

Online Table/Order Booking in a Restaurant

CPSC 6119

Team Members:

Nithisha Reddy Boyapati

Sritej Macharla

Problem Statement:

In this automated world, everyone wishes to get their work done faster so why not with the restaurants? Now a days each restaurant is occupied with people all the time and it is not easy to get out of it with our own estimated time as it will be time consuming depending on the availability of the table or the food to be served after each order. It becomes nightmare for customers to visit each restaurant for their availability. For this reason, automating the process in the restaurants for booking a table helps customers to use their time efficiently by enjoying their favorite food instead of using old method like visiting the restaurant for availability of a table or food.

Proposed Solution:

We propose to build a simple and easy to use online table/order booking system in a restaurant which focus on table booking and management of details, so customers can concentrate on what is important for them.

The system requires to complete a registration to login for the new customers and old(already) customers can directly login. It covers most of the basic functionality required for booking a table. It allows the customer to select from a list of table types. Once the customer selects the table, the system then provides the customer to choose the date, number of persons accompanying and place. The customer will be provided with the total cost considering all venue (indoor, outdoor), food and all the booking details will be stored in database.

Users and Stakeholders:

The users and stakeholders for this application will be customers as it will allow them to reserve a table within seconds by choosing their required venue and number of persons accompanying them within their budget.

Admin will only have access to the details of the food and venue. Customers will only reserve their table by selecting food, venue. The details of their booking can be viewed in booking history.

Scope:

Below is the list of features that will be made available in the application.

Customer Scope:

The registered customers will be able to book their order/table by choosing the food and venue details. Once they book, they can view the data in the booking history. After customer logs into the system, the features made available for him/her on the application are:

- Customer Login – Using this page customer can log into the page/application.

- Customer Register – A new customer/user should register themselves only then they can login and use the full application.
- View Notifications – Customers can view any messages sent by an admin in the form of notifications.
- Update personal details – Customers will be able to update their personal information provided during their registration manually.
- Booking history – Customer will be able to view all the past, current and future bookings along with their payment status.
- Book order – By choosing their preferred choices like food items and venue, customer can book an order.
- View food items – Customer can view the details of the food items like, its ID, name, cost along with an image.
- View venue – Customer can view the details of the venue like, its ID, name, cost along with an image.

Admin Scope:

Admin will only be able to manage venue, food and notifications. Admin will also have access to view all the bookings made by a customer. Whenever admin logs into the system, the features made available for him/her on the application are:

- Admin login – Only admin can authorize this page with a username and password.
- Manage venue – Admin will be able to manage venue by inserting, updating deleting and viewing it along with its cost.
- Manage food - Admin will able to manage food items by inserting, updating, deleting and viewing it along with their cost.
- View bookings – Admin can view bookings made by all the customers at any time.
- Manage notifications – Admin will be able to send messages in the form of notifications only to the subscribed customers.

Design Patterns:

In this project we used MVC, Observer, Singleton, Abstract, Factory design patterns.

Observer pattern:

For 1: N dependency one can implement this pattern. All the subscribed listeners will be notified if there is a change in the subject's state. In this project during registration, all the customers will be provided with an option called subscribe to notifications. Here admin acts as a subject. Array list of customers are present in the admin class. Update method is present in the customer class. In Manage notifications page admin can type a message and send it to all the subscribed customers by calling an update method and looping through each of them. In the Update personal details page customer if not interested in receiving notifications can unsubscribe it any time. In View notifications page, all the notifications sent by an admin can be viewed by customer.

Abstract Pattern:

In this design pattern, without specifying their classes explicitly a factory of related objects will be created by an interface. Based on the factory pattern each factory that is generated can give the objects. In this project we created a DDL and DML interface for not specifying the classes for DB. The interfaces are to create the classes which interact and connect with the DB. Inside DatabaseQueries which is an abstract class each factory will be handled, where for both factories upon request one object will be generated and returned.

Factory Pattern:

Using this pattern without exposing the logic to a client we created objects and using a common interface we referred to the objects that are created newly. Three parts of factory pattern has been added in this project. Firstly, within the DBConnection class which is the parent class of DatabaseQueries, when the project starts running the data base will be created for the first time within it. It will just connect to the data base if already exists. By calling static instance of it, the instantiation of data base is done inside the dataBaseQueries class. We ensured the secrecy of it not being exposed to the user by encapsulating the process of creation. All this was happened using the common interface.

