:
  - a. Pre-Conditions: The customer should be able to customize the pizza.
  - b. Postconditions: Customer cannot add more than a predefined number of toppings, on his pizza.
5. Primary and Alternative Flow of Events:
  - a. Primary: Customer logs in and customizes the pizza to order.
  - b. Alternative: No alternative flow.

#### **Use-case: V**

1. Name: Delivery
2. Actors: New/Returning Customer
3. Narration: Delivery option is provided on extra charge for the customers. If the customer selects the delivery option then pizza would be delivered at their door, else the customer has to walk into the pizza store and collect with no additional delivery charges.
4. Pre-Post Conditions:
  1. Pre-Conditions: Customer must have completed customizing the pizza.
  2. Postconditions: Customer selects delivery option , pizza is delivered to their doorstep. Customer has not selected the delivery option , customer picks up the pizza from the store.
5. Primary and Alternative Flow of Events:
  1. Primary: Customizes the pizza and decides on how to get the delivery.
  2. Alternative: No alternative flow.

#### **Use-case: VI**

1. Name: Place Order
2. Actor: New/Returning Customer
3. Goal: Customer places the order for pizza from the app, the price for the pizza calculated based on the number of toppings, base of pizza and the type of delivery. The total amount is displayed on the screen, before customer places the order.
4. Pre-post conditions:

Pre-Conditions:  
Customer must be logged in, and should have been customized the pizza and made a choice of delivery.

Post-Conditions:  
The total amount of the order is displayed on the customer screen.
6. Primary and alternate flow of event:

Customizes pizzas, made a choice of delivery and the total final price shown on the customer screen.

### **Use-case: VII**

1. Name: Payment
2. Actor: New/Returning Customer
3. Goal: The payment could be done through the QR code generation.
4. Narration: The total amount on the customer order can be paid through various kinds of payment options. The customer should be able to pay through the QR code generated for the total amount displayed on the screen.
5. Pre-Post Conditions:  
Pre-Conditions: The customer finalizes the order.  
Post-Condition: A QR code is generated for the total amount displayed on the customer screen.
6. Primary and Alternative Flow of events: Primary flow is the customer places the order and taken to the payment option through QR code generation.

### **Use-case: VIII**

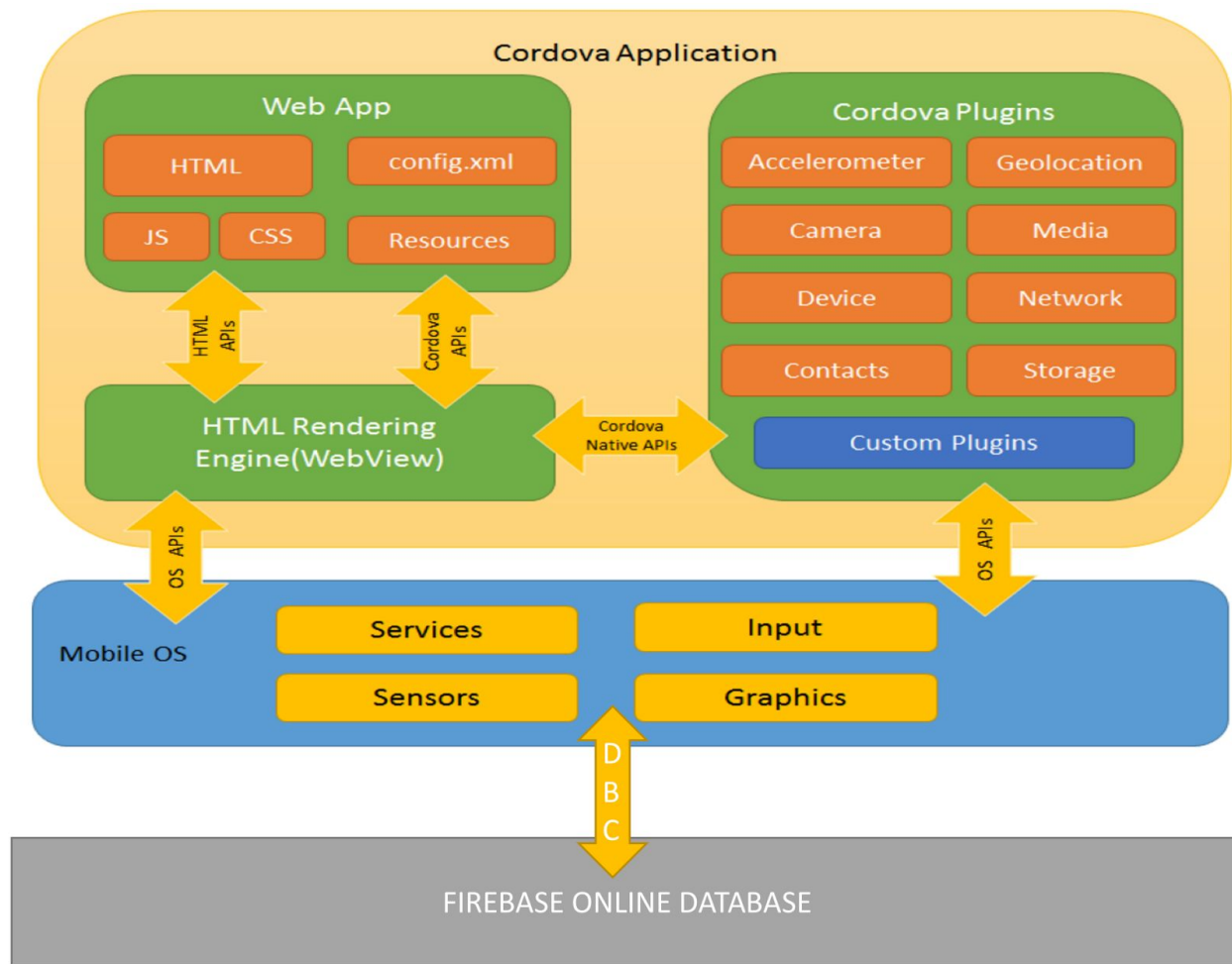
1. Name: Payment
2. Actor: Merchant
3. Goal: Decode the QR code generated by the customer for their order and get the payment through credit/debit cards or cash.
4. Narration: Merchant should be able to use the pizza store app and scan the QR code, the amount should show up on the merchant screen. The amount can be taken from the customer's Credit/Debit cards or cash.
5. Pre-Post Conditions:  
Pre-Conditions: The customer pays for the order.  
Post-Condition: Merchant should be able to scan the QR code and read the amount. Merchant should be able to charge the customer for the amount shown in the QR code.
6. Primary and Alternative Flow of events: Primary flow is the customer pays for the order and Merchant scans the QR code generated for the payment.

