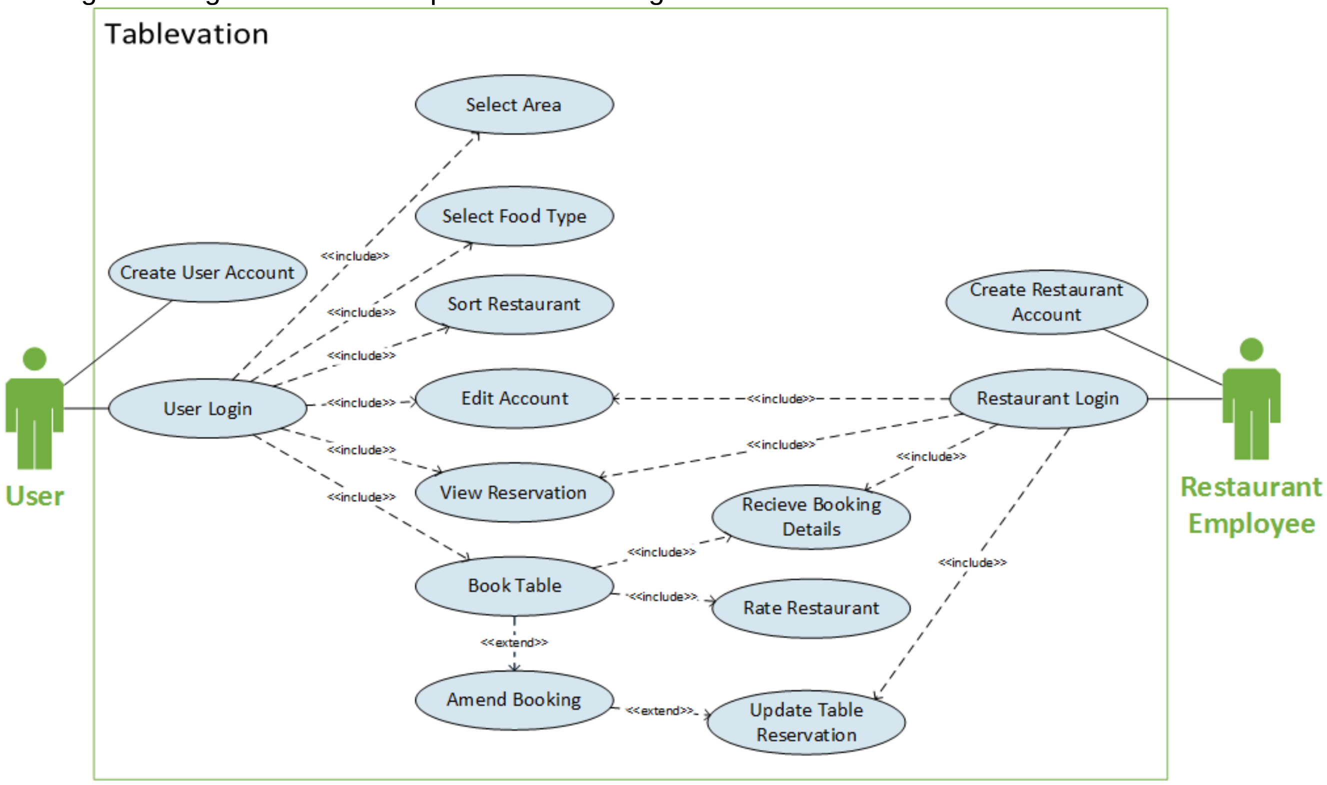
Design Documentation

Use Case

Below is the Use case diagram, this presents all of the scenarios within Tablevation. The table after this diagram will give a more in depth understanding of how each scenario works.



Use Case Description

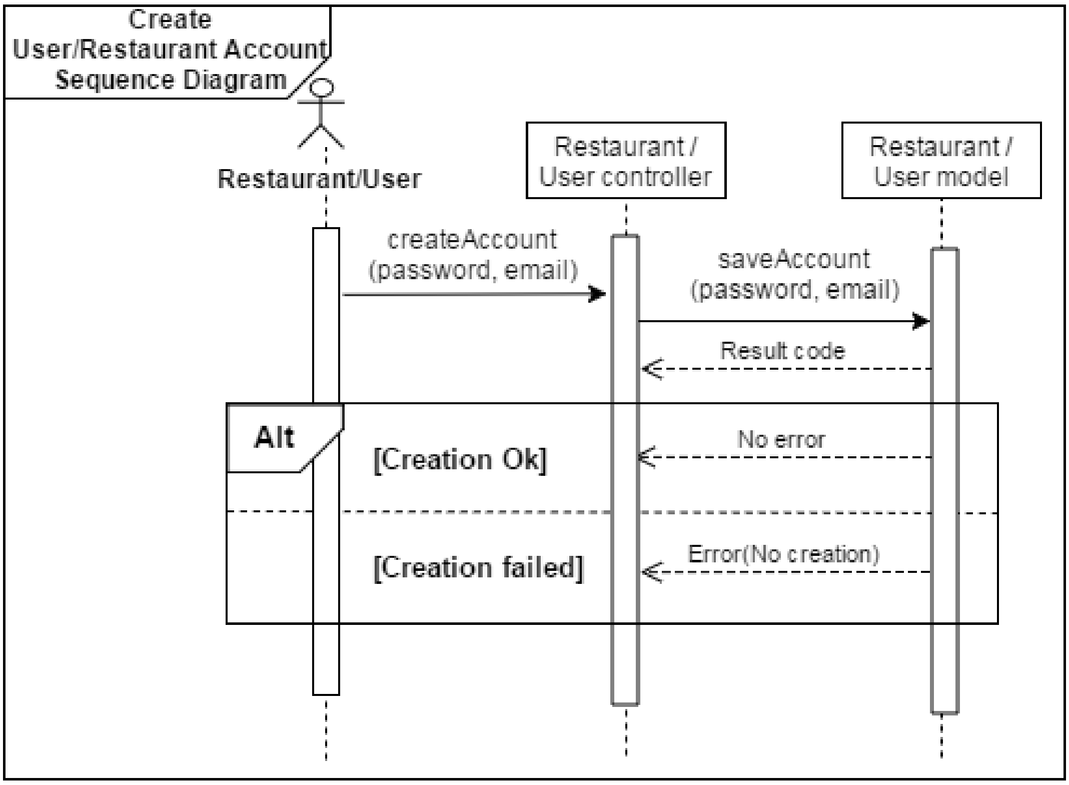
Tablevation System: Create User Account / Create Restaurant Account	
Actors	User, Restaurant employee/ Owner
Description	**User account:** The user will need to create an account in order to access the system. **Restaurant Account:** The restaurant will need to create an account in order to populate the system as a restaurant that can be booked via this systems services.
Data	**User account:** Forename, Surname, Email (Username), contact number, password and confirm password. There will also be a checkbox on whether or not they would like to receive emails advertising promotions. **Restaurant Account:** Restaurant Name, Address, Email (Username), contact number, Portsmouth area, opening times (Monday to Sunday), Food Type, password and confirm password.
Stimulus	**User account:** They want to use the accounts services. They will enter in personal details. **Restaurant Account:** They would like to expand their business and get more people to come to their restaurant. They will enter details about the restaurant.
Response	Receive Confirmation email about account creation.
Tablevation System: User Login / Restaurant Login	
Actors	User, Restaurant employee/ Owner
Description	**User:** To use the system from any device, the user must login to access this systems services of reserving tables in the areas around Portsmouth.

	Restaurant: To use the system from any device, the restaurant must login to access this systems services to view their bookings.
Data	Email (Username) and password.
Stimulus	They will enter in login details.
Response	The actor will be permitted access to the system if the details are valid.
Tablevation System: Edit Account	
Actors	Restaurant employee/Owner, User
Description	This will allow whoever is logged in to edit the fields that were entered to create the account in the first place.
Data	**If user is logged in:** Forename, Surname, Email, contact number, username, password and confirm password. **If Restaurant Employee/Owner is logged in**: Restaurant Name, Address, Email, contact number, Portsmouth area, opening times (Monday to Sunday), promotions, username, password and confirm password.
Stimulus	They will enter in account details.
Response	Their account details will be updated and they will receive a confirmation email.
Tablevation System: Sort Restaurant	
Actors	User
Description	The users will be able to sort the generated restaurants from the specified area given. The sort can be by ratings or alphabetically, it can also be a search..
Data	Drop down list: Lowest to Highest Rating, Highest to Lowest Rating or Alphabetically. Search bar: Restaurant Name.
Stimulus	User selects a Sort drop down list option which in turn sorts the generated restaurants. They can also do a fast sort which includes the search bar.
Tablevation System: View Reservation	
Actors	Restaurant employee/Owner, User
Description	The user, or restaurant employee/owner will be able to view all of their reservations in a calendar type view.
Data	Time, date, number of people and Extra details (allergies etc.), Restaurant Name (If actor is a user)
Stimulus	The actor will select the option, View Reservations.
Response	All of the up to date reservations will appear on the screen.
Tablevation System: Select Area	
Actors	User
Description	The user will be able to specify the Portsmouth area they would like to generate restaurants in.
Data	Drop down list: Portsea, Cosham, Eastney, Southsea etc.
Stimulus	User selects an Area drop down list option.

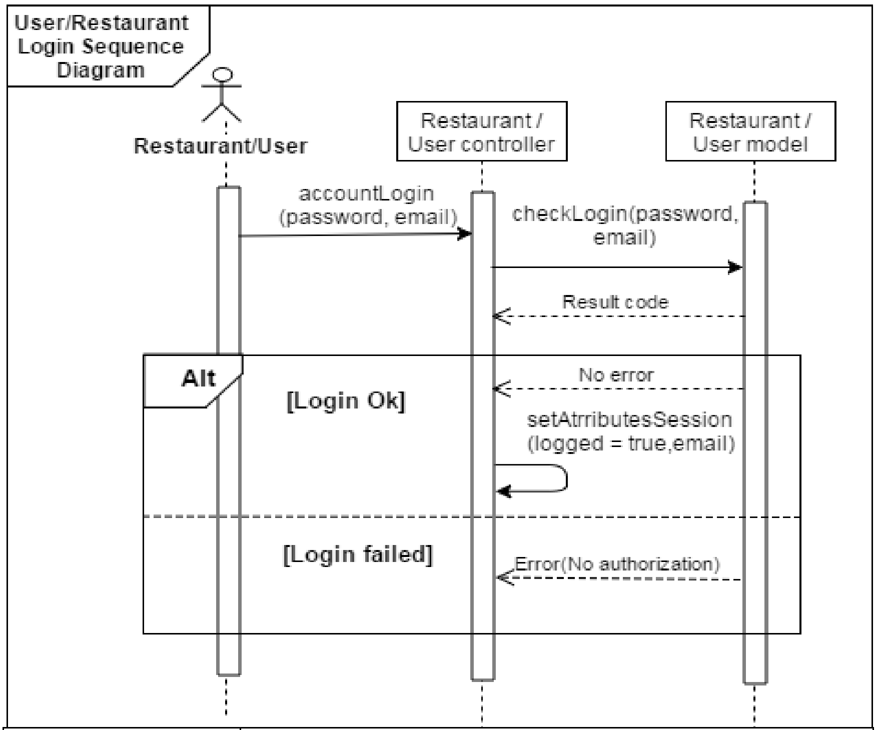
Response	The system will only generate restaurants within the specified area in Portsmouth.
Tablevation System: Select Food Type	
Actors	User
Description	The user will be able to specify the food type they would like to generate.
Data	Drop down list: Chinese, Indian, Italian, British, Portuguese, Thai etc.
Stimulus	User selects a Type drop down list option.
Response	The system will only generate restaurants within the specified area which corresponds to that food type.
Tablevation System: Book Table	
Actors	User
Description	The user will be able to select a restaurant from the generated restaurant list, this will then take them to a page which will allow the user to make a reservation at that restaurant.
Data	Time, date, number of people and Extra details (allergies etc.)
Stimulus	The user will enter the details required to book a table.
Response	The user will receive an email confirmation if the booking is successful. If the booking slot is unavailable then the user will be notified to change one of the date or time fields to find an available slot.
Tablevation System: Amend Booking	
Actors	User
Description	If the user decides to change their mind or invite more people along, they will need to notify the restaurant. They can also cancel the booking.
Data	Time, date, number of people and Extra details (allergies etc.) Check Box tick (against the words 'Delete Reservation')
Stimulus	The user changes some of or even all of the booking details already confirmed. Also, the user checks the box that confirms they would like to delete the reservation and then clicks submit to ensure the process was done intentionally if cancelling.
Response	The user will receive an email confirming the new details of the reservation. The system will also automatically make that time slot available to other users if the reservation is cancelled.
Tablevation System: Rate Restaurant	
Actors	User
Description	The user will be able to rate the restaurant out of five stars. The average rating will be used to help sort the restaurants for other users.
Data	Number of stars out of 5, Restaurant Name
Stimulus	The user will enter their rating of the restaurant by selecting up to 5 stars in a drop down list.
Response	The database will update the average rating of the specified restaurant for future sorting.
Tablevation System: Receive Booking Details	
Actors	Restaurant Employee/Owner

Description	The restaurant will be notified when a new reservation is made for their restaurant.
Data	All Booking details for specific restaurant logged in.
Stimulus	A user makes a reservation at this restaurant.
Response	Notification with the details of the new reservation will appear.
Tablevation System: Update Table Reservation	
Actors	Restaurant Employee/Owner
Description	The restaurant will be able to update the reservations available in case people use other means to book tables, for instance, someone could call.
Data	Booking details
Stimulus	The restaurant employee/ owner will select the date and time specified and enter the reservation details of users not using the Tablevation method to book.
Response	The system will update available time slots.

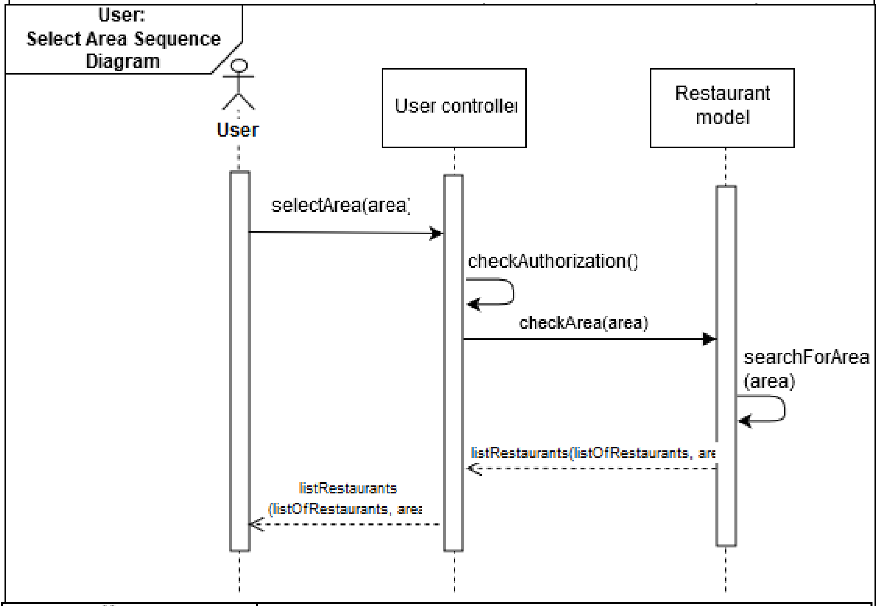
Sequence Diagram


Create User/Restaurant Account Sequence Diagram
Restaurant/User
Restaurant / User controller
Restaurant / User model
createAccount (password, email)
saveAccount (password, email)
Result code
Alt
[Creation Ok]
No error
[Creation failed]
Error(No creation)

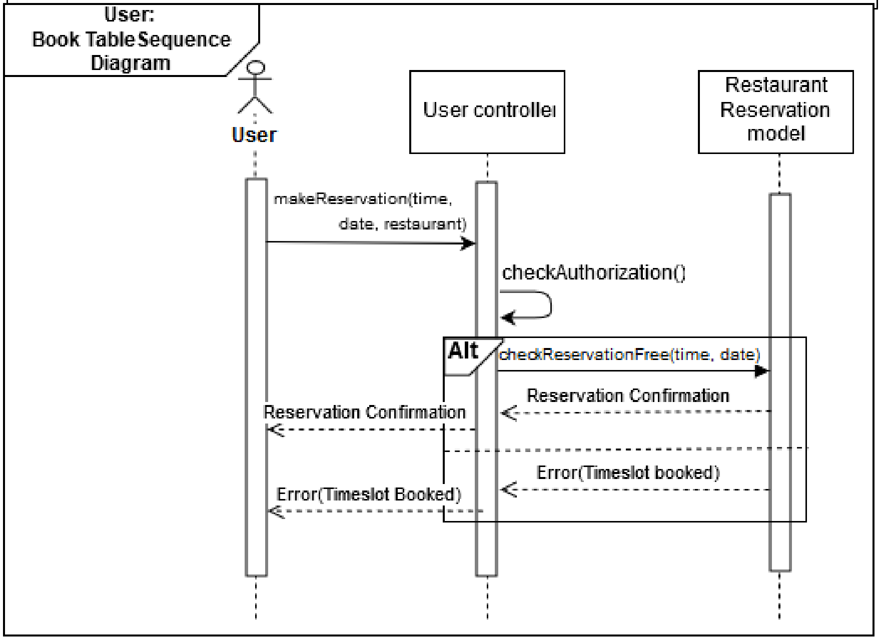
This is the create account sequence diagram. As it applies to both the Restaurant Employee/ owner and User, it has been made to accommodate both in one diagram. It illustrates how the actor will create an account. The system checks the input details and if an error is returned the account creation fails. See Use Case description for more in depth data details.


User/Restaurant Login Sequence Diagram
Restaurant/User
Restaurant / User controller
Restaurant / User model
accountLogin (password, email)
checkLogin(password, email)
Result code
Alt
[Login Ok]
No error
setAtrributesSession (logged = true,email)
[Login failed]
Error(No authorization)

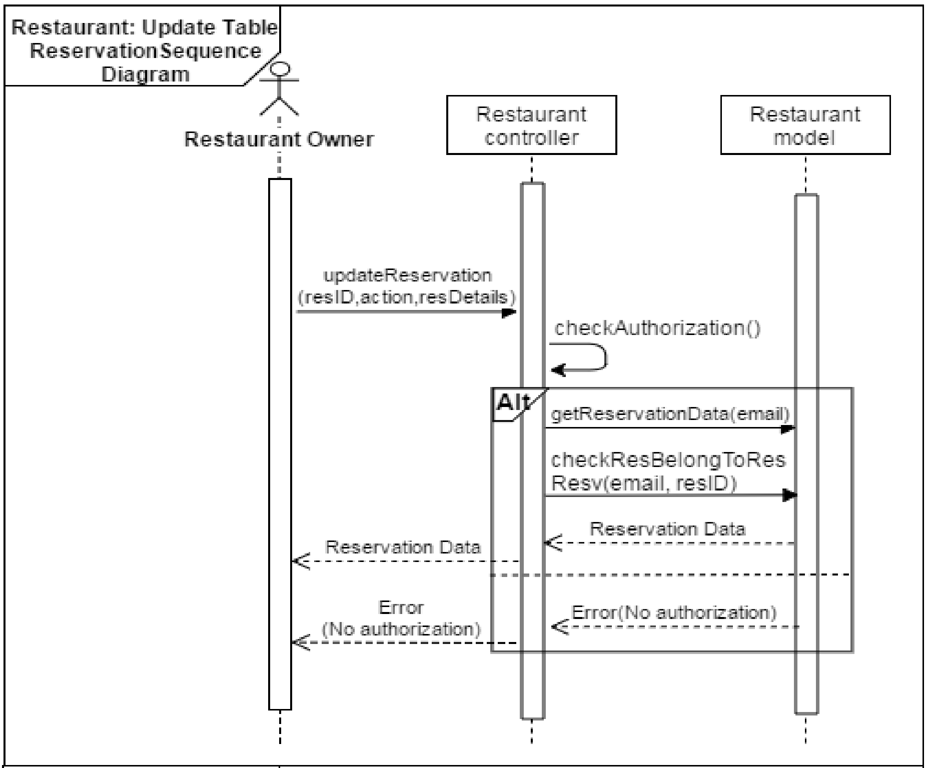
This is the Login sequence diagram. Again, this process is covered by both the User and Restaurant Employee/ Owner. It requires the actors email and password to enter the system. If there is an error in the email or password, then the login will fail.


User: Select Area Sequence Diagram
User
User controller
Restaurant model
selectArea(area)
checkAuthorization()
checkArea(area)
searchForArea (area)
listRestaurants(listOfRestaurants, are
listRestaurants (listOfRestaurants, area

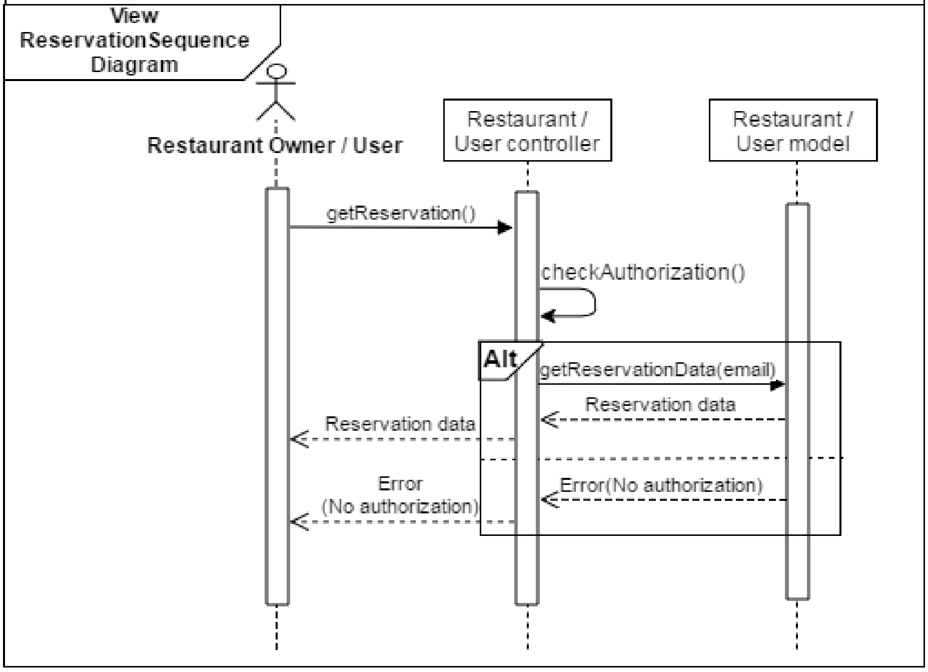
The user selects an area which is fed into the database. The database then searches known restaurants for the area selected and outputs a list of restaurants in that area to the user.



The user selects a time/date slot at a specific restaurant, the details of which are then sent to the restaurant in question's account. The system checks the reservation details against current reservations made with the restaurant then either sends back a confirmation to the user if the time slot is free or returns an error if the timeslot is already taken.


Restaurant: Update Table ReservationSequence Diagram
Restaurant Owner
Restaurant controller
Restaurant model
updateReservation (resID,action,resDetails)
checkAuthorization()
Alt
getReservationData(email)
checkResBelongToRes Resv(email, resID)
Reservation Data
Reservation Data
Error (No authorization)
Error(No authorization)

Once the Restaurant employee/ Owner is logged in, they are able to update table reservations, this is because there are still other forms of people booking reservations. The process checks for their authorization on the account and then allows them to update on the condition that no errors occur.



Only once logged in can either the Restaurant employee/ owner or user can view the reservations.

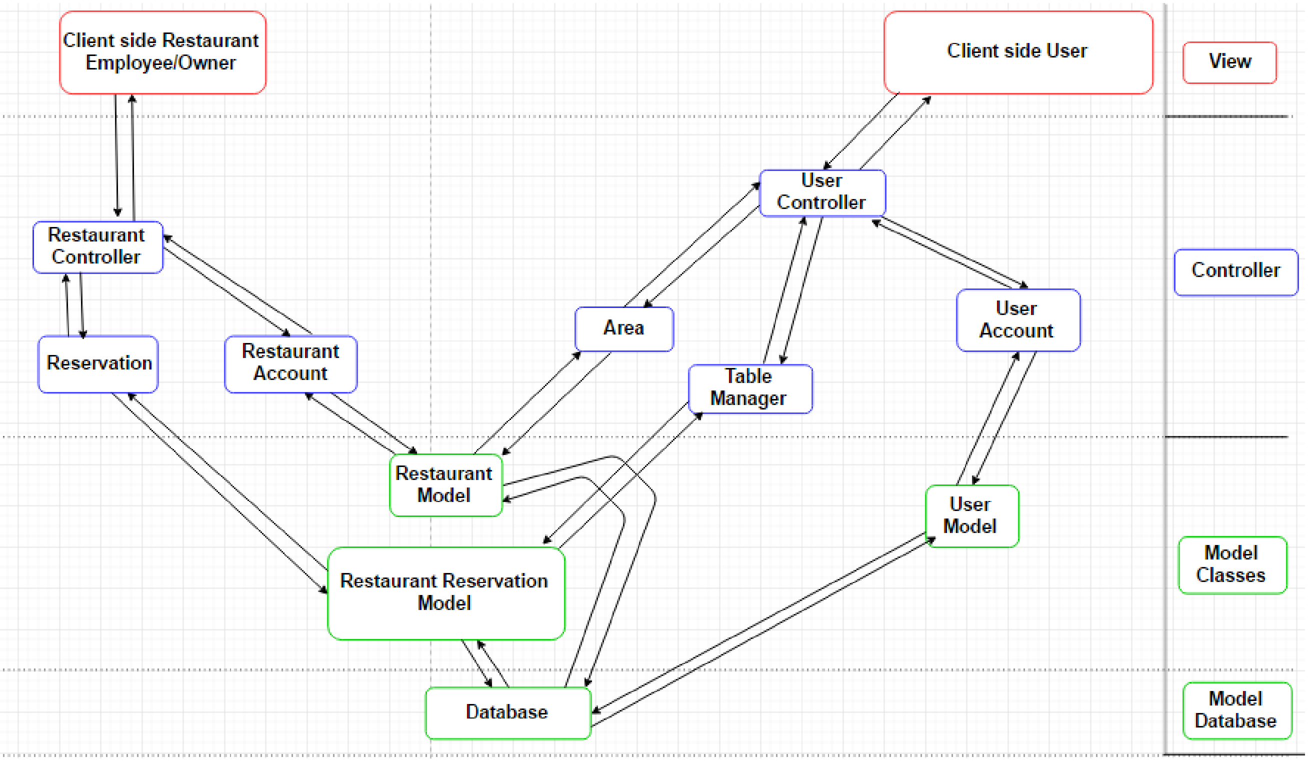
System Architecture

There are two main models to be used as a reference to this system, the first is the event driven model where data is updated in real time data in both user and Restaurant Employee/Owner GUI. The second one is MVC (Model View Controller) pattern in this case is simpler to update data not in real time but only when an user refreshes the page; furthermore this pattern ensures a modular system enabling people to work on interfaces between model, view and controller. The event driven model could use the new HTML5 technology called web sockets; however, the integration of it in both server and client require a deep knowledge of some reusable components of external libraries, therefore due to current knowledge of the technologies in the group this model will not be used. The MVC pattern instead could be used because in the case of a web application is generally more appropriate, in fact the model could be used in this way:

- Model: database and server classes which interfacing with the database.
- View: browser GUI, therefore HTML, CSS, JavaScript and all the technologies that could regard presentation data (XML, JSON)
- Controller: server classes that manage client requests, validate data, manage the application logic and interact with the model classes and the view

Nevertheless this approach could lead to an increasing difficulty of the overall system, therefore at first glance only the requirements with higher priority will be satisfied.
Considering the requirements with higher priority a first architecture of the system based on the MVC pattern is proposed, in the figure below the operations considered are:

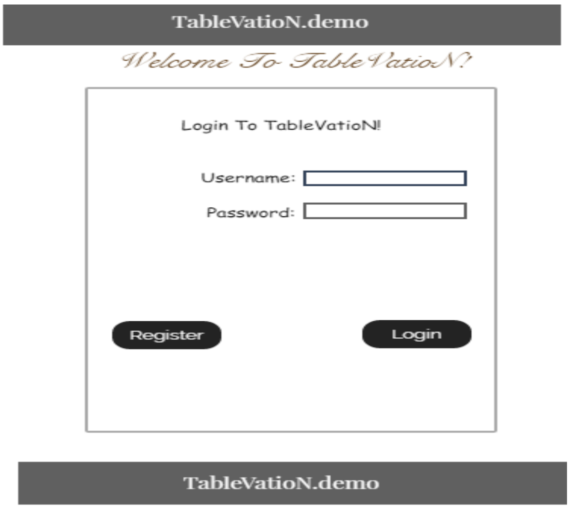
Restaurant Employee/Owner Operations	User Operations
• Create an account • Login • The view of reservations • Update of the reservations	• Create an account • Login • View restaurants in a specific area • Make a reservation for a specific restaurant • View the reservations made



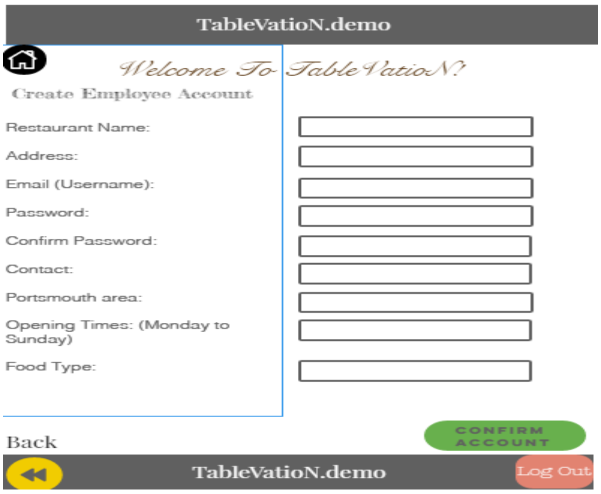
In this figure the view is represented by the restaurant view and the client side view, this is because in Tablevation there are two different users. The controller is represented by the classes that manage the HTTP requests from the client that are ‘Restaurant controller’ and ‘User controller’ and the classes that are the application logic, validate data and interact with the model classes (Reservation, Restaurant account, Area, Table manager, User account). By definition the model is divided in two parts the business logic and the database; the business logic is represented by the classes that manage data and interact with the database (Restaurant model, Restaurant reservation model, User model), while the database simply responds to the queries of the model classes.

To give an idea of how this architecture works in the context of Tablevation system we propose an example of an operation executed on the system. The view reservations operation made by a restaurant user is executed starting from the ‘client side restaurant employee/owner’ (the browser) that send a HTTP request to the server, the ‘Restaurant controller’ manage the request and asks to the ‘Reservation’ class to interact with the ‘Restaurant reservation model’ that will get the data from the database send it back to the view that will update the GUI.

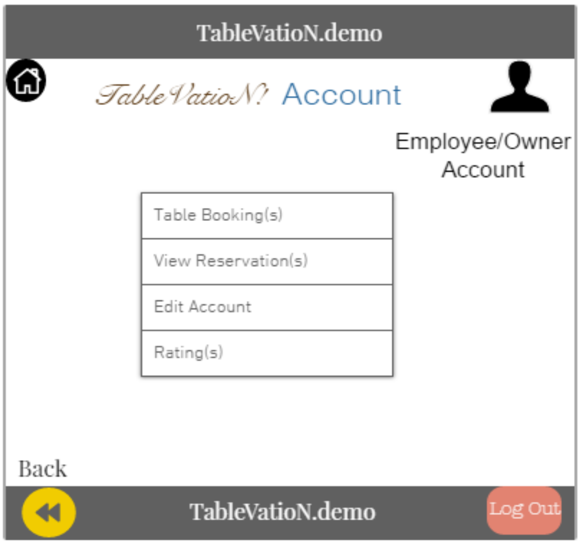
Mock ups of the System's UI for the Restaurant Side



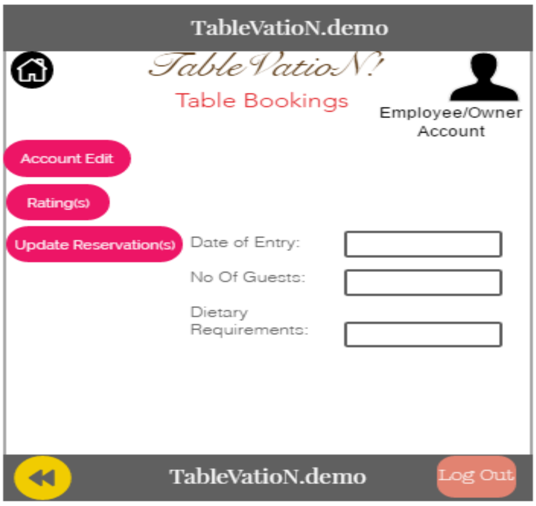
1. Asks the Employee/Owner or user to enter their login details or to register if they do not already have an account. This UI will be used for both views.



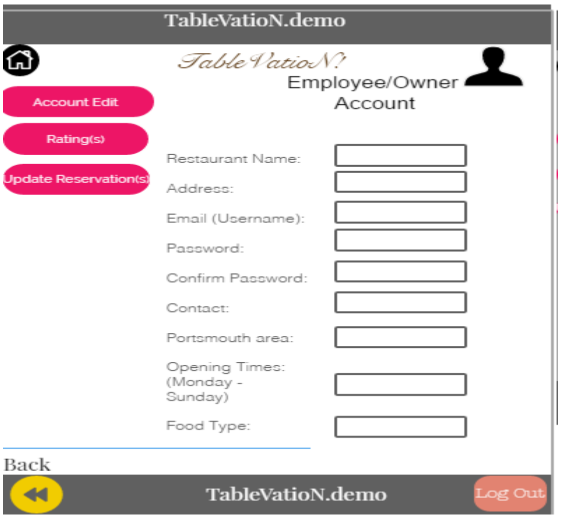
2. If the Employee/Owner selects register, they will have to enter the details required.



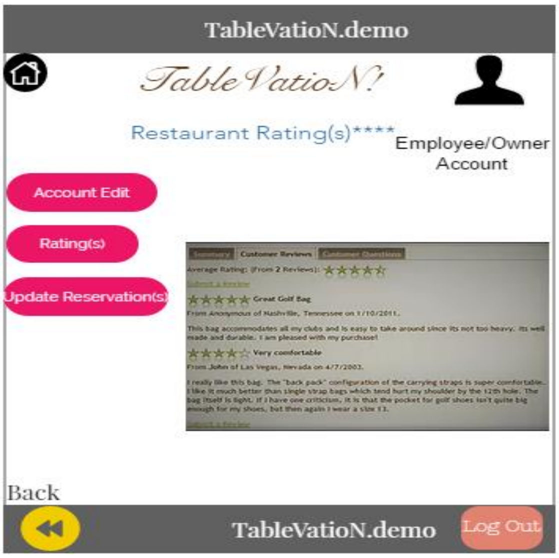
3. When the Employee/Owner is logged on they will see this screen where they can select one of the options.



4. If the Employee/Owner clicks on the first option they will see this screen. Which will show the date/ number of guests and the dietary requirement needed for the meal.



5. When the Employee/Owner clicks on the Account Edit they can edit their details they first initially put in.



Summary | Customer Reviews | Customer Questions

Average Rating: (From 2 Reviews): ★★★★☆

Submit a Review

★★★★★ Great Golf Bag

From Anonymous of Nashville, Tennessee on 1/10/2011.

This bag accommodates all my clubs and is easy to take around since its not too heavy. Its well made and durable. I am pleased with my purchase!

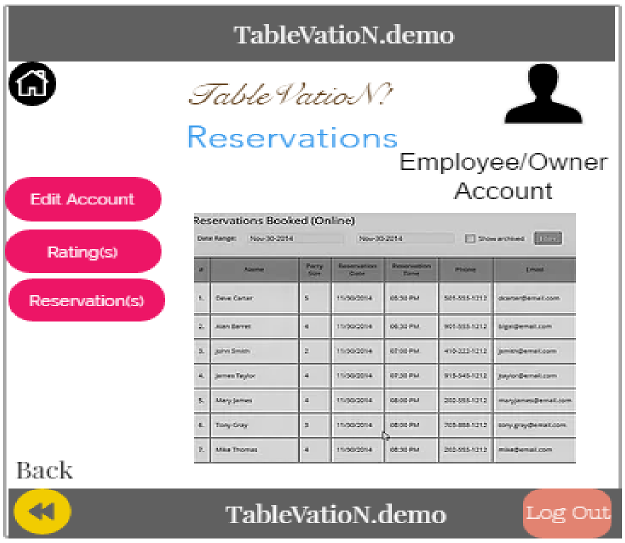
★★★★☆ Very comfortable

From John of Las Vegas, Nevada on 4/7/2003.

I really like this bag. The "back pack" configuration of the carrying straps is super comfortable. I like it much better than single strap bags which tend hurt my shoulder by the 12th hole. The bag itself is light. If I have one criticism, it is that the pocket for golf shoes isn't quite big enough for my shoes, but then again I wear a size 13.

Submit a Review

6. When the Employee/Owner clicks on the Ratings they can see what the customer have said or thought about the restaurant and can give it a star rating.

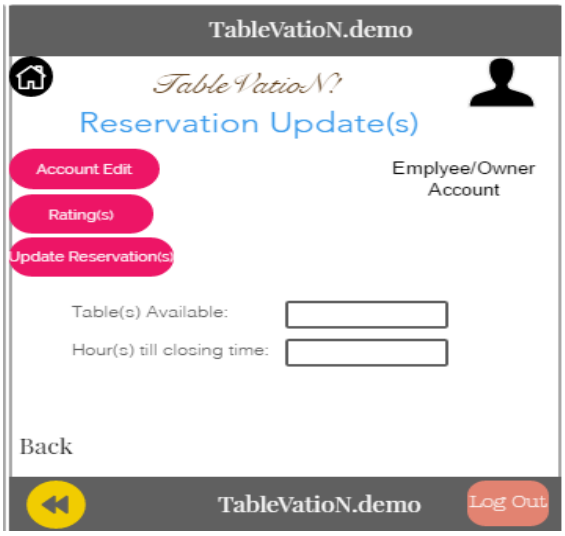


Reservations Booked (Online)

Date Range: Nov-30-2014 Nov-30-2014 □ Show archived

#	Name	Party Size	Reservation Date	Reservation Time	Phone	Email
1.	Dave Carter	5	11/30/2014	05:30 PM	501-555-1212	dcarter@email.com
2.	Alan Barret	4	11/30/2014	06:30 PM	901-555-1212	bigal@email.com
3.	John Smith	2	11/30/2014	07:00 PM	410-222-1212	jsmith@email.com
4.	James Taylor	4	11/30/2014	07:30 PM	915-545-1212	jtaylor@email.com
5.	Mary James	4	11/30/2014	08:00 PM	202-555-1212	maryjames@email.com
6.	Tony Gray	3	11/30/2014	08:00 PM	703-888-1212	tony.gray@email.com
7.	Mike Thomas	4	11/30/2014	08:30 PM	202-555-1212	mike@email.com

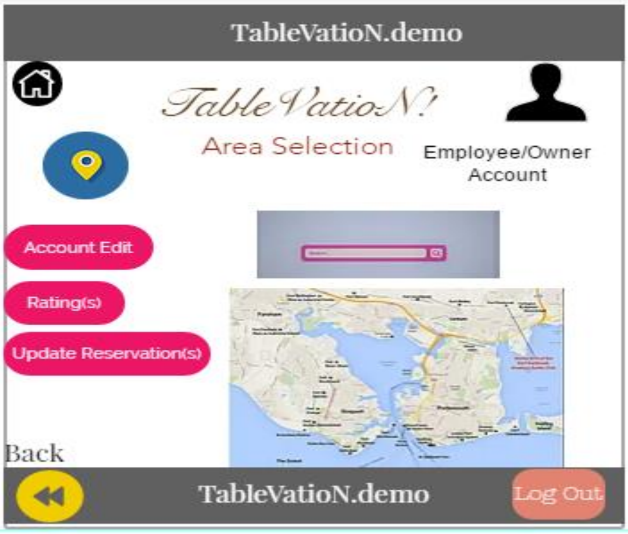
7. When the Employee/Owner selects the View Reservation option this screen will appear and they will be able to see what tables have been reserved/ were reserved and can update the reservation.



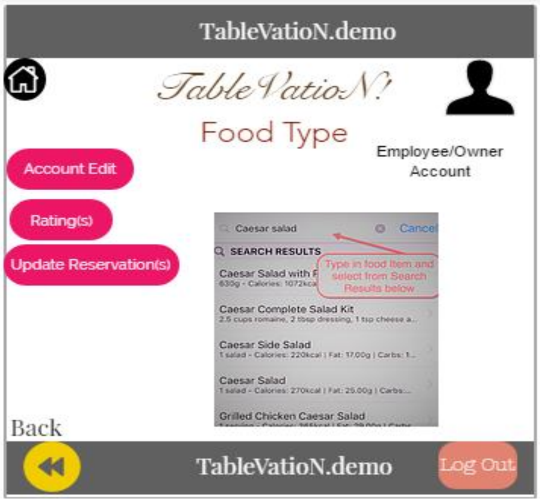
Table(s) Available:

Hour(s) till closing time:

8. When the Employee/Owner clicks on Update Reservation they can edit the table availability.



9.On this page the Employee/ Owner can select the area in which the restaurant will be based.



Caesar Salad with F
630g - Calories: 1072kca

Caesar Complete Salad Kit
2.5 cups romaine, 2 tbsp dressing, 1 tsp cheese a...

Caesar Side Salad
1 salad - Calories: 220kcal | Fat: 17.00g | Carbs: 1...

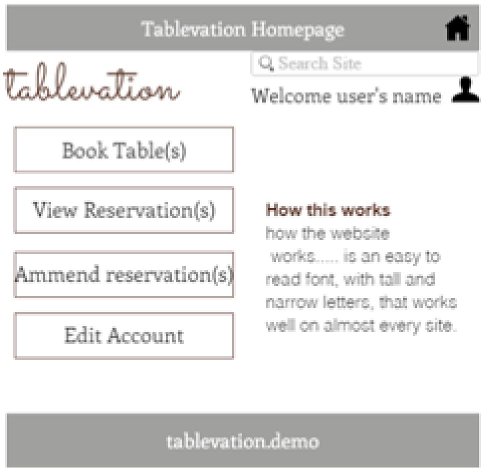
Caesar Salad
1 salad - Calories: 270kcal | Fat: 25.00g | Carbs:...

Grilled Chicken Caesar Salad

10. In this page the employee/Owner can update the food types of their food.

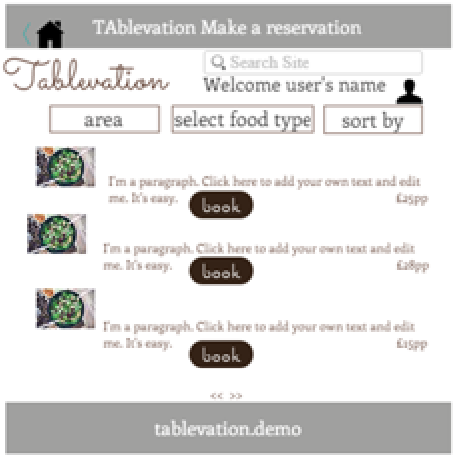
Mock ups of the System's UI for the User Side

When the user logs in
the system should
give these options in

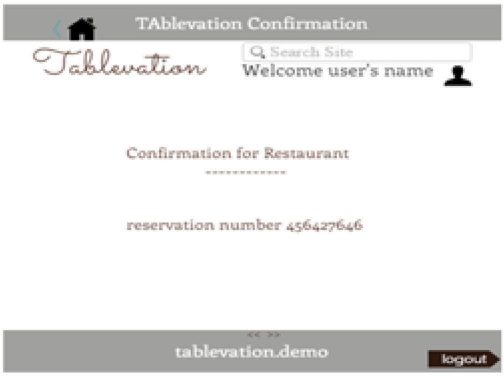


When Book table

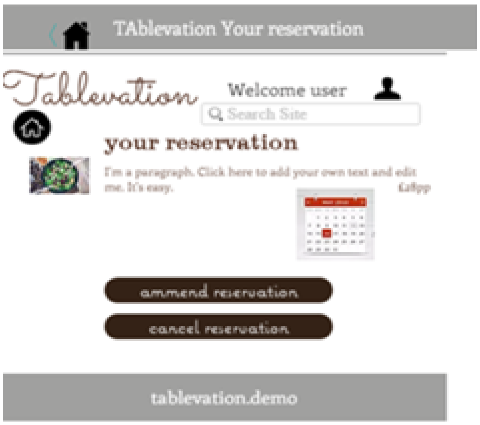
When the user enters the required information the system should generate a list of available places in the chosen area


TAblevation Make a reservation
Tablevation
Search Site
Welcome user's name
area
select food type
sort by
I'm a paragraph. Click here to add your own text and edit me. It's easy.
book
£25pp
I'm a paragraph. Click here to add your own text and edit me. It's easy.
book
£28pp
I'm a paragraph. Click here to add your own text and edit me. It's easy.
book
£15pp
<< >>
tablevation.demo

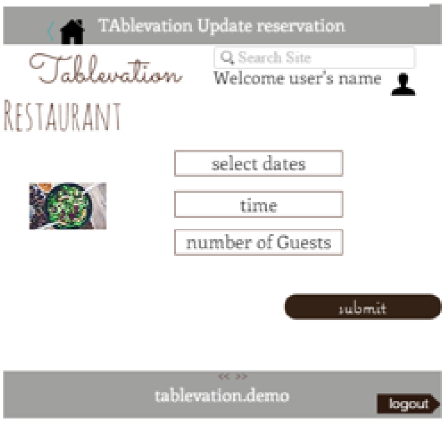
The user should receive confirmation when they have booked a place



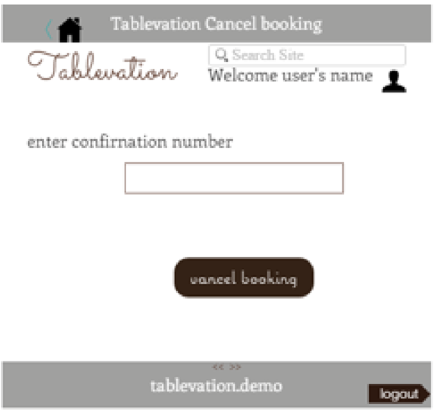
View reservation page should show the user the dates and information of their booking and an option to either cancel or amend booking if they are not happy with anything



The user should have the following options when they select update reservation/ amend reservation



The user can use their confirmation number to cancel booking



Declarations:

- I, Lucinda Ashdown, declare my contribution to the Use Case and Description of this Design Documentation.

Lucinda Ashdown LAshdown

- I, Jamie Toloui, declare my contribution to the Sequence Diagrams of this Design Documentation.

Jamie Toloui JToloi

- I, Benjamin Harris, declare my contribution to the Sequence Diagrams of this Design Documentation.

Benjamin Harris BHarris

- I, Francesco David Nota, declare my contribution to the System Architecture and Sequence Diagrams of this Design Documentation.

Francesco David Nota Francesco David Nota

- I, Mitchel Tendai Gwaze, declare my contribution to the UI Mock ups of this Design Documentation.

Mitchel Tendai Gwaze: Mitchel

- I, Abraham Nwokolo, declare my contribution to the UI Mock ups of this Design Documentation.

Abraham Nwokolo Abraham