Singleton Pattern:

This pattern makes sure that there is only one instance of an object any time. In this project an instance of a class is provided for the classes DDL_Queries and DML_Queries. The class can be accessed several times by calling this instance and declaring the object once. So, the class can be used whenever required as it is already instantiated once.

MVC Pattern:

For separating the application's concerns, we used this design pattern.

- The model updates the controller if there is any change in the data and represents the objects that carries the information.
- The view is responsible for viewing the data that is present in the model.
- The action of controller will be on model and view. It updates the view when there is a change in the data and controls the flow of data into the model objects. It also separated model from the view.

The code has been split into three different packages namely model, view and controller for this design pattern.

Model:

- Admin and customer/user are the main users of Online Order Management System.
- Venue and FoodItem are maintained in separate classes and we require these while booking an order.

- Notification contains all the notification details.
- Foodorder contains the complete information about the order.

In the MYSQL database a table have been created for each model that contains all the main attributes.

- The attributes of admin like adminID and password are present in Admin.java.
- The attributes of customer like customerID and password are present in Customer.java.
- The attributes of venue like its ID, name, place, image, cost etc. are present in Venue.java.
- The attributes of food order like its booking ID, ordertype, venue, fooditems etc. are present in Foodorder.java
- The attributes of food items like ID, name, cost, images are present in FoodItem.java
- The attributes of notification like sent date and message are present in Notification.java.

During the creation of these models, encapsulation principle is applied where each field contains getters and setters for modifying and accessing the attributes of an object as it is private field.

View:

User interfaces have been created in this package where there are separate logins for admin and customer.

- OrderHomeGUI – used by admin and customer.
- CustomerRegisterGUI, CustomerLoginGUI, UpdateCustomerGUI and CustomerHomeGUI – used only by the customers/users.
- AdminLoginGUI and AdminHomePageGUI – used only by the admin.
- ManageVenueGUI and ManageFoodGUI – we are managing this based on the login. Admin has read/write options where as customer has only read access to these. So, buttons will be hidden for customer not for admin. This is used by admin and customer.
- ManageNotificationsGUI – like ManageVenueGUI and ManageFoodGUI we are managing UI based on the login. Here admin has read/write options, so admin can send messages in the form of notifications to the customer and read those messages as well where as for the customers unlike admin buttons will be disabled so they can only view/read the notifications. This page is used by admin and customers as well.
- BookingHistoryGUI – UI is managed based on login. The pay for order button will be enabled for customer as they can pay for their bookings where as it will be disabled for the admin. Here admin can view bookings of all the customers where as a customer can only view his/her bookings. This page is used by admin and customer as well.
- BookOrderGUI, OrderPaymentGUI and OrderInvoiceGul – used only by the customer for their bookings, payments and generating their bills.

Controller:

Linking model with the view, handling and connecting the activities related to database are the two main functions controller package classes handles. DBConnection.java class was created for handling database work. This class creates ordermanagemebtdb by connecting to MYSQL driver. We have tables created for all the model classes in MYSQL. A default username 'root' and password 'root' was created upon database creation. All the DDL and DML queries are handled by DatabaseQueries.java with the database.

Software Required:

- NetBeans IDE
- Mysql-connector.jar
- JDK 1.7 or above
- MySQL Server

The project starts running from the OrderHomeGUI.java which is the main class in the view. The username and password we provided for our MySQL server is 'root'. Based on your requirement the username and password can be changed in DBConnection.java class.

