**Food delivery Application**

The aim of the application is to form a bridge between customer and Restaurant owner. The Restaurant owner must be able to sign up and list out their food items along with price, where the customers can login and view the restaurants nearby and order food items from selected restaurants and makes a payment.

Technologies used:

Backend: Spring REST, MySQL

Frontend: Angular/ html/CSS

Deployment: Web

Functional requirement:

Following is the list of functionalities in Food Delivery

There are two types of users who would access the system viz. Customer and restaurant owner. Each one of them would have some exclusive privileges as follows

1. Customer
2. Restaurant owner

**Customer:**

The customer of the restaurant must have following functionality:

When the customer visits the Food delivery webpage, he/she needs to sign in. Once they are signed in they are presented with an interactive menu. After making a selection, the item is added to their order, which the customer can review the details at any time and place the order.

* **Signup**: If a customer wants to place an order then he/she must be registered with their valid credentials. If a customer is not registered then he/she cannot place the order.
* **Login**:

The customer should login with his/her valid username and password for placing the order.

* **Search for a Restaurant** :

The customer should search for a restaurant in order to place an order.

* **Select items** :

The customer can select the items for ordering

* **Place order** :

The customer place the order after selecting the items

* **Update order** :

If the customer want to update his items list before placing an order, he can update.

* **Delete order**:

The customer can delete his/her order if required.

* **Make a payment** :

Once the order is placed, the customer can make a payment by debit/credit card

**Admin (Restaurant owner):**

The restaurant owner must have the following list of functionality:

The restaurant owner will have the access to manage the menu to create the new items or update the food items or delete the items. The owner also have ability to accept or reject the order of the customer.

* **Login**:

The Restaurant owner needs to be login by entering the valid username and password to give the restaurant menu. If the restaurant owner doesn’t login with his/her valid credentials he cannot add the items to his restaurant.

* **Create list of items**:

The owner can add his list of items to menu. So, customers can place an order.

* **Update items**:

The owner can update his list of food items in his menu.

* **Delete items**:

The owner can delete his list of food items from his menu

* **Reject/Accept order**:

The restaurant owner has right to reject or accept the customer order.

* **Payment Received:**

Therestaurant owner can check whether the customer’s payment had been received successfully or not.

**Work** **Flow**:

* The Restaurant owner will be sign in to the home page and add list of items to the menu.
* Customer navigates to the Home Page to register and login.
* Once the customer logins they can view for a restaurant to place an order.
* Customer will add their food items to the cart providing the required details.
* Select a method of payment.
* Once the order is confirmed, customer will receive a message of order details.
* Restaurant owner will check the order and payment details of the customer.
* Customer will be provided with his/her order.
* Customer will logout from the application.

Modules:

* Admin

In this module the owner can manage services (add, view, delete, update list of items).

* Customer

In this module the customer register with their details, search nearby restaurants and places the order or cancels the order.

* Services

The services are managed by the restaurant owner so that any customer issues are handled by the owner.

* Request

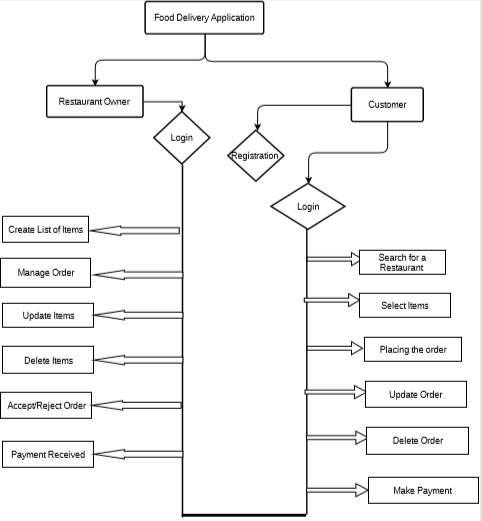
The Restaurant owner have the ability to accept or reject the customer orders

* Payment

The customer has a privilege to select a method of payment by Credit/debit card.

