

# **Grp:P003 Design Document**

for

## **FOODEX**

By

Neel K.Parikh(190001044)

Bhoomil Gohel(190001008)

Parul Mogre(190001037)

Yash Suvarna(190001068)

Shivesh Dave(190001057)

**May 07, 2021**

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Scope . . . . .	3
<b>2</b>	<b>DESIGN OVERVIEW</b>	<b>3</b>
2.1	Technologies Used . . . . .	3
2.2	System Architecture . . . . .	4
2.3	System Operation . . . . .	4
<b>3</b>	<b>The Customer Interface</b>	<b>6</b>
<b>4</b>	<b>Data Model And Storage</b>	<b>8</b>
4.1	Customer . . . . .	9
4.1.1	Attributes . . . . .	9
4.1.2	Methods . . . . .	9
4.2	Admin . . . . .	10
4.2.1	Attributes . . . . .	10
4.2.2	Methods . . . . .	10
4.3	Payment . . . . .	11
4.3.1	Attributes . . . . .	11
4.4	Menu-Items . . . . .	11
4.4.1	Attributes . . . . .	11
4.4.2	Methods . . . . .	11
4.5	Cash On Delivery . . . . .	12
4.5.1	Attributes . . . . .	12
4.6	System orders . . . . .	12
4.6.1	Methods . . . . .	12
4.7	Relational Schema . . . . .	14
4.8	ER diagram . . . . .	15

# 1 INTRODUCTION

## 1.1 Purpose

This document aims to describe the implementation of the design of the Foodex web application described in the SRS document. The Foodex Web Application is designed for the management of a restaurant.

## 1.2 Scope

This document describes the implementation details of the Foodex Web Application. Foodex is a restaurant management system created to automate everyday restaurant tasks such as order and inventory management and feedback collection. This version of the program would only deal with these activities. The key goal is to boost the restaurant's efficiency by eliminating regular paperwork. Tasks will be completed in less time and with greater efficiency with this method.

# 2 DESIGN OVERVIEW

## 2.1 Technologies Used

The website can be accessed using any of the modern internet browsers and can run on any consumer-based operating system (Windows, Mac, Linux, etc.). The website was prepared on Microsoft Visual Studio Code and MySQL Workbench. The website's backend is built using NodeJS and MySQL, whereas the front end is prepared using HTML5, Bootstrap, JavaScript.

## 2.2 System Architecture

- **The Customer Interface** - The windowed Interface to check menu, place order for the customer and update Menu and other settings for The Admin.
- **Data Model** - The classes needed to organize Orders, Menu Items, Customer Profiles, Ratings, etc
- **Data Storage** - The interface for storing, importing and exporting the data model and raw collected data.

## 2.3 System Operation

The following figure represents the sequence of events while placing an order or performing other operations on the interface.

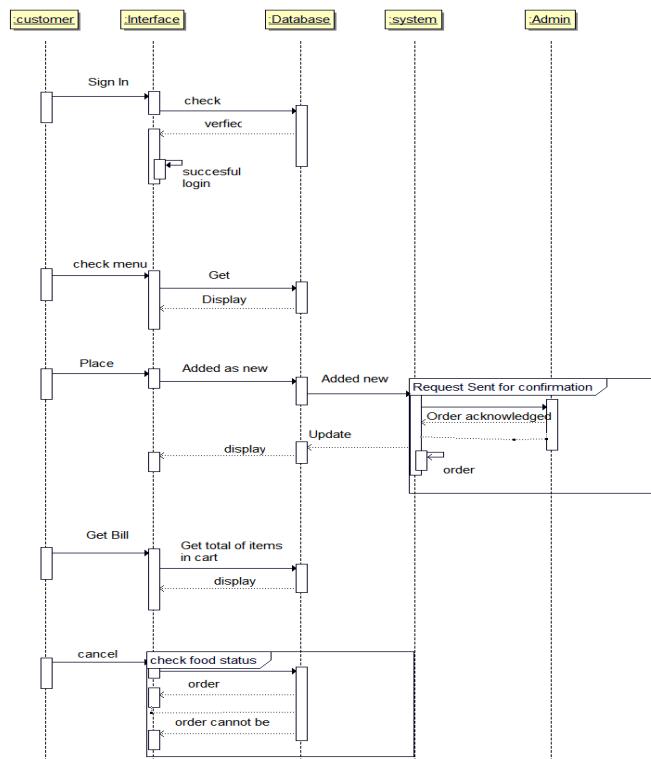


Figure 1: Foodex Sequence Diagram

The following diagram represents the activity flow while placing the order and interacting with the interface.

