**Online Food Ordering System**

**Statement of Problem:**

Developing a Rapid Food Order system to promote a greater count of food lovers to splurge into the field of Restaurant. It integrates the benefits of an ordering food with the convenience of an online excitement and going with the technology, minus the commuting hazards and expenses. It will usher in the immense flexibility and sophistication in the existing manual platform structures, with the perfect blend of synchronous and asynchronous interaction. It provides a means of collaborative E-ordering for the customers.

Generally, clients want to access updated menus including all food items, prices, and to receive instant order confirmation when they place an order. Moreover, customers need restaurants’ details such as postcode, since they preferentially order food from local restaurants because they have an idea of the food proposed by these restaurants and also because the ordered meals could be delivered within a reasonable wait time.

The project takes into account several issues regarding companie’s strategic business plans enabling customer “*to* *order takeaways in a smart way”*. Indeed, all restaurants have their own strategy including unique food items and origins meal deals allowing the customer to order more food at a lower price, and restaurant’s details including postcode, opening hours and areas for food delivery to make the order process quick and easy.

Hence, our system will need to represent the restaurants, menus, and prices in an organised manner. Indeed, among the different dishes of restaurants, some items are similar to each others; some have the same ingredients, or same prices. To solve these issues, the E-ordering system is required to provide meaningful concepts and relations to identify the knowledge and to represent it in a formal way.

Apart from the food management, the system is also going to integrate knowledge to enable the customer to place an order. Indeed, the system will provide some information to choose meal dishes such as rice, chicken, beef, mutton, burger, pizza, etc. by including concepts about the different ingredients and toppings.

The system will also consist of concepts related to restaurants’ contact details in order to facilitate the food delivery. Moreover, the user will be able to select the restaurants not only by postcode, but also by cities and addresses.

Software Requirements Specification

For

Online Food Ordering System

Version 1.0

By

131071045

Chetana Patel

T.Y.BTech (Sem VI)

VJTI, Mumbai

**Table Of Contents**

1. **Introduction** 
   1. Purpose
   2. Document Conventions
   3. Intended Audience and Reading Suggestions
   4. Product Scope
   5. References
2. **Overall Description** 
   1. Product Perspective
   2. Product Features
   3. User Classes and Characteristics
   4. Operating Environment
   5. Design and Implementation Constraints
   6. User Documentation
   7. Assumptions and Dependencies
3. **System Features and Functional Specification**
4. **External Interface Requirements** 
   1. User Interfaces
   2. Hardware Interfaces
   3. Software Interfaces
   4. Communications Interfaces
5. **Other Nonfunctional Requirements** 
   1. Performance Requirements
   2. Safety Requirements
   3. Security Requirements
   4. Software Quality Attributes
6. **Diagram**
   1. Use Case Diagram
   2. Class Diagram
   3. State Transition Diagram
   4. Interaction Diagram
7. **Design**
   1. Data Structure Design
   2. Algorithm Design
   3. Component Design
   4. Deployment Design
8. **Web Modeling**
   1. Hypertext Modeling
   2. Navigation Modeling
   3. Presentation Modeling

# Introduction

## Purpose

The purpose of the system is ordering food from a local restaurant or food cooperative through a web page. Much like ordering consumer goods online, many of these allow customers to keep accounts with them in order to make frequent ordering convenient.This document explains how online food ordering system can be implemented efficiently so that customers can easily contact the restaurants.

## Document Conventions

Throughout the document, the font used is Times New Roman and the font sizes are such that the topic is easily understood and followed as in the case of any conventional document practice. IEEE Standard Software Requirement Specification is used.

Title:

Font Style: Times New Roman

Size: 16

Heading:

Font Style: Times New Roman

Size: 16

Subheading:

Font Style: Times New Roman

Size: 14

Content:

Font Style: Times New Roman

Size: 12

## Intended Audience and Reading Suggestions

This document is intended for readers that include developers, project managers, users, students and researchers working in this domain. This SRS is categorized into five sections, viz. Introduction, Overall Description, System Features, External Interface Requirements, Other Non-functional Requirements and Other Requirements.

## Product Scope

The online food Ordering System will permit Process Impact customers to order food from the nearby restaurants online to be delivered to specified locations.The objectives associated with this system are:

1. To order food rapidly
2. To make it convenient for people who have limited time.
3. Cost reduction
4. Reduced paper work
5. Computerized Order and billing System

This system can be used by a registered customer. He/she can order food immediately.

## References

1. Standard IEEE 830 recommended practice for software requirements specification
2. Wiegers, Karl. *Process Impact Intranet Development Standard, Version 1.3*,

# Overall Description

## Product Perspective

This online application 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 cooking the food for you. By using this application the work of waiter is reduced and we can also say that the work is nullified. The benefit of this is that if there is rush in the Restaurant then there will be chances that the waiters will be unavailable and the users can directly order the food to the chief online by using application. The user will be given a username and a password, by signing in overtime user logs in he/she gets 2%discount in total bill and may the amazing coupons codes .

## Product Features

The website will provide the following features:

* Simple user-Interface Admin panel for creation and configuration for menu groups, menu items,etc.
* Built-in-facility to set modifiers on different menu items.
* Facility to create modifier groups, individual modifier items and assign items into different groups.
* Centralized Food Ordering System to control and supervise multiple restaurants centrally as well as individuals.
* Single and Individual Admin Panel and login for each restaurant.
* Custom Regional Settings for Date Format , Time Format, Currency and Languages.
* Online Payment processing for orders.
* Integration of multiple payment gateways including Credit cards,Paypal,etc.
* Discount feature to provide seasonal discounts on menu items.
* Coupons module for restaurant promotions
* Completely automated online ordering of food for a restaurant
* Unlimited orders
* Food ordering pages that look and feel exactly the same as the existing restaurant website.
* Food ordering pages hosted on secure and special server so no risk of customers getting redirected to servers where competitor’s website are listed.
* Developed using the latest website programming protocols for minimum server loads and ultra fast food loading and processing.

## User Classes and Characteristics

The users of this system will be the customers, the manager and the administrator. The characteristics of each of these users are mentioned below.

**Customer:**

The customer is a primary user of the system. A person has to login and can order food by seeing e-menu and can even do payment through payment gateway .

**Administrator:**

The administrator manages the working of the entire system. He can devise new plans and services. The admin alone has the responsibility to delete old and outdated plans.He can add new plans or update existing plans to the website.

**Manager:**

The staff work under the admin. The staff gives the order to the chief according to the received online orders. And when the order is ready order is set to deliver. The manager then updates the status of order and then delivered to given address.

## Operating Environment

Hardware Requirements:

* Processor: Pentium IV
* RAM: 512 MB or higher
* HDD: 1 GB or higher

Software Requirements:

* Technology: JavaScript, HTML,PHP
* Operating system : Windows XP or Windows 7

## Design and Implementation Constraints

Only a registered customer can use the system. For registering the customer needs to enter the mobile number and password in the login and password panel respectively. The customer then need to enter the location and area, so that the nearby restaurant can be viewed.Then,from e-menu customer can order food. Only the admin is allowed to modify the system database. He can add plans and services and display them on the website.

## User Documentation

The user will be provided with manual or user guide to understand how to use the system. These documents will be freely available on the website. The first time a new user accesses the system and on user demand thereafter, the system shall provide an online tutorial to allow users to practice ordering meals using a static e-menu.

## Assumptions and Dependencies

Various assumptions and dependencies of the system are described as follows:

* The most important assumption is that the restaurant is open for breakfast, lunch, and dinner every company business day in which employees are expected to be on site.
* The website will function in a standalone way when the user wants to browse through the services or subscribe them.
* The operation of the online food ordering system depends on changes being made in the Payroll System to accept payment requests for meals ordered with the system.
* The operation of the online system depends on changes being made in the Online food Inventory System to update the availability of food items as online food ordering accepts orders .

# System Features and Functional Specification

The system provides a number of features like placing a phone call, sending messages, quick messenger, subscribing for services and privacy, etc. The functional specifications are mentioned as follows:

|  |  |
| --- | --- |
| **Users** | **System features and functional specifications** |
| Customer | * Registers by entering mobile no. or email-id and password * Enters the location and area * Checks the e-menu * Orders the food in unlimited quantities * Enters the address * Proceeds for online payment |
| Administrator | * Launches new plans, services etc. * Monitors all systems and managers * Can delete old services and plans * Updates databases with respect to new services * Adds a new user to system database * Notifies user on updates |
| Manager | * Gives the orders to chief according to the First serve * Dispatches the order * Updates the status as order left |