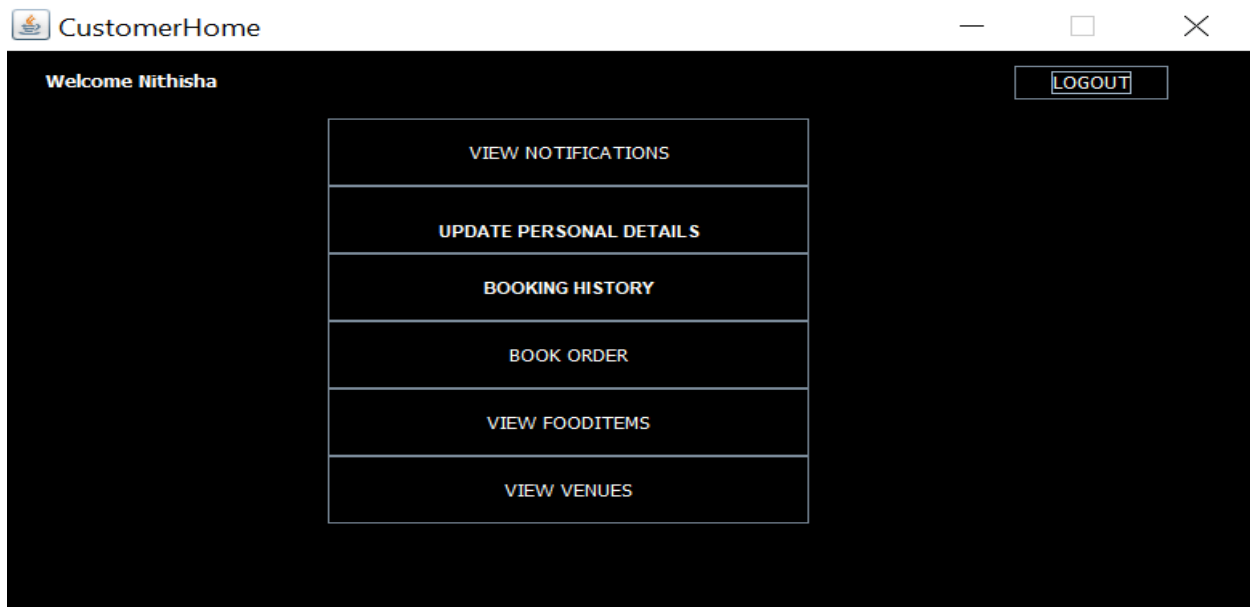
OrderHomeGUI:

It contains three pages within it. Login page helps the registered customers to login with their own username and password provided at the time of registration. Register page helps customer creating their account and providing with customerID after successful completion of registration. Admin page helps admin to login with admin credentials and validation checking's are applied here.

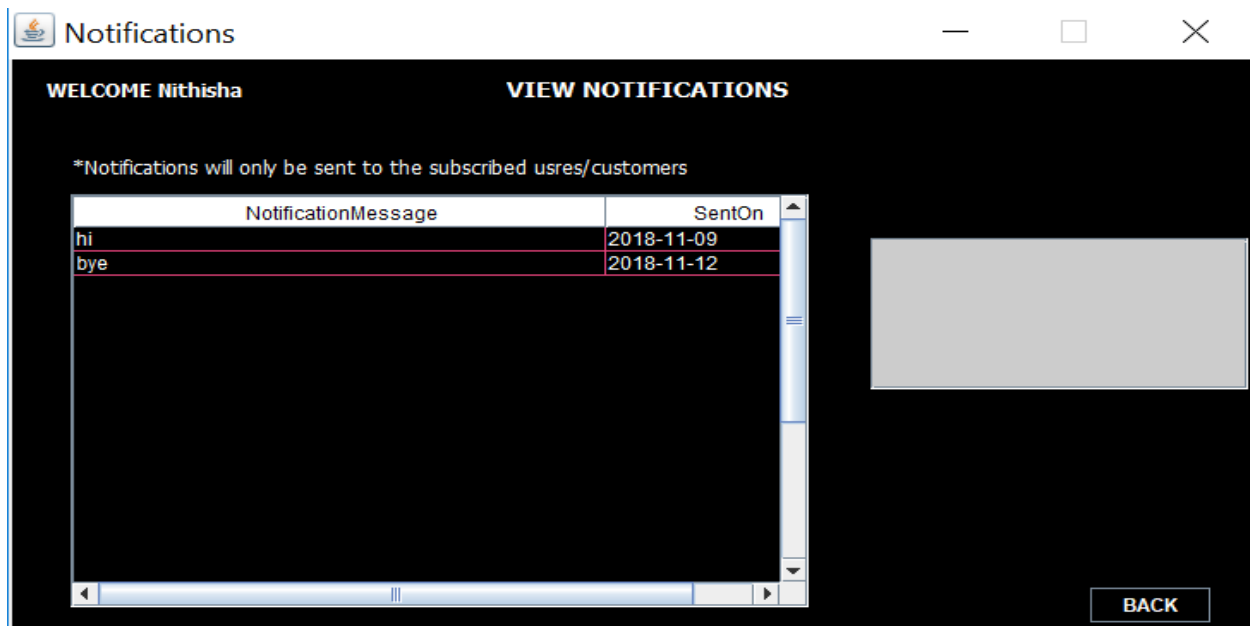
The screenshot displays the login interface of an 'ONLINE ORDER MANAGEMENT SYSTEM'. At the top, the title 'ONLINE ORDER MANAGEMENT SYSTEM' is centered in a bold, black font. Below the title, there are three navigation buttons: 'LOGIN', 'REGISTER', and 'ADMIN', each enclosed in a black rectangular box. The 'LOGIN' button is highlighted with a white border. The main content area has a black background. It features a white text block that reads: 'Order operations and logistics is where all the time spent. Planning the order comes together in the execution of the order. Save your time by booking for any order Online using our system'. Below this text, there are two input fields: 'Customer ID' and 'Password', each with a white rectangular input box. At the bottom, there are two buttons: 'SUBMIT' and 'CANCEL', each in a white rectangular box.

CustomerHomeGUI:

The main functionality of the application exists after successful logging of the customer. By clicking on the logout button on this page the customer will be redirected back to the OrderHomeGUI. Here the customer will have options for viewing the information by clicking on each appropriate button.



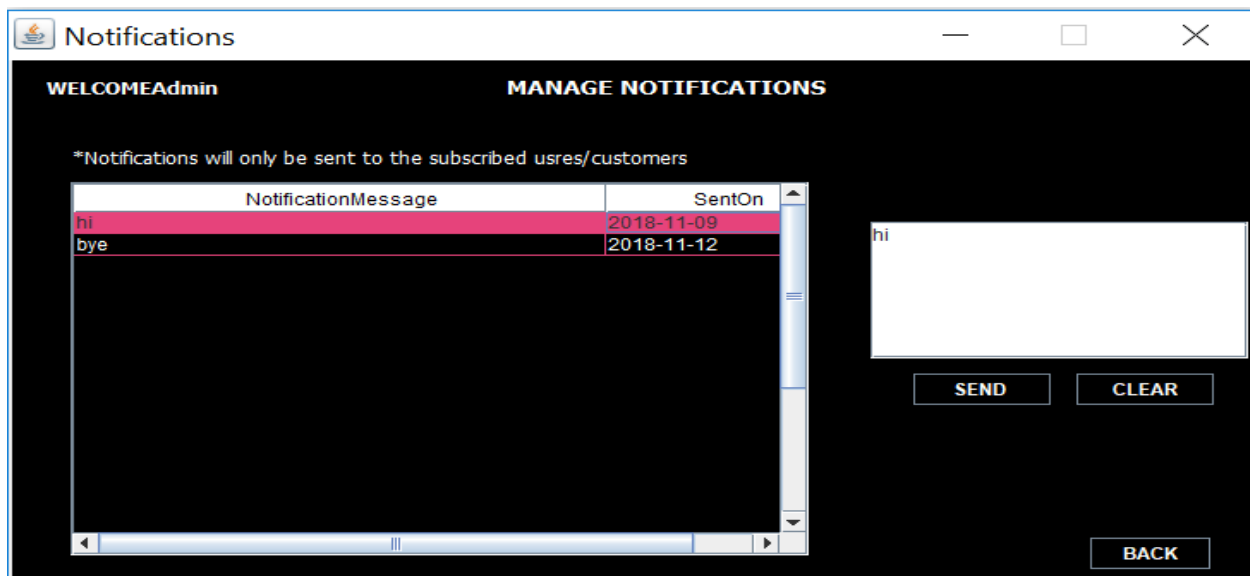
ManageNotificationsGUI:



The customer will be redirected to this page when he/she clicks on view notifications button. Only the subscribed customer will be able to view and read all the messages sent by an admin in the form of notifications. The customer can also disable this subscription by unchecking the box

next to subscribe notifications in the UpdatingPersonalDetailsGUI. The back button navigates customer back to the CustomerHomeGUI.

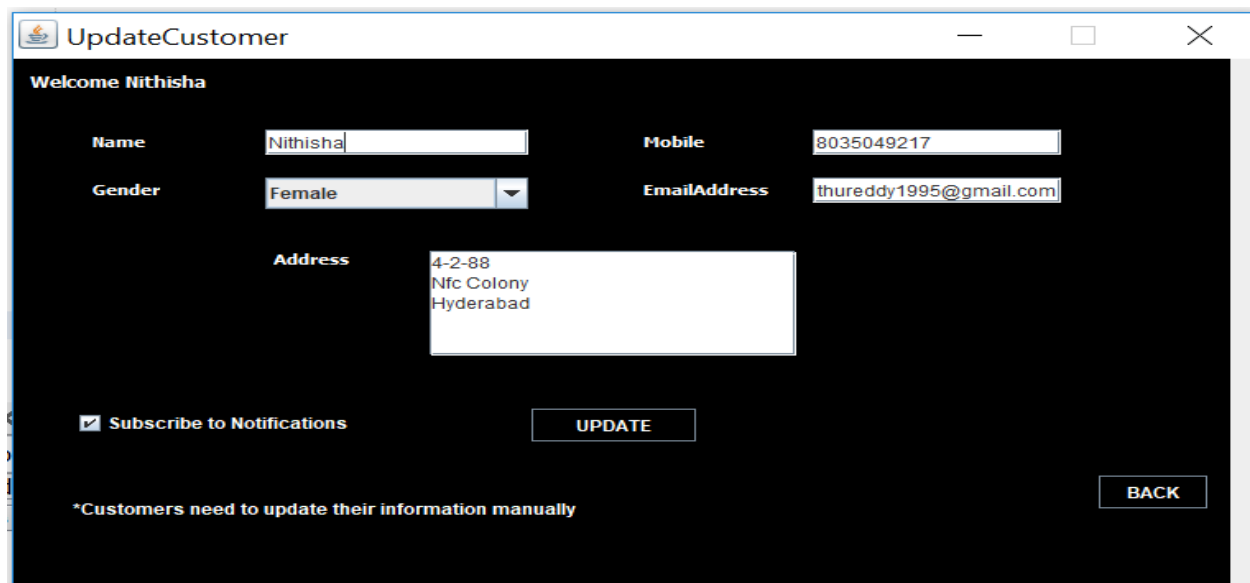
The admin will be redirected to this page when he/she clicks on manage notifications button. This page helps admin in sending messages to all the subscribed customers in the form of notifications. By clicking on back button, the admin will be redirected back to the AdminHomeGUI.



The screenshot shows a window titled "Notifications" with a dark background. At the top, it says "WELCOMEAdmin" and "MANAGE NOTIFICATIONS". Below this, a note states: "*Notifications will only be sent to the subscribed usres/customers". There is a table with two columns: "NotificationMessage" and "SentOn". The table contains two rows: one with "hi" and "2018-11-09", and another with "bye" and "2018-11-12". To the right of the table is a text input field containing "hi". Below the input field are two buttons: "SEND" and "CLEAR". At the bottom right of the window is a "BACK" button.

NotificationMessage	SentOn
hi	2018-11-09
bye	2018-11-12

UpdateCustomerGUI:



The screenshot shows a window titled "UpdateCustomer" with a dark background. At the top, it says "Welcome Nithisha". Below this, there are four input fields: "Name" (containing "Nithisha"), "Mobile" (containing "8035049217"), "Gender" (a dropdown menu showing "Female"), and "EmailAddress" (containing "thureddy1995@gmail.com"). Below these fields is a larger text input field for "Address" containing "4-2-88", "Nfc Colony", and "Hyderabad". At the bottom left, there is a checkbox labeled "Subscribe to Notifications" which is checked. To the right of the checkbox is an "UPDATE" button. At the bottom right of the window is a "BACK" button. A note at the bottom left states: "*Customers need to update their information manually".

The customer will be redirected to this page when he/she clicks on Update personal details button. This page helps customer updating all their personal details provided during registration. A popup appears after successful updating and by clicking on back button the customer will be redirected back to the CustomerHomeGUI.

BookingHistoryGUI:

Welcome Nithisha

BookingID 548928

BACK

Pay for Order

BookingID	CustomerID	OrderType	TableType	VenueType	OrderDate	OrderTime	FoodItems	Guests	VenueCost	FoodItems...
548928	1000	both	Three chair...	Outdoor	2018-11-15	7:00 PM	Italian veg ...	3	41.0	120.0
548929	1000	Non-vegitar...	sofa_Water...	Outdoor	2018-11-21	8:00 PM	Italian Non...	7	70.0	455.0

The customer will be redirected to this page when he/she clicks on Booking history button. This page helps customer view all their past, present and future bookings along with the payment status. In order to pay for the pending payment, it contains a button pay for order which navigates to other page where payment will be made. If the customer tries to pay again for the for the payment status which shows as completed a pop up appears saying payment is done for the order. By clicking on back button, the customer will be redirected back to the CustomerHomeGUI.

Welcome Admin

BookingID 548930

BACK

BookingID	CustomerID	OrderType	TableType	VenueType	OrderDate	OrderTime	FoodItems	Guests	VenueCost	FoodItems...
548928	1000	both	Three chair...	Outdoor	2018-11-15	7:00 PM	Italian veg ...	3	41.0	120.0
548929	1000	Non-vegitar...	sofa_Water...	Outdoor	2018-11-21	8:00 PM	Italian Non...	7	70.0	455.0
548930	1000	Vegitarian	two chair_J...	Indoor	2018-11-20	7:00 PM	Italian veg ...	2	30.0	80.0

The admin will be redirected to this page when he/she clicks on view bookings button. This page helps admin to view all the bookings made by the customers. By clicking on back button, the admin will be redirected back to the AdminHomeGUI.

BookOrderGUI:

The customer will be redirected to this page when he/she clicks on Book order button. This page helps the customer book their order by providing all their requirements. The booking ID will be generated after choosing all the required options by clicking on book button. This allows customer to the payment page. By clicking on back button, the customer will be redirected back to the CustomerHomeGUI.

ManageFoodGUI:

FoodItemID	FoodItemName	FoodItemCost	FoodItemImage
1002	American Breakfast	30.0	C:\Users\nithu\Do
1003	American Buffet	80.0	C:\Users\nithu\Do
1004	American nonveg ...	35.0	C:\Users\nithu\Do
1005	American nonveg ...	70.0	C:\Users\nithu\Do
1006	American Veg Bre...	25.0	C:\Users\nithu\Do
1007	American Veg Buff...	50.0	C:\Users\nithu\Do
1008	Italian Breakfast	35.0	C:\Users\nithu\Do
1009	Italian Nonveg Buf...	65.0	C:\Users\nithu\Do
1010	Italian nonveg Bre...	30.0	C:\Users\nithu\Do
1011	Italian veg Breakfa...	20.0	C:\Users\nithu\Do
1012	Italian veg Buffet	40.0	C:\Users\nithu\Do
1013	Italian Buffet	70.0	C:\Users\nithu\Do

The customer will be redirected to this page when he/she clicks on view fooditems buttons. This page helps customer to view all the food items inserted by the admin along with their images. By clicking on back button, the customer will be redirected back to the CustomerHomeGUI.

The admin will be redirected to this page when he/she clicks on manage food button. This page helps admin in modifying the food details. By clicking on back button, the admin will be redirected back to the AdminHomeGUI.

FoodItems

Welcome Admin

FOOD MANAGEMENT

FoodItemID: 1007

FoodItemName: American Veg Buffet

FoodItemCost: 50.0

FoodItemImage

SELECT PICTURE

CLEAR

FoodItemID	FoodItemName	FoodItemCost	FoodItemImage
1002	American Breakfast	30.0	C:\Users\nithuDo...
1003	American Buffet	80.0	C:\Users\nithuDo...
1004	American nonveg ...	35.0	C:\Users\nithuDo...
1005	AMERICAN nonveg ...	70.0	C:\Users\nithuDo...
1006	American Veg Bre...	25.0	C:\Users\nithuDo...
1007	American Veg Buff...	50.0	C:\Users\nithuDo...
1008	Italian Breakfast	35.0	C:\Users\nithuDo...
1009	Italian Nonveg Buf...	65.0	C:\Users\nithuDo...
1010	Italian nonveg Bre...	30.0	C:\Users\nithuDo...
1011	Italian veg Breakfa...	20.0	C:\Users\nithuDo...
1012	Italian veg Buffet	40.0	C:\Users\nithuDo...
1013	Italian Buffet	70.0	C:\Users\nithuDo...

INSERT

UPDATE

DELETE

BACK

ManageVenueGUI:

The customer will be redirected to this page when he/she clicks on view venues buttons. This page helps customer to view all the venue details inserted by the admin along with their images. By clicking on back button, the customer will be redirected back to the CustomerHomeGUI.

Venues

Welcome Nithisha

VIEW VENUES

VenueID: 16

TableType: two chair_Indoor

VenuePlace: Indoor

VenueCost: 30.0

VenueImage

BACK

VenueID	TableType	VenuePlace	VenueCost	VenueImage
8	sofa_Waterside	Outdoor	70.0	C:\Users\nithuDo...
9	Two chair_Ro...	Outdoor	40.0	C:\Users\nithuDo...
10	Three chair_R...	Outdoor	41.0	C:\Users\nithuDo...
11	Four chair_Ro...	Outdoor	42.0	C:\Users\nithuDo...
12	Six chair_Roof...	Outdoor	43.0	C:\Users\nithuDo...
13	four chair_Wat...	Outdoor	69.0	C:\Users\nithuDo...
14	three chair_W...	Outdoor	67.0	C:\Users\nithuDo...
15	four chair_Ind...	Indoor	34.0	C:\Users\nithuDo...
16	two chair_Indo...	Indoor	30.0	C:\Users\nithuDo...
17	six chair_Indoor	Indoor	38.0	C:\Users\nithuDo...
18	sofa_Indoor	Indoor	42.0	C:\Users\nithuDo...

The admin will be redirected to this page when he/she clicks on manage venue button. This page helps admin in modifying the venue details. By clicking on back button, the admin will be redirected back to the AdminHomeGUI.

Welcome Admin

VENUE MANAGEMENT

VenueID: 13
 TableType: four chair_Waterside
 VenuePlace: Outdoor
 VenueCost: 69.0

VenueImage:

SELECT PICTURE
 CLEAR

VenueID	TableType	VenuePlace	VenueCost	VenueImage
3	Two Chair_Ga...	Outdoor	50.0	C:\Users\mithu...
4	Three Chair_G...	Outdoor	51.0	C:\Users\mithu...
5	Four Chair_Ga...	Outdoor	52.0	C:\Users\mithu...
6	Six Chair_Gar...	Outdoor	53.0	C:\Users\mithu...
7	Sofa_Garden	Outdoor	54.0	C:\Users\mithu...
8	sofa_Waterside	Outdoor	70.0	C:\Users\mithu...
9	Two chair_Ro...	Outdoor	40.0	C:\Users\mithu...
10	Three chair_R...	Outdoor	41.0	C:\Users\mithu...
11	Four chair_Ro...	Outdoor	42.0	C:\Users\mithu...
12	Six chair_Roof...	Outdoor	43.0	C:\Users\mithu...
13	four chair_Wat...	Outdoor	69.0	C:\Users\mithu...
14	three chair_W...	Outdoor	67.0	C:\Users\mithu...

INSERT
 UPDATE
 DELETE

BACK

AdminHomePageGUI:

After successful login of an admin, he/she will be redirected to this page. It contains the main functionality of the application. This page contains some buttons which can be accessed and managed by admin. By clicking logout button admin will be redirected back to the home page where it asks to login again for accessing the application.

WELCOME ADMIN

LOGOUT

MANAGE VENUE

MANAGE FOOD

VIEW BOOKINGS

MANAGE NOTIFICATIONS

OrderPaymentGUI:

The customer will be automatically redirected to this page after clicking on book button in BookOrderGUI. This page asks for the credit/debit card details of the customer through which they are willing to pay. An invoice will be generated after successful payment of the bill. If the customer is willing to pay later, he/she can choose back button which redirects to the CustomerHomeGUI and they can pay for their booking later by clicking on booking history which contains their payment status as pending.

OrderPayment

Welcome Nithisha

Pay for Order

Booking ID: 548930

Card No:

CVV No:

Name on Card:

Expiration Date: November 2018

Total Cost: 110.0

PAY CLEAR BACK

OrderInvoiceGUI:

Soon after paying the bill by clicking on pay button in OrderPaymentGUI, the customer will automatically be redirected to this page. It helps customer to view all the purchased items. By clicking on save invoice button, the invoice can be saved in the form of pdf. By selecting home button, customer will be navigated back to the CustomerHomeGUI page.

OrderInvoice

CUSTOMER NAME : NITHISHA

HOME

ORDER BOOKING INVOICE

CUSTOMER ID : 1000

Booking Date: 2018-11-18 13:07:33

BOOKING ID	ORDER TYPE	VENUE	GUESTS COUNT	ITEMS	COST
548930	VEGITARIAN	TWO CHAIR_INDOOR	2	TWO CHAIR_INDOOR	30.0
				ITALIAN VEG BUFFET	80.0
					110.0

SAVE INVOICE

THANK YOU FOR YOUR TIME

Conclusion:

Finally, we conclude that all the functionality that was stated earlier in the proposal was added in this project.