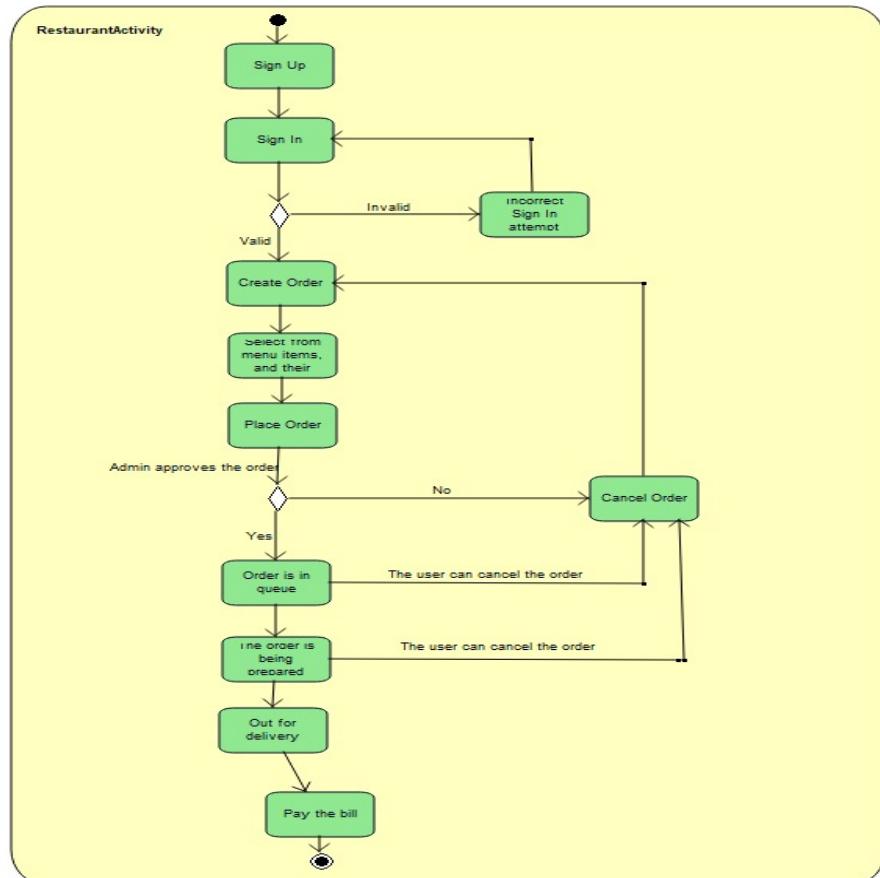


Figure 2: Activity Diagram

Following diagram represents the use case diagram of the system. It represents a discrete task that involves external interaction with a system. Actors in this use case are the customers and the admin.

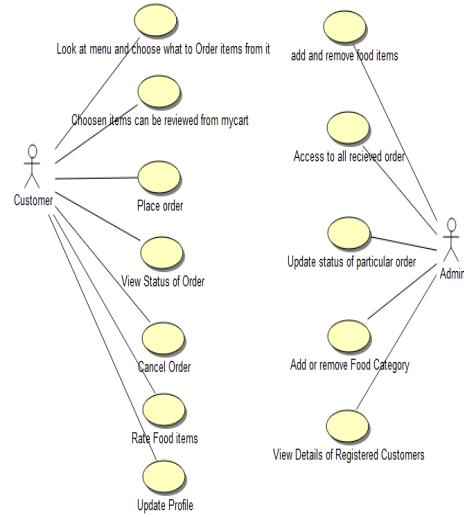


Figure 3: UseCase Diagram

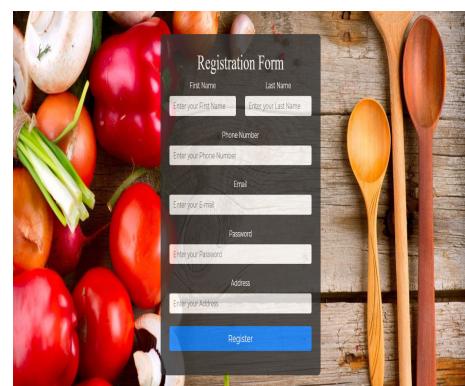
### 3 The Customer Interface

Following figures represent the Mockup of the Customer Interface.