**Functional Requirements**:

The system should support the following use cases:

|  |
| --- |
| Order.Place: The system shall let a Customer who is logged into the Online Food Ordering System place an order for one or more meals.  Order.Place.Register: The system shall confirm that the Customer is registered for payroll deduction to place an order.  Order.Place.Register.No If the Customer is not registered for payroll deduction, the system shall give the Customer options to register now and continue placing an order  Order.Cutoff: The Customer may either change the meal date or cancel the order. |
| Order.Deliver.Select: After the parcel is ready it is dispatched from the restaurant. |
| Order.Menu.Date: The system shall display a menu for the specified date.  Order.Menu.Available: The menu for the current date shall display only those food items for which at least one unit is available in the cafeteria’s inventory. |
| Order.Units.Food: The system shall allow the Customer to indicate the number of units of each menu item that he wishes to order.  Order.Units.Multiple: The system shall permit the user to order multiple identical meals, up to the fewest available units of any menu item in the order.  Order.Units.TooMany: If the Customer orders more units of a menu item than are presently in the Food inventory, the system shall inform the Customer of the maximum number of units of that food item that he can order.  Order.Units.Change: If the available inventory cannot fulfill the number of units ordered, the Customer may change the number of units ordered, change the number of identical meals being ordered, or cancel the meal order. |
| Order.Confirm.Display: When the Customer indicates that he does not wish to order any more food items, the system shall display the food items ordered, the individual food item prices, and the payment amount.  Order.Confirm.Prompt: The system shall prompt the Customer to confirm the meal order.  Order.Confirm.Not: If the Customer does not confirm the meal order, the Customer may either edit or cancel the order.  Order.Confirm.More: The system shall let the Customer order additional meals for the same or for different date. |
| Order.Pay.Method: When the Customer indicates that he is done placing orders, the system shall ask the user to select a payment method.  Order.Pay.Pickup: If the meal is to be picked up in the restaurant the system shall let the Customer choose to pay by payroll deduction or by paying cash at the time of pickup.  Order.Pay.Details: The system shall display the food items ordered, payment amount, payment method, and delivery instructions.  Order.Pay.Confirm: The Customer shall either confirm the order, request to edit the order, or request to cancel the order.  Order.Pay.Confirm.Deduct: If the Customer confirmed the order and selected payment by payroll deduction, the system shall issue a payment request to the Payroll System.  Order.Pay.Confirm.OK: If the payment request is accepted, the system shall display a message confirming acceptance of the order with the payroll deduction transaction number.  Order.Pay.Confirm.NG: If the payment request is rejected, the system shall display a message with the reason for the rejection. The Customer shall either cancel the order, or change the payment method to cash and request to pick up the order at the cafeteria. |
| Order.Done: When the Customer has confirmed the order, the system shall do the following as a single transaction:  Order.Done.Store Assign the next available meal order number to the meal and store the meal order with an initial status of “Accepted.”  Order.Done.Inventory: Send a message to the Online Food Ordering System with the number of units of each food item in the order.  Order.Done.Menu: Update the menu for the current order’s order date to reflect any items that are now out of stock in the cafeteria inventory.  Order.Done.Times: Update the remaining available delivery times for the date of this order.  Order.Done.Customer: Send an e-mail message to the Customer with the meal order and meal payment information.  Order.Done.Cafeteria: Send an e-mail message to the Restaurant Staff with the meal order information.  Order.Done.Failure: If any step of Order.Done fails, the system shall roll back the transaction and notify the user that the order was unsuccessful, along with the reason for failure. |

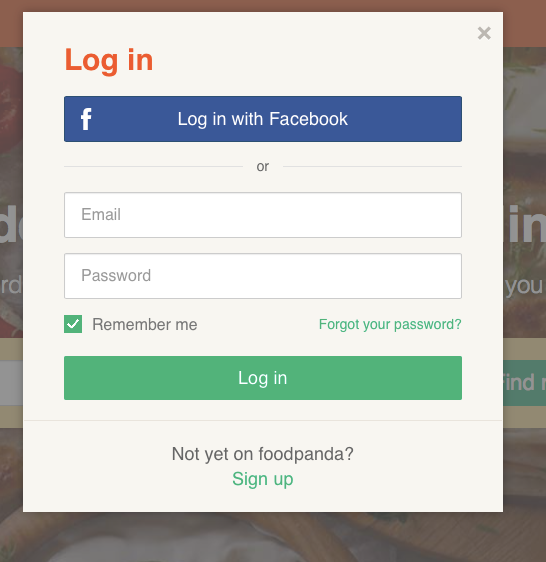
# External Interface Requirements

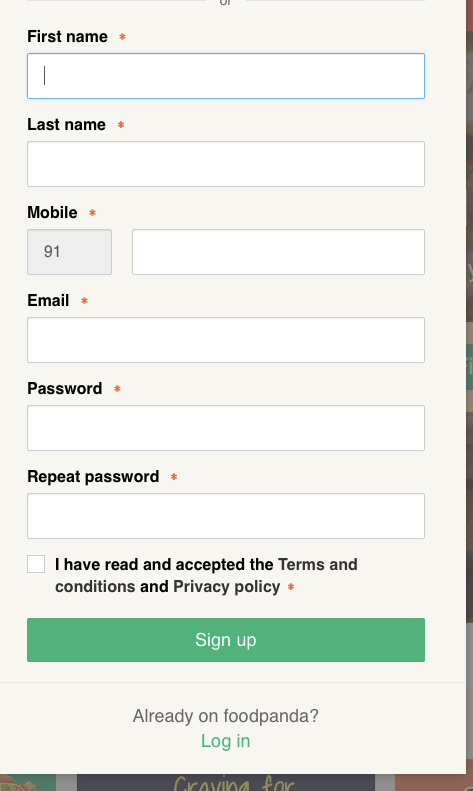
## User Interfaces

The website will provide the features of registration(logging in), exiting vouchers, searching for new food corners .

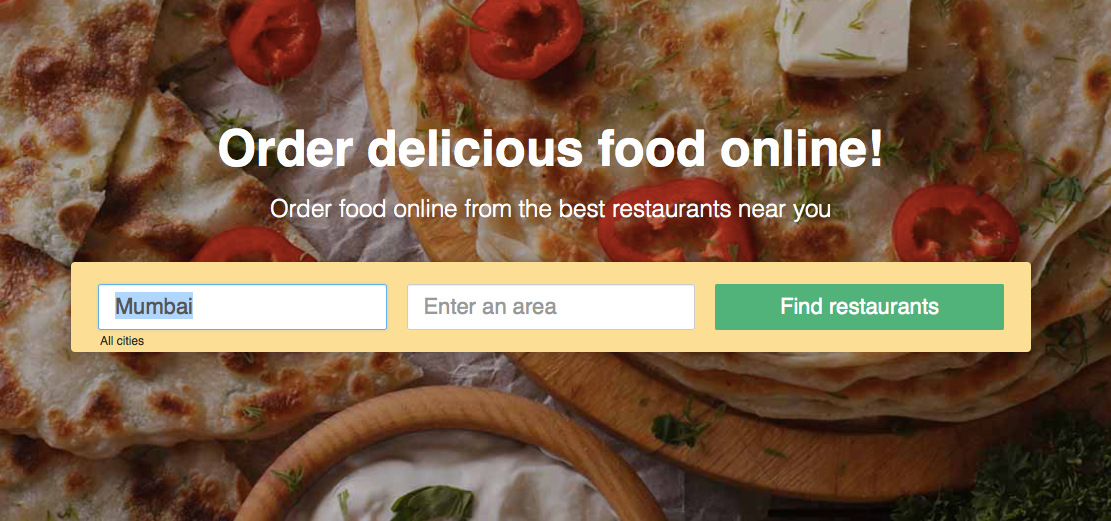
1. The initial user interface is making an account
2. Selecting the location and area
3. Ordering the food
4. Payment

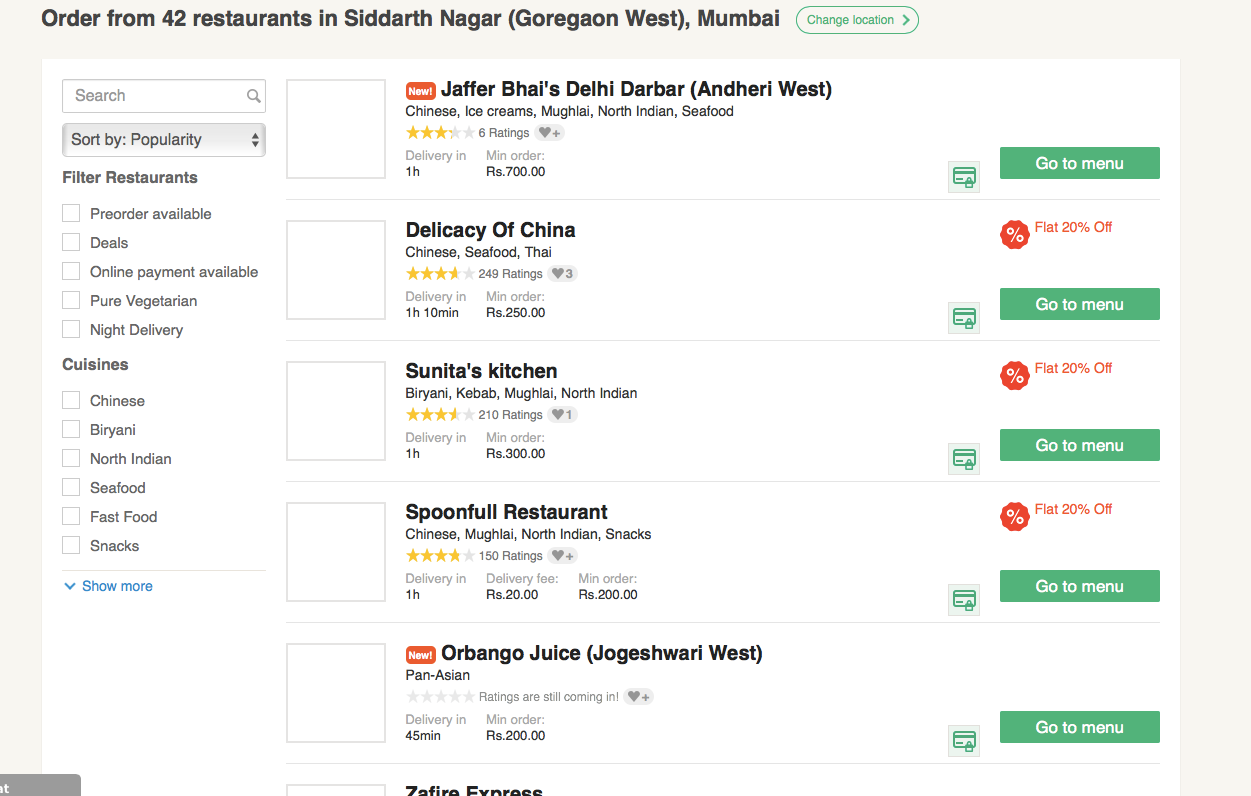
**LOG IN:** If person already have an account then continue logging in