* Over all summary: The restaurant owner can view an overall summary regarding the order history of their customers

Flow Chart:



Schema Used:

Schema to be used:

**Restaurant**:

restaurant\_Id Number(10) (Pk), restaurant\_Name VARCHAR(10), restaurant\_address VARCHAR(10), restaurant \_mail\_Id VARCHAR(20), admin\_Id number(FK)

**Resturant\_Item** :

item\_Id Number(10) (PK), restaurant\_Id Number(10) (FK), item\_name VARCHAR(10), item\_description VARCHAR(10), item\_price Number(10)

**Customer**:

Customer\_Id Number(10) (PK), first\_Name VARCHAR (20), last\_Name VARCHAR (20), cust\_Email\_Id VARCHAR (20), cust\_address VARCHAR (20), cust\_mobile\_Number VARCHAR (20).

**Order**:

Order\_Id Number(10) (PK), Customer\_Id Number(10) (FK),order\_date VARCHAR (20), order\_total\_amount Number(10)

**Payment**:

Payment\_Id Number(10) (PK), Order\_Id Number(10) (FK), payment\_date VARCHAR (20), payment\_price Number(10)

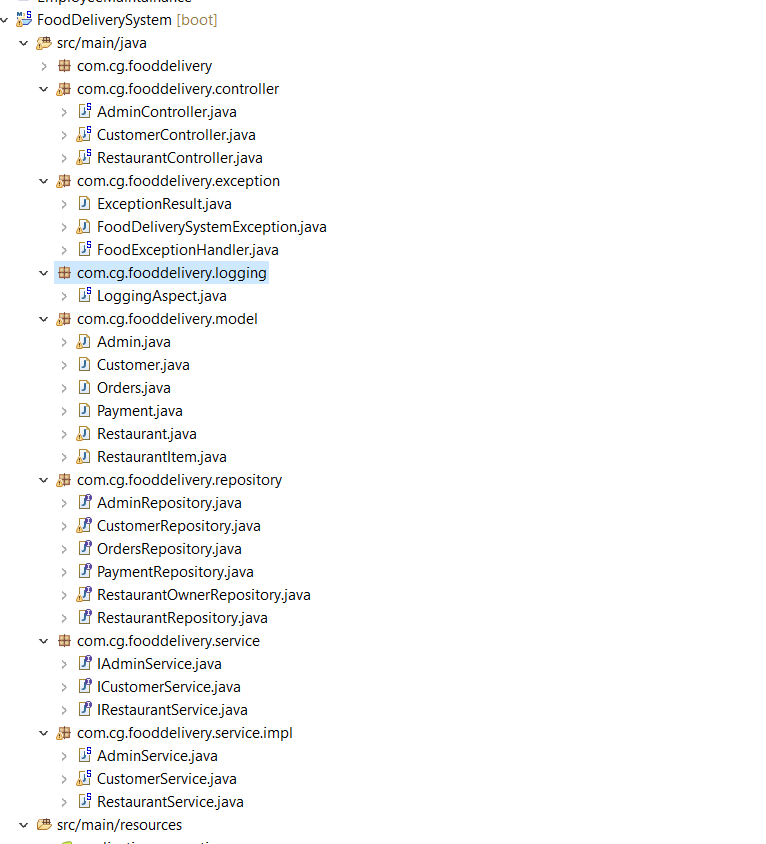
**Admin** :

Admin\_Id Number(10) (Pk), admin\_Username VARCHAR(10), admin\_Password(10)

SpringBoot Structure:

We have included 3 controllers and 3 service layers and respective Repositories.

* Admin Controller
* Customer Controller
* Restaurant Controller
* Admin Service
* Customer Service
* Restaurant Service and services implementation



Team:

1. Manga Sneha
2. Palakurthy Spandana
3. Kongara Bindusri
4. Boreddy Swapna
5. Kamble Aditya
6. Pooja Tiwari