A new user encounters the home page of the website, when visiting for the first time.



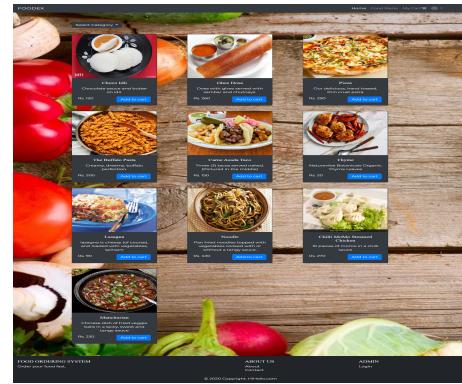
(a) Home Page



(b) Sign Up



(a) Login Page



(b) Menu

- The registered user can directly login into the website while the new user is directed to the sign-up page.
- After Logging In, the menu page is displayed. One can pick items from the menu card and add to my cart.
- To place the order, user is directed to my cart page which displays the sum total of all items in the cart.

 A screenshot of the "My cart" page. At the top, there's a header with the FOODEX logo and navigation links for "Home", "Food Menu", "My Cart", and "Logout". The main content area is titled "Your Order" and lists three items in a grid:
 

- Choco Muffin**: Chocolate sauce and butter on silk. Price: 120, Quantity: 1. Buttons: "Remove From Cart" (red)
- Garnish Dosa**: Dosa with green salad with sambar and chutney. Price: 260, Quantity: 2. Buttons: "Remove From Cart" (red)
- Pasta**: Our delicious hand tossed thin crust pizza. Price: 360, Quantity: 1. Buttons: "Remove From Cart" (red)

 To the right of the order list is a sidebar with a form for entering delivery details:
 

- "Flat or Building Name": Street Name, Apartment, Landmark Key, City
- "Your Total is: 930"
- Buttons: "Place Order" (blue), "Check Status" (blue)

 At the bottom of the page is a footer with links for "FOOD ORDERING SYSTEM", "ABOUT US", and "ADMIN".

Figure 6: My cart



(a) Admin Login Page

Fooder	
Food Ordering System	
Total Orders	New Orders
25 Total Orders	5 New Orders
Confirmed Orders	Food Dispatched
6 Confirmed Orders	5 Food Dispatched
Food Delivered	Cancelled
6 FoodDelivered	1 Cancelled

(b) Admin Dashboard

Figure 7: Admin Portal

## 4 Data Model And Storage

The following diagram represents UML model for System Class.

The model depicts the object-oriented nature of the system and illustrates the classes in a system and the associations between them.

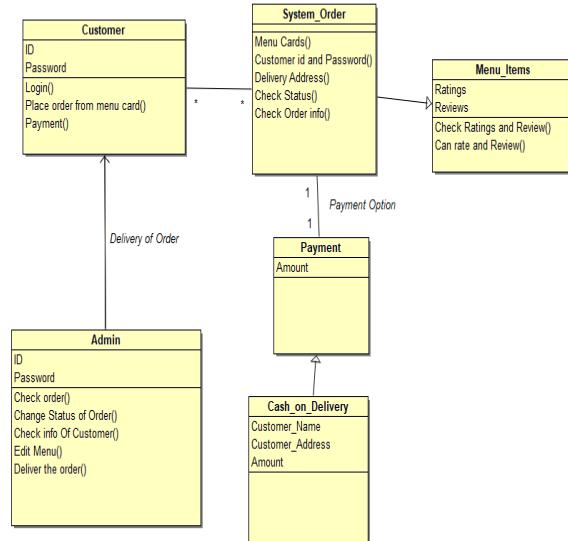


Figure 8: Class Diagram

## 4.1 Customer

### 4.1.1 Attributes

Name	Type	Description
ID	Integer	Stores the customer Id of the user.
Password	VarChar	Required Login credential into the account.

### 4.1.2 Methods

Login()	.
Input	Takes email-id and password as input.
Output	Menu page for the customer and Admin Dashboard for the admin.

Place Order From Menu card()	
Input	Menu item id and name from the storage directory.
Output	Directs to the my cart page of the website.