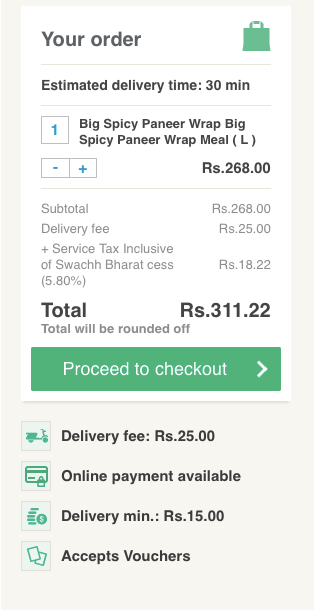


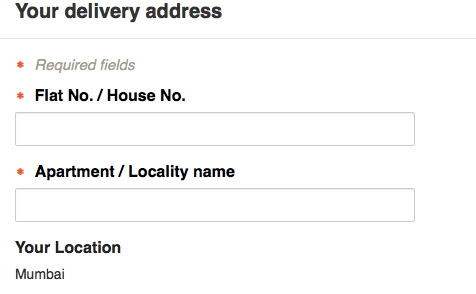


**SIGN UP:** If person does not have an account then make an account by entering the details.

**ADDING LOCATIONS:** Add location and areas so that nearby restaurants can be found

**AVAILABLE FOOD CORNERS:** According to the added location and area the nearby restaurants are displayed along with the menu.

**ORDER THE FOOD ITEM:** From the menu, add the food items .****

**PROCEED TO CHECKOUT:** Now, enter the delivery address.

**PAYMENT GATEWAY:** Proceed for payment i.e. either online or while delivery time

## Hardware Interfaces

The hardware interfaces used by this systems are :laptop/desktop/mobile for accessing the website.Server for providing website and database services ,a printer and scanner for documents.

## Software Interfaces

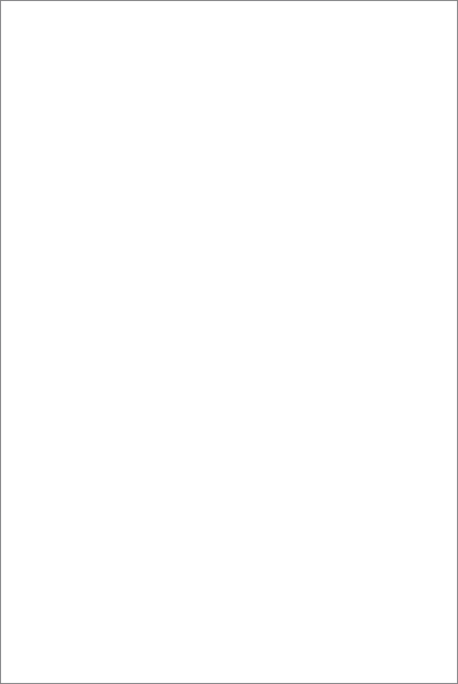
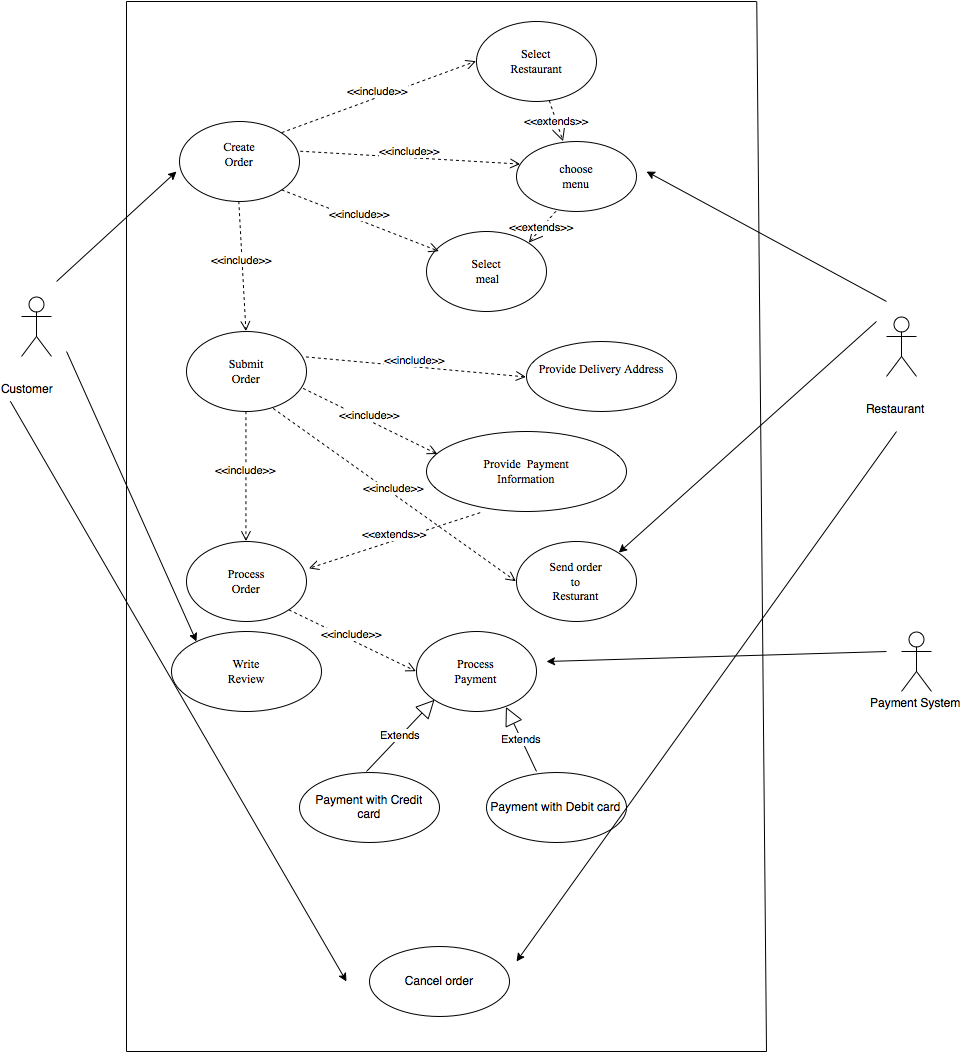
Software requirement of the system are very nominal. The system does not require any additional except JAVA and hence is economically feasible. JAVA is freely available and s platform independent which makes the system user friendly.

## Communications Interfaces

The website is accessed using the web browser which uses the HTTP protocol.

* The Online Food Ordering System shall send an e-mail message to the Customer to confirm acceptance of an order, price, and delivery instructions.
* The Online Food Ordering System shall send an e-mail message to the Customers to report any problems with the meal order or delivery after the order is accepted.

# Other Nonfunctional Requirements



## Performance Requirements

* + The system should have a quick response time. In other words the application should be submitted immediately. The system should have nominal delay for voice . The updates done to the database should quickly be reflected in the functioning of the network. For e.g. when a student apply for concession the database should be updated immediately. Web pages should be accessed immediately when the link is clicked. Delay should be tolerable.

## Safety Requirements

Uninterrupted internet connectivity is expected to be there.

## Security Requirements

The user account and system database should be secured from tampering and should be accessible only to the admin. Anti spoofing is to be implemented as mentioned before. The security requirements for the system are as follows:

* + **Access Control and authentication :** Authentication of user is necessary and login and password should be kept secret
  + **Confidentiality :**All user passwords is required to be encrypted. Being transmitted wirelessly through the air medium, any attacker can easily capture this data and eavesdrop. Thus encryption is required to prevent any intrusion.

## Software Quality Attributes

The quality of the system is maintained in such a manner that it would be user friendly and efficient. The following attributes are to be considered :

* + **Availability** : The availability of internet is the centre point for assurance of quality.
  + **Portability :** The system is platform independent hence will be available from any platform.
  + **Usability :** The system should be user friendly, and easy to use.