## 5. Architectural Design

The following architectural design showcases how our application will communicate. We decided to create a mobile application using Cordova, JavaScript, HTML5 and CSS. View is developed using HTML5 , CSS and JavaScript , that handle the application design and data input. The Javascript will then be able to communicate with the domain objects that are processed by service layer (business logic) or DAO layer (database operation). Once the data is determined, the FireBase online database will be contacted to retrieve information and/or update records.

We have used Framework 7 for a base design in our application, then used cordova (a phonegap application) to build the mobile application from a HTML5,CSS and JavaScript. Cordova could be run in node.js environment. We can add both iOS and Android platforms using cordova for our application.

Below is the architectural diagram that we have put together to best describe our application process.



## **5.1 Subsystem Architecture**

A customer opens a pizza application and tries to access the application for ordering pizza. The application page created using HTML5, CSS and JavaScript opens up and let's the customer enter the input. The input is sent to the next activity of the application (service layer), which will allow the user to customize the pizza and select a delivery option. All the information is passed on to the service layer of the application where the customer total bill and the payment options are listed. During all these activities we are constantly connected to an online Firebase database to save all the customer preferences and customer profile data.

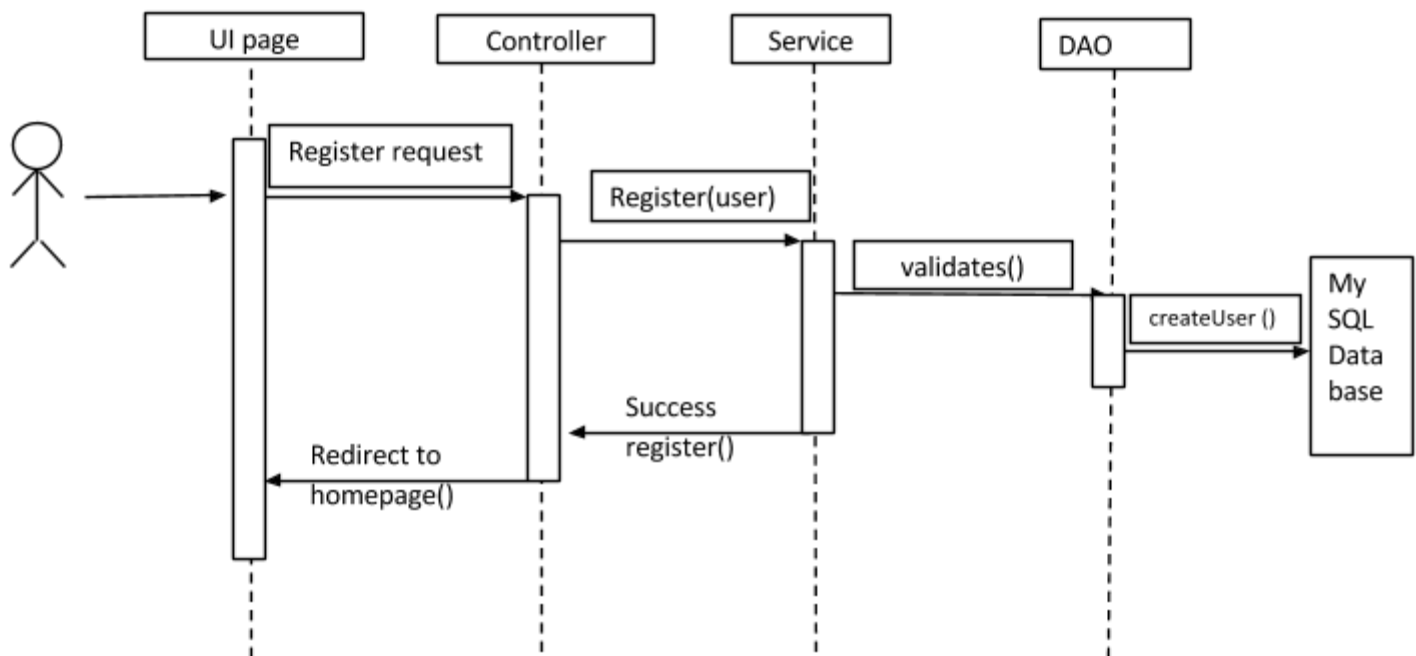
## **5.2 Deployment Model**

The following discussion describes the mapping of subsystems to deployment. The application UI would be the front facing application used in an iOS or Android platforms. The UI looks similar to a new customer and to a returning customer who has access to the application. The merchant has different UI for accepting the payment from the customers. All controllers and handlers will be deployed to the Servlet (Business Logic) portion of the deployment model. These handlers and controllers contain all the business computation of our mobile application. They will handle all business computation given to them. Mobile device (Android or iOS, Customer, Mobile Inventory, Customer Inventory, Payment Info, Schedule, and Hardware entities will be in an online Firebase database. These entities will hold all information pertaining to their respective responsibilities.