Payment()	
Input	Price of the individual item in my-cart,
Output	Bill of the order placed.

## 4.2 Admin

### 4.2.1 Attributes

Name	Type	Description
ID	Integer	Stores the customer Id of the user.
Password	VarChar	Required Login credential into the account.

### 4.2.2 Methods

Check order()	
Input	void
Output	List of all the new, confirmed, delivered and cancelled orders.

Check status of order()	
Input	void
Output	Returns the current status of a particular order.

Check info of customer()	
Input	Customer id
Output	Returns the details of user with entered customer-id.

Edit Menu()	
Input	Input of type String.
Output	Adds, deletes or updates the menu as per the data entered.

Deliver the order()	
Input	order id
Output	changes the status of order to delivered.

## 4.3 Payment

### 4.3.1 Attributes

Name	Type	Description
Amount	Integer	Stores the bill total of the items in my cart.

## 4.4 Menu-Items

### 4.4.1 Attributes

Name	Type	Description
Review	String	Stores the review of the menu-item,
Rating	Number	Stores the average rating of the menu-item given by the customer.

### 4.4.2 Methods

Check Ratings and Review()	
Input	Item-id
Output	All the reviews and rating on the item.

Can Rate and Review()	
Input	Varchar
Output	rating and review entered by the user.

## 4.5 Cash On Delivery

### 4.5.1 Attributes

Name	Type	Description
Customer Name	String	Stores the name of the customer.
Customer Address	Varchar	Stores the address of the customer.
Amount	Number	Price total of the food ordered.

## 4.6 System orders

### 4.6.1 Methods

Menu cards()	
Input	void
Output	Returns the menu card of the restaurant.

Customer Id and Password()	
Input	Login credentials
Output	Returns Menu page of the website

Delivery Address()	
Input	Customer Id
Output	Returns delivery address of the user with given user-id.

Check Status()	
Input	order
Output	Returns current status of the order.

Check order Info()	
Input	order id
Output	Returns Details about the order.

## 4.7 Relational Schema

Relation schema defines the design and structure of the relation like it consists of the relation name, set of attributes/field names/column names. every attribute would have an associated domain.

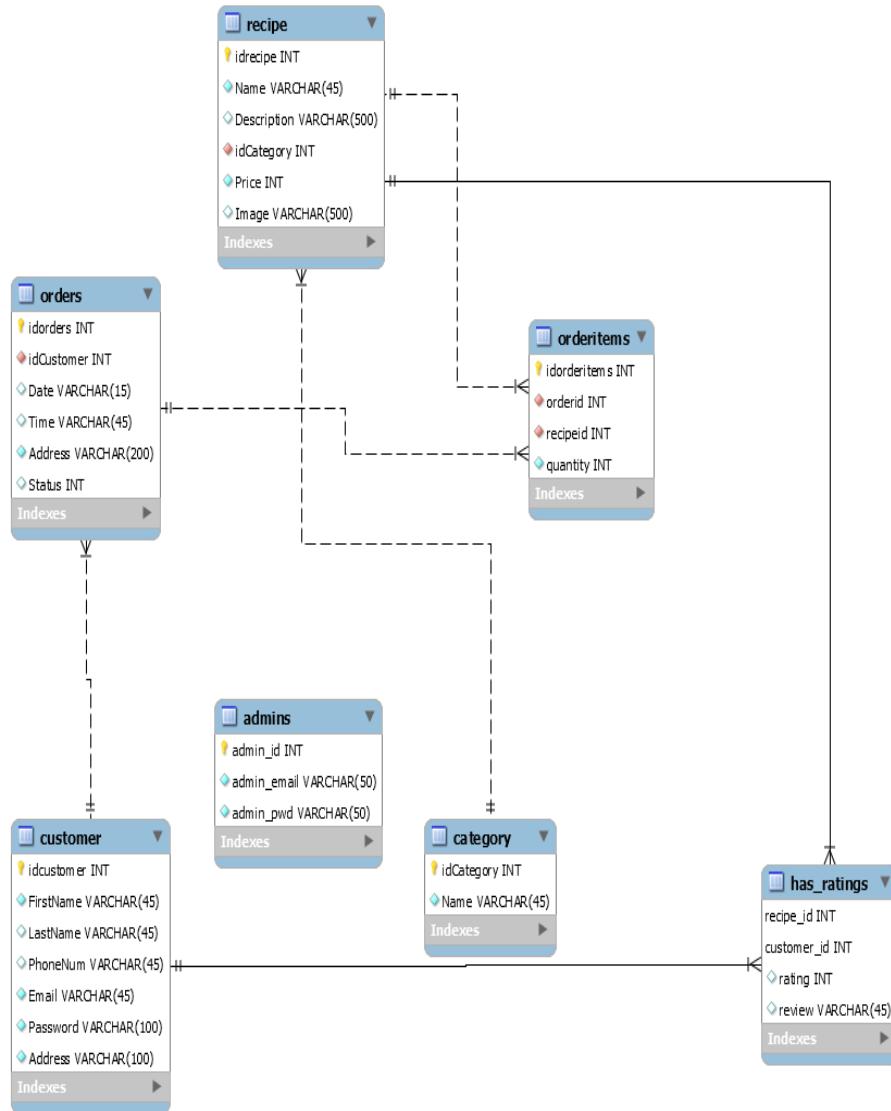


Figure 9: Relational Schema

## 4.8 ER diagram

ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.

In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

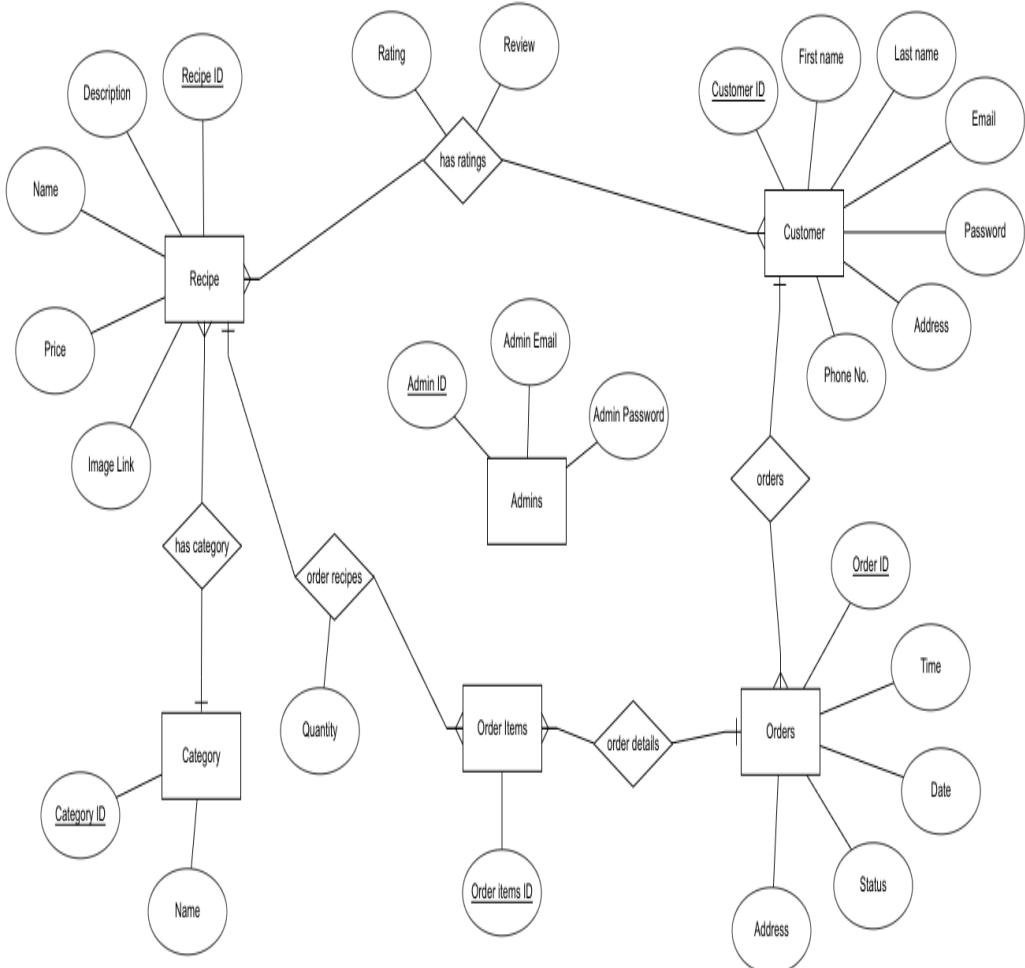


Figure 10: ER-Diagram