**USE CASE DIAGRAM**

**Use case Description:**

1. **Login**

**Description:** The user login’s into online food ordering system by providing username and password.

**Pre-condition:** A registered user.

**Post-condition:** The user has been authorised to order food.

**Algorithm:**

1. Enter the username and password
2. Press login
3. If(username and password are valid)
4. Enter the user’s account
5. else
6. Invalid username or Password
7. End if
8. **Create order**

**Description:** The user can select restaurant by entering location name.

**Pre-condition:** The restaurant list is sorted according to the location entered.

**Post-condition:** There are restaurant present in the given location or no restaurant.

**Algorithm:**

1. Enter the location from drop downcast where user wants to order food.
2. List of restaurant are displayed
3. Select the restaurant
4. Choose the menu according to the cuisine
5. Select and add the meal
6. Create order
7. **Submit order**

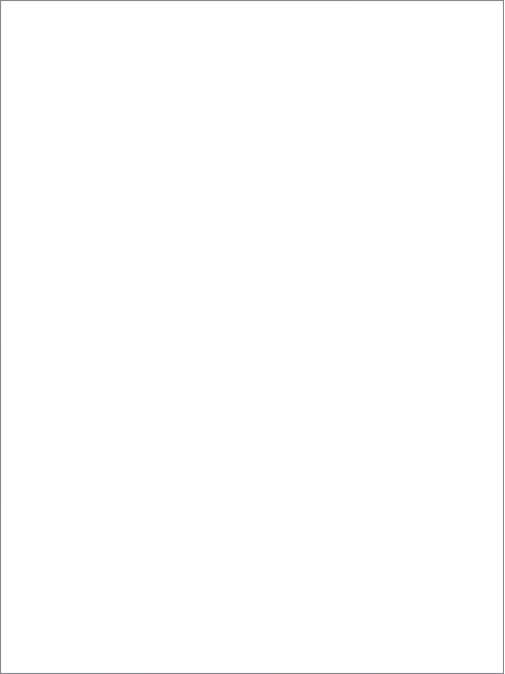
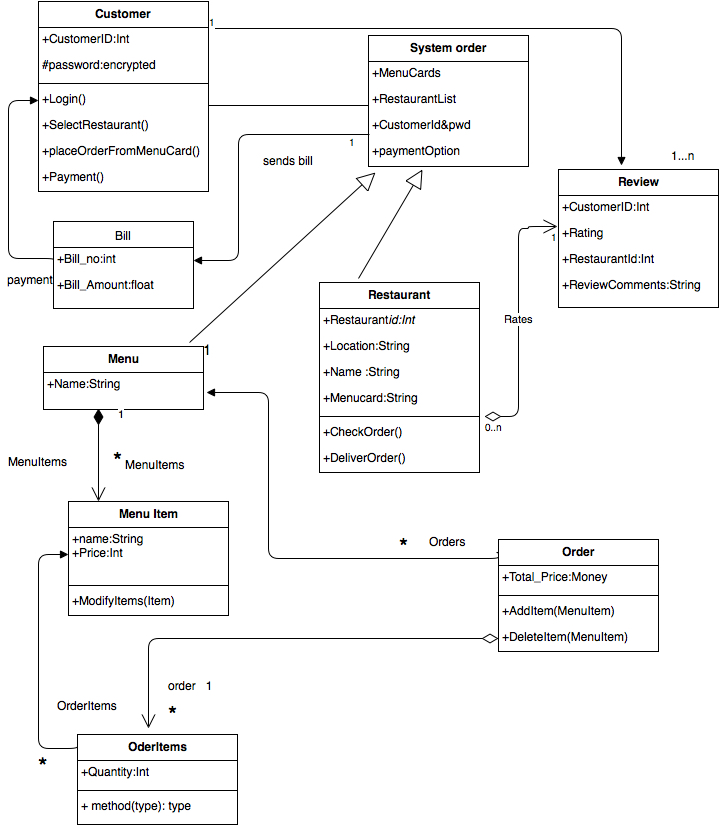
**Description:** The user add’s the order in the cart

**Pre-condition:** The list of meal is shown to the user

**Post-condition:** The order is submitted to the restaurant if cart is not empty.

**Algorithm:**

1. Click the next button after completing Create order
2. The list of ordered item is displayed.
3. If(there are items present in the cart)



1. Click Submit order
2. Then ordered is sent to the selected Restaurant entered by user
3. else
4. No items in the list order cannot be submitted.
5. End if
6. **Make Payment**

**Description:** User makes the payment after submitting the order.

**Pre-condition:** User successfully is entered to the payment page

**Post-condition:** Transaction completed and bill is generated

**Algorithm:**

1. Enter the mode of payment by debit card or credit card
2. Enter the card details
3. If( card details are valid)
4. Payment successful and bill is generated
5. else
6. Invalid information
7. End if
8. **Cancel Oder**

**Description:** User cancel’s the order

**Pre-condition:** User entered into the cancel order page after login to it’s account.

**Post-condition:** Order cancel or not canceled

**Algorithm:**

1. Enter the user name and password
2. If(user name and password are valid)
3. Displays the user’s page
4. Click the cancel order button
5. If( order is not dispatched )
6. Order canceled and money refunded after cutting tax.
7. Else
8. Display “Order cannot be canceled as already time out”
9. else
10. Invalid information.
11. End if

**CLASS DIAGRAM**

**Class diagram Description:**

1. **Identify the classes and their types**

* Customer: Entity
* System: Controls
* Restaurant: Entity
* Menu:Entity
* Menu Item: Entity
* Order : Entity
* Order Items: Entity
* Review: Entity
* Bill: Entity

1. **Identifying association between the classes**

* Customer orders from System orders
* Customer Gives Reviews
* System Givers order to Restaurant
* Customer Orders from Menu
* Menu contains MenuItems
* OderItem contains order from Customer from MenuItems
* OderItems send list to Order
* Order sends list to Restaurant

1. **Identifying attributes and operation of the classes**

**Class name:** Customer

**Attributes:**

* Customer\_id:int
* Customer\_pwd:encrypted

**Methods:**

* Login()
* LogOut()
* Select Restaurant()
* placeOrderfromMenu()
* Payment()

**Class name:** System

**Methode:**

* Menu()
* RestaurantList()
* Customerid&pwd()
* PaymentOption()

**Class name:** Restaurant

**Attributes:**

* Restaurant\_id:Int
* Location:String
* Name:String
* MenuCard:String

**Methods:**

* CheckOrder()
* DeliverOrder()

**Class name:** Menu

**Attributes:**

* Name:String

**Class name:** MenuItem

**Attributes:**

* Name:String
* Price:Int

**Methods:**

* ModifyItems()

**Class name:** Order

**Attributes:**

* Total\_price:Float

**Methods:**

* OrderItem(name)
* DeleteItem(name)

**Class name:** OrderItem

**Attributes:**

* Quantity:varchar(20)

**Methods:**

* method(type)

**Class name:** Review

**Attributes:**

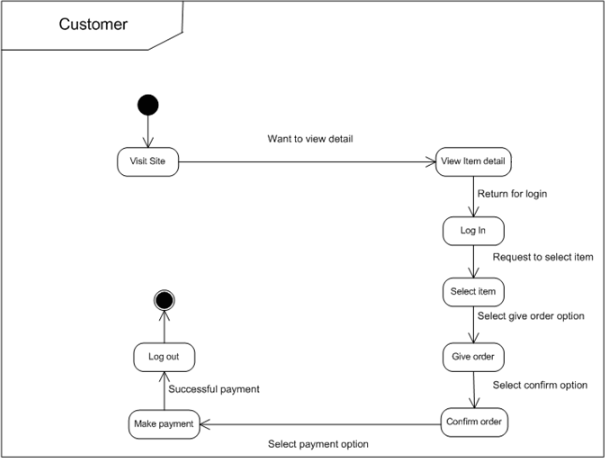
* Customer\_id:varchar(10)
* Rating:Int
* Restaurant\_id:Int
* ReviewComments:varchar(30)

**Class name:** Bill

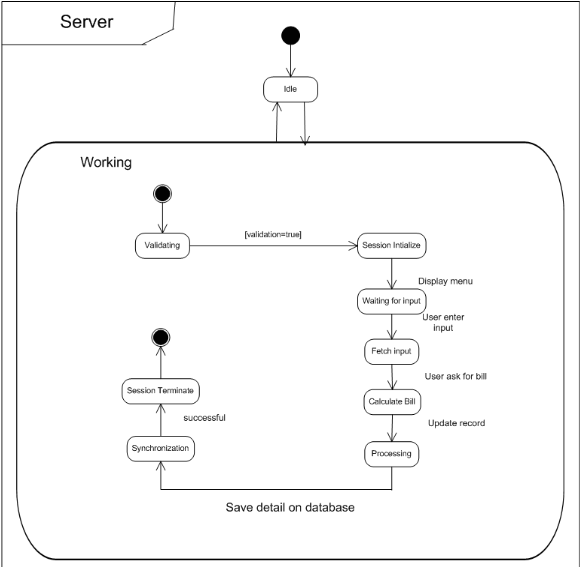
**Attributes:**

* Bill\_no:Int()
* Bill\_amount:Float

**STATE TRANSITION DIAGRAM**

**State Transition Diagram For Customer:**

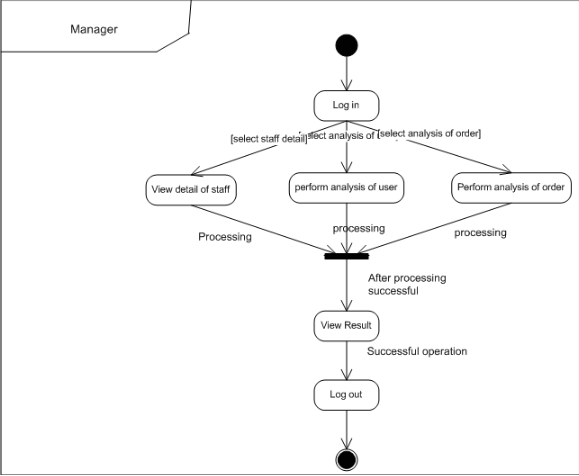
**Description:**

Customer visit’s the website.Initially customer can view the website and then proceed for registration . Customer has created customer\_id and password. Then customer login’s into the website by entering customer\_id and password. Customer Enters the location and Restaurant list is displayed accordingly. A Restaurant is selected and then customer selects the meal from the menu and places the order.When the order is confirmed customer is proceeded for the payment. After all the states are completed the last state is executed log out.

**State Transition Diagram For Server(System)**

**Description:**

The server is the system in class diagram running the website.When the customer enters the website Server works start here. Firstly,validation of the customer details are checked. After validation is true the server grants the permit to the customer and waits for the completion of activity of customer. After receiving the order the bill is calculated and sended to customer and records are updated. After receiving the payment the Session is completed.

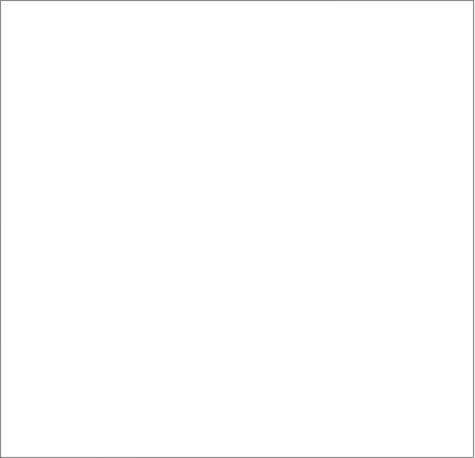
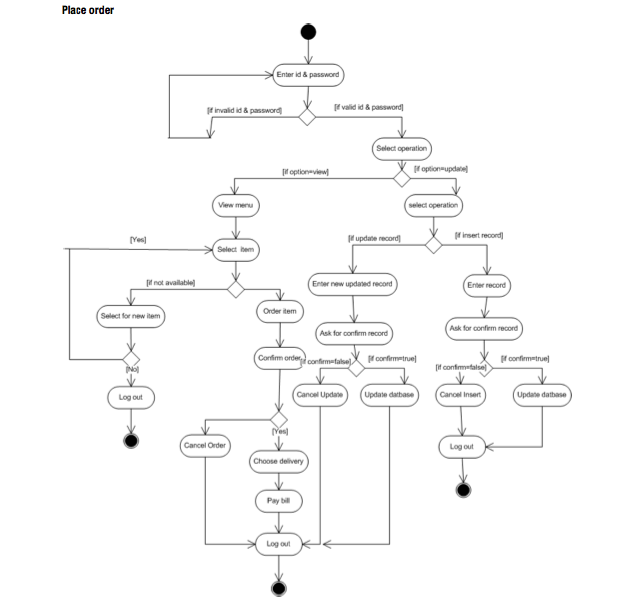
**State Transition Diagram For Manager(Restaurant)**

**Description:**

Restaurant works as manager . Manager login’s into its account. Checks if any order, If manager receives order from customer then the order is passed to chief and order is made ready .Once the processing is completed the Order is dispatched and updated to the customer about the status of order.After the delivery of order to the customer the manager log out from the corresponding customer and receives other customer’s order.

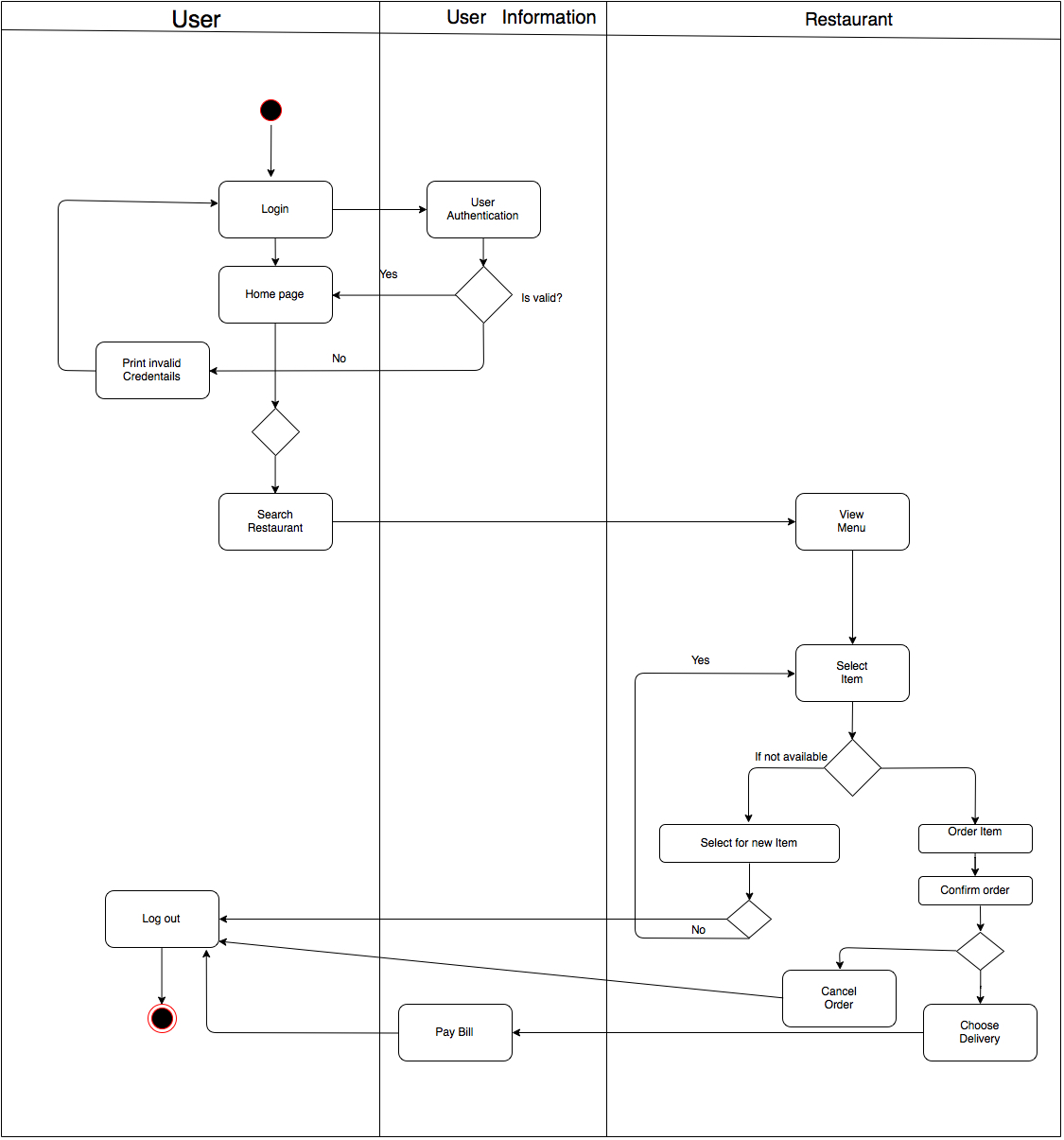
**ACTIVITY DIAGRAM**

**Activity diagram for Online Food Ordering System :**



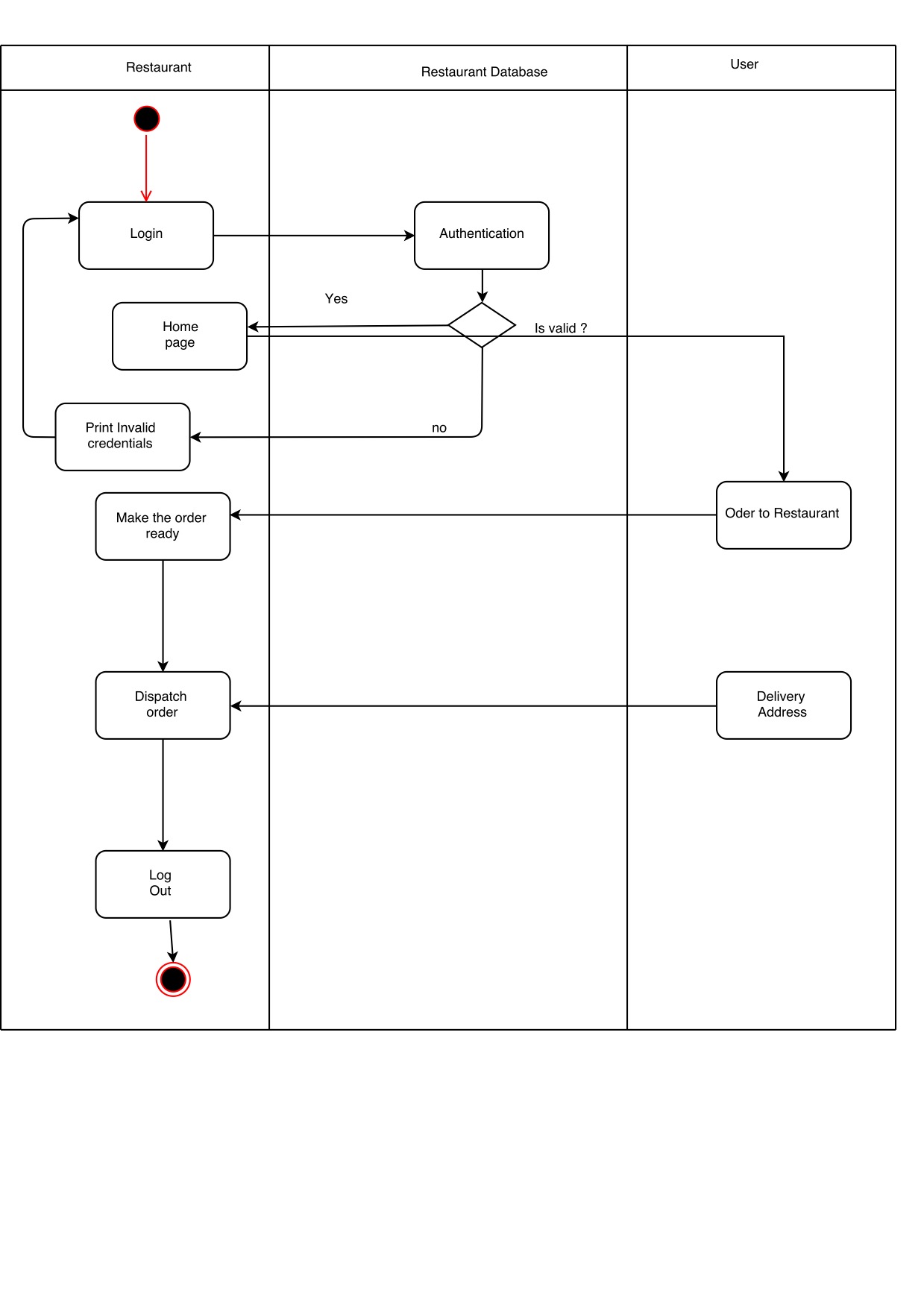
**Description:**

The above activity diagram is the defined for whole system.Initially, the activity starts with login state where password need to enter details. and then there are three end state i.e. logout based on the conditions of the system.

**Activity Diagram For User:**

**Description:**

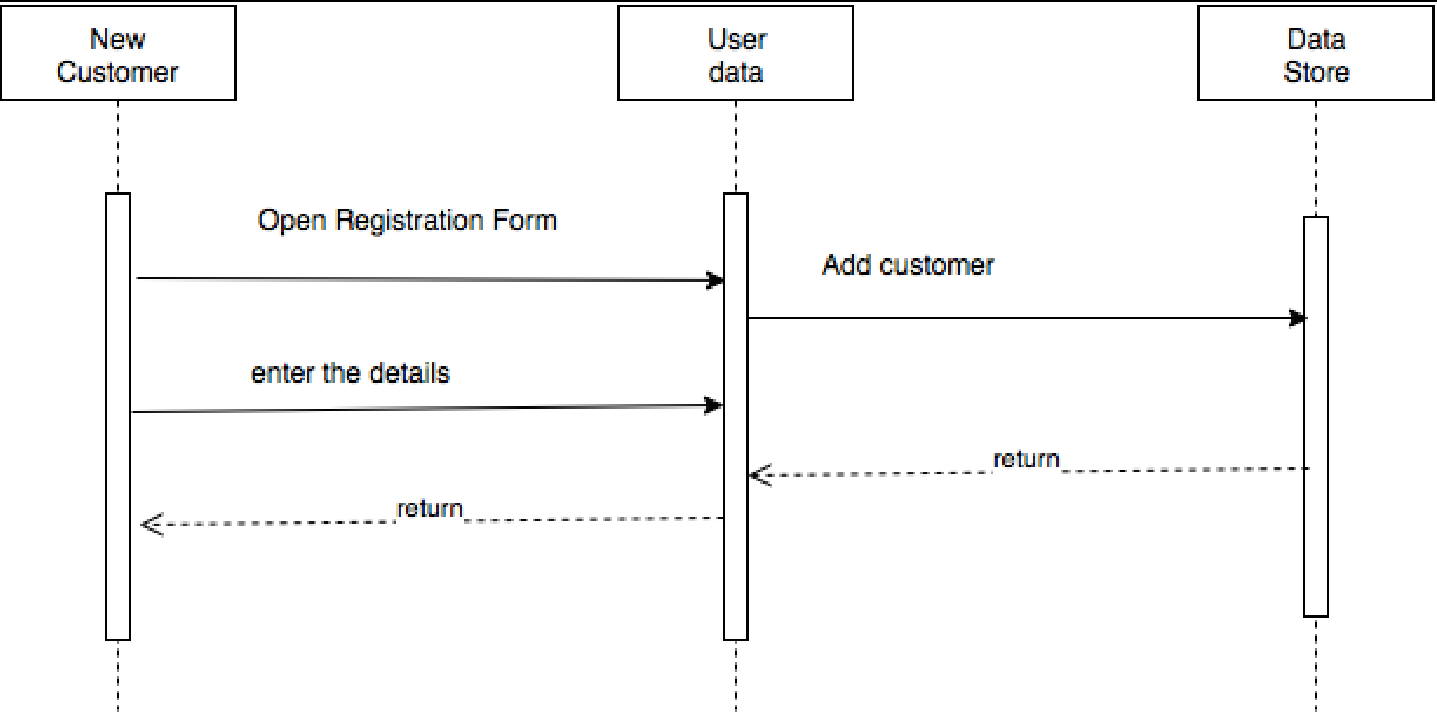
The initial state is login then after confirming the user details it enters the system.Were there are states such as view menu,search restaurant,select item and cancel orderIn select item if the item is available then it is forwarded further flow and if not available then select new menu. After all, the ordering the state is paybill and end state log out.

**Activity Diagram For Restaurant:**

**Description:**

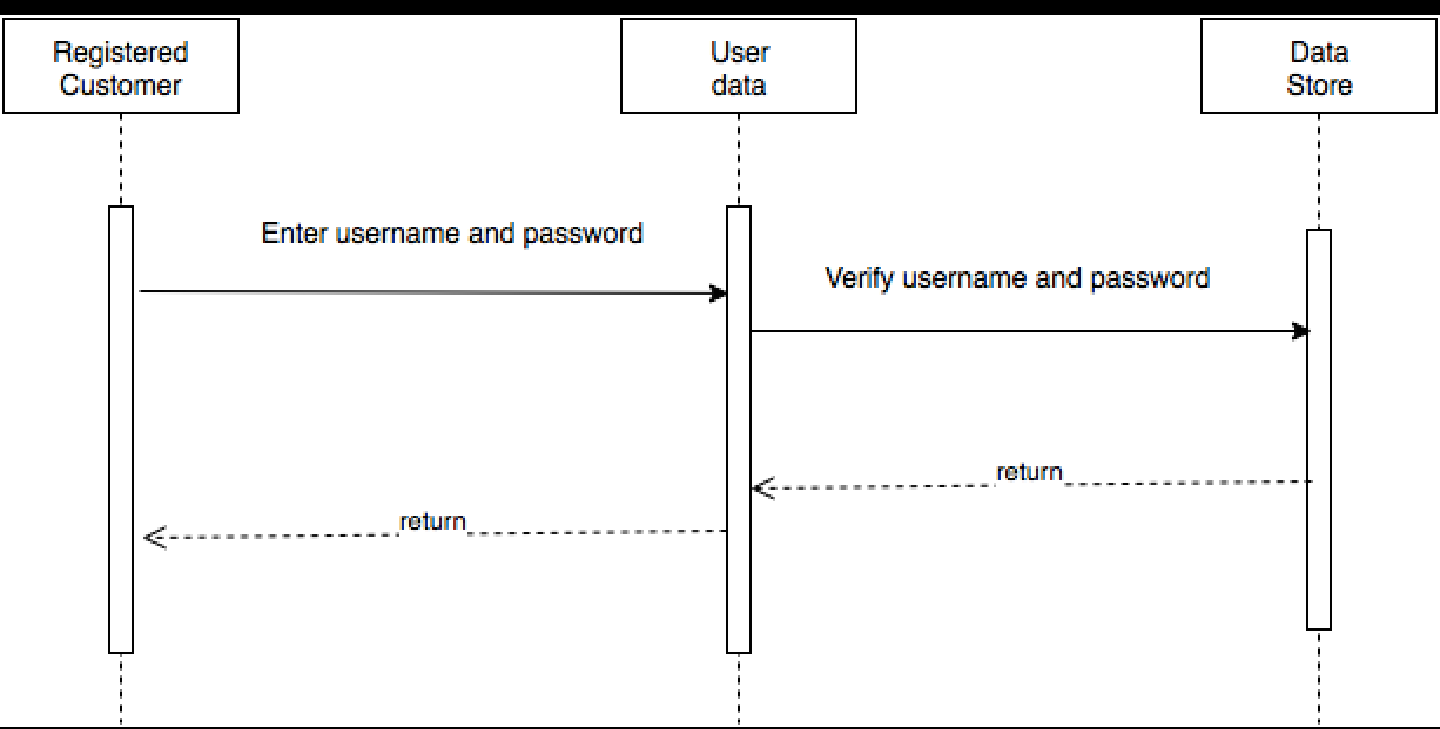
Initial and final state are same as the customer and intermediate states make order ready,dispatch order and then deliver order.

**INTERACTION DIAGRAM**

**Interaction for Sign up:**

**Description:**

The initial very first state of the customer registration where the details are filled and then is registered by the system.

**Interaction Diagram for Login:**

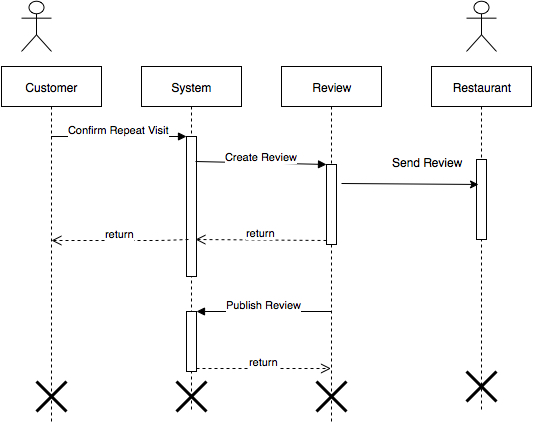
**Description:**

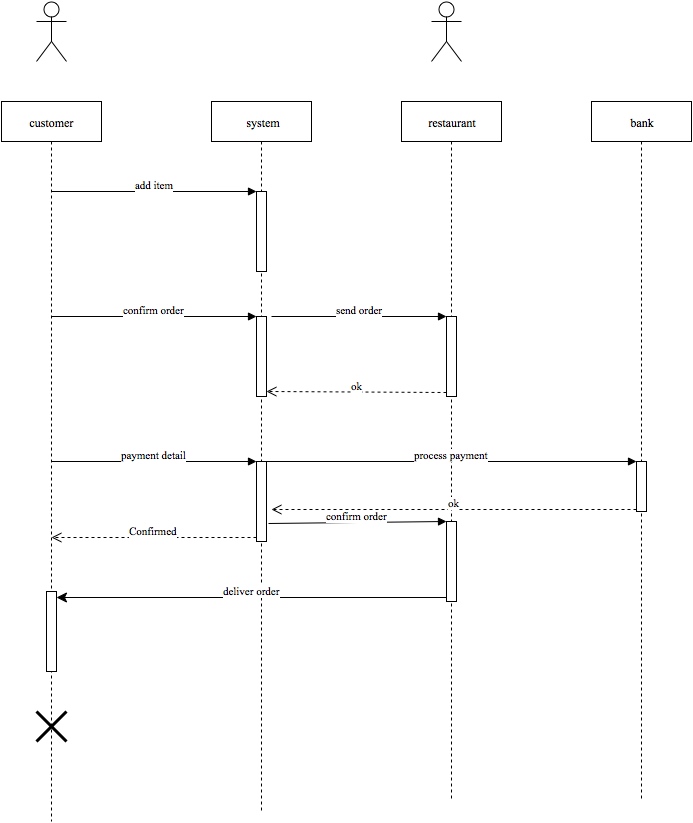
The Interaction diagram for login is shown above:

The customer enters the username and password and are verified by user authentication. If valid the access is granted or denied.

**Interaction Diagram for Create Order:**

**Description:**

The customer adds the items to the cart from the menu and then after confirm order the order is sent to Restaurant. For the payment the payment details are provided by customer and the payment is processed and finally order is delivered.

**Interaction Diagram For Review:**

**Description:**

Customer can give its review after create order and the reviews are published on the website.

**DATA STRUCTURE DESIGN**

Create Table **Customer**

( Customer\_id INT,

Password VARCHAR(20),

PRIMARY KEY(Customer id)

)

Create Table **SystemOrder**

(MenuCards VARCHAR(100),

RestaurantList VARCHAR(100),

FOREIGN KEY(CUSTOMER ID)

)

Create Table **Restaurant**

( Restaurant\_id INT,

LocationVARCHAR(20),

Name VARCHAR(20),

FOREIGN KEY(MenuCard)

PRIMARY KEY(Customer id)

)

Create Table **MenuItem**

(Name VARCHAR(10),

Price INT(10)

)

Create Table **OrderItem**

(Quantity INT

)

Create Table **Order**

(Total\_Price INT

)Create Table **Review**

(

Rating INT,

ReviewComments VARCHAR(50),

FOREIGN KEY(Restaurant\_id),

FOREIGN KEY(Customer\_id),

)

Create Table **BILL**  
(

Bill\_id INT,

Bill\_Amount VARCHAR(10),

PRIMARY KEY(Customer\_id)

)

**ALGORITHM DESIGN:**

1. Enter Home Page
2. If(Registred)

LOGIN:

1. Enter the username and password
2. Press login
3. If(username and password are valid)
4. Enter the user’s account
5. else
6. Invalid username or Password
7. End if

3.else

REGISTER

1.Enter the details

2.Username and password generated.

3.Go to Step 1

4.CREATE ORDER:

1. Enter the location from drop downcast where user wants to order food.
2. List of restaurant are displayed
3. Select the restaurant
4. Choose the menu according to the cuisine
5. Select and add the meal
6. Create order

5.SUBMIT ORDER

1. Click the next button after completing Create order
2. The list of ordered item is displayed.
3. If(there are items present in the cart)
4. Click Submit order
5. Then ordered is sent to the selected Restaurant entered by user
6. else
7. No items in the list order cannot be submitted.
8. End if

6.MAKE PAYMENT

1. Enter the mode of payment by debit card or credit card
2. Enter the card details
3. If( card details are valid)
4. Payment successful and bill is generated
5. else
6. Invalid information
7. End if

**7.I**f (CANCEL ORDER)

1. Enter the user name and password
2. If(user name and password are valid)
3. Displays the user’s page
4. Click the cancel order button
5. If( order is not dispatched )
6. Order canceled and money refunded after cutting tax.
7. Else
8. Display “Order cannot be canceled as already time out”
9. else
10. Invalid information.
11. End if

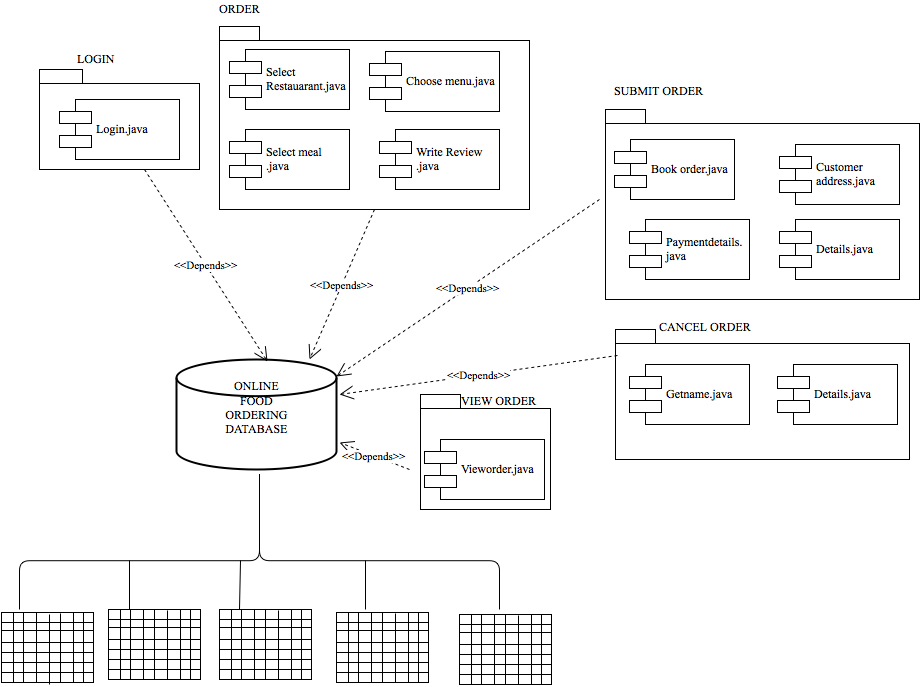
8.else

Logout

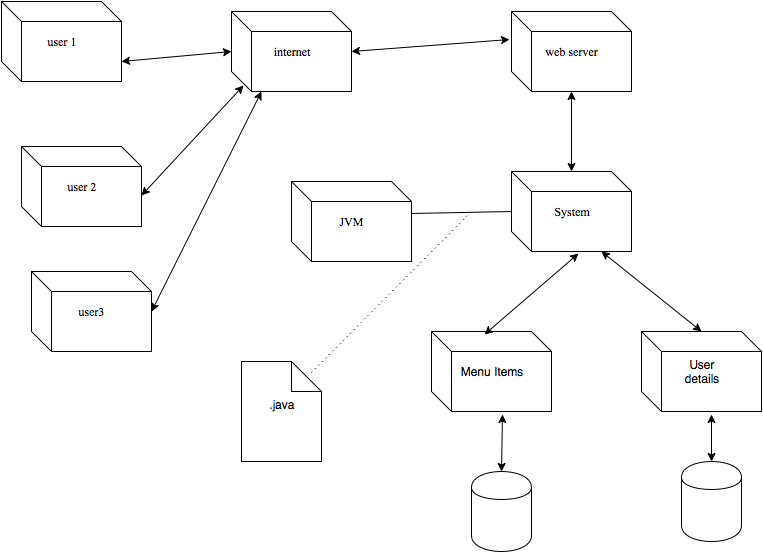
**COMPONENT DIAGRAM**

A component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Component diagrams are different in terms of nature and behaviour. Component diagrams are used to model physical aspects of a system. It does not describe the functionality of the system but it describes the components used to make those functionalities. So from that point component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

****

**DEPLOYMENT DIAGRAM**

The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

The purpose of deployment diagrams can be described as:

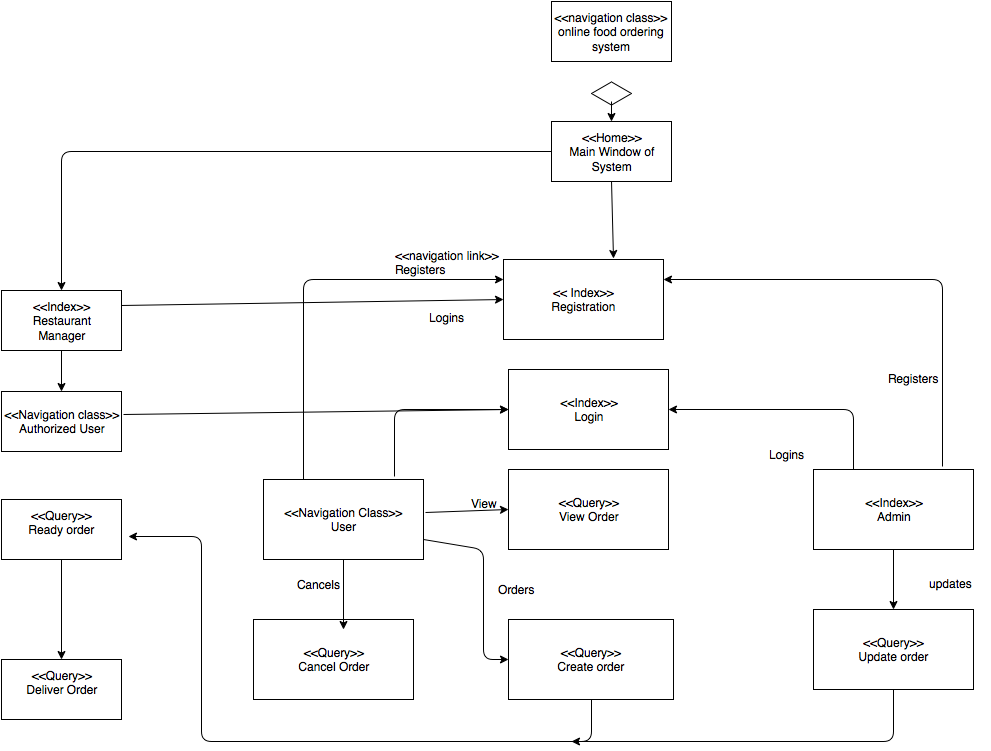
* Visualize hardware topology of a system.
* Describe the hardware components used to deploy software components.
* Describe runtime processing nodes.