## 6. Use Case Realization Design

### 6.1 Use Case 1: Register

#### Sequence Diagram:

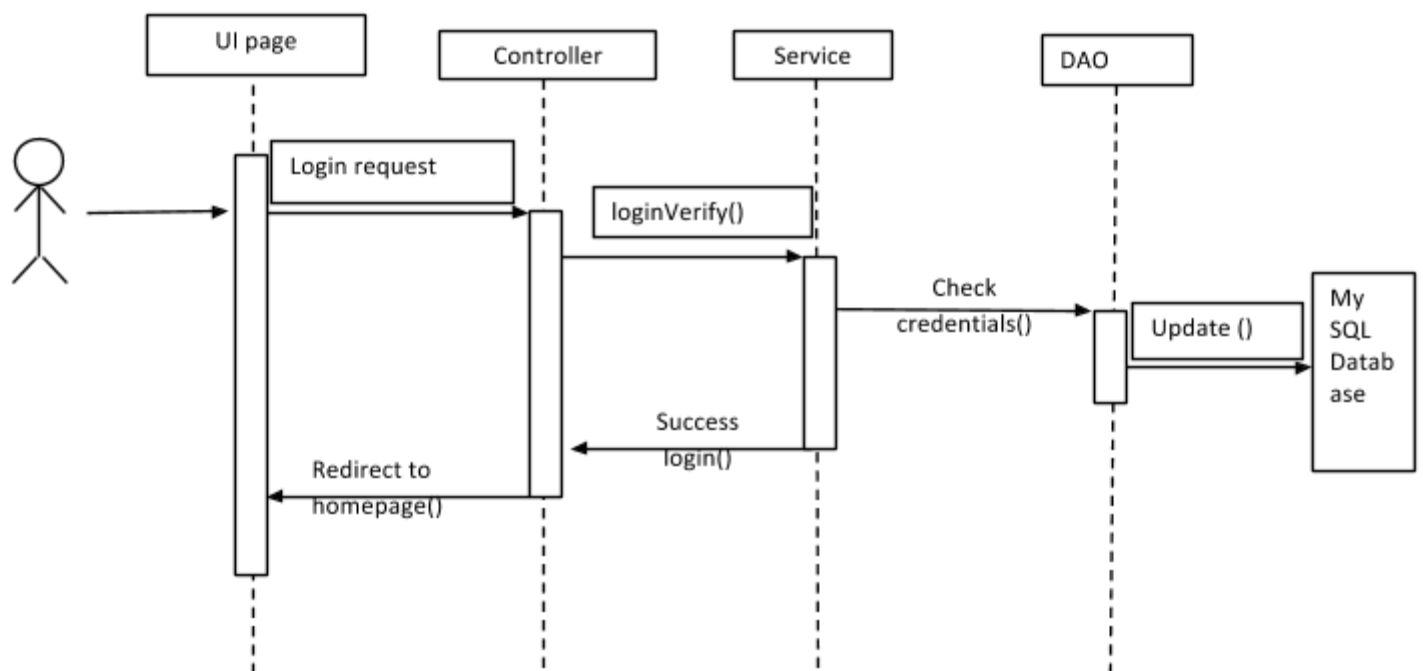


**Narration:** Actor in this is unregistered customer or unregistered merchant. When the user attempts to access the site he will be prompted to login to the site. As he is an unregistered user, he will submit a registration request. He will be redirected to the registration UI page. He then enters his details which will be passed to the service layer which are validated and then inserted to the database through the DAO layer. For very customer, registration form will save the user name, first name, last name, Email, Account no., Pin, Mobile no. Once the details are updated he will be redirected to the main page and his details are stored in session to be available throughout the application. If any details are incorrectly entered he will remain in the register page until he submits are values appropriately. Also, if a user tries to create a user name which is already taken by some other user, system displays an error message 'Invalid login' on screen and indicates user to choose another username.

## 6.2 Use Case 2: Login

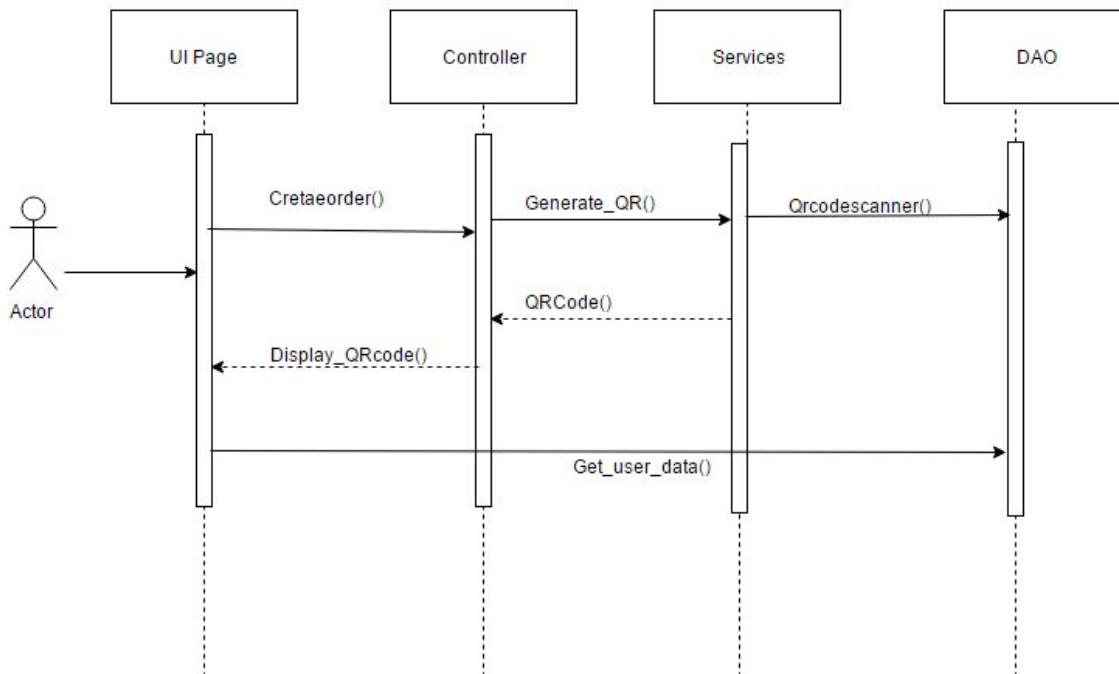
### Sequence diagram:

**Narration:** If a customer who has already registered to site comes to access the site. He need not register again. He can use his user name and the password he chose during the time of register and can login to the site. Similar is the case with merchant. While the user prompts to login to the site, his request is passed to the controller where his credentials are validated along with the details in the database which are brought to the application through the DAO layer and verified in the service layer. If the credentials are correct the user is redirect to the home page of the site and his details are stored in session to be available throughout the application, else he is prompted to login again.



### 6.3 Use Case 3: Client Side QR Pay

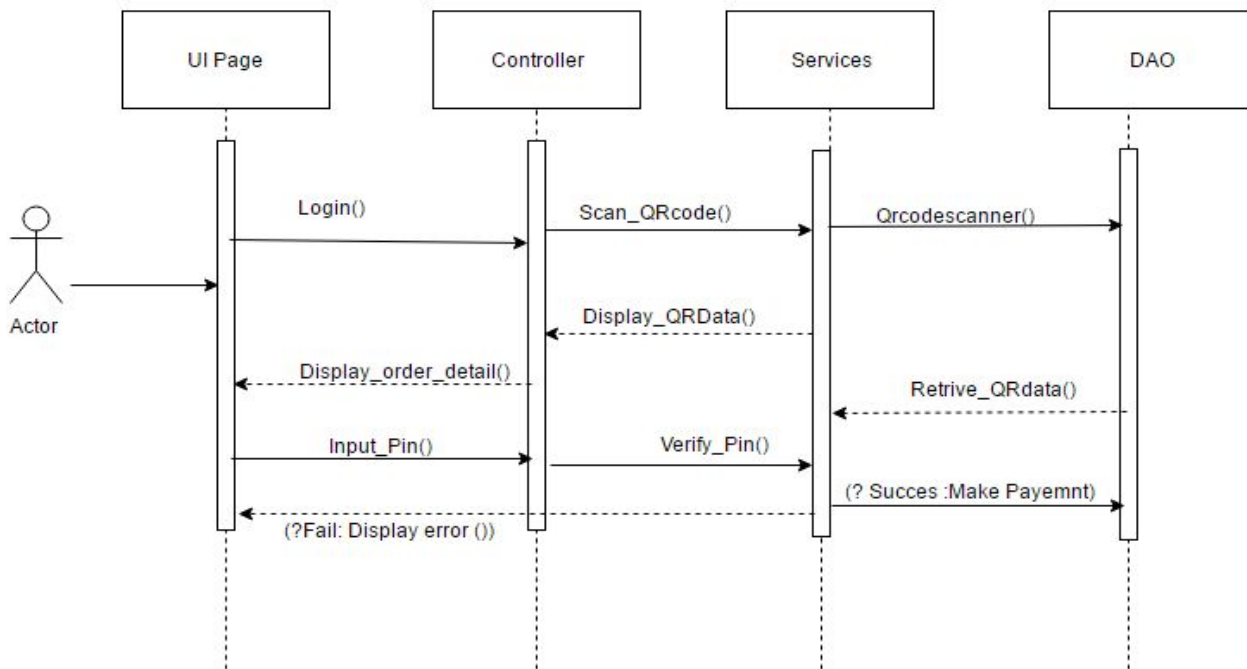
Sequence diagram:



**Narration:** Once the customer is successfully logged in or registered to the site he will be able to access the site and place an order. He can browse through the app to choose from a wide variety of pizza selection such as pan pizza, hand tossed pizza, thin 'n' crispy and so on. Once the type of crust is selecte, he can select the size from small, medium and large. Also, Customize it with different selection of topping provided such as Yellow onion, Sun dried tomato, Spinach etc. When the user creates the order,his order details are read by the controller and corresponding OR code is generated which passed on to the service layer where it interacts with the DAO layer. QR code contains order id, customer id and price for the order. All user details are stored in Sql database via DAO layer.

## 6.4 Use Case 4: Merchant Side QR Pay

### Sequence diagram:

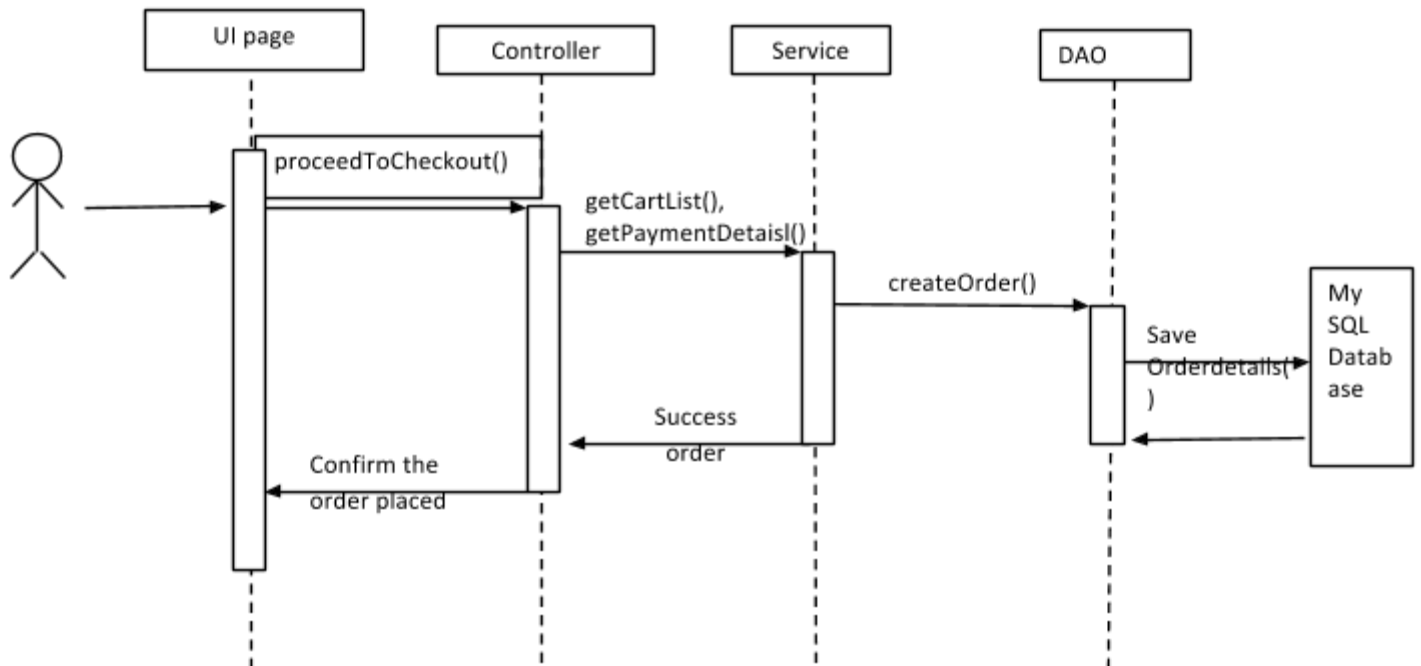


**Narration:** When the merchant logs in, with his account information, he gets an option to verify QR code. As soon as he presses verify code, qrcode scanner is automatically opened by app and QR code is scanned. Once the QR code is successfully scanned, it displays the value hybridized in QR code which are order id, user id and amount. On UI page, an option to enter the pin is also displayed. This is to deduct the amount displayed on screen from the account info. Provided earlier during registration. Customer is prompted to input the pin on merchant's device. This pin is a user defined password which is entered during registration. Letting the user enter the pin adds an extra protection layer from security perspective. If the pin entered, matches with the pin stored in database, corresponding to that user, request is passed to the controller which interacts with the service layer which places a request to the DAO layer to make the payment. This request is further fulfilled by a third party which is considered out of scope for this project. Once the confirmation is received from third party reg. Payment success is acknowledged to customer and merchant by displaying a success message on screen. If transaction was not successful, an error popup appears with invalid pin on screen.



## 6.5 Use Case 6: Place Order

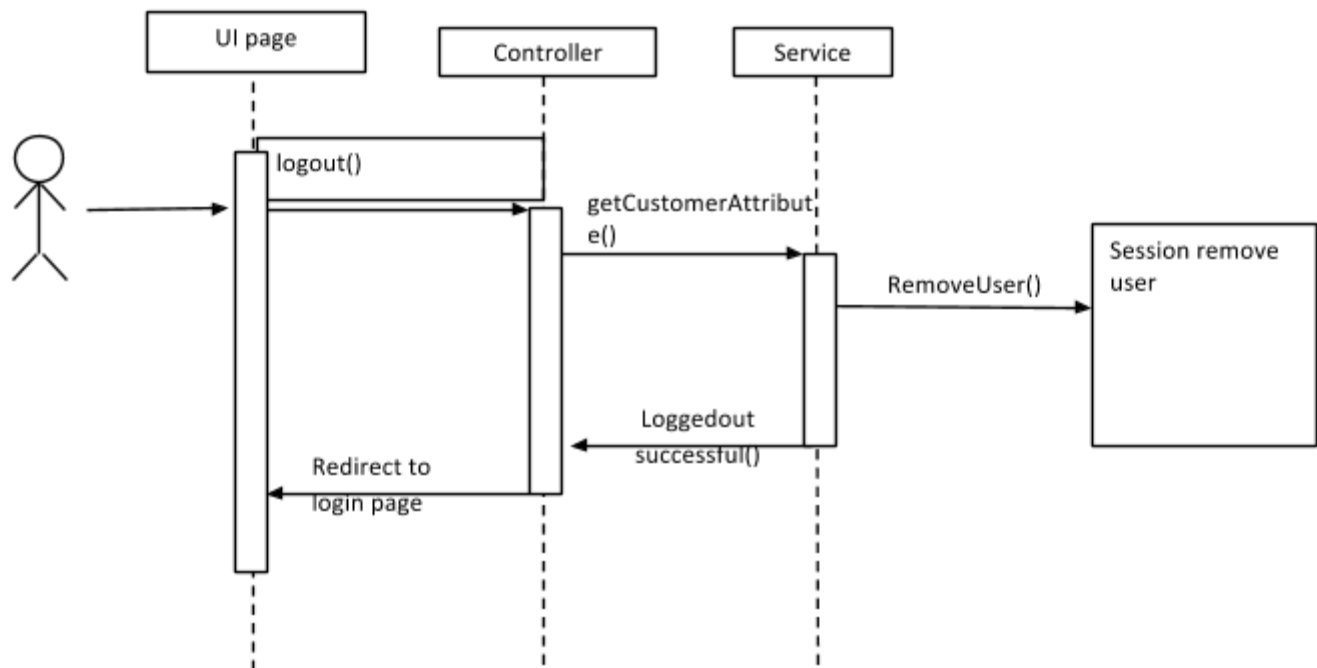
Sequence diagram:



**Narration:** Once the user is logged in the android application. He will browse the menu available. If he wishes to order a pizza, he selects it and places a request to add to the cart. When he prompts to add to the cart, the details of the order are passed to the controller which interacts with the service layer and stores the selected order in the cart object which is stored in the session. And the user will be redirected to the cart page where he will be shown all the items he has added to the cart. He then places a request to proceed to checkout where the user generates a QR code for the entire order. Then Merchant logs in and scans the QR code for payment purpose that's when the user details are verified in the service layer and the order details along with the card details are entered into the database using the DAO layer. Once the payment is done, a popup on merchant's screen will confirm the order as success and the transaction completes.

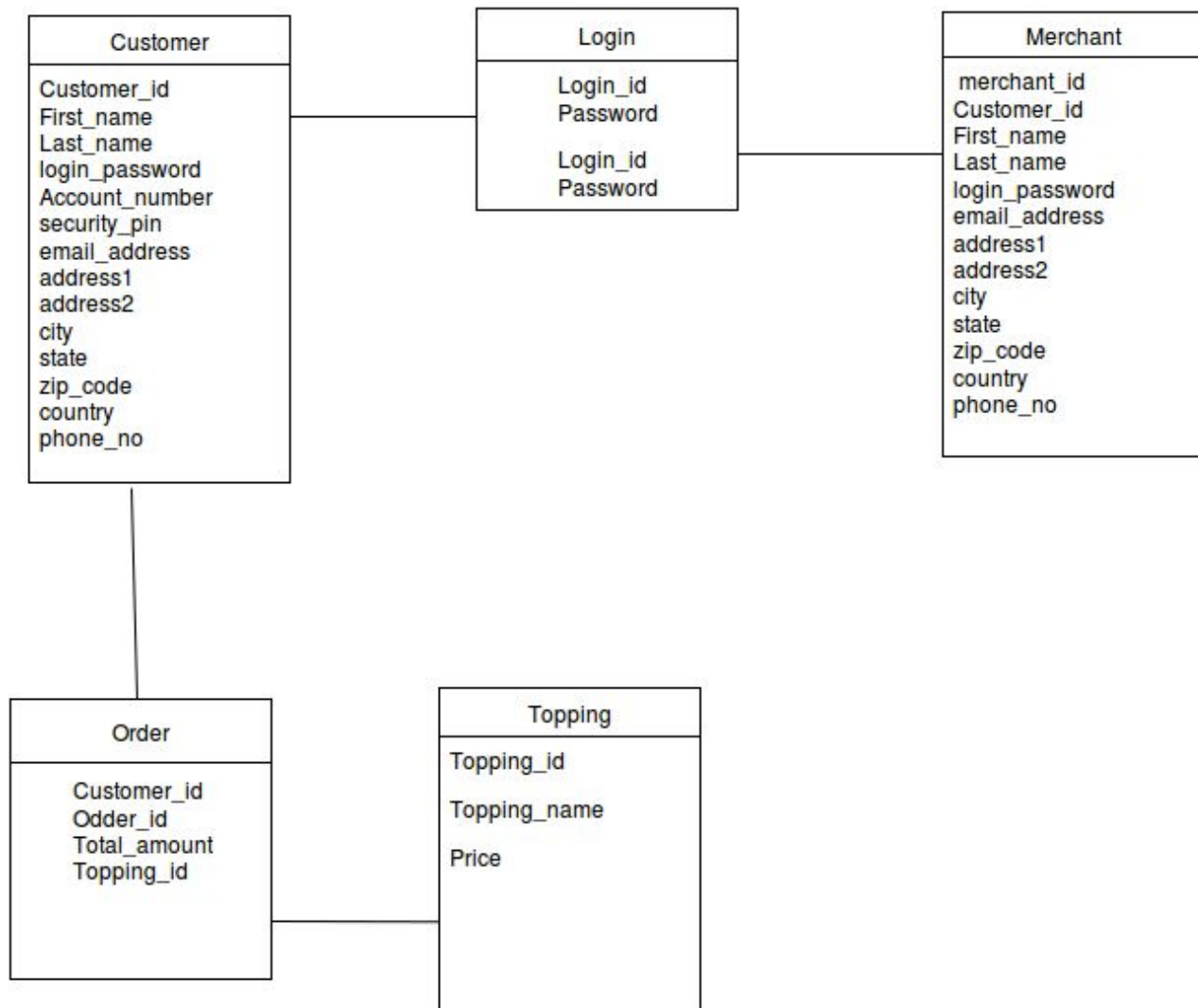
## 6.6 Use Case 7: Logout

Sequence diagram:



**Narration:** At any point of time if the user wishes to logout of the site, he can place of request for it. This request will be handled by the controller which then removes the user info from the session and redirects the user to the login page.

## 7. Database System:



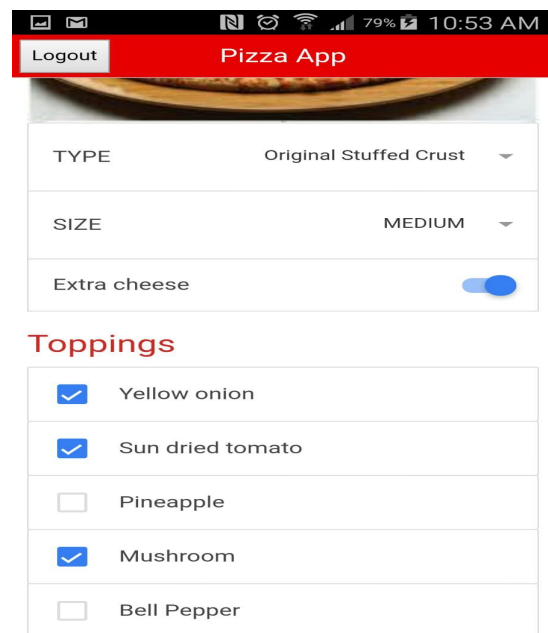
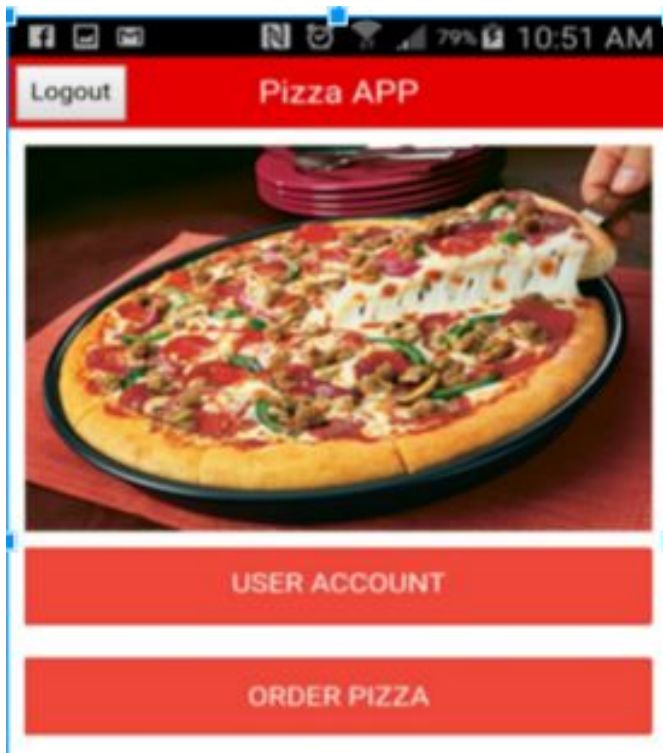
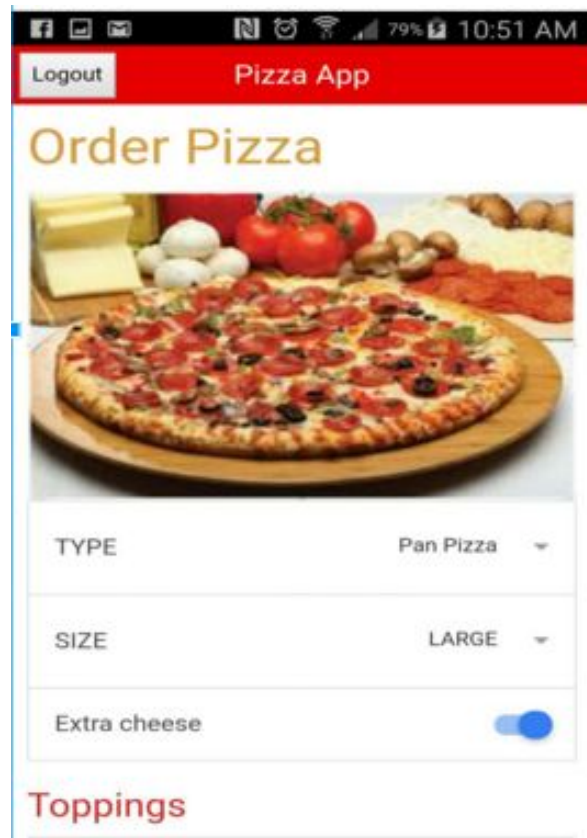
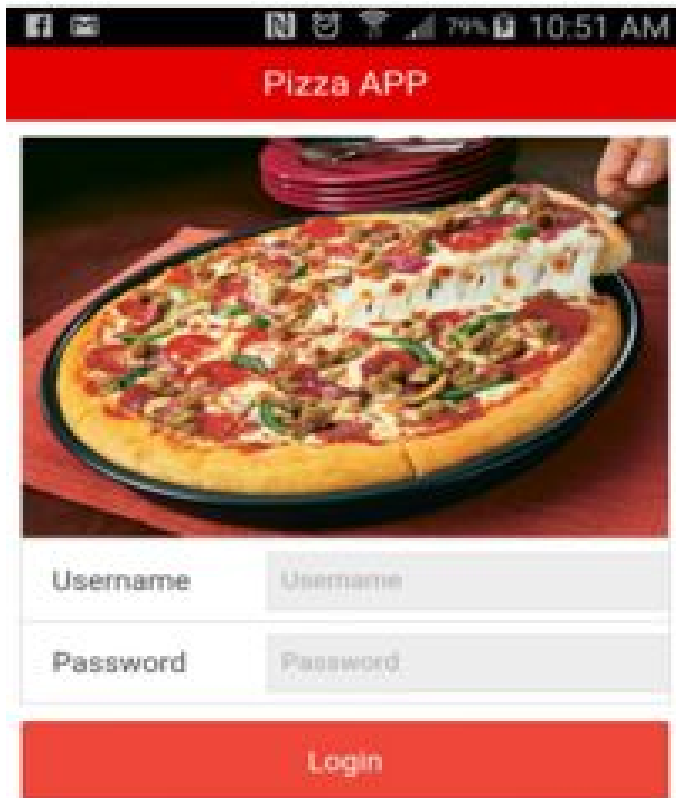
**Customer:** A customer table stores the information of the user entered during registration to the site. The details include the customer id, email address, password, first name, last name, address, contact number, account number, and security pin chosen.

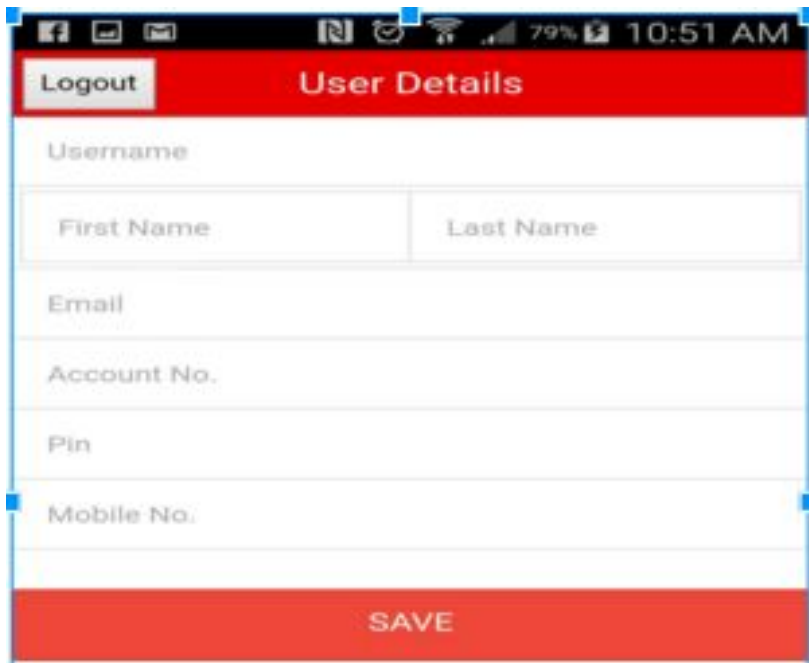
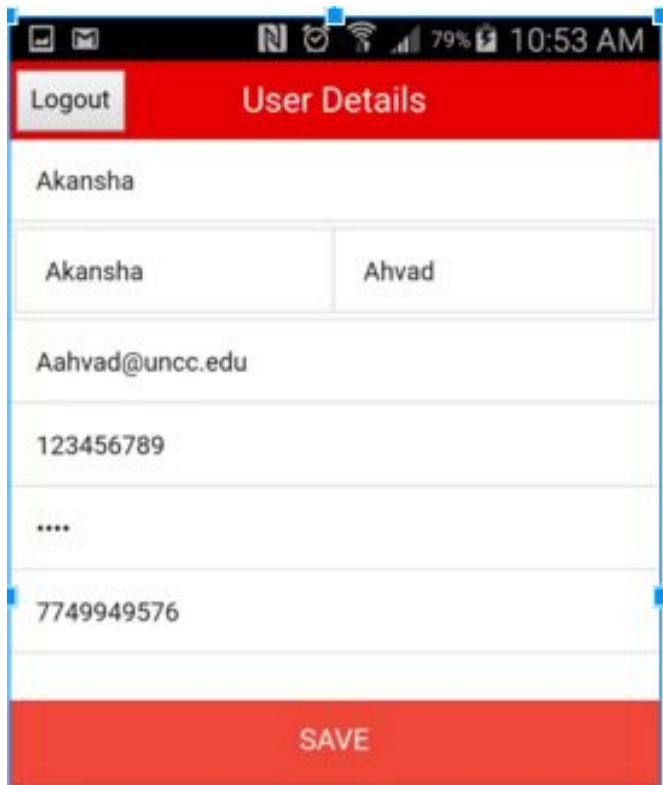
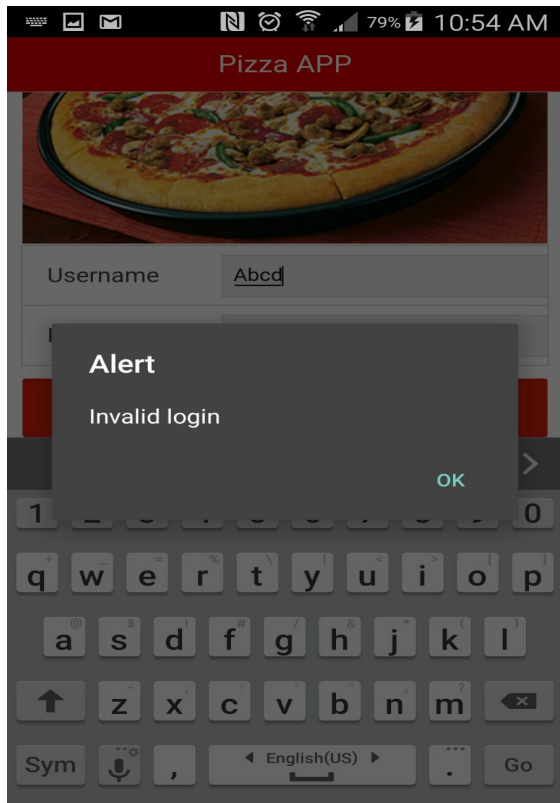
**Merchant:** A merchant table stores the information about the merchant and its personal information. The table consists of merchant id, customer id, first and last name, login password, email id and address.

**Order:** This table stores the order placed by the user. It maps each line to the account Id of the user from the customer table. The table consists of customer id, order id, topping id and total amount.

**Login:** This table stores information about login id and corresponding password.

### Client Side Pages:





## Merchant side pages:

