Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

B. Tech. CE Semester - VI

Subject: (CE-620) Object Oriented Software Engineering

Project Title:

Food Ordering System

By:

Priyanka Dave CE-023 (17ICUOG024)

Trupti Gandhi CE-034 (17CEUON042)

Guided By:

Prof. Jatayu Baxi

Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

CERTIFICATE

This is to certify that Service Oriented Computing Project entitled “**Food Ordering System**” is the bonafied report of work carried out by

1) Priyanka Dave (17ICUOG024)

2) Trupti Gandhi (17CEUON042)

Of Department of Computer Engineering , Semester VI , academic year 2019-2020, under our supervision and guidance.

Guide	**HOD**
Prof. Jatayu Baxi	**Dr. C. K. Bhensdadia**
Assistant Professor of Department of Computer Engineering, Dharmsinh Desai University, Nadiad	Head of Department of Computer Engineering, Dharmsinh Desai University,Nadiad

Table of Contents

1.Abstract

Food is necessity for every living being. The “**online food ordering system**” provides convenience for the customers. It overcomes the disadvantages of the traditional queuing system. This system increases the takeaway of foods than visitors. Therefore, this system enhances the speed and standardization of taking the order from the customer. It provides a better communication platform. The online food ordering system set up menu online and the customers easily places the order with a simple mouse click. Also with a food menu online you can easily track the orders, maintain customer's database and improve your food delivery service. This system allows the user to select the desired food items from the displayed menu. The user orders the food items. The payment can be made online or pay-on-delivery system. The user’s details are maintained confidential because it maintains a separate account for each user. An id and password is provided for each user. Therefore it provides a more secured ordering.

2.Introduction

2.1. Brief Introduction

This is a Windows Form Application. It enables the end users to register online, select the food from the e-menu card, read the E-menu card and order food online. By just selecting the food that the user want to have. The results after selecting the food from the E-menu card will directly appear in the screen near the Chef who is going to cook the food for you. It provides lightning fast delivery so that users can have hot and fresh food. This application allows you to order food from your favorite restaurants any time.

2.2 Technologies Used

Operating Systems: Microsoft Windows

Web designing language: C sharp

Database: SQL Server Database

Software Tool: Visual Studio

Operating Environment : The product can run on any browser.

3.Software Requirement Specifications

3.1 Product Scope

This system is designed to enable customer to order food from anywhere any time. It enables restaurant to advertise themselves. It provides interface between customer and restaurant so that they can give feedback to them. It provides confirmation before place order.

3.2 Types of Users

1. User(Customer)
2. Restaurant Manager
3. Admin

3.3 System Functional Requirements

R.1 User

R.1.1 Login

Description: If you already have an account, login here.

Input: Username, password.

Output: Successfully logged in.

Processing: User will enter username and password to login, if already registered.

R.1.2 Register

Description: If you are a new user, register yourself.

Input: Username, email, password, contact number.

Output: Successfully registered.

Processing: User will enter username and password to register, if not registered.

R.1.3 Navigate Menu

Description: user can search food item.

R.1.3.1 Select Food Item

Input: User selection.

Output: Item added successfully.

R.1.3.1.1 Add Item

Input: User selection.

Output: Item added successfully.

R.1.3.1.2 Delete Item

Input: User selection.

Output: Item deleted successfully.

R.1.3.1.3 Edit Item

Input: User selection.

Output: Item updated successfully.

R.1.4 Search Restaurant

Description: Customer can search restaurants.

R.1.4.1 Search manually

Input: user selection.

Output: restaurant selected.

R.1.4.2 Location based.

Input: user selection.

Output: restaurant selected.

R.1.5 Make payment

Description: Customer can make payment through different methods.

R.1.5.1 Make payment via Cash on Delivery

Input: User selection.

Output: Food Item delivered successfully.

Processing: Customer will make the payment through cash on delivery.

R.1.5.2 Make payment via Credit card/Debit card

Input: User selection.

Output: Order Placed.

Processing: Amount will transfer from Customer's account to Manager's account.

R.1.5.1 Make payment via UPI

Input: User selection.

Output: Order Placed.

Processing: Amount will transfer from Customer's account to Manager's account.

R.1.6 Add to Cart

Description: User can add food item to cart.

Input: User selection.

Output: Added successfully to cart.

R.1.6.1 View Cart

Input: User selection.

Output: Display user’s cart.

R.1.6.2 Delete Cart

Input: Cart details.

Output: Deleted successfully from cart.

R.1.7 Give Feedback

Description: Customer can give the feedback either good or bad.

R.1.7.1 About Restaurant

Input: User selection.

Output: Display feedback.

R.1.7.2 About Application

Input: User selection.

Output: Display feedback.

R.1.8 Logout

Description: User logout from the application.

Input: User selection.

Output: Successfully logged out.

R.1.9 View offers

Description: Customer can view the offers.

Input: User selection.

Output: viewed offers.

R.1.10 View ads

Description: Customer can view the ads.

Input: User selection.

Output: viewed ads.

R.1.11 Place Order

Description: Customer will order food items by entering address details.

Input: Name, contact-no, address, city, pin-code.

Output: Ordered successfully.

R.1.12 Cancel Order

Description: Customer can cancel the order.

Input: User selection.

Output: Order cancelled.

R.2 Restaurant Manager

R.2.1 Login

Description: If you already have an account, login here.

Input: Username, password.

Output: Successfully logged in.

Processing: User will enter username and password to login, if already registered.

R.2.3 View feedback

Description: Restaurant manager will view feedback about Restaurant.

Input: User selection.

Output: Display feedback.

R.2.4 Manage Food Item

Description: Restaurant manager will add, remove and view his food menu.

R.2.4.1 Add Food Item

Input: Food item name, price, description.

Output: Food item added successfully.

Processing: When Manager will click on the add food item, it will show all the food item details.

R.2.4.2 Remove Food Item

Input: Food Item details.

Output: Food Item deleted successfully.

R.2.4.3 View Food Items

Input: User selection.

Output: Display Food items.

R.2.5 Edit Restaurant Details

Description: Restaurant manager will edit his restaurant details.

Input: Edit Restaurant details.

Output: Display Restaurant details.

R.2.6 View feedback

Description: Restaurant Manager will view the feedback of Restaurant.

Input: User selection.

Output: Display feedback.

R.2.7 Logout

Description: Restaurant manager will logout.

Input: User selection.

Output: Successfully logged out.

R.2.8 Manage Request

Description: Restaurant manager will send the request to admin.

R.2.8.1 Send request

Description: Manager will send the request of managing the ads, offers, food items to admin.

Input: Send the request.

Output: Request Sent.

R.2.8.2 Delete Request

Input: Delete Request.

Output: Request deleted.

R.3 Admin

R.3.1 Login

Description: admin login here.

Input: Username, password.

Output: Successfully logged in.

Processing: admin will enter username and password to login

R.3.2 Manage Restaurant

Description: Admin will add, remove and view the Restaurant.

R.3.2.1 Add Restaurant

Input: Restaurant details.

Output: Restaurant added successfully.

Processing: When admin will click on the add Restaurant, it will show all the Restaurant details.

R.3.2.2 Remove Restaurant

Input: Restaurant details.

Output: Restaurant removed successfully.

R.3.2.3 View Restaurant

Input: Restaurant details.

Output: Display Restaurant details.

R.3.3 View feedback

Description: Admin will view the feedback of Application.

Input: User selection.

Output: Display feedback.

R.3.4 Manage Restaurant Manager

Description: Admin will add, remove and view the Restaurant.

R.3.4.1 Add Restaurant Manager

Input: Restaurant details.

Output: Restaurant added successfully.

Processing: When admin will click on the add Restaurant, it will show all the Restaurant details.

R.3.4.2 Remove Restaurant Manager

Input: Restaurant details.

Output: Restaurant removed successfully.

R.3.4.3 View Restaurant Manager

Input: Restaurant details.

Output: Display Restaurant details.

R.3.5 Logout

Description: Admin can logout.

Input: User selection.

Output: Successfully logged out.

R.3.6 Receive Request

Description: Admin will receive the request which is sent by manager.

R.3.6.1 Send Confirmation

Input: Send the confirmation.

Output: Request Confirmed successfully.

3.4 Other Non-Functional Requirements

1. Performance

Performance of the system must be interactive and the delay involved must be less. So in every action-response of the system, there are no immediate delays. In case of opening App component, of popping error messages and saving the settings or there is delay of some seconds.

2. Safety

User details should be securely stored to the server. The main security concern is for account hence proper login mechanism should

3. Reliability

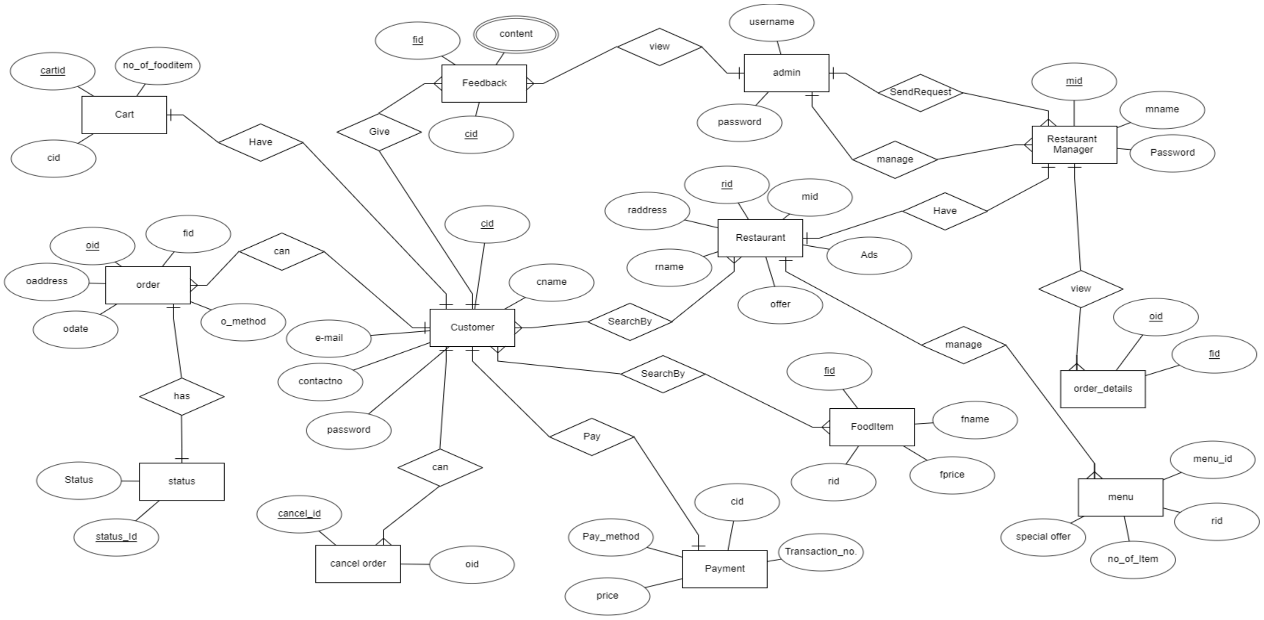
As the system provide the right tools for discussion, Problem solving it must be made sure that system is reliable.

4. Database

System requires to access users data fastly to maintain the performance.

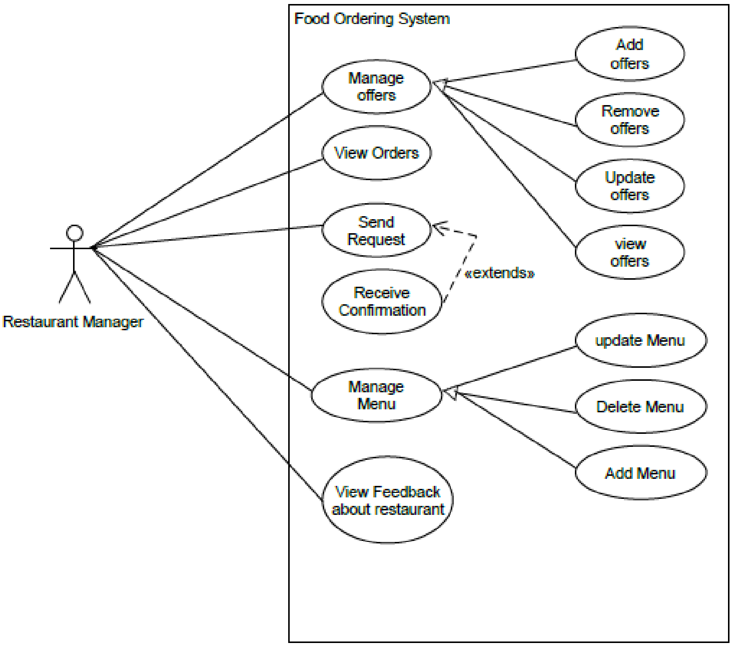
4.Design

4.1 Entity Relationship Diagram:

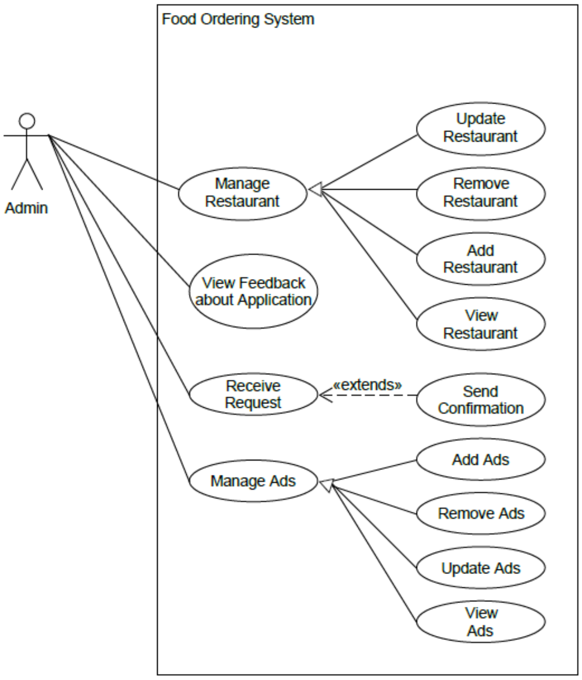


4.2 Use case Diagram

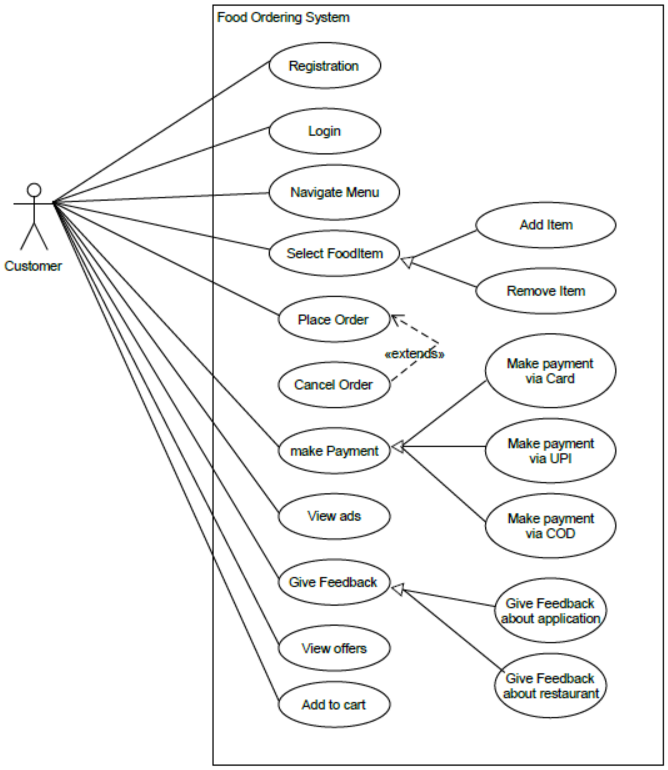
1. Restaurant Manager’s interaction with the system



2. Admin’s interaction with the system



3. Customer’s interaction with the system



User Stories:

Here are some user stories:

1. Administrator:

Use-case: Login into the website

Goal in context: Gain access to the website

Brief Description: This use case is used when the administrator wants to access the website to enable/disable/update the menu details.

Preconditions: The Administrator must be logged onto the website inorder for this use case to begin.

Basic Flow:

The Website prompts the administrator for the username and password.

The Administrator enters the user name and password.

The Website verifies the password and sets the user’s authorization.

The Administrator is given access to the Website to perform his/her tasks.

Alternative Flow:

The administrator enters invalid username and password then he will not be alowed to enter the website.

Post conditions: The website state is unchanged by this use case.

Use Case : Display menu details

Goal in context: View the details of a menu

Brief Description: This use case is used when the administrator wants to view the menu details of the restaurants already existing in the database on the screen.

Preconditions:

The Administrator must be logged into the system in order for this usecase to begin

The details of the student must be pre-exist in the database.

Basic Flow:

The Administrator logs onto the System.

The Administrator search the student from Student id.

The System prompts for the student detail from one of the id.

The menu details are displayed on the screen.

Alternative Flow: Menu details not found If in the Display a menu sub-flows, a menu with the specified id number does not exist, The system displays an error message. The Administrator can then enter a different id number or cancel the operation, at which point the use case ends.

Post conditions: The menu details are displayed on the screen already existing in the system. The state of the system remains unchanged.

2. Customer

Use Case: Customer registration

Goal in context: Registration of a Customer

Brief Description: This use case is used when the Customer register himself/herself in the database online.

Preconditions:

The Customer must accessed the website in order for this use case to begin.

The user id must be unique and entered correctly.

Basic Flow:

The Customer enters into the website.

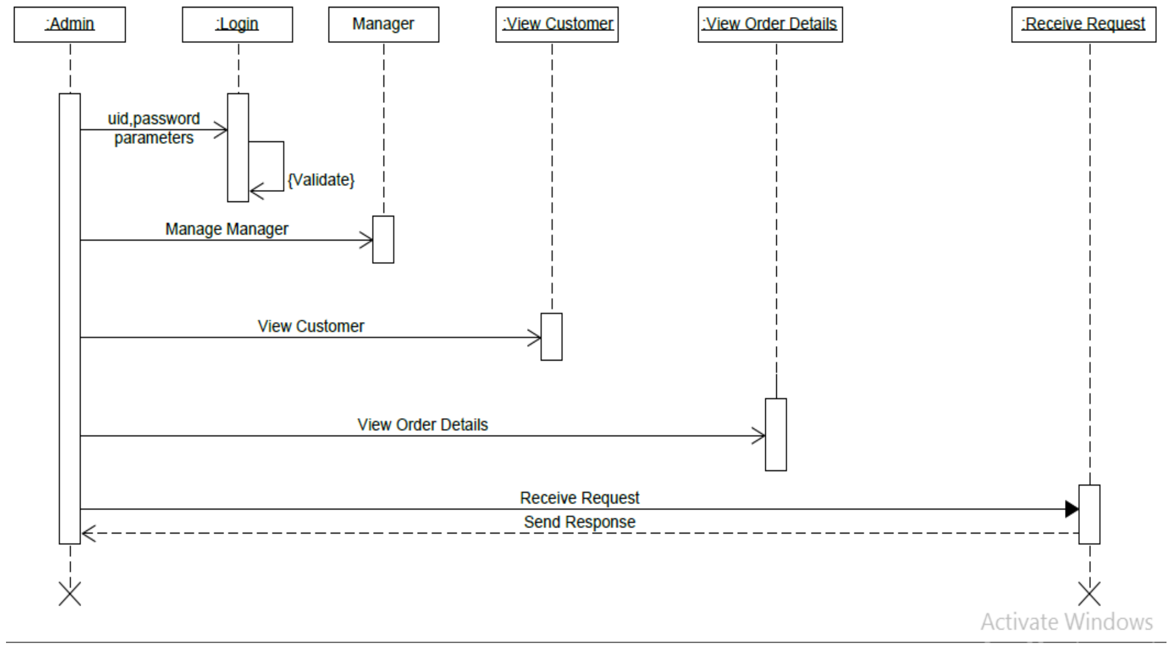
The Customer fills their details .
The details are added to the database. Customer details are displayed on the screen.

Alternative Flow: User ID not unique: if the user id entered is not unique then it will show an error message.

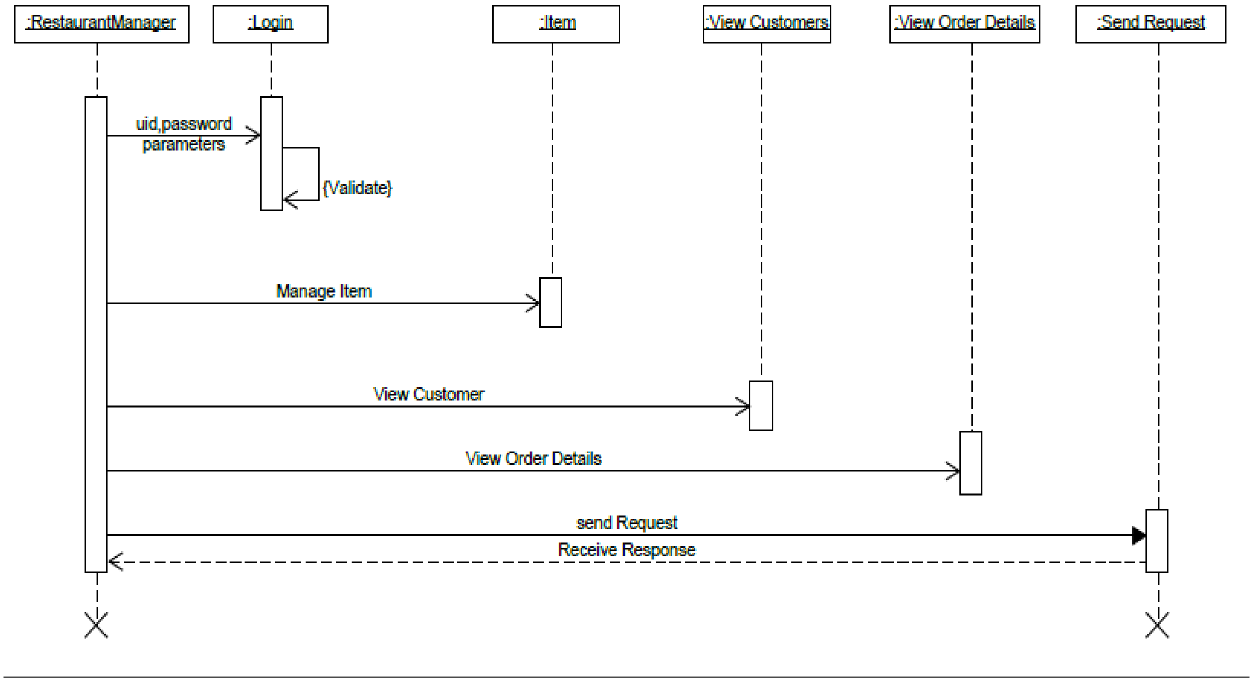
Post conditions: The Customer get registered on the website and to login into that particular the administrator must enable it.

4.3 Sequence Diagram

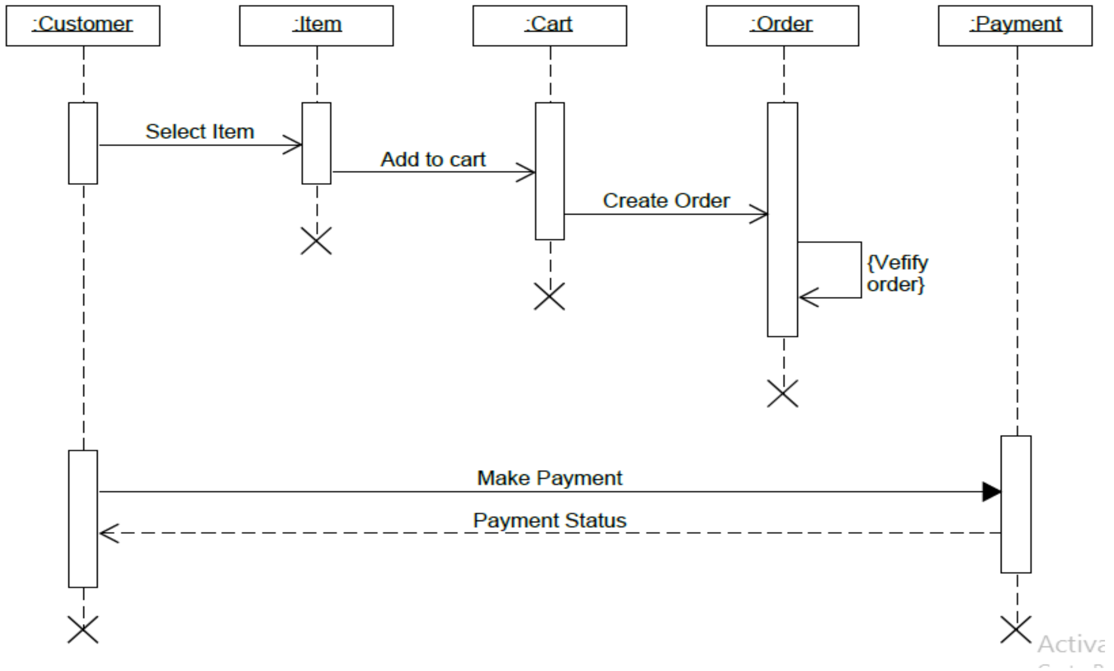
1. Admin Interaction in time Sequence



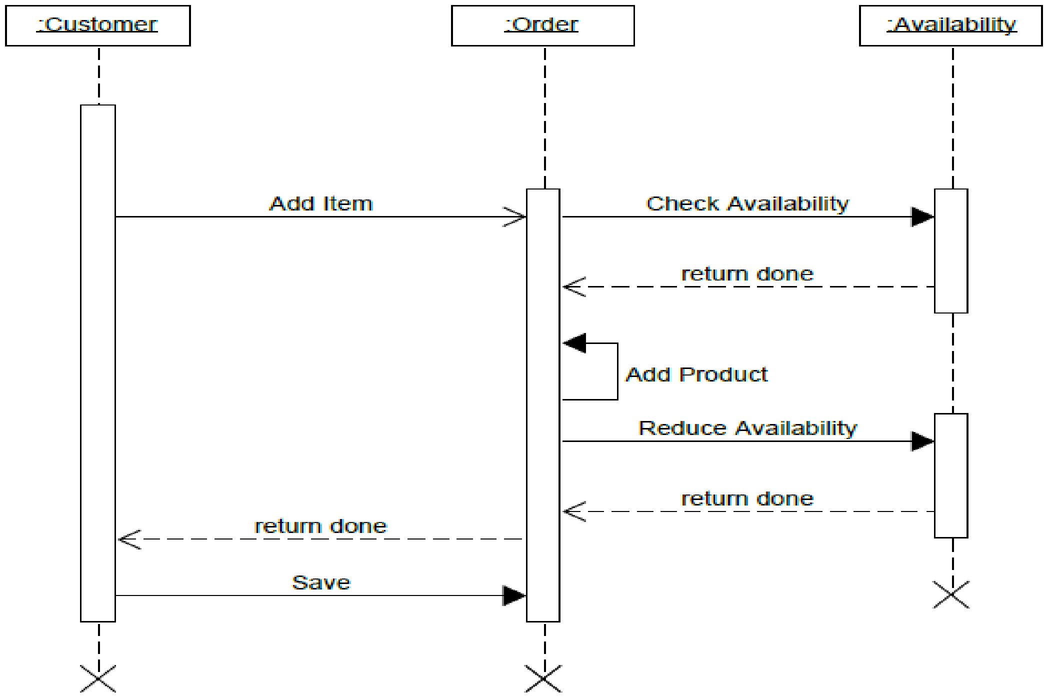
2. Restaurant Manager interaction in time sequence



3. Customer interaction in time sequence

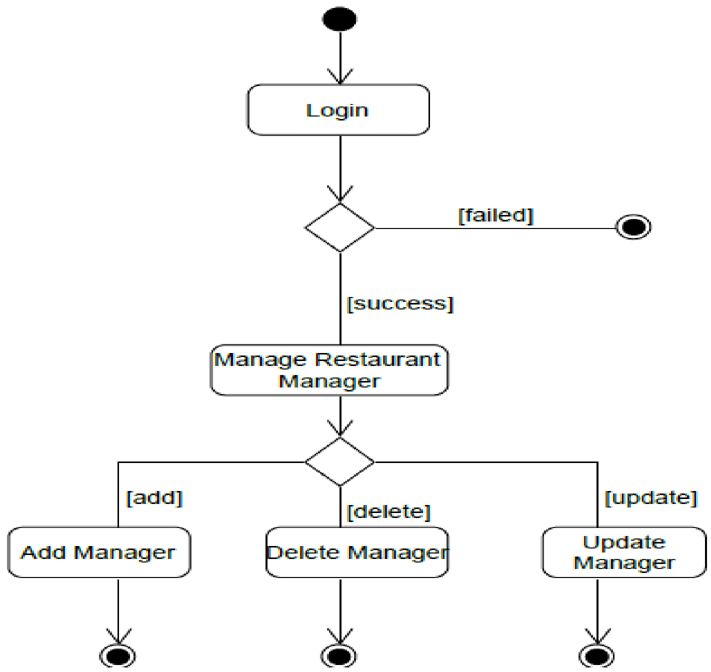


4. Customer interaction in time sequence

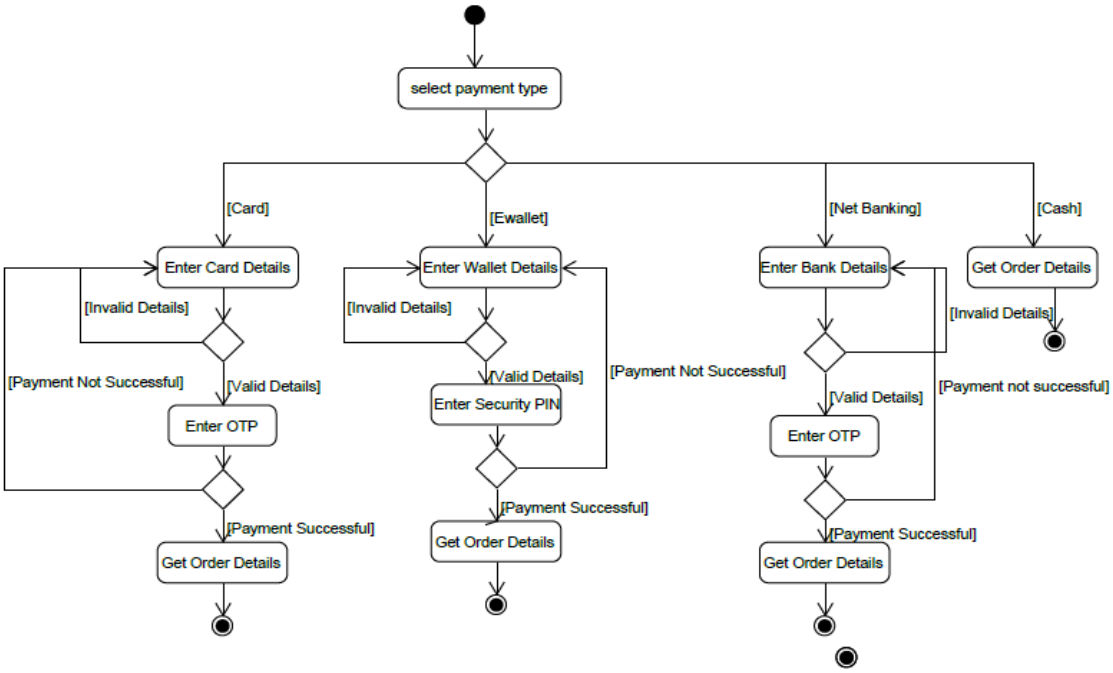


4.4 Activity Diagram

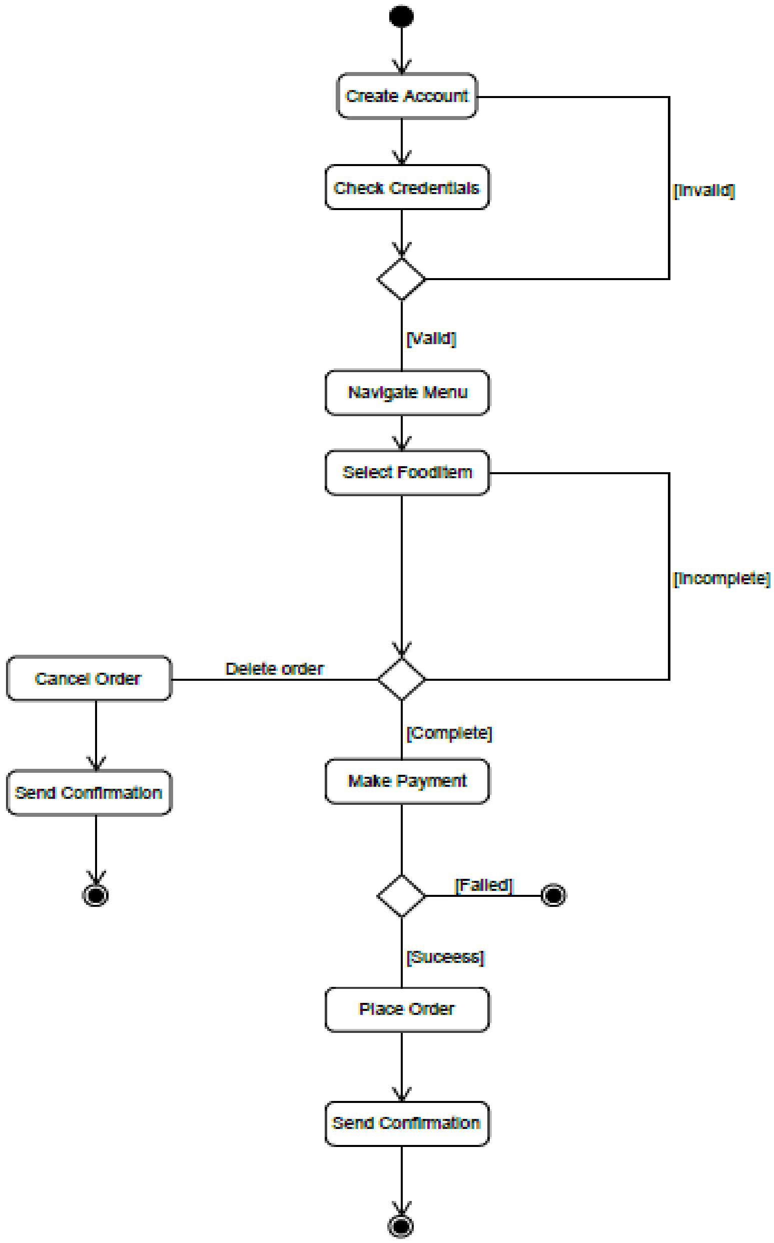
1. Manage Restaurant Manager



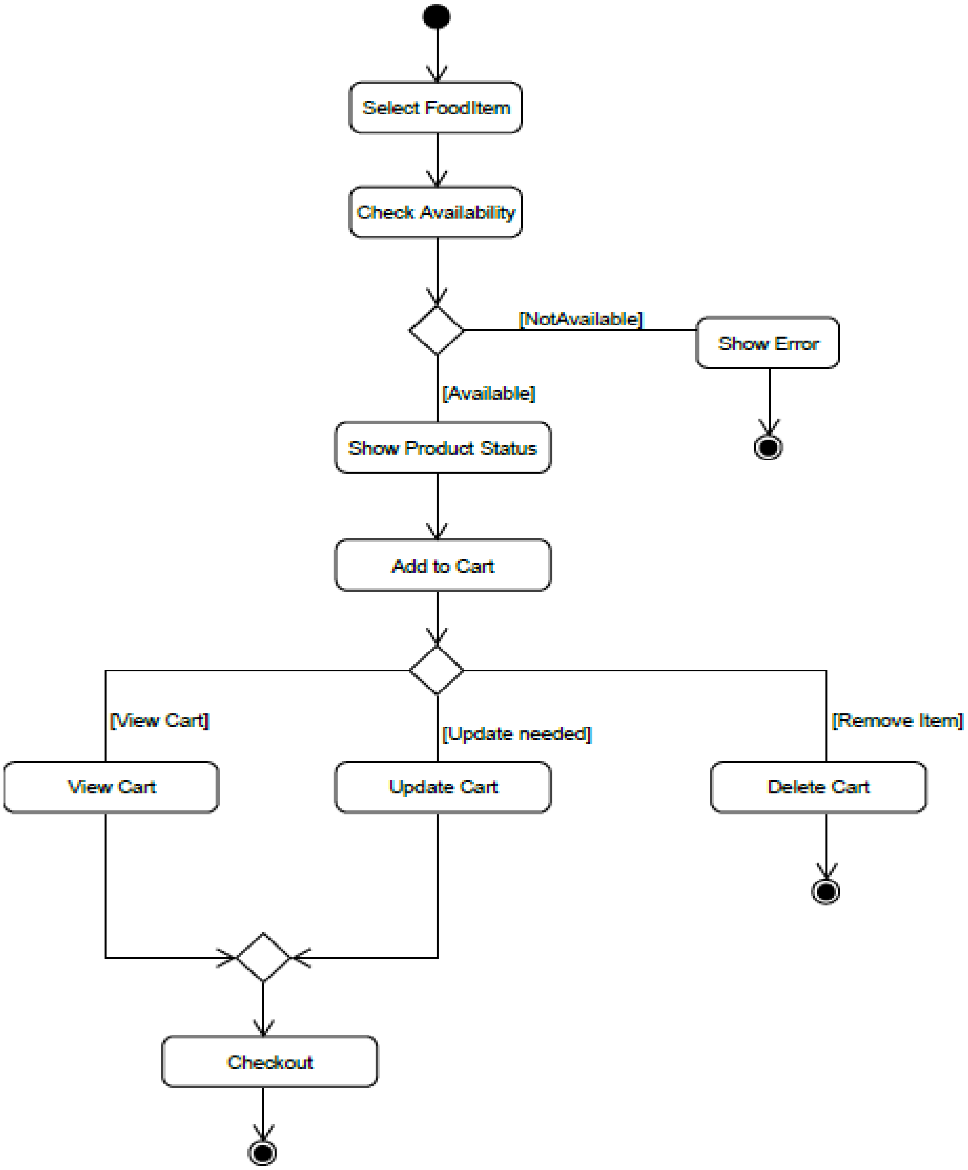
2. Payment Processing



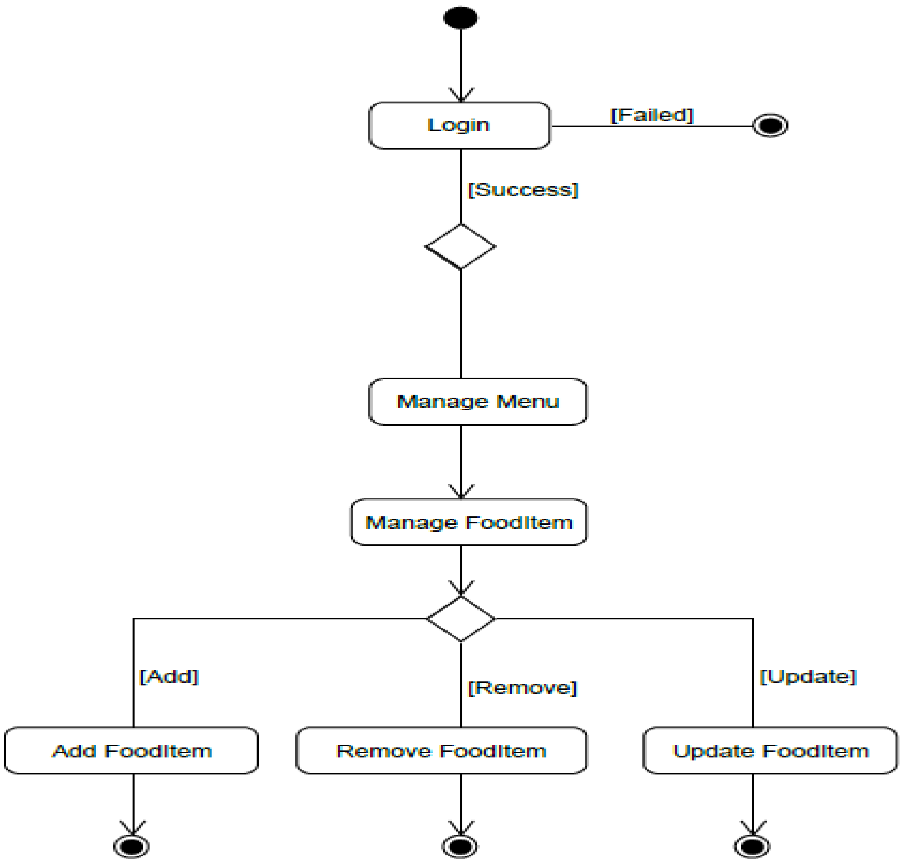
3. Basic Workflow of Customer


Create Account
Check Credentials
[Invalid]
[Valid]
Navigate Menu
Select FoodItem
[Incomplete]
Delete order
Cancel Order
Send Confirmation
[Complete]
Make Payment
[Failed]
[Suceess]
Place Order
Send Confirmation

4. Order Processing

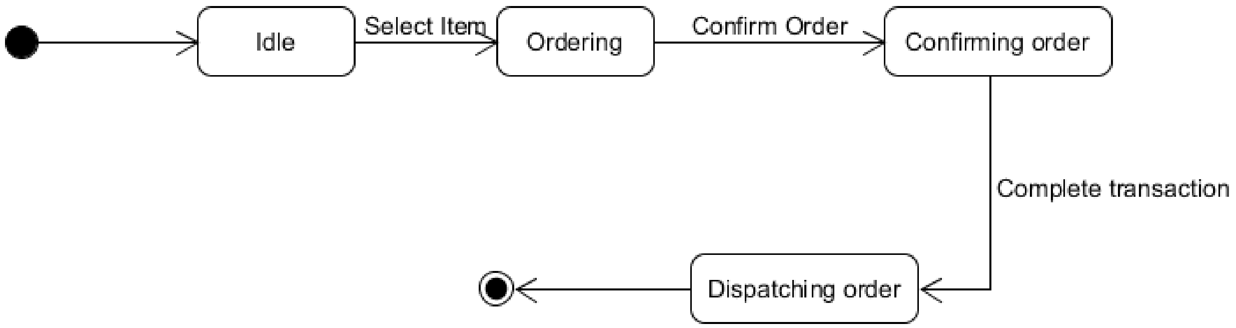


5. Managing Food Item

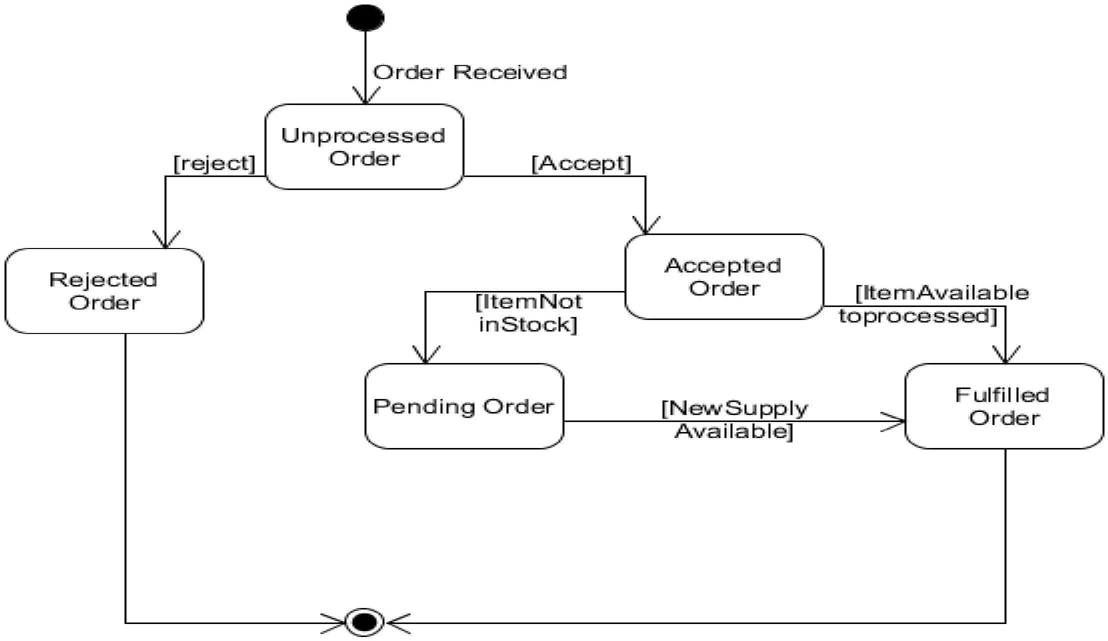


4.5 State Diagram

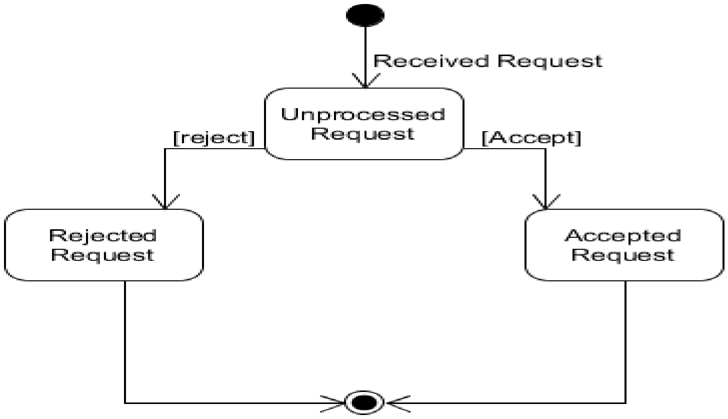
1. Ordering Event



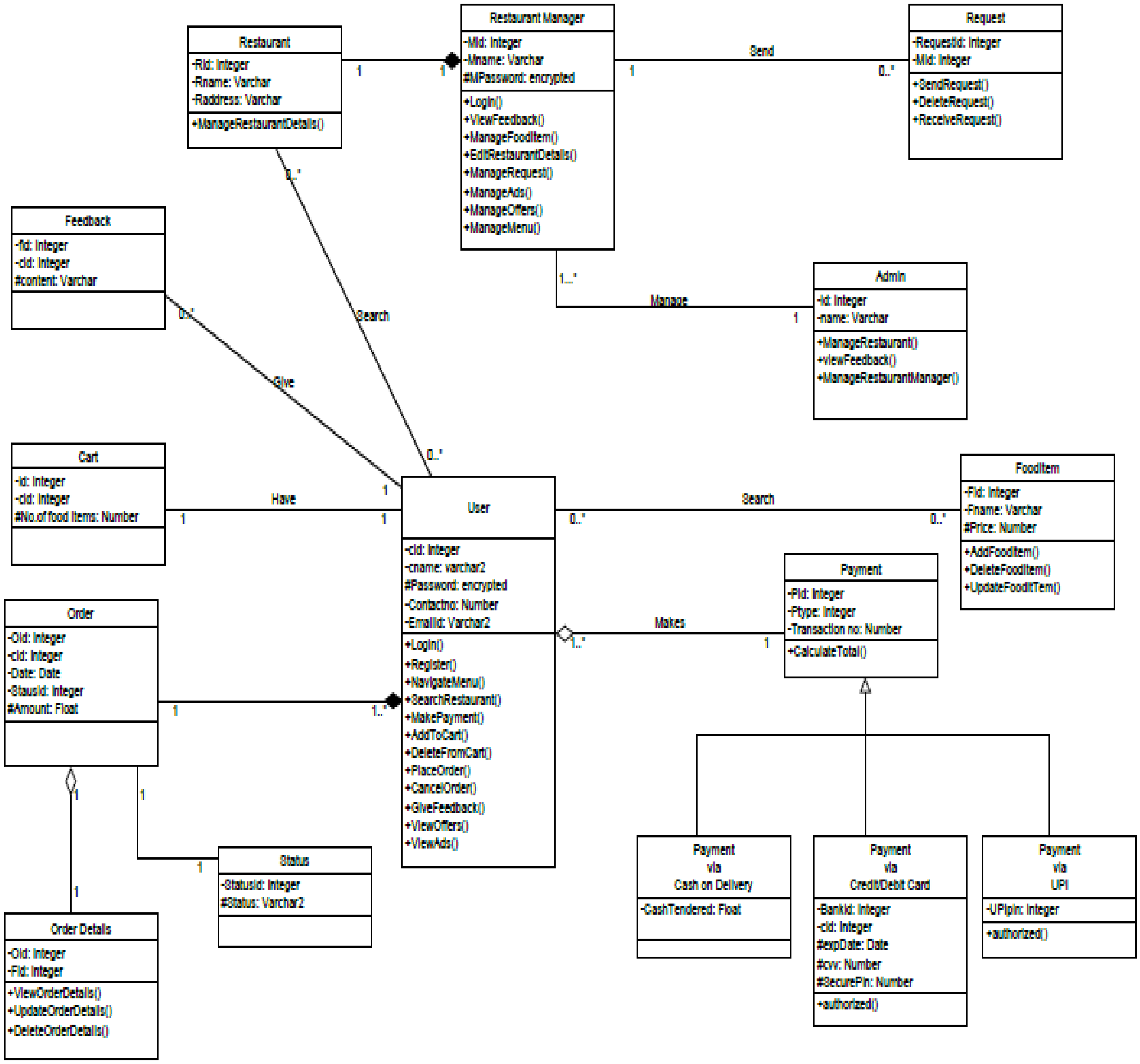
2. Unprocessed order Event



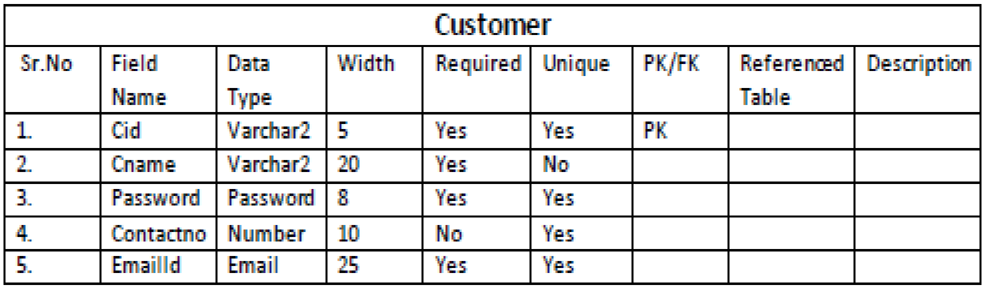
3. Requesting Event



4.6 Class Diagram

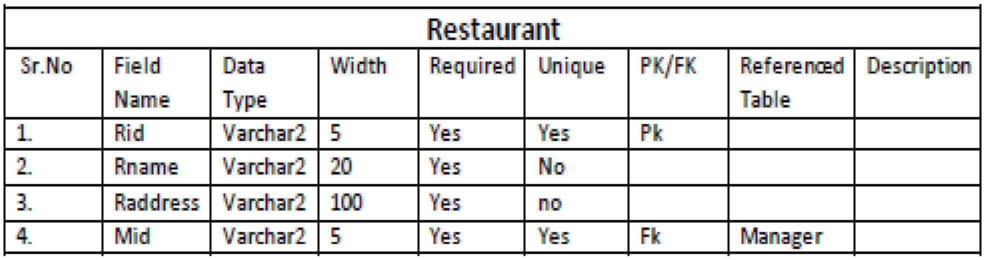


4.7 Data Dictionary



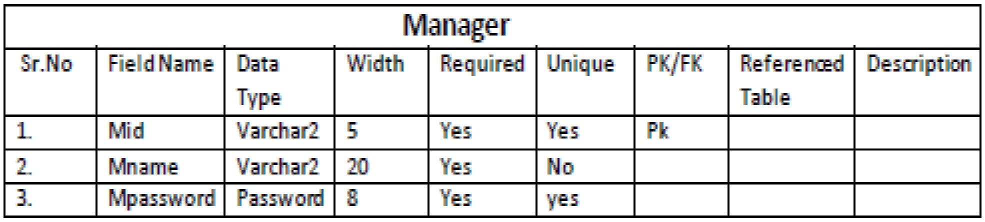
Customer

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Cid	Varchar2	5	Yes	Yes	PK		
2.	Cname	Varchar2	20	Yes	No			
3.	Password	Password	8	Yes	Yes			
4.	Contactno	Number	10	No	Yes			
5.	EmailId	Email	25	Yes	Yes			



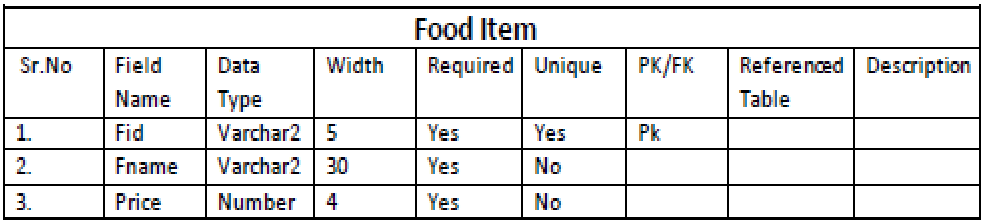
Restaurant

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Rid	Varchar2	5	Yes	Yes	Pk		
2.	Rname	Varchar2	20	Yes	No			
3.	Raddress	Varchar2	100	Yes	no			
4.	Mid	Varchar2	5	Yes	Yes	Fk	Manager	



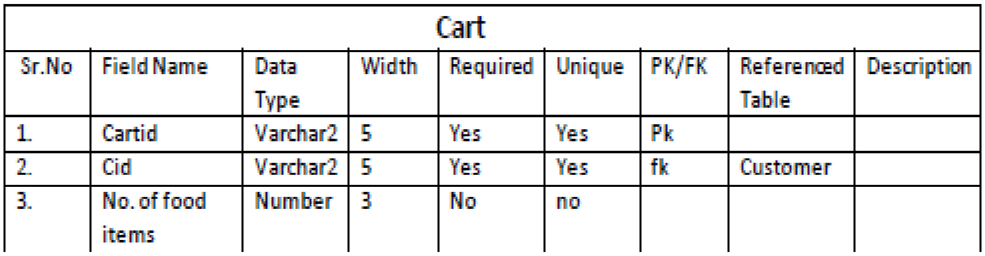
Manager

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Mid	Varchar2	5	Yes	Yes	Pk		
2.	Mname	Varchar2	20	Yes	No			
3.	Mpassword	Password	8	Yes	yes			



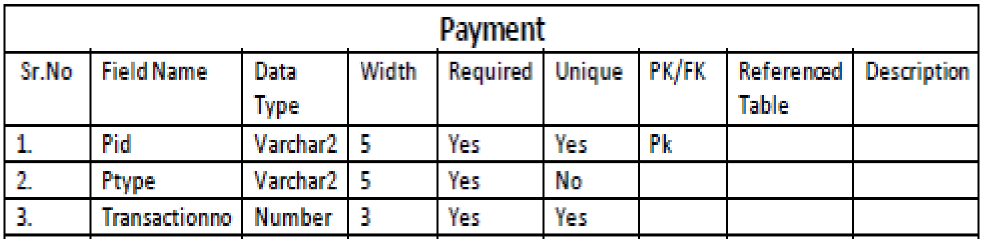
Food Item

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Fid	Varchar2	5	Yes	Yes	Pk		
2.	Fname	Varchar2	30	Yes	No			
3.	Price	Number	4	Yes	No			



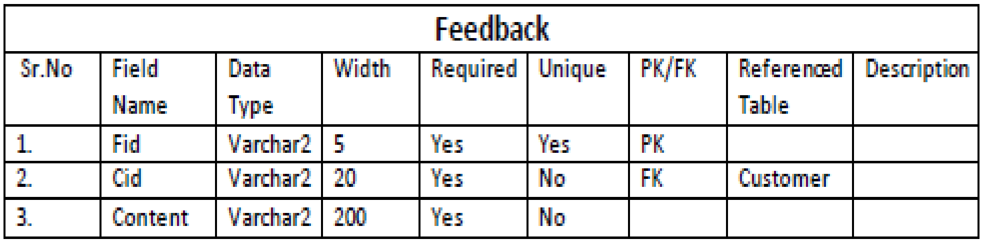
Cart

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Cartid	Varchar2	5	Yes	Yes	Pk		
2.	Cid	Varchar2	5	Yes	Yes	fk	Customer	
3.	No. of food items	Number	3	No	no			



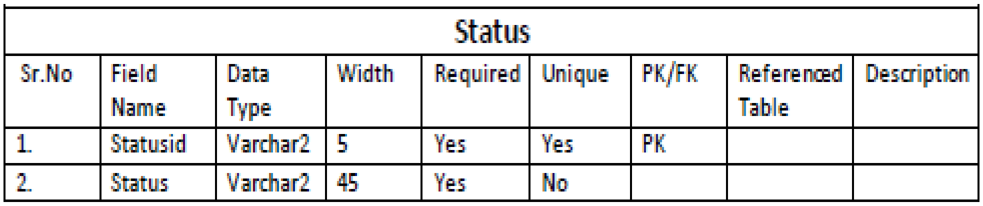
Payment

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Pid	Varchar2	5	Yes	Yes	Pk		
2.	Ptype	Varchar2	5	Yes	No			
3.	Transactionno	Number	3	Yes	Yes			



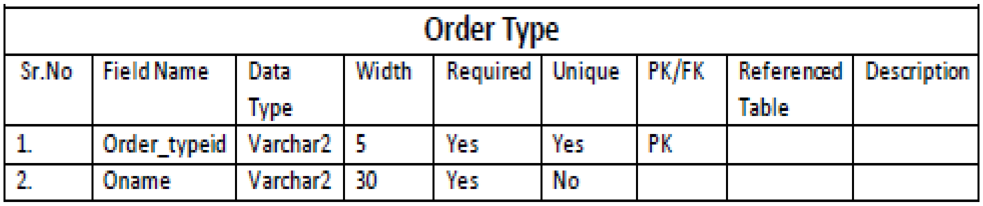
Feedback

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Fid	Varchar2	5	Yes	Yes	PK		
2.	Cid	Varchar2	20	Yes	No	FK	Customer	
3.	Content	Varchar2	200	Yes	No			



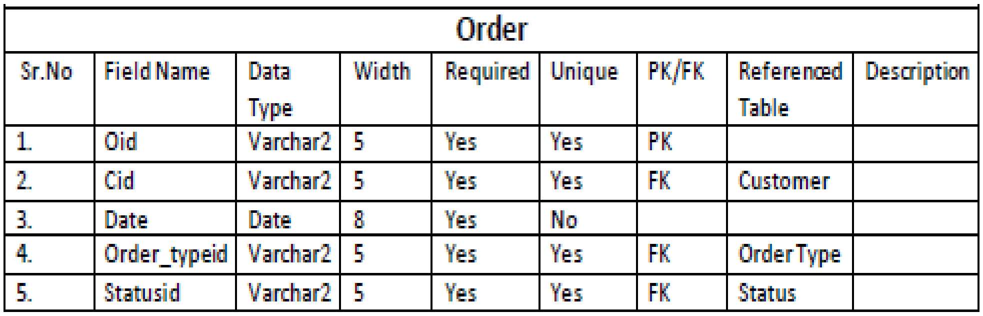
Status

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Statusid	Varchar2	5	Yes	Yes	PK		
2.	Status	Varchar2	45	Yes	No			



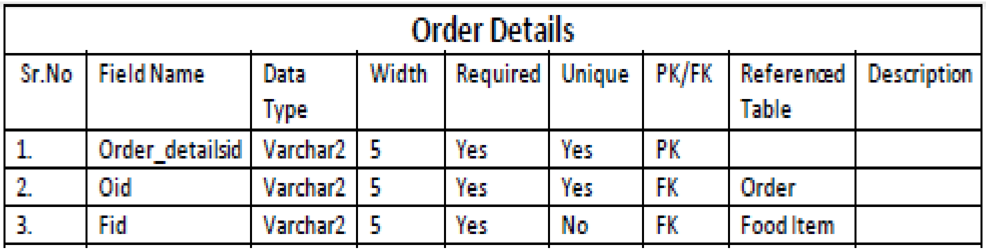
Order Type

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Order_typeid	Varchar2	5	Yes	Yes	PK		
2.	Oname	Varchar2	30	Yes	No			



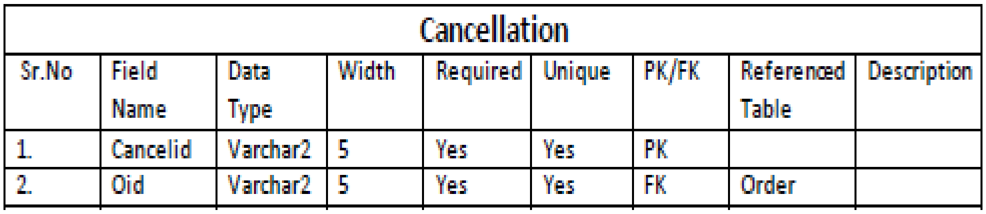
Order

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Oid	Varchar2	5	Yes	Yes	PK		
2.	Cid	Varchar2	5	Yes	Yes	FK	Customer	
3.	Date	Date	8	Yes	No			
4.	Order_typeid	Varchar2	5	Yes	Yes	FK	Order Type	
5.	Statusid	Varchar2	5	Yes	Yes	FK	Status	



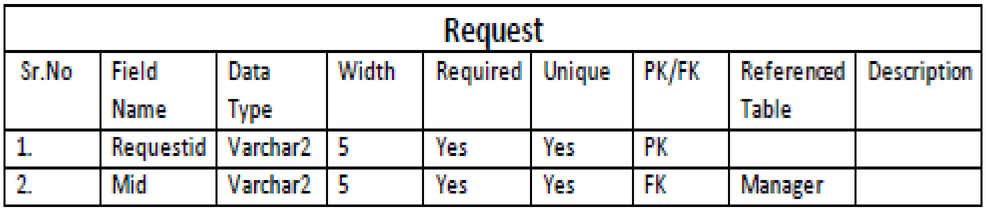
Order Details

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Order_detailsid	Varchar2	5	Yes	Yes	PK		
2.	Oid	Varchar2	5	Yes	Yes	FK	Order	
3.	Fid	Varchar2	5	Yes	No	FK	Food Item	



Cancellation

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Cancelid	Varchar2	5	Yes	Yes	PK		
2.	Oid	Varchar2	5	Yes	Yes	FK	Order	



Request

Sr.No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1.	Requestid	Varchar2	5	Yes	Yes	PK		
2.	Mid	Varchar2	5	Yes	Yes	FK	Manager	

5.Implementation Details

5.1 Module Description

Login module:
Login module will help in authentication of user accounts. This module takes users credentials and then verifies it with registered users, if user is not registered the invalid credentials is shown else if they match with database then login user.

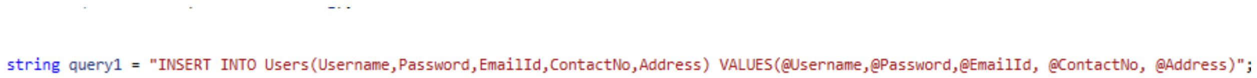
Registration Module and Account Management:
This module will help the Customer to get registered from anywhere if internet is present and enables the user to login to the system .This module will really simplify the task of on paper registration. Also after successful registration the user can update information and change their password as and when required.

User Management:
This module will help the administrator in enabling/disabling a user account and updating user information as required.

5.2 Function Prototypes

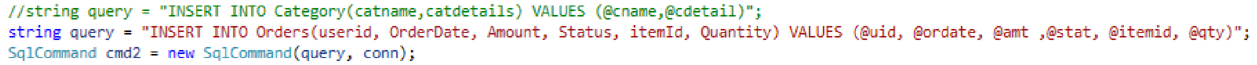
CRUD Operation

- Creating user



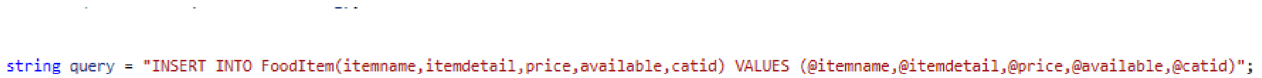
```
string query1 = "INSERT INTO Users(Username,Password,EmailId,ContactNo,Address) VALUES(@Username,@Password,@EmailId, @ContactNo, @Address)";
```

- Creating Order



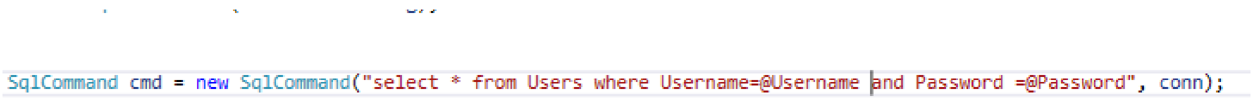
```
//string query = "INSERT INTO Category(catname,catdetails) VALUES (@cname,@cdetail)";
string query = "INSERT INTO Orders(userid, OrderDate, Amount, Status, itemId, Quantity) VALUES (@uid, @ordate, @amt ,@stat, @itemid, @qty)";
SqlCommand cmd2 = new SqlCommand(query, conn);
```

- Creating Food Item



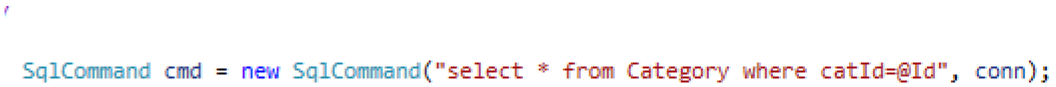
```
string query = "INSERT INTO FoodItem(itemname,itemdetail,price,available,catid) VALUES (@itemname,@itemdetail,@price,@available,@catid)";
```

- Reading User



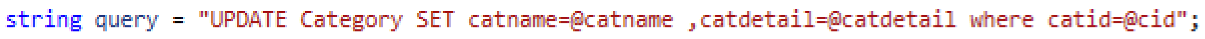
```
SqlCommand cmd = new SqlCommand("select * from Users where Username=@Username and Password =@Password", conn);
```

- Reading Category



```
SqlCommand cmd = new SqlCommand("select * from Category where catId=@Id", conn);
```

- Updating Category



```
string query = "UPDATE Category SET catname=@catname ,catdetail=@catdetail where catid=@cid";
```

6.Testing

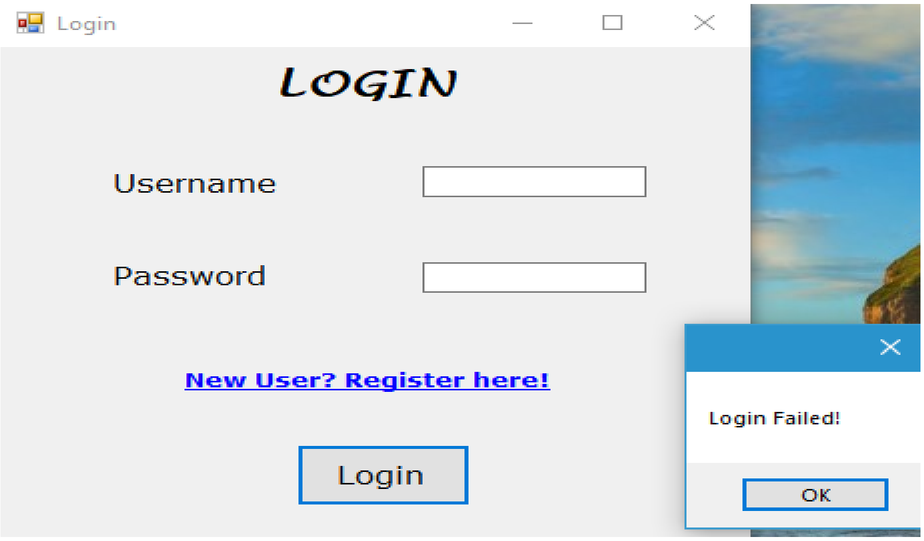
6.1 Testing Method

We have performed Black-Box Testing for the testing purpose.

6.2 Test Cases

For Login:

Username and password must be valid.

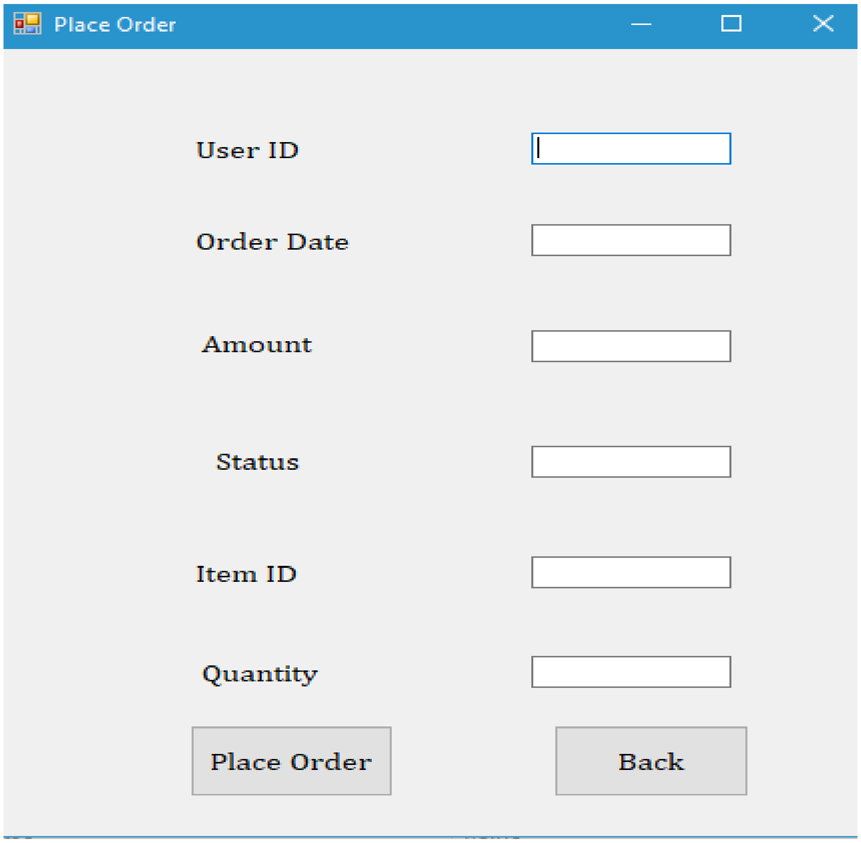


Login Failed!

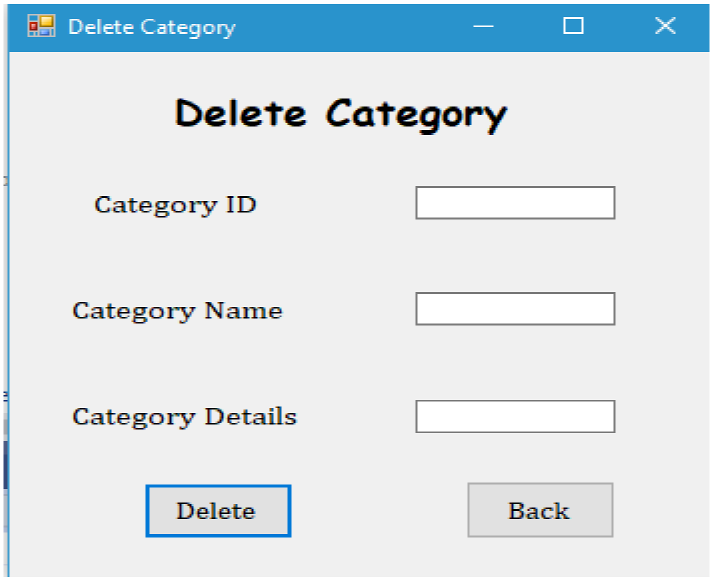
Input: Null

Output: Login Failed

If any required field is empty then process failed because of the authentication.

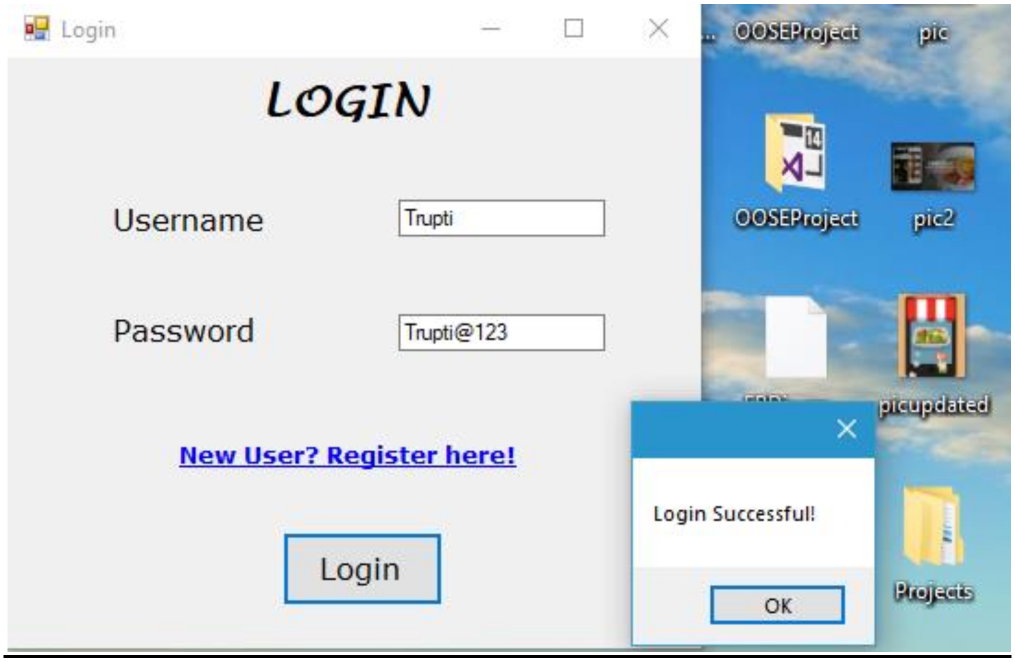


Here UserID and ItemID must be provided by the User.



Here CategoryID is Required.

7.Screenshots



Username Trupti

Password Trupti@123

Login Successful!

Figure 7.1 : User Login

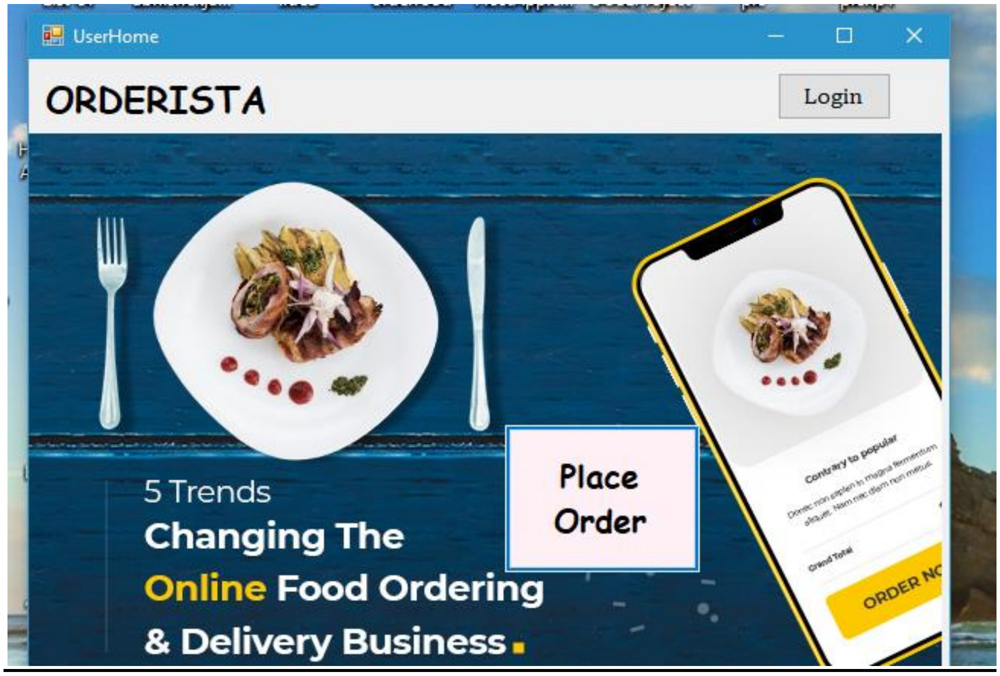
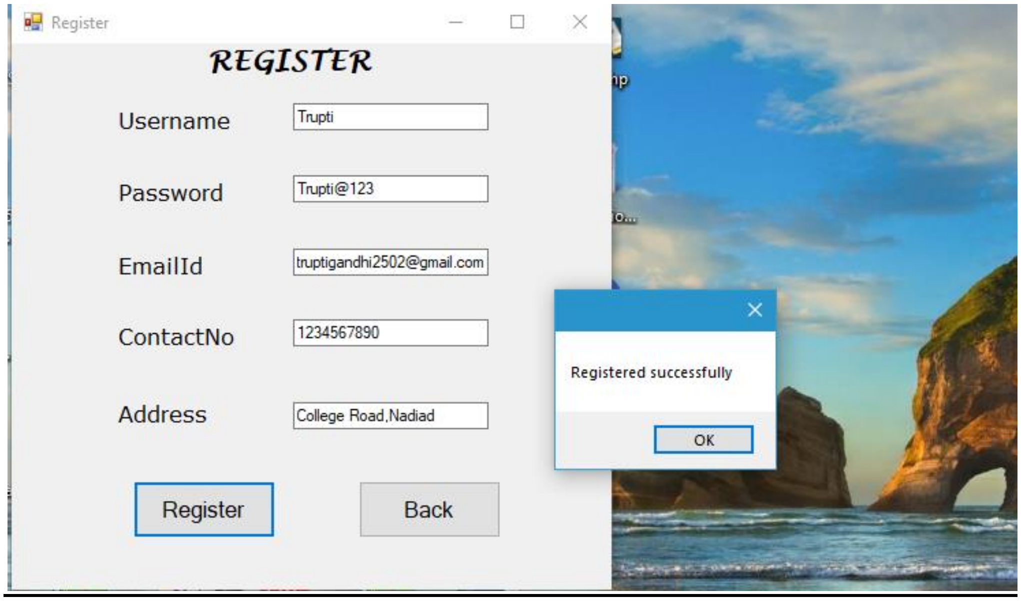


Figure 7.2 : User Home Page



REGISTER

Username: Trupti

Password: Trupti@123

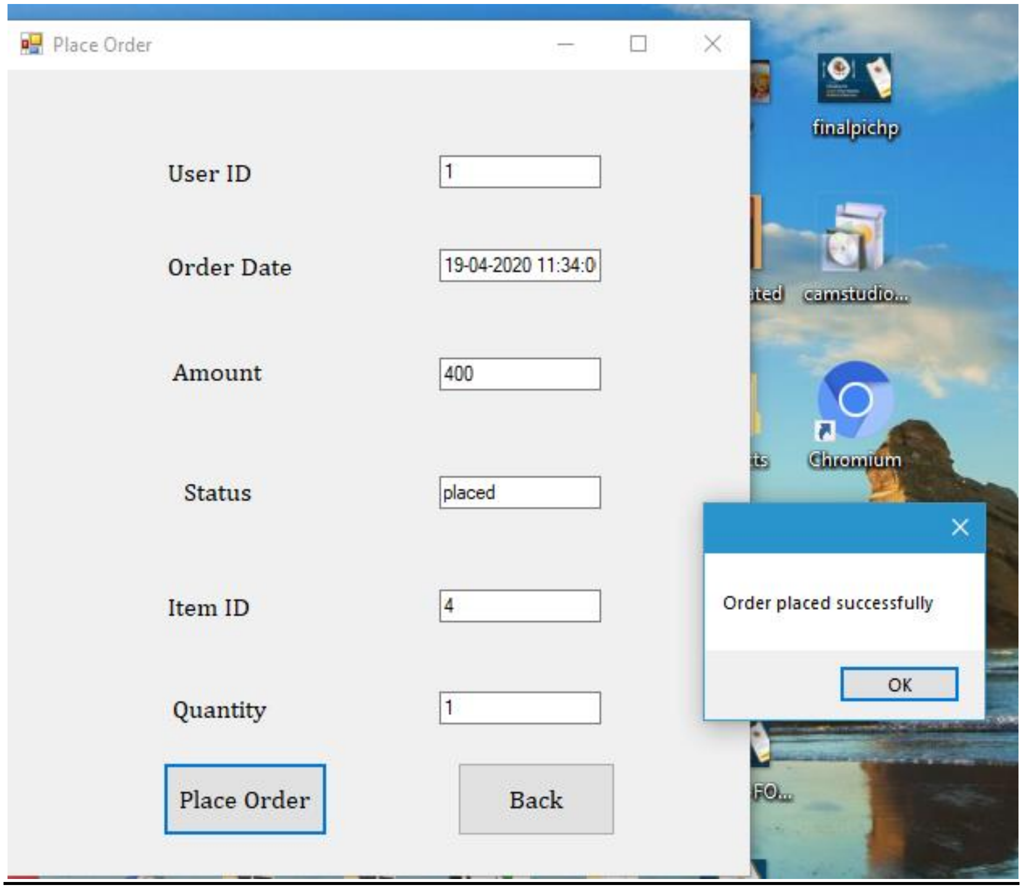
EmailId: truptigandhi2502@gmail.com

ContactNo: 1234567890

Address: College Road,Nadiad

Registered successfully

Figure 7.3 : User Registration



User ID: 1

Order Date: 19-04-2020 11:34:0

Amount: 400

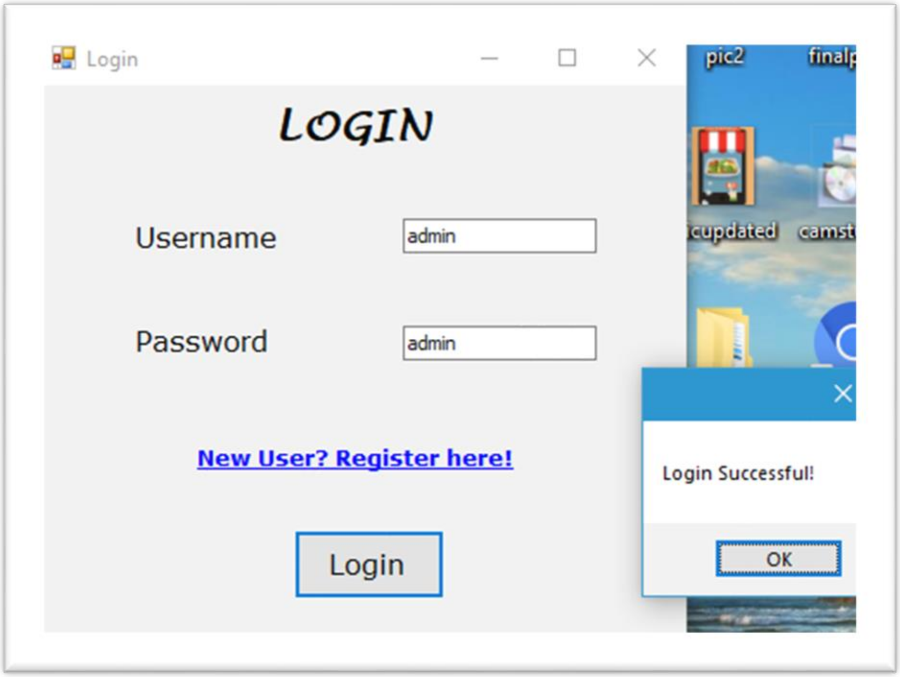
Status: placed

Item ID: 4

Quantity: 1

Order placed successfully

Figure 7.4 : User Page after Placing Order



Login Successful!

Figure 7.5 : Admin Login

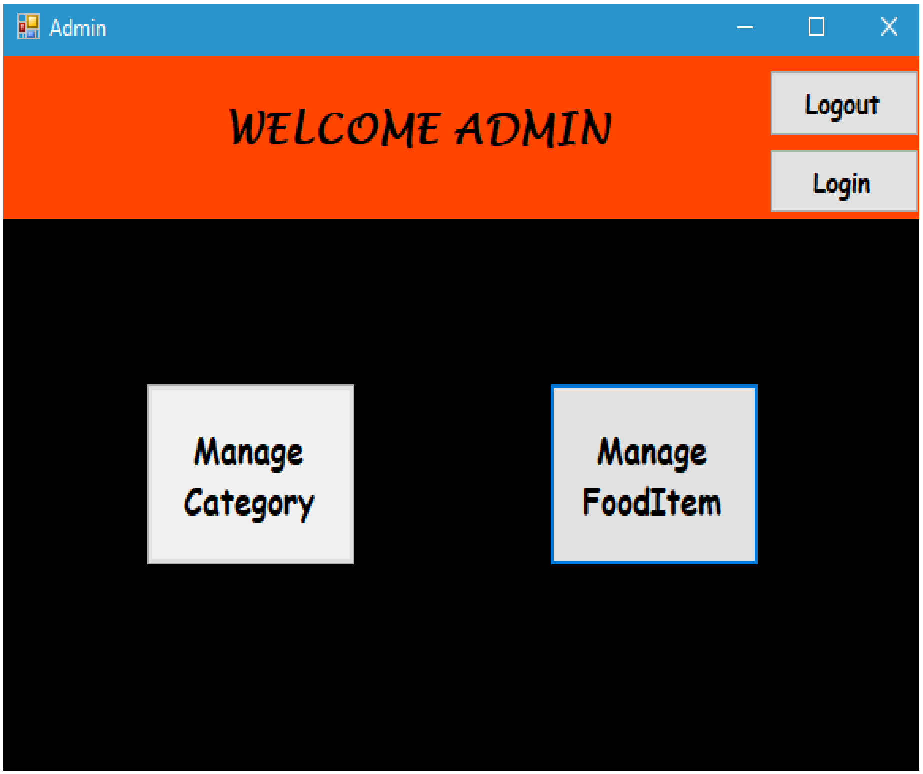


Figure 7.6 : Admin Home Page

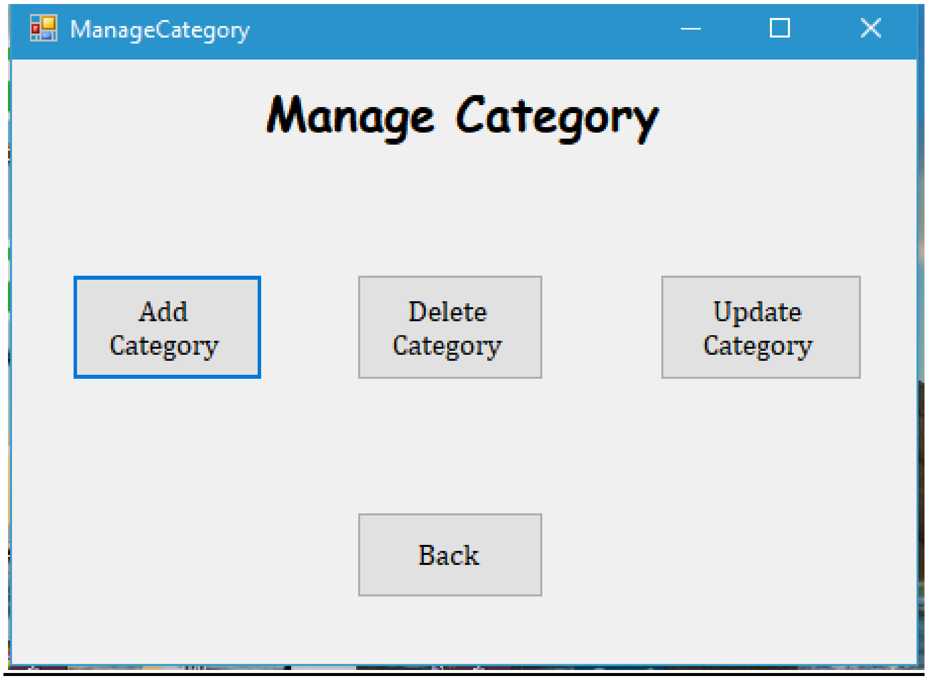


Figure 7.7 : Admin Manage Category Page

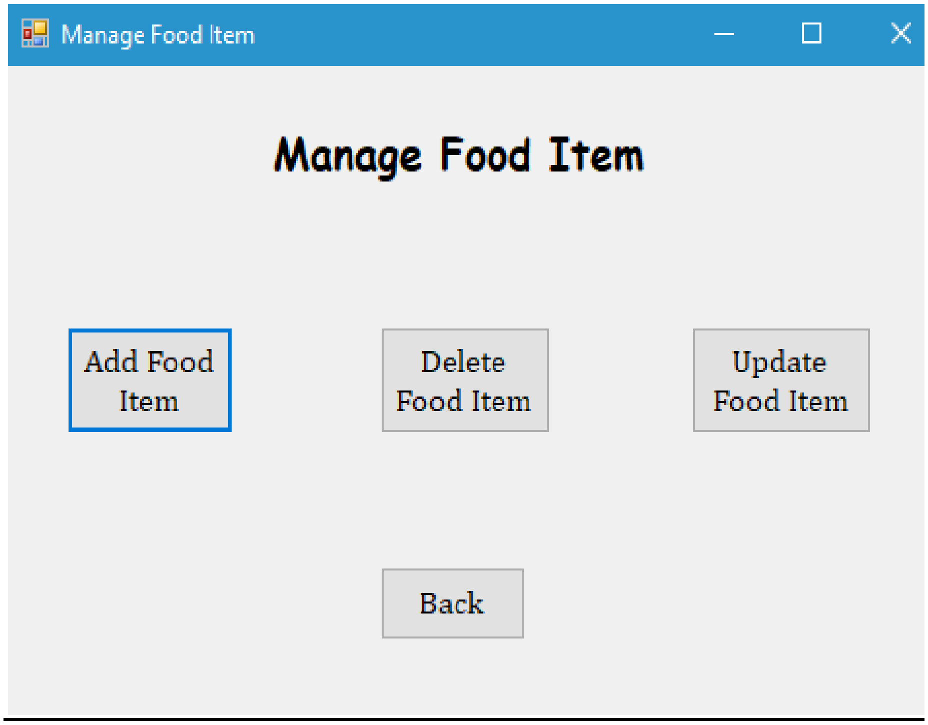
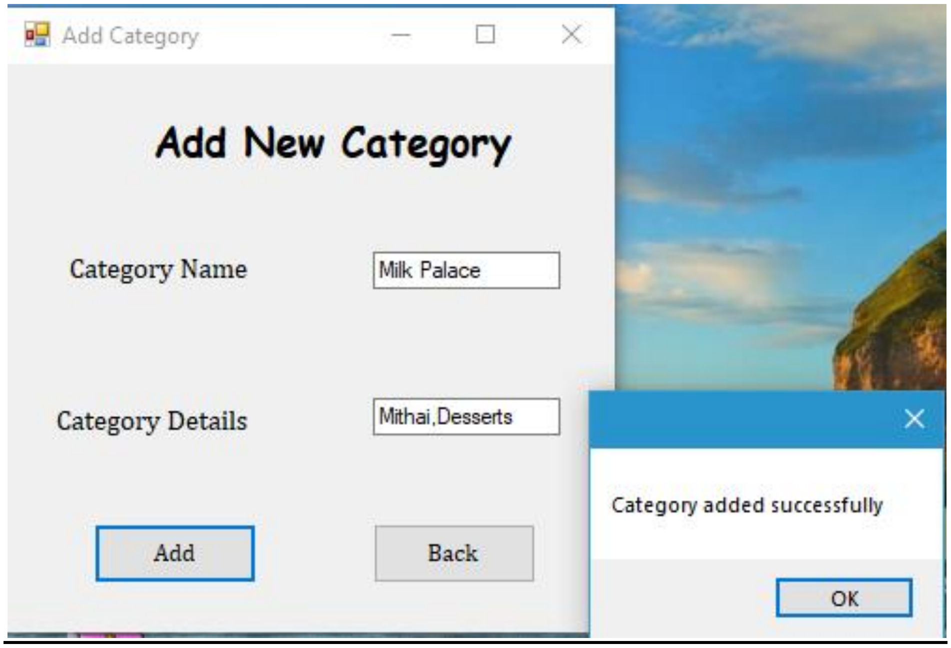


Figure 7.8 : Admin Mange Food Item Page



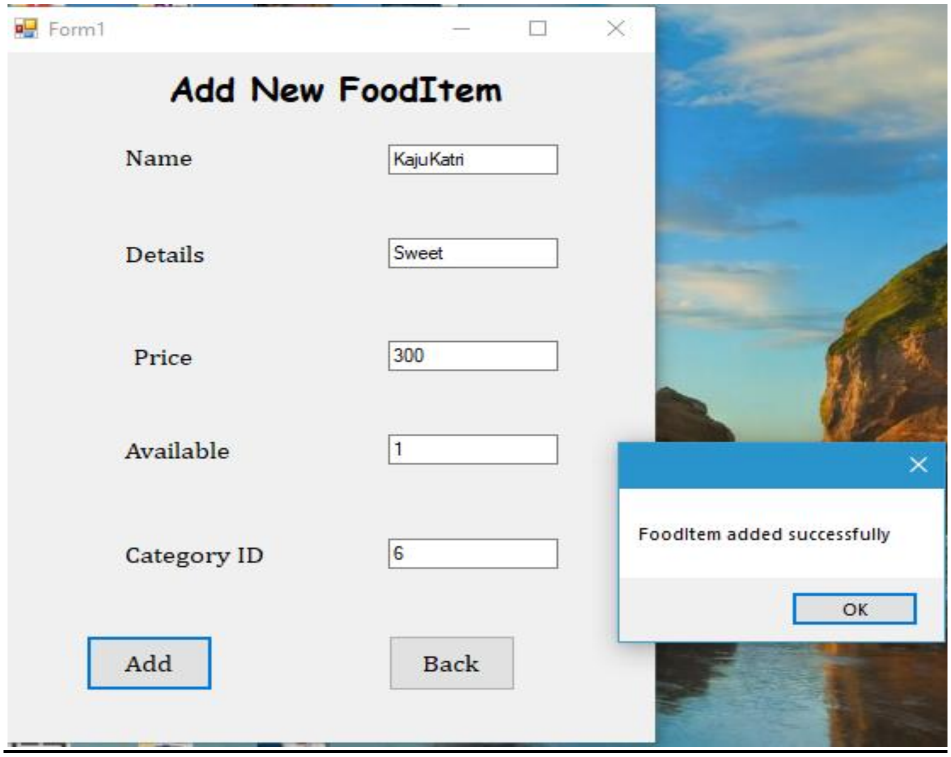
Add New Category

Category Name: Milk Palace

Category Details: Mithai,Desserts

Category added successfully

Figure 7.9 : Admin Add Category Page



Add New FoodItem

Name: KajuKatri

Details: Sweet

Price: 300

Available: 1

Category ID: 6

FoodItem added successfully

Figure 7.10 : Admin Add Food Item Page

8.Conclusion

The Functionality implemented in the system was done after understanding all the system modules according to the requirements

Functionalities that are Successfully implemented in the system are:

- User Registration containing all the necessary validation on field.
- Login
- User Authentication
- Logout
- Update Details
- Place an order
- Profile Details
- Crud Operation

Assumptions & dependencies

•Administrator is created in the system already.

•Roles and tasks are predefined.

After the implementation and coding of system, comprehensive testing was performed on the system to determine the loopholes and possible flow in the system.

9.Limitation and Future Extension

Limitations:

In this system we have assumed that the restaurants will have their own delivery person.

Order cannot be replaced.

There is only COD(Cash On Delivery) Payment method.

Functionality not Implemented:

Providing Offers.

Different Payment Methods.

Having our own delivery person.

Future Extension:

Using Geo-location for live tracking delivery person.

Using online payment methods.

Expanding our reach by providing service to more cities and including more restaurants.

10.Bibliography

Websites

1. https://www.tutorialspoint.com/uml/uml_overview.htm -for understanding UML
2. https://www.geeksforgeeks.org/software-engineering-object-oriented-life-cycle-model/- for understanding basic concepts
3. https://erdplus.com/ -to draw entity relationship diagram

Useful Links

1. https://www.umlet.com/ - to draw software diagram
2. https://www.zomato.com/anand?city_id=11342 - for reference
3. https://www.swiggy.com/ - for reference