**WEB ENGINEERING MODEL**

1. **Hypertext Modeling**

The non-linearity of Navigation is one of the most important properties to be taken into account when modeling Web applications. Thus the Navigation structure has to be designed carefully. This can be achieved by using suitable access structures, i.e., navigation options, to avoid the risk of users getting lost and putting them under excessive cognitive stress.

The objective of Navigation modeling – also known as navigation modeling – is to specify the navigability through the content of a Web application, i.e., the navigation paths available to the users. Navigation modeling generates a two-fold result: First, it produces the Navigation structure model, also known as navigation structure model which defines the structure of the Navigation, i.e., which classes of the content model can be visited by navigation. Second, it refines the Navigation structure model by access elements in the form of an access model.

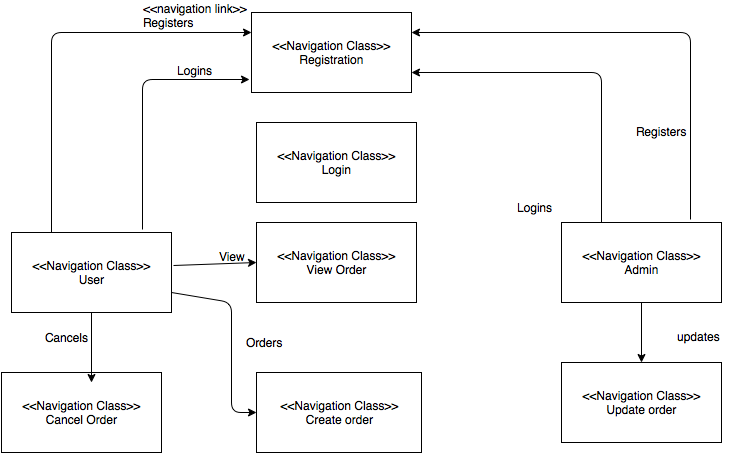


**2.Navigation Structure Model:**

Navigation structure modeling is based on the concepts of Navigation, i.e., on nodes (also called pages or documents) and links between these nodes.

Fig. shows UML stereotype <<navigation class>> is used to mark classes representing nodes

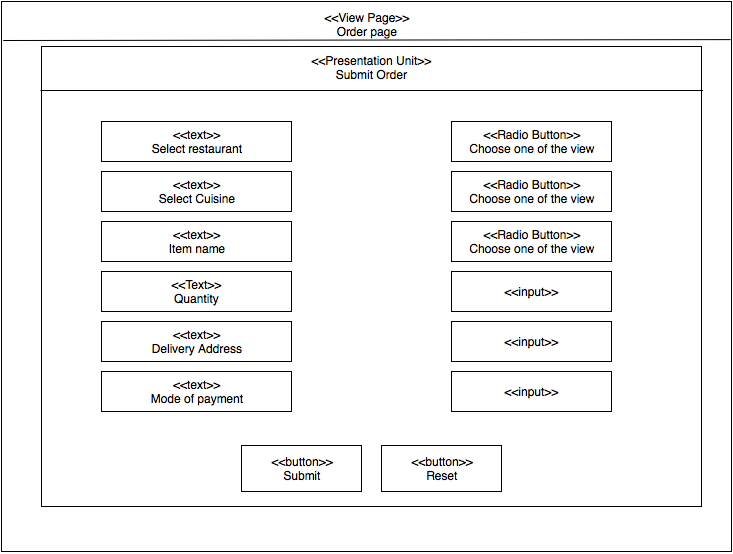
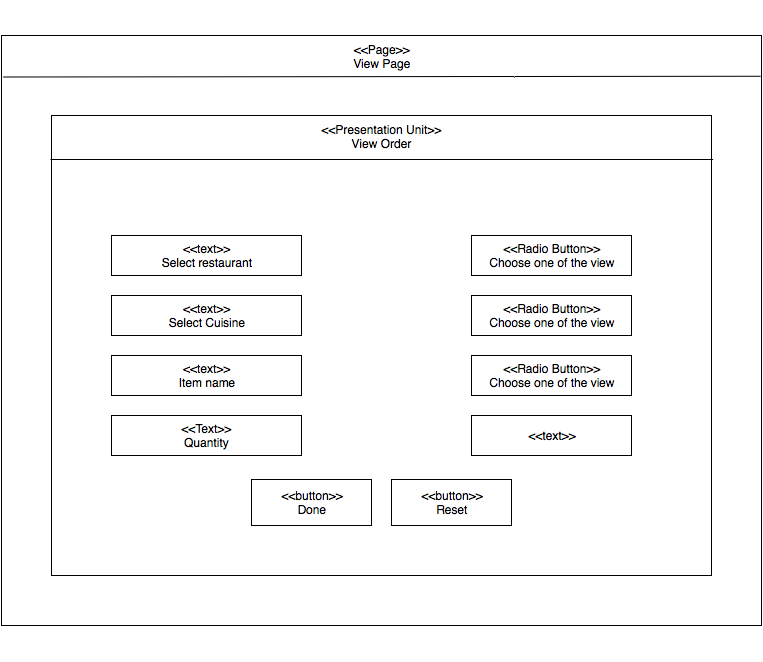
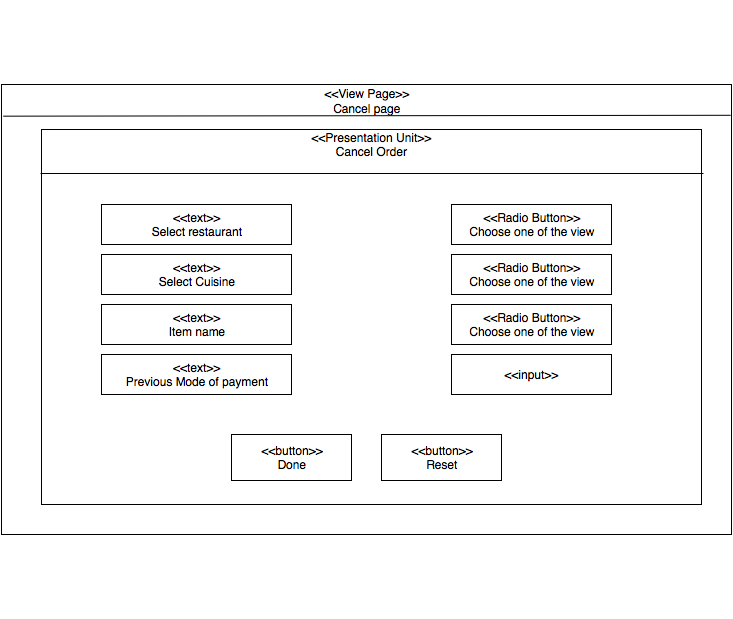
in the Navigation structure model to distinguish them from content classes. Links are modeled by directed associations with the stereotype <<navigation link>>.

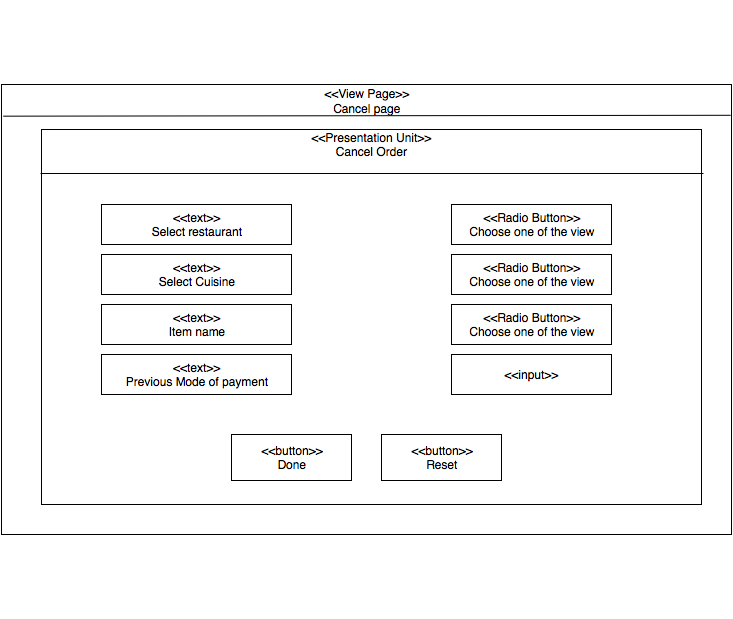


**3.Presentation Modeling**

Presentation modeling is aimed at designing the structure and behaviour of the user interface to ensure that interaction with the Web application is simple and self-explanatory.

Model elements are described on three hierarchical levels:

* A presentation page describes a page presented to the user as a visualization unit. It can be composed of different presentation units.
* A presentation unit serves to group related user interface elements, representing a logical fragment of the page. It presents a node stemming from the Navigation model.
* A presentation element is the basic building block of the presentation model. Presentation elements represent a node’s set of information and can include text, images, audio, etc.



Conclusion: