Restaurant Booking System

Requirements Specification and Analysis

Final Version

17.11.2019

Volkan Davşan

Ali Atahan Topal

Oğuz Kaan Yazan

Berke Biten

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc496873294)

[1.1. Purpose of the System 1](#_Toc496873295)

[1.2. Scope of the System 1](#_Toc496873296)

[1.3. Objectives and Success Criteria of the Project 1](#_Toc496873297)

[1.4. Overview](#_Toc496873299) 2

[2. Current System 2](#_Toc496873300)

[3. Proposed System 3](#_Toc496873301)

[3.1. Overview 3](#_Toc496873302)

[3.2. Functional Requirements 4](#_Toc496873303)

[3.3. Nonfunctional Requirements 5](#_Toc496873304)

[3.4. System Models 5](#_Toc496873313)

[Scenarios 5](#_Toc496873314)

[Use case model 13](#_Toc496873315)

[Object model 30](#_Toc496873316)

[Dynamic model 30](#_Toc496873317)

[User interface—navigational paths and screen mock-ups 30](#_Toc496873318)

[3.5. Project Schedule 30](#_Toc496873319)

[4. Glossary 30](#_Toc496873320)

[5. References 32](#_Toc496873321)

REQUIREMENTS ANALYSIS DOCUMENT[1]

**INTRODUCTION**

## Purpose of the System

Our goal is to make things easier for people that wants to make a reservation from a specific restaurants. When going to a restaurant people face with a lot of discomforts such as you are going to a restaurant but it is full, so you return empty-handed or you go to a restaurant but you regret it because you didn’t have enough information when you go there. By using our site, people can easily avoid these discomforts because you can see whether the restaurant is full or not and you can also see all of the informations about that restaurant. Also, you can look for more than one restaurant at a time this allows people to search restaurants until he/she finds the one that meets their expectations. So, by using our site people can save a lot of time, also they will have informations on that restaurant beforehand.

## Scope of the System

In this project, we designed a website which is about restaurant booking. This project has 4 users and these are Guest, User, RestaurantOwner and Admin. Guests can do sign up and restaurant sign up operations. They can search for a restaurant and also they have the freedom to visit restaurant profiles and read the reviews that has been made before in order to make a decision. Users can do booking operations, can drop a review. They play an important role whether the site is succesfull or not because the site has been made to make things easier for them. RestaurantOwners can register their restaurants to our site if their informations about the restaurant is legit. More restaurants mean that the Users can book from many different sources so this creates diversity. Finally, Admin’s duty is to make sure that the site operates succesfully.He/she can also do operations such as ban or warn, accept/decline restaurant signups. Basically, Admin is responsible for maintaining the site.

## Objectives and Success Criteria of the Project

Our objective is to create a platform which helps users to choose a restaurant according to their wishes because in this site there will be a lot of restaurants to choose from. We are also trying to make the reservation process much more smoother, like instead of going to a restaurant and do the operation there(which can cause a lot of problems like maybe it is full etc.) you can do it at home at any time you want. These are the objectives that we are trying to achieve by doing this project.

Our success criteria of the project can be measured by how many people are using this site, do they have any trouble while using the sites’ functions, how many people are actually do a booking instead of just looking and leaving or are they satisfied by the interface of the site.

## Overview

So in this RAD we have introduction parts, current system and proposed system, functional overview of the system, functional and nonfunctional requirements, system models which includes scenarios, use case models, object models, dynamic models and user interface which includes navigational paths and screen mock-ups. We also have gannt chart which we set up some milestones and glossary to establish a clear terminology and finally we have references part where we provide a complete list of all documents referenced.

We started our RAD by doing introduction part which includes our purpose, scope, objective and success criterias and defitions. In these parts our purpose is to make the reader understand what we are trying to accomplish, what is our purpose and what is our end-game.

After that we do the current system section, in this section we talked about some disadvantages of current system which gets used by a number of users. We can also say that, in this part we listed all of the disadvantages that we are trying to remove or improve in our system.

In proposed system section, we talked about our system, which we are trying to improve all of the current system sections’ disadvantages. We also explained what can be done in order to improve them and how to do it. Basically we explained what is our systems advantages over the current system.

In functional requirements part, we explained what functions each of our users’ can do. There are different functions for User, Guest, Admin, RestaurantOwner and each of them has different functions and same functions and we specified them.

In nonfunctional requirements part, we divided our system to 5 sub-topics and analyze it. These were Usability, Reliability, Performance, Supportability, Implementation and we explained how our site fulfill these conditions.

In System Models part, we first wrote our scenarios then by looking at this scenarios we wrote use-case scenarios which was a formal version of normal scenarios. Then by looking at this we made use-case diagrams. Then we made our object model by creating UML class diagrams. Finally, we finished our dynamic model which consists of sequence diagrams which represents the interactions among a set of objects during a single use case.

We prepared glossary as well to establish a clear terminology and we finished by providing a references which is a complete list of all documents referenced elsewhere in the RAD

**2- Current System**

The reservation system is being used as a calling via telephone. The person who wants to make a reservation is named Ali calls the restaurant he or she wants to book and gives details about a spesific date, time and number of people that will attend to this reservation. The person who represents the restaurant answers by givingthe details about the given information by Ali and lets Ali know that the reservation is available or not in every detail. Such as the date you picked is available or not, the chosen time is available or not and lastly the number of people is acceptable by the restaurant. In addition Ali can go to the restaurant physically and arrange the reservation acording to his wishes at therestaurant’s reception. But the above methods does not give away the qualiy of the restaurant Ali went in order to make the reservation. If a person who don’t know the quality of the restaurant and if Ali went to the restaurant himself to make the reservation he will not have an idea about whether there is an available date and time or not. So, he can not be sure about whether or not reserve this restaurant. Also the caller can be made to wait to make a reservation because of a busy line or a communication errors. The caller even can not reach the restaurant due to some network problems.

**3. Proposed System**

The proposed system to be designed is not going to replace the current system but improve it to make the waiting and the errors lesser than the current system. With the system the person who wants to book a restaurant that is registered in the system, can see when the restaurant is available and the amount people that can attend in a spesific time and date. With use of this The reservation can be made more easier than that of a calling or going to the restaurant phsycially. The system will give information about the quality of the food for each restaurant by determining the comments and reviews made by the previous customers. The customers who wishes to call to make a reservation can still search for the restaurant from the system and have an idea whether to reserve or not and make the call after searching and being sure to book from that restaurant. With these we resolve the uncertainty problem that we spoke earlier in the Current System section. Because the system is online the reservations will be made online. So, the customer will not have to call or come to restaurant physically. With the system the booking can be made faster and more time efficient and the calling customers will not be have to wait to make the reservation through the phone. The restaurant owners will be able to register their restaurant to the system with little effort to fill the registration form. After the administrator accepts the registeration of the restaurant, customers of the restaurant can book from the restaurant through the system. Restaurants that will be in system will have ratings and these rating will be visible to any customer and this will make the desicion of which restaurant to book easier and quicker.

**3.1. Overview**

In this part, we describe the functional requirements and the explanation of the actors that will use the functionalities. We have four users in our system These are Guest, User, Restaurant Owner and Administrator. First, is the Guest. The Guest has access to searching restaurant and filter the search results, see the location of the restaurants, view the reviews to the restaurants, view the highest rated restaurants, sign in, sign up (both for becoming User or Restaurant Owner) and forgot password functions. These steps are not requires a logging in action to work, meaning a customer who doesn’t have an account in the system can look to the restaurants location, reviews, rating ext. Except booking and the functionalities that comes with booking.

Second is the User. Users has access to everything that guests has access to and in addition Users can sign out, book a restaurant, edit his bookings, edit his own profile, comment to a restaurant he has booked before, submit a rquest ticket from the Administrator and view the suggested restaurant for themselves. The functions that the User has access to except for the ones that the guest has access to too are requires a logged in User for to operate. Meaning, without logging in to the system the user will not be able to book a restaurant, edit his/her restaurant bookings, comment about a restaurant he/she has booked before, view the suggested restaurant for themselves, submitting a request ticket from the administrator, changing the current password and lastly logging out.

Third, is the Restaurant Owner. Restaurant Owners has access to his restaurants own profile page and editing this profile page, loggin in as a Restaurant owner and signing out, viewing the booking to their restaurant, replying reviews that left to their restaurant page, view the past reviews for the restaurant he owns and changin the current password. These functions requires a logging in action and on top of that the logged in account should be a restaurant owner account.

Fourth, is the Administrator. Administrators has access to view restaurant sign-ups, accept/decline restaurant sign-up requests, view request tickets, respond request tickets ban a user from the website, warn a user, ban a restaurant from the website, warn a restaurant, edit restaurant profiles and edit user profiles. These functions requires a logging action an on top of that the logged in user must be an administrator.

**3.2. Functional Requirements**

The main function of our system is create bookings for a restaurant. Booking can be done by User only. User must be logged in to his/her account to create a booking.When he/she logged in there will be a searching area in the homepage.In that area , User fills the form and makes a search. After searching , he/she sees the list of reasturants that can be booked by User.In that phase , there will be a filter field in the left of the page and User will be able to filter his/her searching. Then he/she chooses a restaurant from that list and after fills the booking form User will have a booking in that restaurant with informations that given by him. Also our system will give the opportunity of edit the bookings. When User wants change some informations (party size , time etc.) of his/her booking , he/she comes the edit booking page and changes the informations.Users can also cancel his/her booking.There will be a constraint about cancelling and editing of a booking.The time of the booking that will cancelled or editted must be at least 1 hour later of editting/cancelling time.User will be able to drop a review to the restaurant that he/she booked.Also RestaurantOwners can reply to the review of his/her restaurants. When User has a complaint , they can submit a request ticket to our admins. In that way , we can improve our website.

To create a booking , User must follow this steps (Login,Search,Filter…). As we mentioned before , booking can be done by User only. But some of the functions (searching,filtering a search...) that made by User to booking can be also done by Guest.

Booking is most significant function of our system and there should be also Restaurant profiles that will be booked. Restaurant profiles created by RestaurantOwner. RestaurantOwner signes up his restaurant to the system , gives some informations about his/her restaurant and waits for a booking.

Users and RestaurantOwners will communicate with each other in our system. Therefore, someone should manage and control the system. Our administrators stands for this duty. Like we said , administrators manages the system. They can ban/warn a User or RestaurantOwners if they noticed some wrong attitude. Also , when a RestaurantOwner registered to the system they can accept or decline this restaurant. We want to ensure a good service to our users. So , if any RestaurantOwner gives a missing or wrong information our administrators will not accept this restaurant.As we mentioned before , Users and RestaurantOwners can submit a request ticket.Our admins view and responds these tickets.

**3.3 Nonfunctional Requirements**

### **Usability** Any user who knows English can use our system. We made our system very clear.We will not want any unrelated informations from the user. Users can book for a restaurant (our main function) in just 3 clicks. So we can say that our system is easy to use.

### **Reliability** Our system must run 100% of time.So , our users can make a reservation at any time.Also , even if user enters a wrong input our system can still continue to work without any kind of error.

### **Performance** We don’t want our users to wait too long when they’re searching , booking , registering etc. Therefore our system must be able to react in at most 1 minute to the our users.

***Supportability*** Our system can easily adapt to new changes such as restaurant adding, updating

### **Implementation** We will use HTML, CSS, PHP, SQL, javaScript programming languages for this system.

## 3.4. System Models

### Scenarios

**Scenario Name:** Sign In

**Participating Actor Instances:** Berke: Guest

**Flow of Events:**

1- Berke wants to log in to his account and clicks the sign in button on the home page.

2- Berke fills the username and password fields.

3- Berke clicks the confirm button and waits to be redirected to the home page.

4- If username and password are matching, Berke is logged in and redirected to the home page.

5- If username and password are not matching, Berke gets a feedback that says “username or password is incorrect”

6- If Berke leaves one of the fields empty, gets a feedback that declares which field is empty.

**Scenario Name:** Sign Up

**Participating Actor Instances:** Berke: Guest

**Flow of Events:**

1- Berke wants to create an account and clicks to the sign-up button on the home page.

2- Berke fills the first name, last name, username, password, re-enter password, email fields and clicks the submit button.

3- If every entry in the form is appropriate, account is created.

4- If username or email is being used, or password doesn’t meet the requirements, Berke gets a feedback and fills the form again.

5- If one of the fields is empty, Berke gets a feedback that indicates which field is empty.

6- Recovery Code for the account is shown to Berke.

**Scenario name :** Restaurant Sign Up

**Participating actor instances :** Aykut : Guest , Aziz :Administrator

**Flow of events :**

1- Aykut wants to register as a restaurant to the restaurant booking site.

2- Aykut clicks the Restaurant Sign Up button in the home page.

3- Aykut enters the First Name , Last Name , Restaurant Name , Email Address ,Password, Re-enter Password ,Phone Number and Restaurant Address on the opened form.

4- Aykut clicks the submit button and waits for an acknowledgment.

5- If any of the inputs is empty , Aykut get a warning that says “Please fill the gaps”.

6- If password and re-enter password fields does not match , he get a warning that says “Re-enter password field is not same with your password”.

7- If he fills the gaps correctly , he get an acknowledgment that says “Your application is received succesfully”.

8- Aziz gets a notification. He checks the correctness of informations submitted by Aykut and clicks the accept or decline button.

9- Aykut receives a mail about the result of the application.

**Scenario Name:** Sign Out

**Participating Actor Instances:** Volkan: User / Berke: Restaurant Owner/ Ali: Administrator

**Flow of Events:**

1- After completing their job, Volkan/Berke/Ali is done with the site and wants to logout. Clicks the sign-out button on the main page.

2- Volkan/Berke/Ali gets a warning which asks “Are you sure you want to sign-out?” button.

3- If he press “Yes” button, Volkan/Berke/Ali gets a feedback that says “You succesfully logged-out from your account.” If he press “No” button, he is still logged into the site.

4- No matter what happens, Volkan/Berke/Ali is redirected to home page in the end.

**Scenario name :** Restaurant Search

**Participating actor instances :** Oguz : User , Kaan : Guest

**Flow of events :**

1-Oguz or Kaan wants to search a restaurant.

2- He enters the name or the location of the restaurant on the input text area in the home page which is titled with “Search”.

3- He selects the date of the reservation from the date picker.

4- He selects the person number from the combo box and clicks the Search button.

5- If any of the input areas is empty , He get a warning that says “Please fill the gaps”.

**Scenario name :** Filter a Search

**Participating actor instances :** Oguz **:**  User , Kaan : Guest

**Flow of events :**

1- Oguz or Kaan wants to search a restaurant and filter his searching.

2- He makes the search.

3- In the page that comes after search , He sees the list of restaurants.

4- He selects the price, cuisine, seating option, time and rank of the restaurants.

5- After made his selections , He clicks the filter button and the page is updated.

6- In the updated page , He sees the restaurants that filtered according to his selections.

**Scenario Name:** Booking

**Actor Instances:** Özkan: User, Ali: RestaurantOwner

**Flow of Events:**

1- Özkan chooses the restaurant that he is going to book.

2- Decides the party size, date and the time in “Reservation” part.

3- Clicks the “Make a reservation” button.

3a- If the size is too much User gets a warning that says “Party size is too much.”

4- In the new tab, he fills up the necessary informations like first name,last name, phone number and email and he can also add a special request(optional).

5- Clicks the “Complete Reservation” button.

6a- If one of the areas are empty or contains invalid information then the booking can’t be done.

6- Ali gets notified about this booking.

7- Özkan gets notified that the booking is succesfull.

**Scenario Name:** View the Bookings of My Restaurant

**Participating Actor Instances:** Osman: RestaurantOwner

**Flow of Events:**

1- Osman wants to see the bookings that have been made for his restaurant.

2- Osman clicks the “Bookings for My Restaurant” button on the home page.

3- Osman is directed to the page that shows a list of bookings.

**Scenario name:** Booking Editing

**Participating Actor Instances:** User: Ali

**Flow of Events:**

1. Ali wants to edit his booking.
2. Then Ali enters his profile and clicks to the “My Bookings” button.
3. Then the User clicks the “Edit Booking” button of the booking he wants to edit.
4. The User can edit his bookings’ hour and the amount of people that will come until the last one hour of the booking. If the User wants to edit the hour of the booking the available hours will be seen to the user and if the User wants to edit the amount of people that will come, he just enters the new number of people that will attend and clicks to the “Apply” button.
5. If the amount of people that will come is greater than the number of available seats in the restaurant at that hour a notification pops up and says, “The amount you entered is not available at his hour.”.
6. The booking will be edited according to the User’s wishes and a feedback will be given to the related Restaurant Owner that there is a change in one of the bookings.
7. **Scenario Name:** Booking Cancellation
8. **Participating Actor Instances:** Berke: User, Ali: Restaurant Owner
9. **Flow of Events:**
10. 1- Berke wants to cancel one of his bookings (current time is at least 1 hours earlier than the book time)
11. 2- Berke enters his profile and clicks to the “my bookings” button.
12. 3- Berke clicks the “cancel this booking” button near his/her booking.
13. 4- Berke clicks “yes” from the alert that asks “are you sure you want to cancel this booking?”
14. 5- Berke gets a feedback that says “your booking has been cancelled.”
15. 6- Ali receives a notification.
16. 7- If Berke clicks “no” in the alert, nothing happens and Berke returns to his bookings.

**Scenario Name:** Edit Restaurant Profile

**Participating Actor Instances:** Volkan: Restaurant Owner

**Flow of Events:**

1- Volkan wants to change his restaurant profile, enters his restaurant page.

2- Clicks the restaurant profile and selects the “edit profile” button.

3- Volkan edits his restaurant profile informations like, restaurant name, phone number, restaurant state or photos about the restaurant.

4- Clicks the confirm button.

4a- If one of the area is empty Volkan gets a warning which lets him know which area is empty.

5- Volkan receives an acknowledgment that his restaurant is updated.

**Scenario name :** Edit User Profile

**Participating actor instances :** Ceren : User

**Flow of events :**

1-Ceren wants to edit her profile.

2- She clicks the “Profile” button in the home page.

3- In the profile page , She sees her informations. She clicks the “Edit My Profile”.

4- She edits the informations that she wants to change.

5- She clicks the “Confirm” button after changing her informations.

6- If there is a problem with her informations , she get a warning and re-enters the problematic informations.

7- She receives an acknowledgment.

**Scenario name:** Forgot Password

**Participating Actor Instances:** Guest: Ali

**Flow of Events:**

1. Ali forgot his password and wants to change his password from the forgot password page and clicks the “Forgot Password” button.
2. A new page opens and asks the recovery code given in sign up process and accessible from the profile of the User and two new password text fields.
3. Then Ali enters the recovery code and his new passwords.
4. If the recovery code is wrong or the two text fields that asks for the new passwords are not the same Ali will be redirected to the forgot password page and enters the recovery code and the new password fields.
5. If the recovery code is correct and the two new password fields are the same the password changes.
6. A feedback given to Ali that says his password has been changed.

**Scenario name:** Change Password

**Participating Actor Instances:** User: Ali, Restaurant Owner: Berke, Administrator: Volkan

**Flow of Events:**

1. Ali/Berke/Volkan wants to change his password.
2. Ali/Berke/Volkan enters his profile page and clicks to the “Account Settings” button.
3. Ali/Berke/Volkan clicks to “Change Password” button.
4. A page appears that asks for the old password and the new password twice.
5. After filling the fields, Ali/Berke/Volkan clicks to “Apply” button.
6. If the old password is wrong or two text fields that asks for the new password are not same or the password fields are empty, Ali/Berke/Volkan will go back to the Change Password page and enters the old password and new password fields again.
7. If the old password and the two new password fields are correct the password changes.
8. A feedback will be given to Ali/Berke/Volkan that says, “Your password has been changed.”.

**Scenario name:** Drop a Review

**Participating Actor Instances:** User: Ali

**Flow of Events:**

1. Ali wants to drop a review about a restaurant he has eaten before and searches for the restaurant he wants to drop a review to.
2. Then Ali clicks to the “Comment” button on the restaurant’s main page.
3. After clicking the button, a text area pops up.
4. Ali writes his thoughts about the restaurant and clicks to the “Submit” button.
5. If the text area is empty a notification pops up and says, “Text area cannot be empty.”.
6. The comment will be shown in the main page of the restaurant.

**Scenario Name:** Reply to Review

**Participating Actor Instances:** Ali: Restaurant Owner

**Flow of Events:**

1- Ali wants to reply to a review that has been dropped by Users’. Enters his restaurants and clicks the “review” button to see the reviews.

2- Then Ali chooses which review to reply and clicks the “reply” button.

2a- If there is no review about that restaurant then Ali can’t reply.

3- After clicking the button, text area pops up and he writes his “reply” to that review.

4- Clicks the “Send” button.

4a- If the text area is empty then Ali gets a warning “You have to fill the area” and he goes back to the same page again.

5- The reply will be shown in the review’s below in the restaurant page.

**Scenario Name:** View the past reviews

**Participating Actor Instances:** Ali: User, Atahan: Restaurant Owner, Berat: Guest, Volkan: Admin

**Flow of Events:**

1- Ali/Atahan/Berat/Volkan wants to pick a restaurant but in order to do that, he wants to read the reviews about that restaurant first. He enters the restaurant page.

2- Clicks the “reviews” tab to see the reviews that has been made before.

2a- If there is no review about that restaurant, Ali/Atahan/Berat/Volkan can’t see the reviews.

**Scenario Name:** Submit a Request Ticket

**Participating Actor Instances:** Berke: User or Volkan: Restaurant Owner

**Flow of Events:**

1- Berke or Volkan wants to request something from the administration.

2- Berke or Volkan clicks the support button at the top of the page.

3- Berke or Volkan clicks the “submit a ticket button” on the support page.

4- Berke or Volkan selects a category, and describes his request via the text area.

5- Berke or Volkan clicks the submit button.

6- If the text area, category or both is empty Berke or Volkan gets a feedback that indicates which field is empty.

7- Ticket is saved.

**Scenario Name:** Add Favorite Restaurant

**Participating Actor Instances:** Berke: User

**Flow of Events:**

1-Berke wants to add a restaurant to his favorites.

2-Berke finds the restaurant in website by past bookings, searching, most rated restaurants etc.

3- Berke enters the restaurant’s profile.

4- Berke clicks the star icon/button.

5- Berke gets a feedback “Restaurant is added to your favorites”.

**Scenario name:** Ban / Warning User

**Participating Actor Instances:** User: Volkan, Administrator: Ali

**Flow of Events:**

1. Ali wants to ban or warn a user named Volkan.
2. Ali searches Volkan and enters his profile page.
3. Ali clicks the Ban User or Warn User button.
4. In the opened page, Ali explains the reason of ban/warning.
5. Ali clicks the “Apply” button.
6. If the entered reason is empty the Admin receives a notification that says, “Users cannot be banned or warned without a reason.”.
7. Volkan receives a notification about being banned or warned.

**Scenario name :** Ban/Warning Restaurant

**Participating actor instances :** Hakan : Administrator , Büşra : Restaurant Owner

**Flow of events :**

1- Hakan wants to ban or warn a restaurant.

2- He searchs the profile of restaurant and selects the profile that he want to ban or warn.

3- He clicks the “Ban Restaurant” or “Warn Restaurant” button.

4- In the opened page , he explains the reason of ban/warning.

5- He clicks the “Apply” button.

6- Büşra receives a notification about banning or warning of her restaurant.

**Scenario name:** Accept/Decline Restaurant Sign-Ups

**Participating Actor Instances:** Restaurant Owner: Volkan, Administrator: Ali

**Flow of Events:**

1. Ali wants to accept or decline the restaurant sign-ups and clicks to the “View Restaurant Sign-ups” button.
2. Ali view the restaurant sign-up requests.
3. In that page Ali accepts or declines the sign-up requests.
4. Ali receives a notification that says, “Your answer has been saved.”.
5. Volkan receives a mail that the restaurant he owns is registered to the system or not.

**Scenario Name:** View Restaurant Signups

**Participating Actor Instances:** Volkan: Admin

**Flow of Events:**

1. He clicks the button “View Restaurant Signups” and enters the page.
2. In that page, he sees a bunch of signups from different RestaurantOwners.
3. If there is no restaurant signup, the he will see an empty page, but he will still be able to enter the page.

**Scenario name :** View Tickets

**Participating actor instances :** Kemal : Administrator

**Flow of events :**

1- Kemal wants to view the tickets send by Users.

2- Kemal logs in to his administrator account.

3- He opens the admin panel.

4- He clicks the “Tickets” button.

5-He views the tickets.

**Scenario Name:** Respond to Request Ticket

**Participating Actor Instances:** Berat: Administrator, Volkan: User, Ali: Restaurant Owner

**Flow of Events:**

1-Berat wants to respond one of the request tickets.

2-Berat clicks the view the request tickets button on admin panel.

3- If there are any unresponded requests, Berat chooses one of them and clicks on it.

4- Berat reads the request ticket and writes his respond into the text are and clicks OK button.

5- Volkan or Ali gets a notification that says “your request ticket has been responded.”

### Use case model

*Use case name* **Sign In**

*Participating Actors* Initiated by**Guest**

*Flow of events* 1. **Guest** clicks the Sign In button.

2. **SYSTEM** responds by showing the Sign In Form.

3. **Guest** fills the “username” and “password” fields correctly and clicks the Sign in button.

4. **SYSTEM** responds by showing up an acknowledgment that says “Successfully Logged In” and redirects the **Guest** to the homepage.

*Entry condition* **Guest** is in a page that he/she can see the sign in button.

*Exit conditions* **Guest** logs in and returns to the homepage.

*Exceptions* 3a. **Guest** enters the “username” and “password” fields are

incorrect.

4a. **SYSTEM** gives a feedback that says “Username or password is wrong”.

3b. **Guest** doesn’t fill the “username” or “password” field or both.

4b. **SYSTEM** gives a feedback that says “both fields must be filled.”

*Use case name* **Sign Up**

*Participating Actors* Initiated by**Guest**

*Flow of events* 1. **Guest** clicks the Sign Up button.

2. **SYSTEM** responds by showing the Sign Up Form.

3. **Guest** fills the “first name”, “last name”, “username”, “email”, “password” and “re- enter password” fields and clicks the Sign up button.

4. **SYSTEM** responds by showing up an acknowledgment that says “Account has been created”, shows the Recovery Code to the **Guest** and redirects the Guest to the Sign In page.

*Entry condition* **Guest** is in a page that he/she can see the sign up button.

*Exit conditions* Account is created and **Guest** is redirected to the homepage.

*Exceptions* 3a. **Guest** leaves one of the fields empty

4a. **SYSTEM** gives a feedback that says “All fields must be filled”.

3b. **“**username” or “e-mail” that **Guest** has entered are being used by another account.

4b. **SYSTEM** gives a feedback that says “ ‘username’ or ‘e-mail’ is being used by another account”

3c. **“**password” and “re-enter password” that **Guest** has entered are not matching.

4c. **SYSTEM** gives a feedback that says “‘password’ and ‘re-enter password’ fields are not matching.”

*Use case name* **Restaurant Sign Up**

*Participating Actors* Initiated by**Guest** , Communicates with **Administrator**

*Flow of events*

1. **Guest** clicks the Restaurant Sign Up button in the home page.

**2. SYSTEM** responds by showing up the Restaurant Sign Up Form.

3. **Guest** fills the “First Name” , “Last Name”,”Username”,“Restaurant

Name” , “E-mail Address” , “Password” , “Re-enter Password” ,

“Phone Number” and “Restaurant Address” fields and clicks the

“Submit” button.

4. **SYSTEM** responds by showing up an

acknowledgment that says “Your application

received succesfully” and send a notification to

the **Administrator.**

.

*Entry condition* The **Guest** is in the HomePage.

*Exit condition* **Administrator** received the application of **Guest.**

*Exceptions*  3a. **Guest** does not fill all the fields.

4a. **SYSTEM** responds by showing up a warning

window that says “Please fill the gaps.”

3b. **Guest** fills all the inputs but “Password” and “Re-enter Password”

fields are not same.

4b. **SYSTEM** responds by showing up a warning

that says “Re-enter password field is not same with

your password”.

*Use case name* **Sign out**

*Participating Actors* Initiated by**User / Restaurant Owner / Administrator**

*Flow of events*

1. The **User/Restaurant Owner/Administrator** clicks the “Sign out” button on the main page.

2. **SYSTEM** receives the command and responds by

showing a warning in pop-up dialog which says “Are

you sure you want to sign out?”.

1. The **User/Restaurant Owner/Administrator** sees the warning andclicks the “Yes” button.
2. **SYSTEM** acknowledges the command and signs out the

**User/Restaurant Owner/Administrator.**

*Entry condition* The **User / RestaurantOwner/Administrator** must be on the site where he/she can see the sign-out button.

*Exit conditions* - The **User/RestaurantOwner/Administrator** clicks sign out button and successfully signed out from his/her account.

*Exceptions* 3. **The User/Restaurant Owner/Administrator** clicks the

“No” button.

4. **SYSTEM** acknowledges the command and

cancels the sign-out operation.

*Use case name* **Restaurant Search**

*Participating Actors* Initiated by**User** or **Guest**

*Flow of events*

1. The **User** or **Guest** fills the “Name or Location” , “Pary Size” and “Date” fields that is in the home page (He/She fills the Search Form) and clicks the search button.

**2. SYSTEM** responds by showing up the list of restaurants according to informations that is given in the Search form.

*Entry condition* The **User or Guest** is in the HomePage.

*Exit condition* The **User** clicks to Search button and views the list of

restaurants.

*Exceptions*  1. **User** or **Guest** does not fill all the fields.

2.**SYSTEM** responds by showing up a warning

window that says “Please fill the gaps.”

*Use case name* **Filter a Search**

*Participating Actors* Initiated by**User** *or* **Guest**

*Flow of events*

1. **User** or **Guest** marks any of the checkbox(es) from “Cuisine” , “Price” , “Seating Option” , “Time” and “Rank” that is in the Filter part. Then he/she clicks the “Filter” button.

**2. SYSTEM** responds by updating the page with new restaurant list according to filters.

*Entry condition* **User** or **Guest** must have searched and be in the page that

comes after search.

*Exit condition* **User** or **Guest** views the new restaurant list.

*Use case name* **Booking**

*Participating Actors* Initiated by**User,** Communicates with **RestaurantOwner**

*Flow of events* 1.**User** decides the party size, date and the time from the

filters in “Reservation” part and clicks the “Make a reservation” button.

2. **SYSTEM** responds to it by opening up a new tab

which has the informations like first name,last name,

phone number and e-mail and also a special request which is

optional

3. **User** fills up these following informations and clicks the

“Complete Reservation” button.

1. **SYSTEM** responds to it by creating an alert which

says “You successfully reserved a place.”

5. RestaurantOwner gets notified about the booking.

*Entry condition* The **User** must choose the restaurant he is going to book.

*Exit conditions* -  **RestaurantOwner** gets notified about the booking and

**User** successfully booked a place from the restaurant .

*Exceptions 1.* **User’s**party size may not be fit into the Restaurant’s quota

*2.***SYSTEM** responds to it by showing a warning which

says “Party size is too much.”

*3.* **User** doesn’t fill atleast one of the following

informations and click the “Complete Reservation” button.

*4.* ***SYSTEM*** *responds to it by a warning which says*

“You have to fill all of the areas.”

*Use case name* **View the Bookings of My Restaurant**

*Participating Actors* Initiated by**Restaurant Owner**

*Flow of events* 1. **Restaurant Owner** clicks the “Bookings for My Restaurant” button.

2. **SYSTEM** responds by showing the List of Bookings.

*Entry condition* **Restaurant Owner** is in homepage.

*Exit conditions* **Restaurant Owner** is seeing the list of bookings.

*Use case name* **Booking Editing**

*Participating Actors* Initiated by**User**

Communicates with **RestaurantOwner**

*Flow of events* 1. The **User** enters his profile page and finds his bookings then

clicks to the “Edit Booking” button.

2. **SYSTEM** responds by showing a form and available hours, dates for the restaurant and the number of people available in that date and time. *(The form contains one date field, one time field and one number field for the User to fill.)*

3. The **User** receives the formfills it then clicks to “Confirm” button.

4. **SYSTEM** gets the filled form and gives a feedback that says, “Your booking has been edited successfully.” to the **User.**

5. The **User** receives the feedback and redirected to “My Bookings” page.

6.**SYSTEM** responds by sending a notification to the **RestaurantOwner** that says, “One of the bookings in your restaurant has been edited.”.

*Entry condition* The **User** must have at least one booking in the restaurant that he wants to edit before the deadline.

*Exit conditions* - The **User’s** booking successfully edited.

*Exceptions* 3a. The **User** submits an empty or at least one of the fields is empty form.

4a. **SYSTEM** gives a feedback that says, “At least one of the fields is empty.”. Redirects the **User** to the Booking Editing page.

3b. The **User** enters the number of people that will attend to the reservation more than available at that time.

4b. **SYSTEM** gives a feedback that says, “Number of people that is available is less than your entry.”. Redirects the **User** to the Booking Editing page.

*Use case name* **Booking Cancellation**

*Participating Actors* Initiated by**User**

Communicates with **Restaurant Owner**

*Flow of events* 1. **User** clicks the My Bookings button from his/her profile.

2. **SYSTEM** responds by showing the My Bookings page.

3. **User** finds the booking which he/she wants to cancel and clicks the cancel this booking button near the booking.

4. **SYSTEM** responds by showing up an alert that asks “Are you sure you want to cancel this booking?”

5. **User** clicks “Yes”.

6. **SYSTEM** responds by showing up an acknowledgment that says “Your booking has been cancelled.” And **SYSTEM** sends a notification to the related **Restaurant Owner.**

*Entry condition* **User** has at least one active booking.

*Exit conditions* Booking has been cancelled.

*Exceptions* 3a. Current time is not at least one hour earlier than the Booking time.

4a. **SYSTEM** gives a feedback that says “You can’t cancel a booking within last hour”.

5a. **User** clicks “No”.

6a. **SYSTEM** doesn’t execute the function and **User** returns to My Bookings page.

*Use case name* **Edit Restaurant Profile**

*Participating Actors* Initiated by**RestaurantOwner**

*Flow of events* 1. **RestaurantOwner** clicks the “Edit Profile” button.

2.**SYSTEM** receives the command and responds to

It by showing all of the restaurant profile informations.

3. **RestaurantOwner** edits his restaurant profile informations like,

restaurant name, phone number, restaurant state and photos

about the restaurant and clicks the “Confirm” button.

4. **SYSTEM** receives the command and responds to it by changing the informations which the RestaurantOwner edited.

5.After that **RestaurantOwner** directed to his edited version of restaurant profile.

*Entry condition* **RestaurantOwner** must be in the “Profile” page.

*Exit conditions* - The **RestaurantOwner’s** restaurant profile is successfully edited.

*Exceptions 3*. **RestaurantOwner** erase the information that is in one of the text-

areas and doesn’t fill it back. Clicks the confirm button.

4. **SYSTEM** responds to it by creating a warning which

says“Atleast one of the text-area is empty.”

*Use case name* **Edit User Profile**

*Participating Actors* Initiated by**User**

*Flow of events*

1. **User** clicks the “Edit My Profile” button in the profile

page.

2. **SYSTEM** responds by showing Edit Profile

Form. In that form there are filled fields with

**User**’s current informations.(Username,

First Name , Last Name , E-mail)

3. **User** edits his/her editable informations that he/she want

to change and clicks the “Confirm” button.

4. **SYSTEM** responds by changing the

Informations with new ones and view an

acknowledgment that says “Your profile editted sucessfuly”.

5. **User** directed to the profile page.

*Entry condition*  **User** must be in the Profile page.

*Exit condition* **User**’s profile is editted.

*Exceptions*  3. **User** erases any of the fields and does not fill.

4. **SYSTEM** responds by showing up a warning

that says “Please fill all the fields”.

*Use case name* **Forgot Password**

*Participating Actors* Initiated by**Guest**

*Flow of events* 1. The **Guest** clicks the “Forgot Password” button.

2. **SYSTEM** responds by showing up the forgot password form. *(the form contains one text field for the recovery code and two password fields for the new password).*

3. The **Guest** enters the recovery code and fills the two password fields with the new password and clicks to the “Done” button.

4. **SYSTEM** gets the filled form and gives a feedback that says, “Your password has been changed.” to the **Guest** and redirects to login page.*(the feedback will be given as a pop-up message.).*

*Entry condition* The **Guest** could not log in with the password he/she remembers.

*Exit conditions* - The **Guest** password has been successfully changed.

*Exceptions*  3a. The **Guest** enters the wrong recovery code.

4a. **SYSTEM** gives a feedback that says, “Your recovery code is not correct.”. Redirects the **Guest** to the “Forgot Password” page.

3b. The **Guest** enters the correct recovery code but the password

do not match.

4b. **SYSTEM** gives a feedback that says, “New

entered passwords do not match. Redirects the

**Guest** to the “Forgot Password” page.

3c. The **Guest** submits a form that at least one of the fields is empty.

4c. **SYSTEM** gives a feedback that says, “At least one of the fields are not filled.”. Redirects the

**Guest** to the “Forgot Password” page.

*Use case name* **Change Password**

*Participating Actors* Initiated by**User** OR **RestaurantOwner** OR **Administrator**

*Flow of events* 1. The **Actor** enters his account

settings from the profile page and clicks to the “Change Password”

button.

2. **SYSTEM** responds by showing the change

password form. *(the form contains three*

*password fields. One for the current password and two for*

*the new password.).*

3. The **Actor** fills the text fields for themself and clicks to the “Apply” button.

4. **SYSTEM** gets the filled form and gives a feedback that says, “Your password has been changed.” to the **Actor.** *(the feedback will be given as a pop-up message.).*

5. The **Actor** views the feedback and redirected to the account settings.

*Entry condition* The **Actor** must be logged in to system.

*Exit conditions* - The **Actor’s** password has been successfully changed.

*Exceptions* 3a. The **Actor** does not fill all the fields or leaves all fields empty.

4a. **SYSTEM** gives a feedback that says, “At least one field is not filled.”. Redirects the **Actor** to the Change Password page.

3b. The **Actor** fills the current password field incorrect.

4b. **SYSTEM** gives a feedback that says, “Current

password field is incorrect.”. Redirects the **Actor** to the Change Password page.

3c. The **Actor** fills the fields, but the new passwords do not match.

4c. **SYSTEM** gives a feedback that says, “New

password fields do not match.”. Redirects the **Actor** to the Change password page.

*Use case name* **Drop a Review**

*Participating Actors* Initiated by**User**

*Flow of events* 1. The **User** clicks the “Comment” button from the

home page of the restaurant.

**2. SYSTEM** responds by showing up a new text area. *(the text area is where the comments will be written).*

3. The **User** fills out the text area with his thoughts and once finished submits the form by clicking the “Submit” button.

**4. SYSTEM** gets the form and gives a feedback that says, “Your review has been saved.” to the **User.** *(the feedback will be given as a pop-up message).*

*Entry condition* The **User** must have at least one past booking to the restaurant he/she going to drop a review to.

*Exit conditions* - The **User** successfully dropped a review for the restaurant.

*Exceptions* 3. The **User** doesn’t fill the text area and submits it empty.

4. **SYSTEM** gives a feedback that says “Text

area cannot be empty.”.

*Use case name* **Reply to Review**

*Participating Actors* Initiated by**RestaurantOwner**

*Flow of events* 1. **RestaurantOwner** clicks the “Reply” button under the reviews of

other people.

2.**SYSTEM** receives the command and responds by

showing up an text-area for the **RestaurantOwner** to

write his/her thoughts.

3. **RestaurantOwner** writes his/her thoughts and clicks the “Send”

button.

4. **SYSTEM** receives the command and responds by

transmitting the reply message to the review page.

*Entry condition* **RestaurantOwner** should be in the “Review Page” where he can see

The reviews andthere must be a “review” in order for **RestaurantOwner**

to

*Exit conditions* **RestaurantOwner** is succesfully replies to a review and it is showing in

the “review” page.

*Exceptions* 1. If there is no review, **RestaurantOwne**r cannot reply.

3. **RestaurantOwner** clicks the “Send” button without writing anything.

4. **SYSTEM** receives the command and responds by

initiating a warning that says “ You have to fill the text-

area”

*Use case name* **See the Past Reviews**

*Participating Actors* Initiated by**User/RestaurantOwner/Guest/Admin**

*Flow of events 1.* **User/RestaurantOwner/Guest/Admin** clicks the

“reviews” button to see the reviews that has been made before.

2. **SYSTEM** receives the command and

responds it by showing the reviews(if any).

*Entry condition* The **User/RestaurantOwner/Guest/Admin** must be in the

Restaurant Page in order to see the “Reviews” tab.

*Exit conditions* - The **User/RestaurantOwner/Guest/Admin** can

successfully see the reviews about a restaurant (if there is any).

*Use case name* **Submit a Request Ticket**

*Participating Actors* Initiated by**User** OR **Restaurant Owner**

*Flow of events* 1. **User** or **Restaurant Owner** clicks the Support button.

2. **SYSTEM** responds by showing the Support page.

3. **User** or **Restaurant Owner** clicks the “Submit a ticket” button on the support page.

4. **SYSTEM** responds by showing up a form that contains “category” and “description” fields.

5. **User** or **Restaurant Owner** selects a category and describes his request via the text area and clicks the submit button.

6. **SYSTEM** responds by showing up an acknowledgment that says “Your request is saved and will be reviewed by an Admin.”, **User** or **Restaurant Owner** is redirected to the homepage.

*Entry condition* **User** or **Restaurant Owner** is in homepage.

*Exit conditions* Ticket is saved.

*Exceptions* 5a. **User** or **Restaurant Owner** clicks the submit button without selecting a category or leaves the text area empty or both.

6a. **SYSTEM** gives a feedback that warns user by declaring which field is empty.

*Use case name* **Add Favorite Restaurant**

*Participating Actors* Initiated by**User**

*Flow of events* 1. **User** clicks the star icon/button in the restaurant’s profile

2. **SYSTEM** responds by showing an alert box that contains a text which is “Restaurant is added to your favorites”.

*Entry condition* **User** is in a restaurant’s profile

*Exit conditions* Restaurant added to the favorites.

*Use case name* **Ban/Warning a Restaurant**

*Participating Actors* Initiated by**Administrator ,** Communicates with **RestaurantOwner**

*Flow of events*

**1. Administrator** searchs the profile of restaurant that he/she want to

ban/warn.

2. **SYSTEM** responds by showing relative restaurant

profile.

3. Administrator clicks the ban or warn button in the restaurant profile

page and writes the reason of ban/warn to the text area in the opened

tab.

4. **SYSTEM** responds by sending an e-mail about

banning or warning to the **RestaurantOwner.**

*Entry condition*  **Administrator** must be logged in to the system.

*Exit condition* **RestaurantOwner** banned or warned.

*Exceptions*  3. **Administrator** does not write the reason of ban/warn.

4. **SYSTEM** responds by showing up a warning

that says “You can not ban/warn a restaurant

without a reason”.

*Use case name* **Accept/Decline Restaurant Sign-ups**

*Participating Actors* Initiated by**Administrator**

Communicated with **Restaurant Owner**

*Flow of events* 1. The **Administrator** clicks the “View the Restaurant Sign-ups” button

from the home page of the restaurant.

**2. SYSTEM** responds by showing up the Restaurant Sign-up requests.

3. The **Administrator** views the requests and clicks “Accept” button.

**4. SYSTEM** gets the answer and gives a feedback that says, “Your answer has been saved.” to the **Administrator.** *(the feedback will be given as a pop-up message).*

5. The **User** receives an e-mail that says, “Your Restaurant has been officially registered to our system.”.

*Entry condition* The **Administrator** must have at least one Restaurant sign-up request to accept or decline.

*Exit conditions* - The **Administrator** successfully accepted or declined a request.

*Exceptions* 3. The **Administrator** declines the Restaurant Sign-up Request.

5. The **User** receives an e-mail that says, “Your Restaurant Sign-up request been declined.”.

*Use case name* **View Restaurant Signups**

*Participating Actors* Initiated by**Admin**

*Flow of events* 1. **Admin** clicks the “View Restaurant Signups” button.

2.**SYSTEM** receives the command and responds to

it by showing all of the restaurant signups in the open

page.

3.After that **Admin** will be directed into the page where he can

see all of the restaurant signups.

.

*Entry condition* **RestaurantOwner** must be in the Homepage where he can see the

“View Restaurant Signups” button.

*Exit conditions* - **Admin** successfully view the restaurant signups even though there

is none.

*Use case name* **View Tickets**

*Participating Actors* Initiated by**Administrator**

*Flow of events*

**1. Administrator** clicks the “Tickets” button on the admin panel.

**2. SYSTEM** responds by showing the tickets

send by Users.

*Entry condition*  **Administrator** must be logged in to the system.

*Exit condition* **Administrator** views the tickets

*Use case name* **Respond to Request Ticket**

*Participating Actors* Initiated by**Administrator**

Communicates with **User or Restaurant Owner**

*Flow of events* 1. **Administrator** clicks a request ticket from the Tickets page.

2. **SYSTEM** responds by showing the details of the ticket.

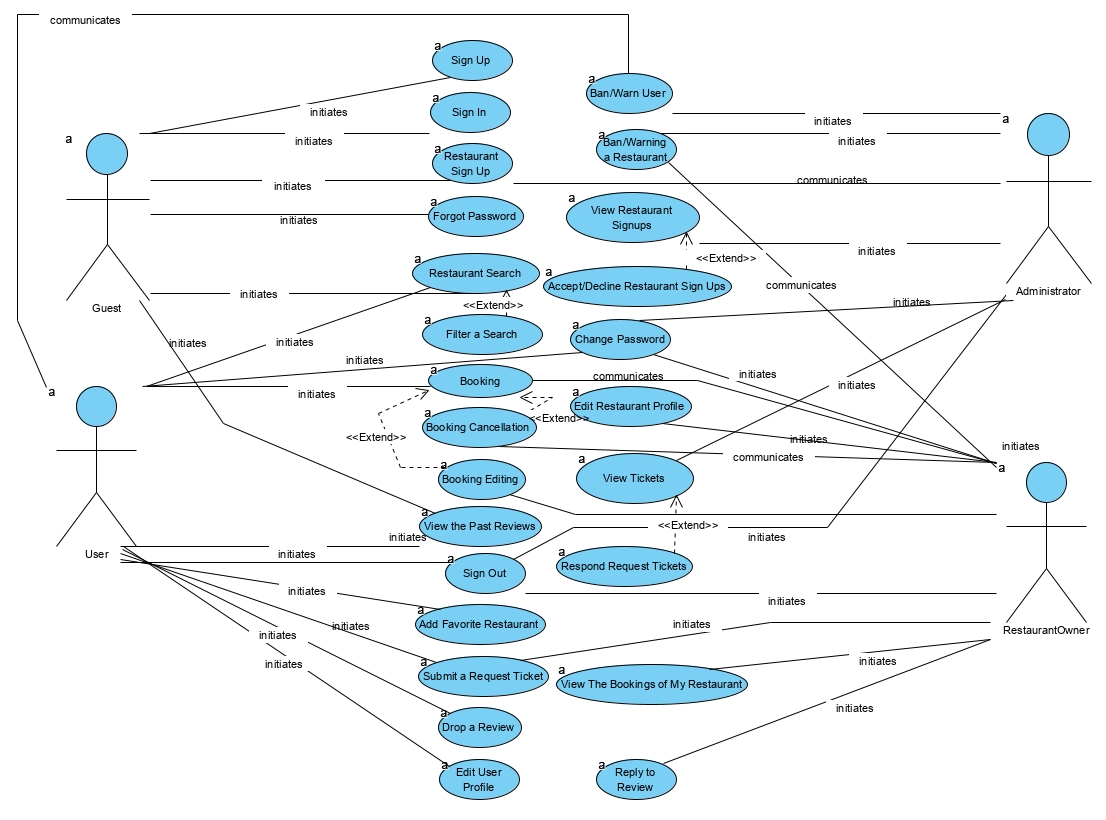
3.**Administrator** reads the ticket, writes a reply to the text area and clicks OK button.

4.**SYSTEM** responds by showing an acknowledgment that says “respond has been sent” and sends a notification to the related **User** or **Restaurant Owner**.

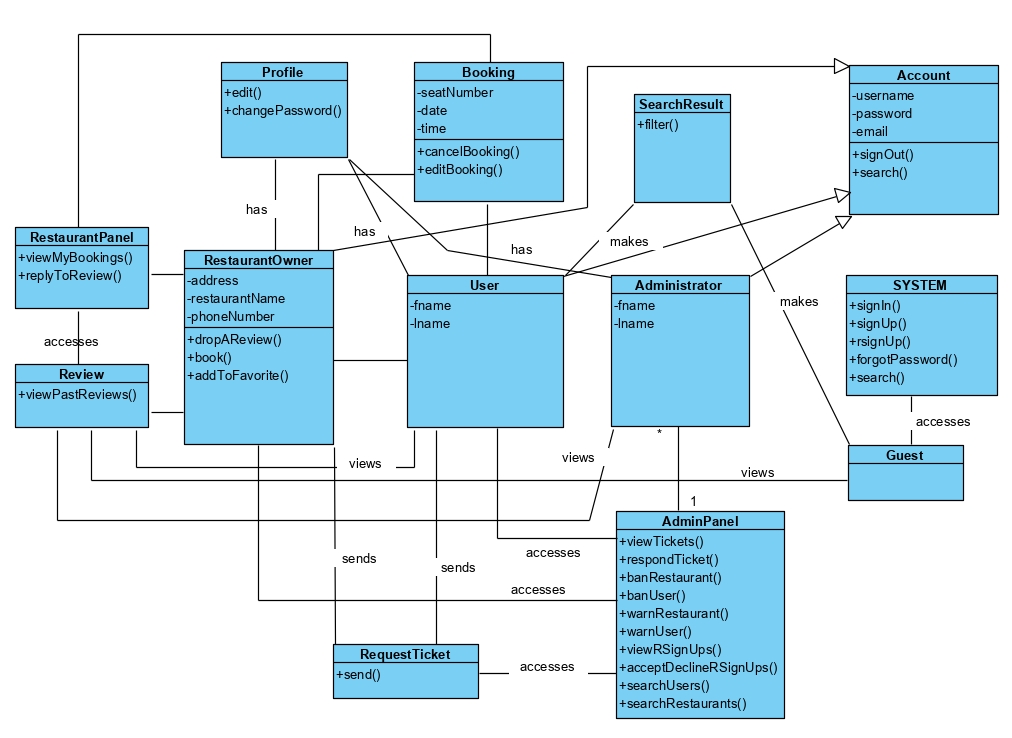
*Entry condition* **Administrator** must be viewing the request tickets.

*Exit conditions* **Administrator** responds to a ticket and **User or Restaurant Owner** receives a notification.

*Use-case Diagram:*



### Object model

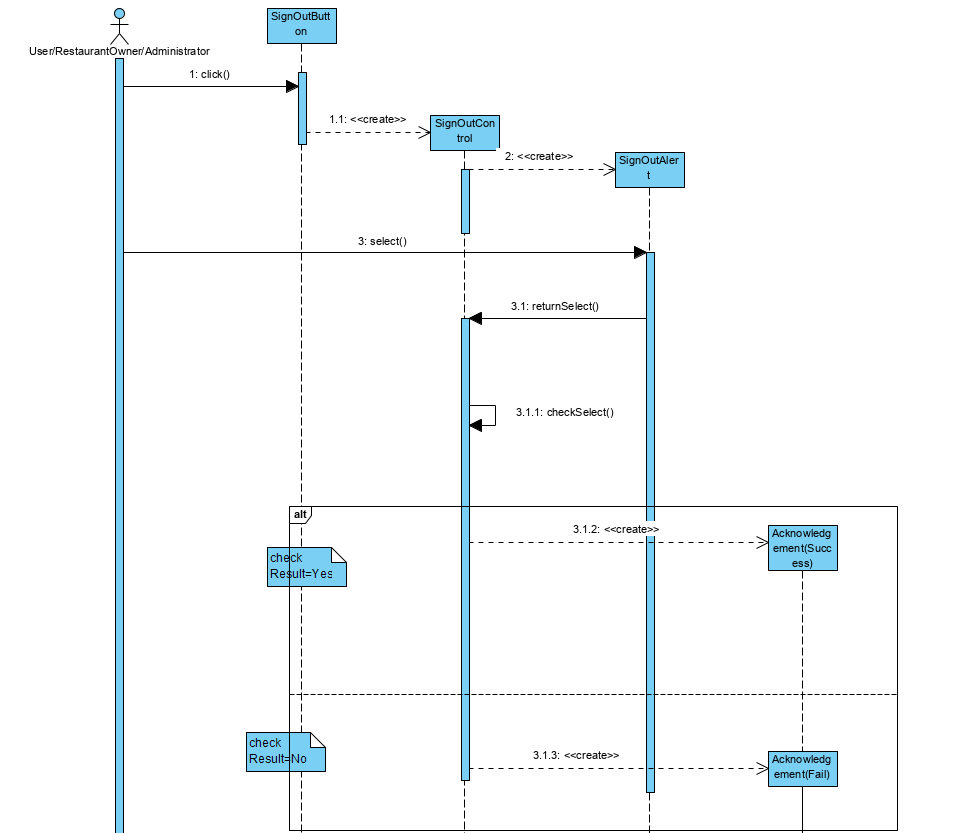


Dynamic model

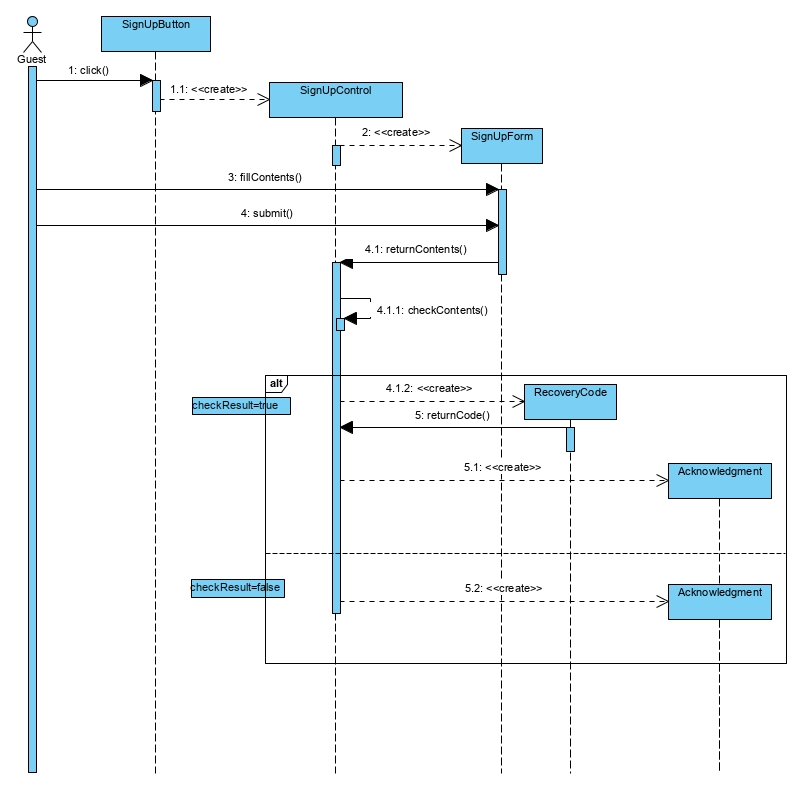
### **1-Sign In**

### 

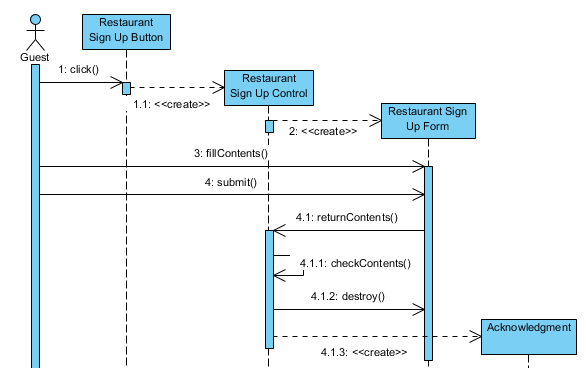
2- **Sign Out**



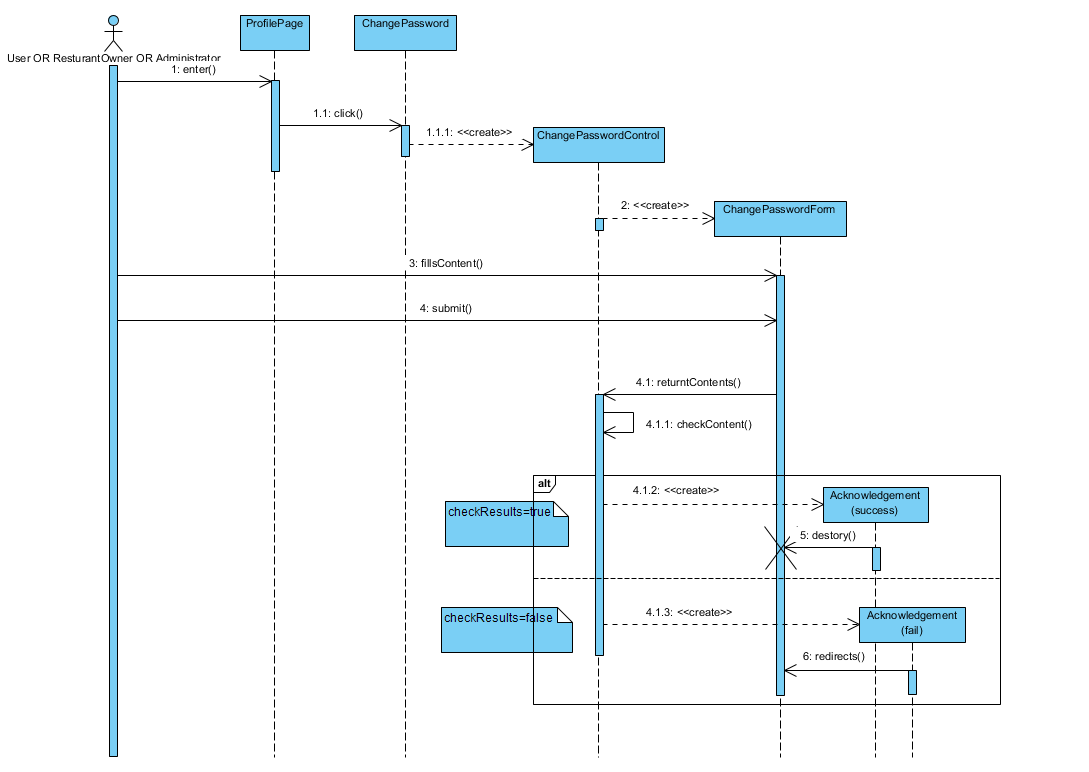
***3-Sign Up***



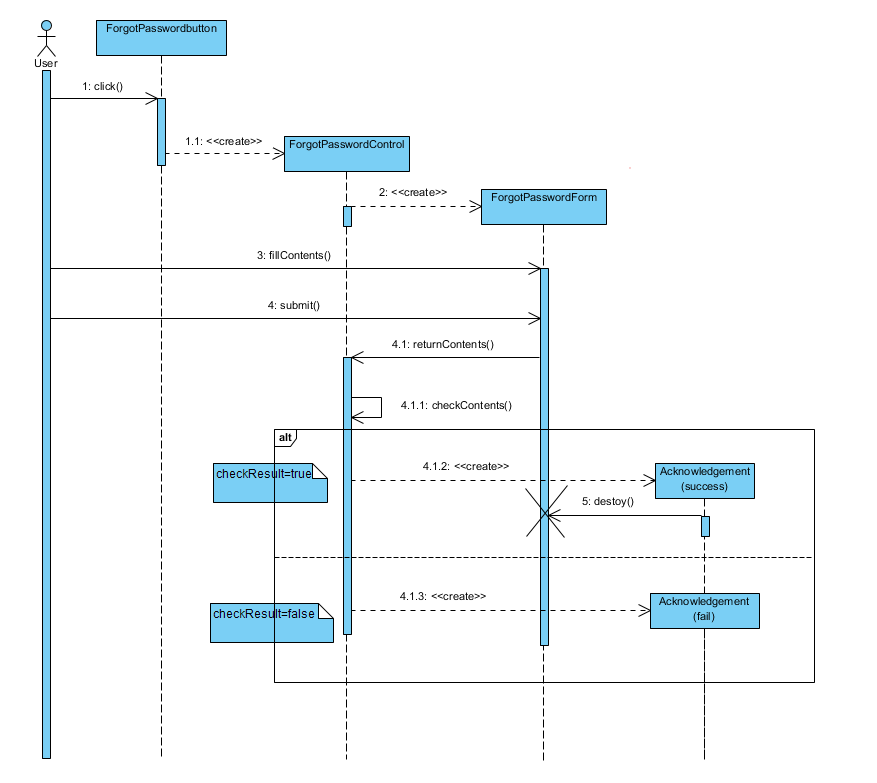
***4-Restaurant Sign Up***



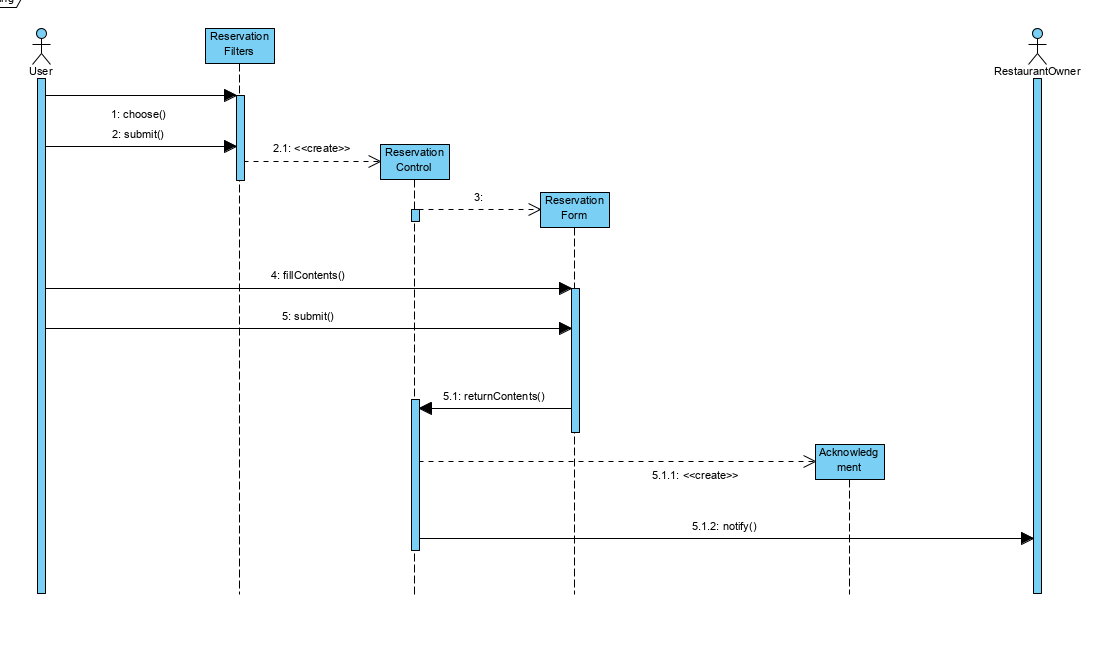
***5-Change Password***



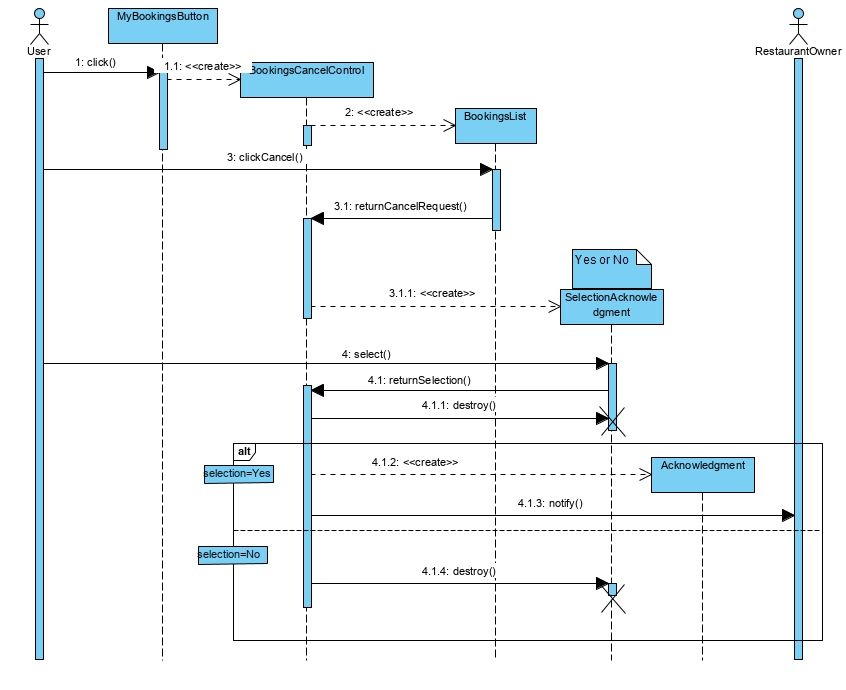
***6- Forgot Password***



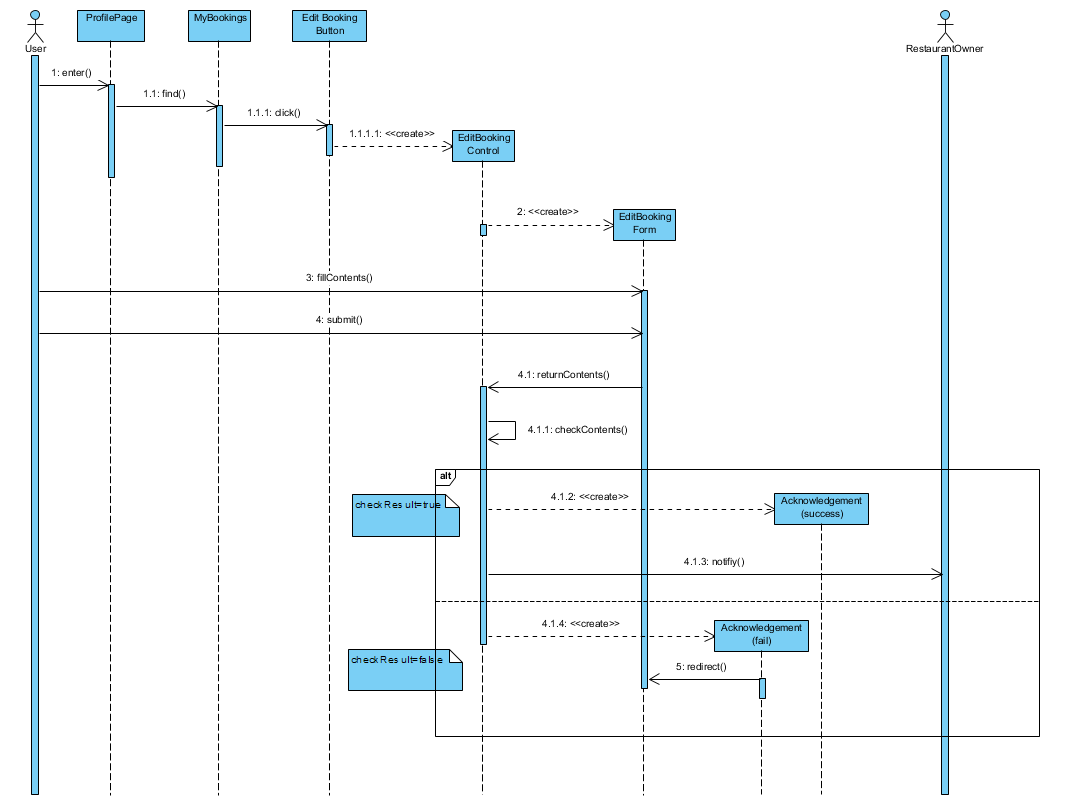
***7- Booking***



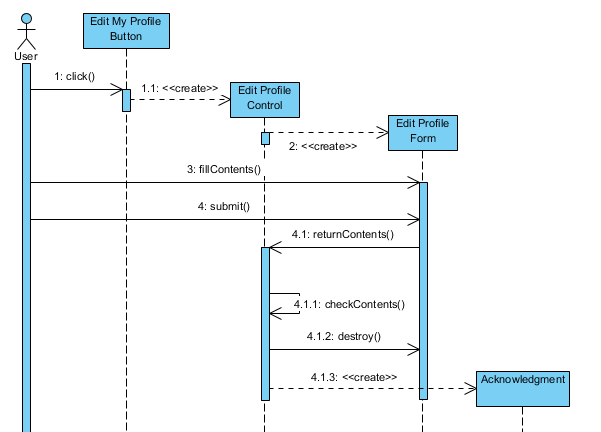
***8-BookingCancellation***



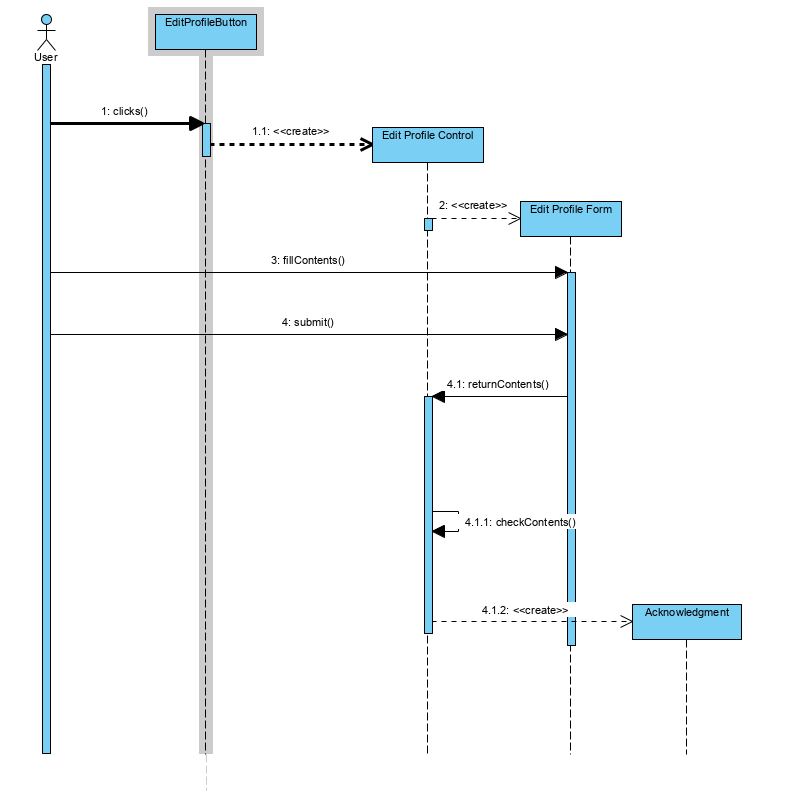
***9-Booking Editing***



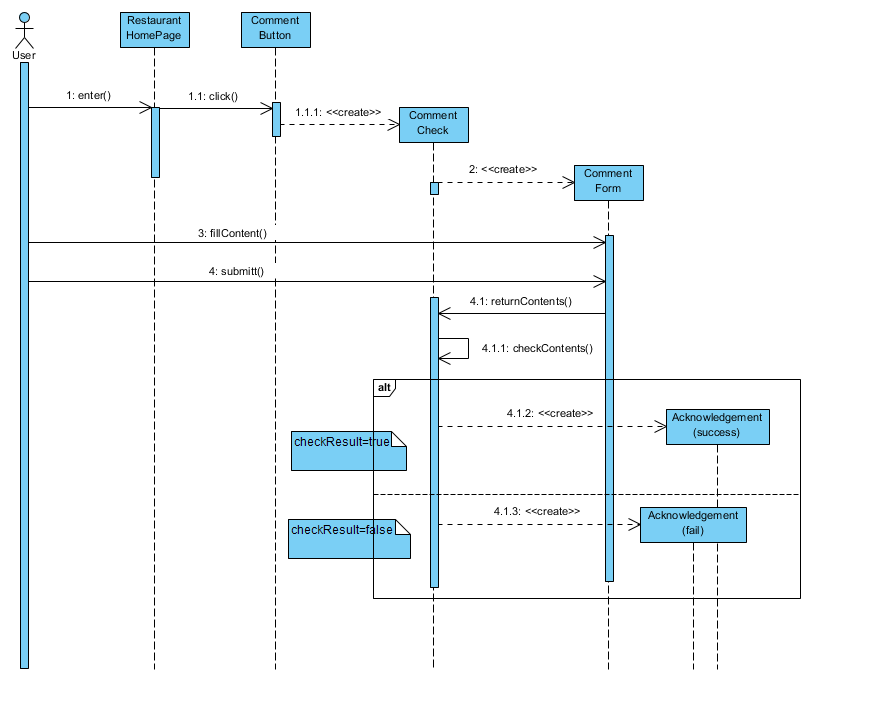
***10-Edit User Profile***



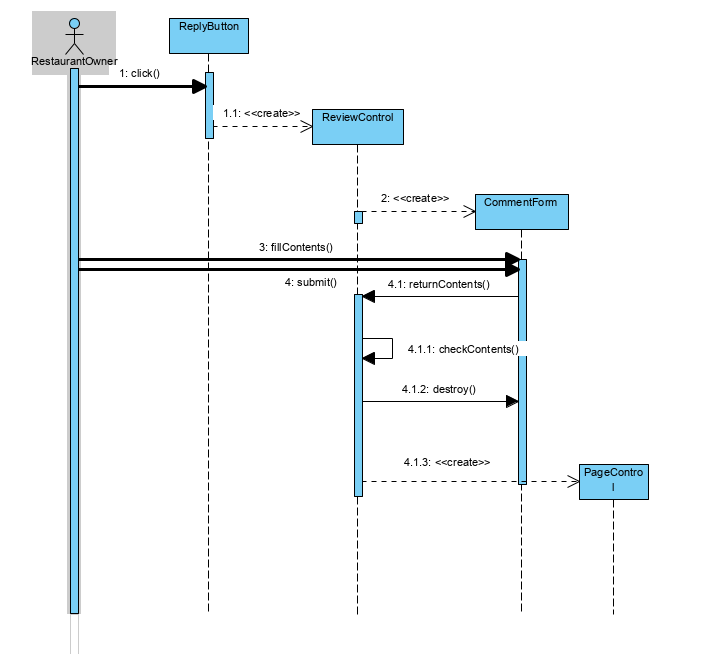
***11- Edit Restaurant Profile***



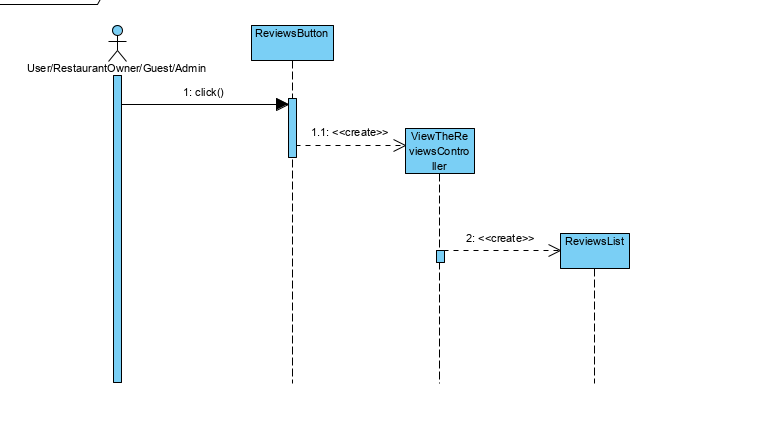
***12-Drop a Review***



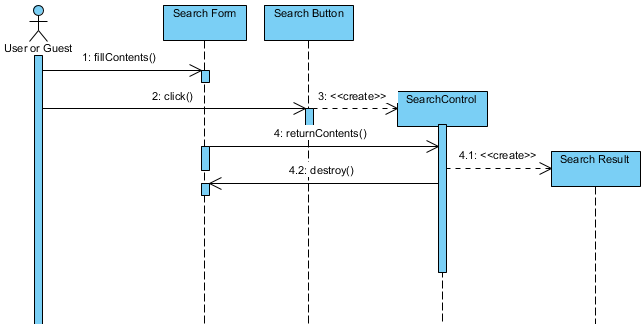
***13- Reply to Review***



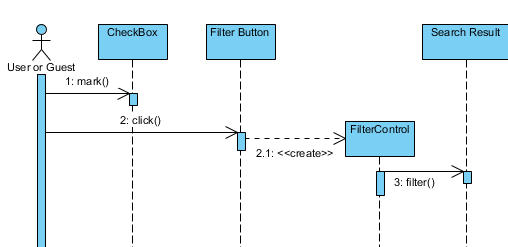
***14- View the Past Reviews***



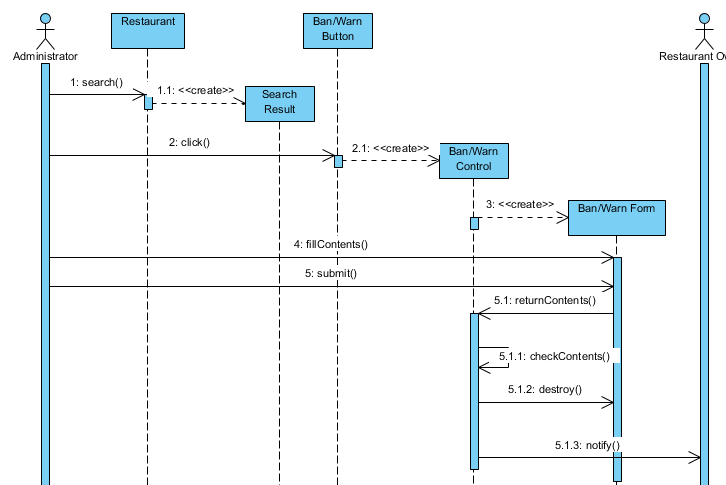
***15- Restaurant Search***



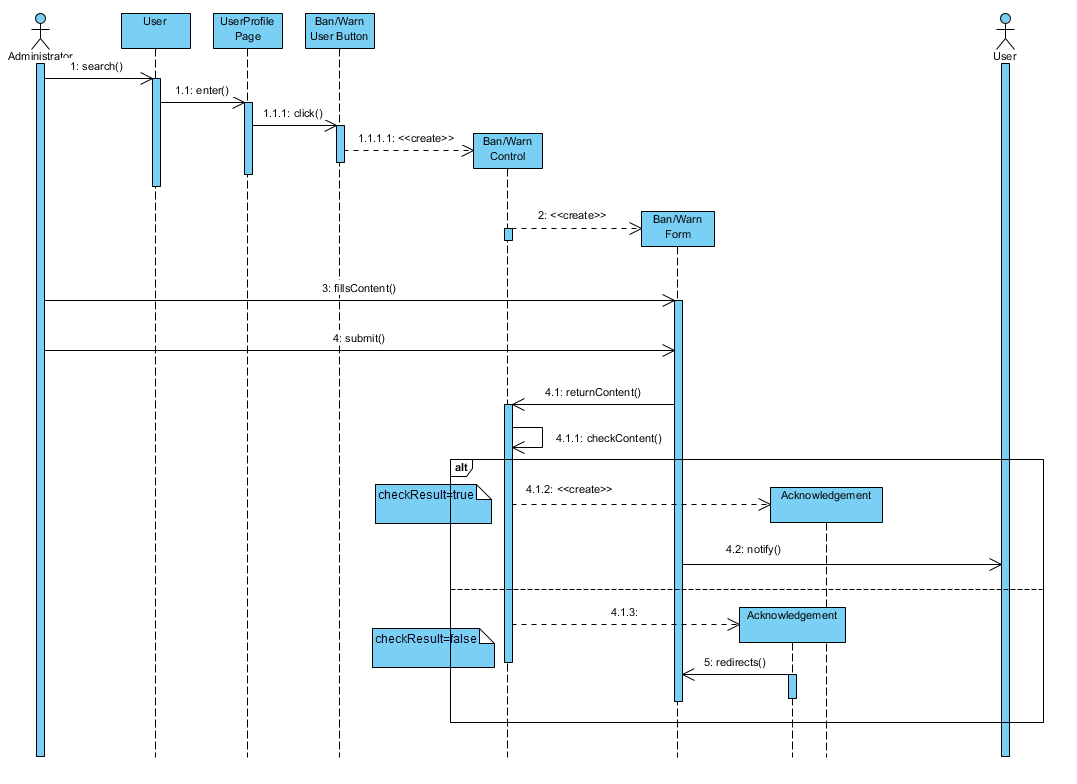
***16- Filter a Search***



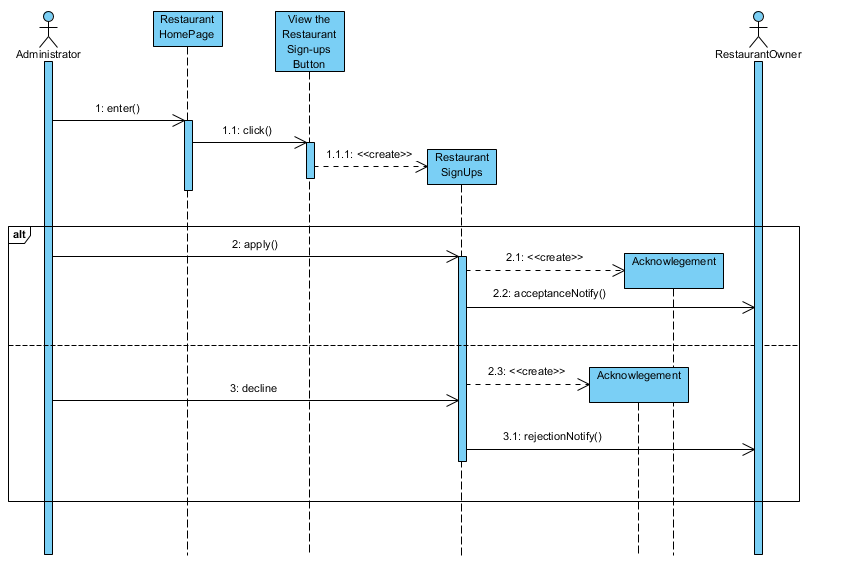
***17-Ban/Warn Restaurant***



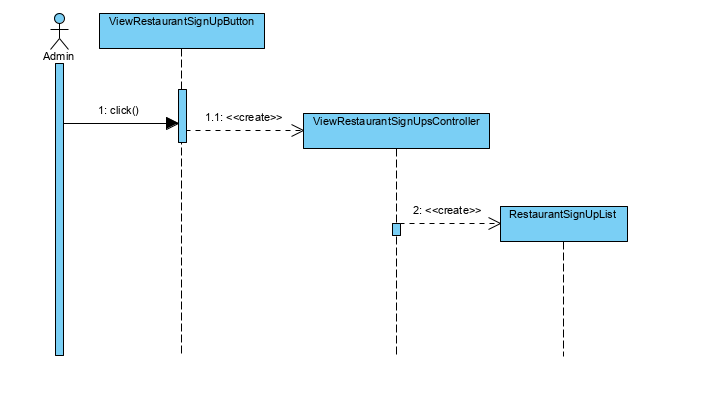
***18-Ban/Warn User***



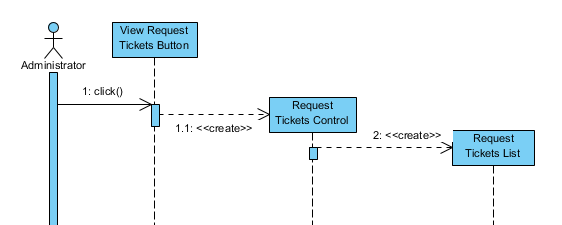
***19-Accept/Decline Restaurant Signups***



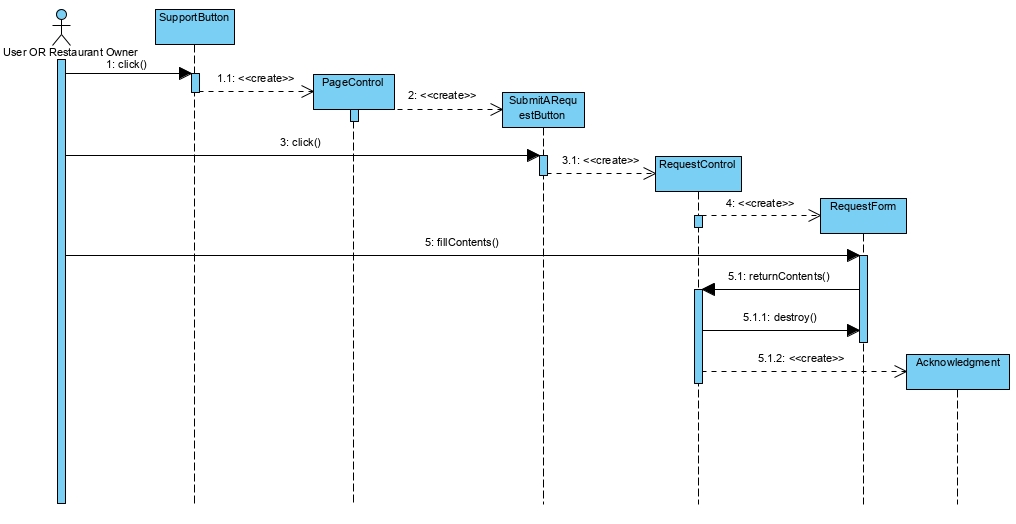
***20-View Restaurant Signups***



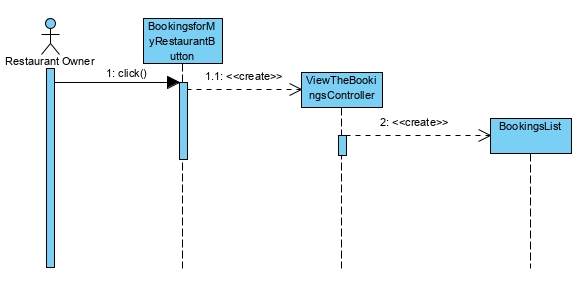
***21- View Tickets***



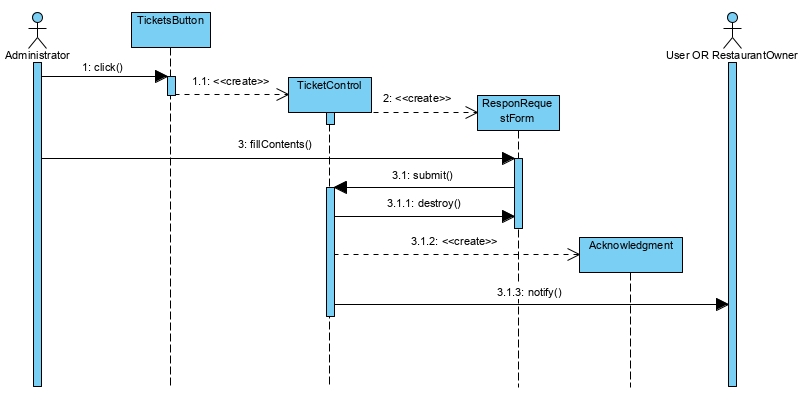
***22-Submit a Request Ticket***



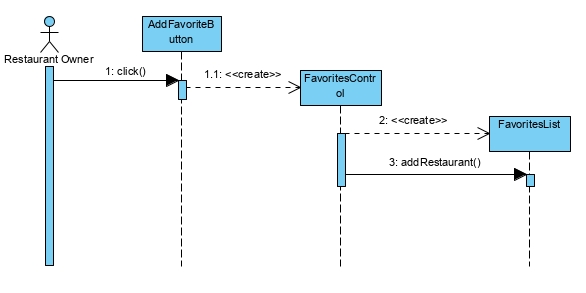
***23-View the Bookings of My Restaurant***



***24-Respond to Request Ticket***

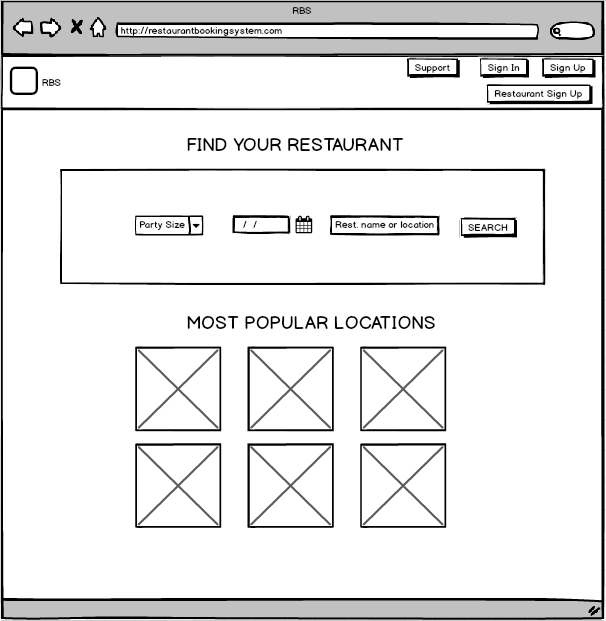


***25- Add Favorite Restaurant***

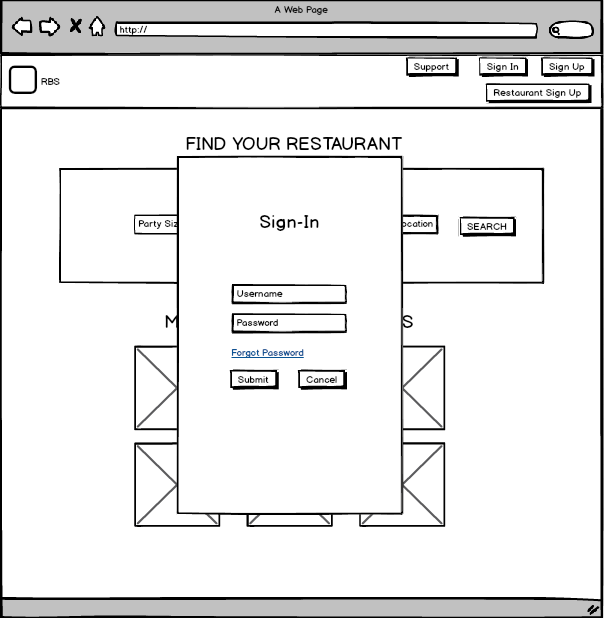


***User Interface Navigational Paths and Screen Mock-ups***

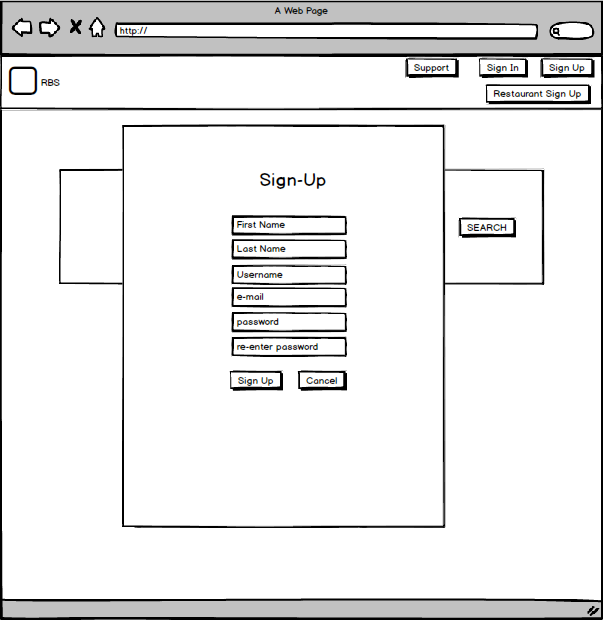
Homepage(Guest)



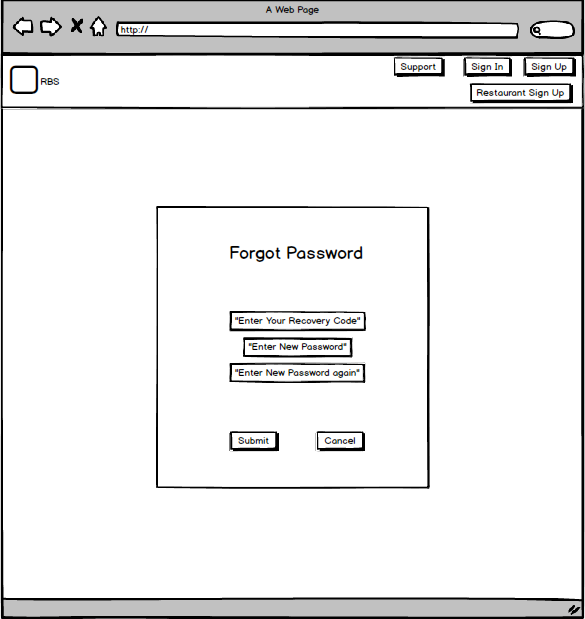
Sign In



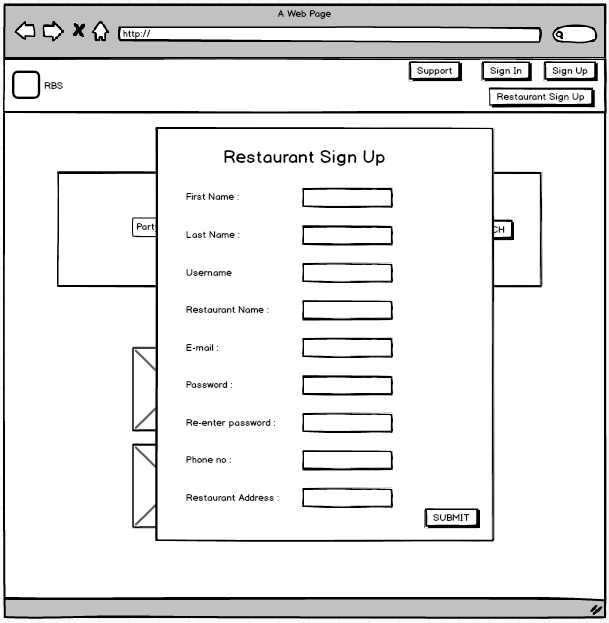
Sign Up



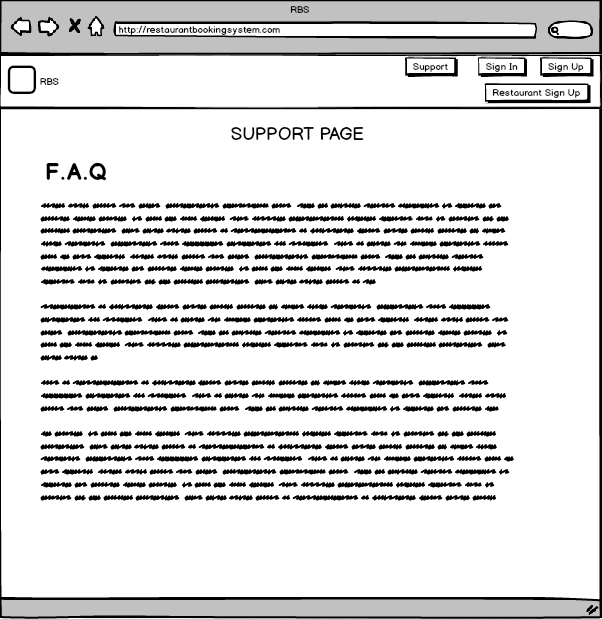
Forgot Password



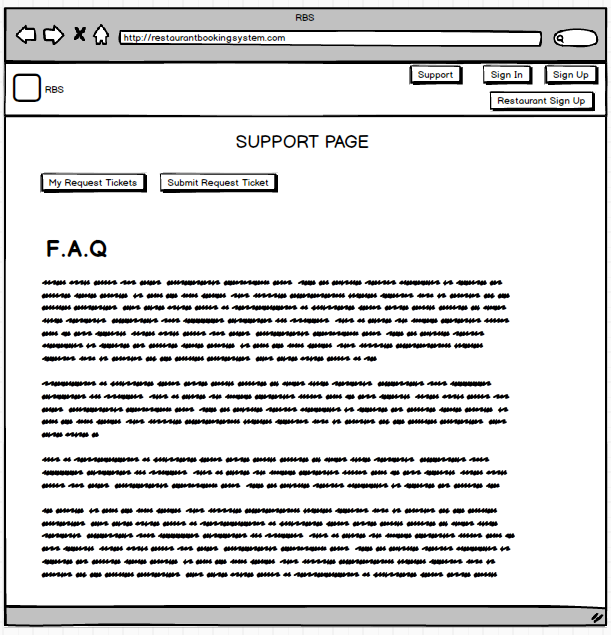
Restaurant Sign-Up



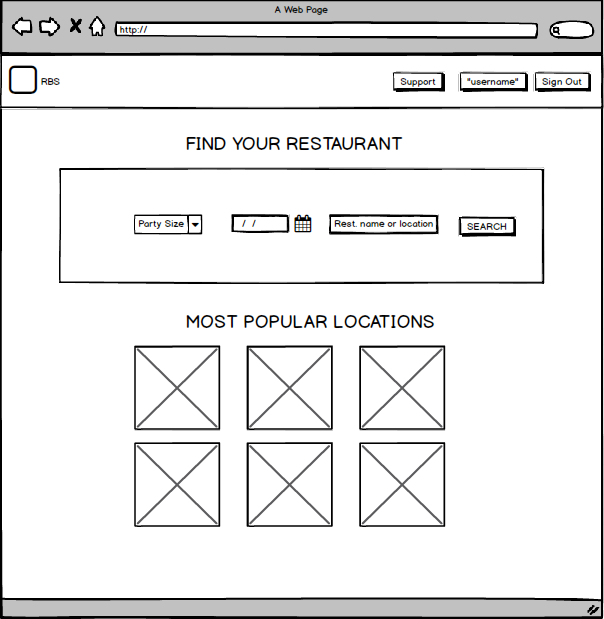
Support Page(Guest)



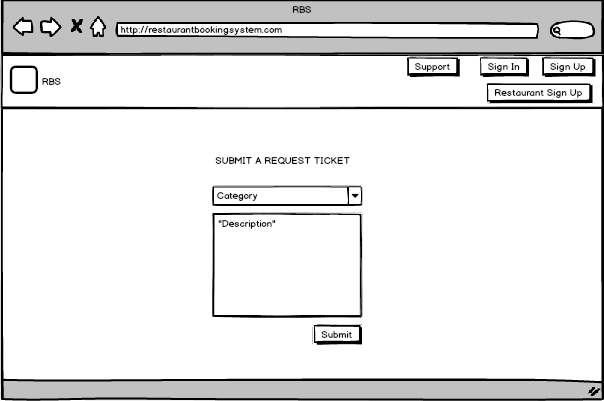
SupportPage(Others)



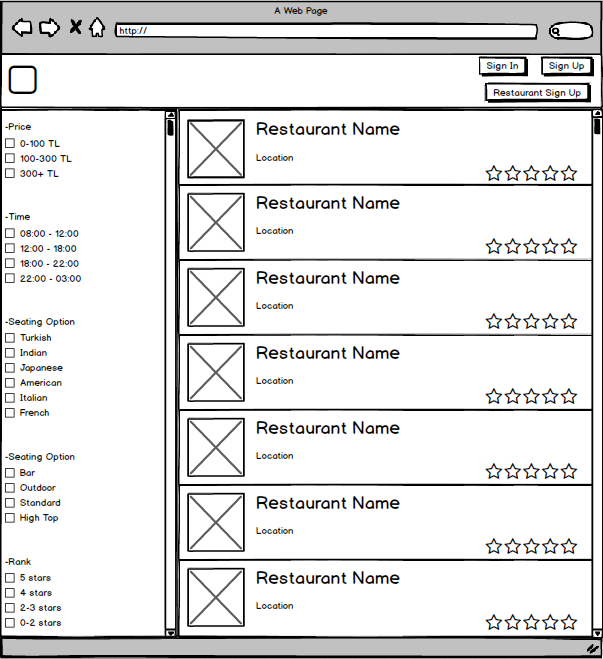
Homepage(User)



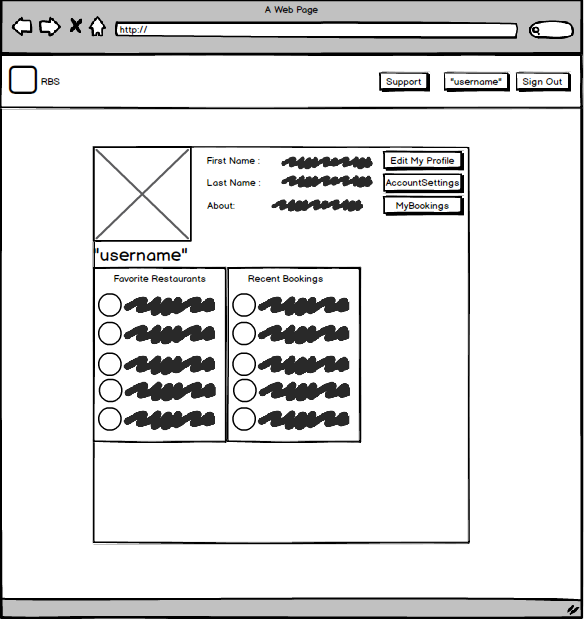
Submit a Request Ticket



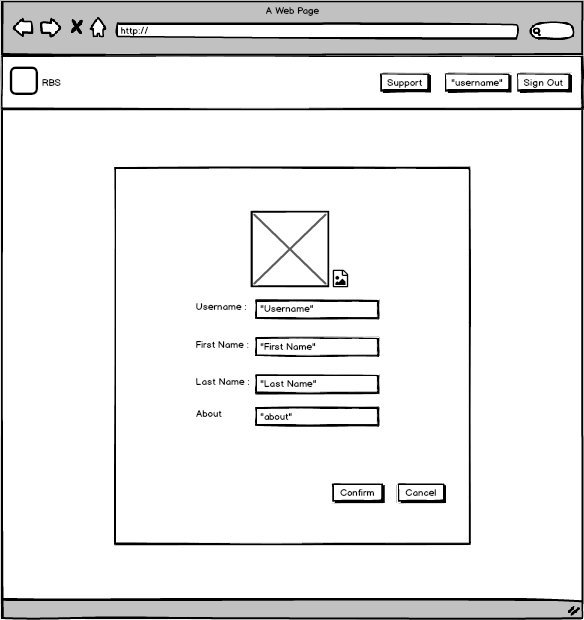
Search Result



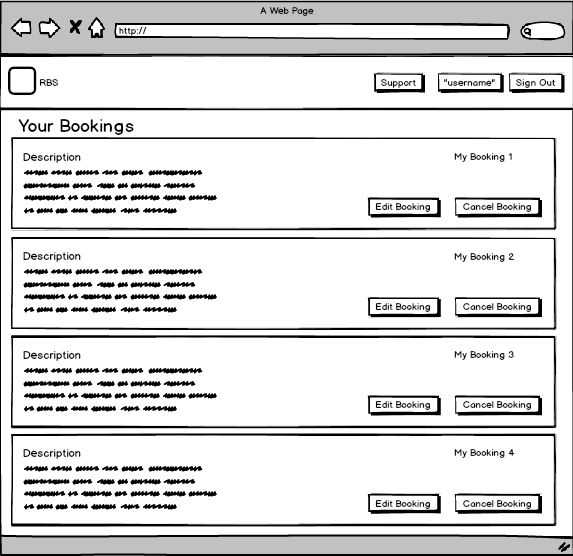
MyProfile(User)



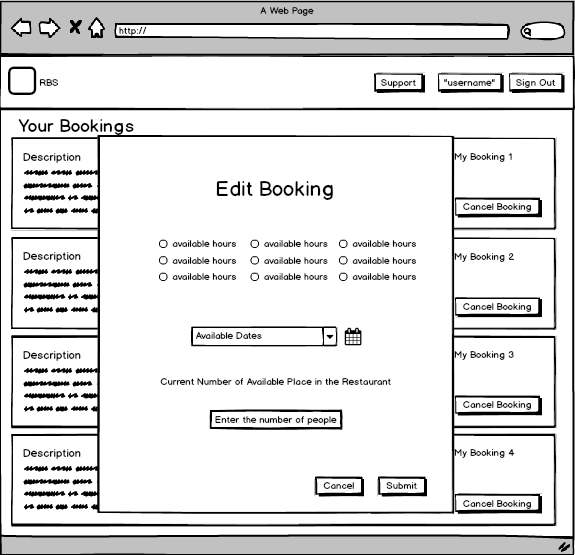
EditProfile(User)



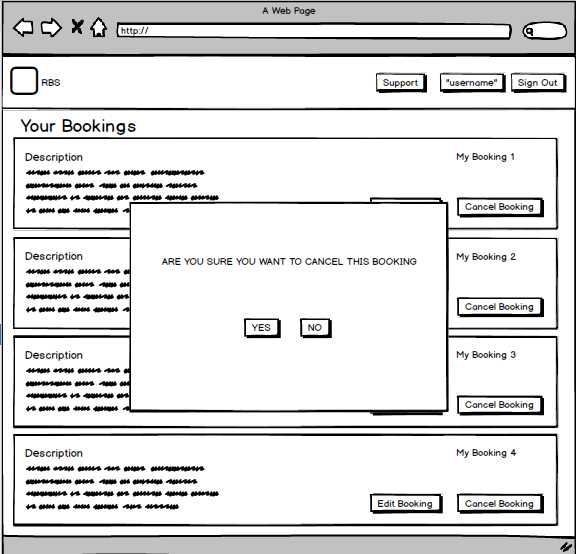
MyBookings



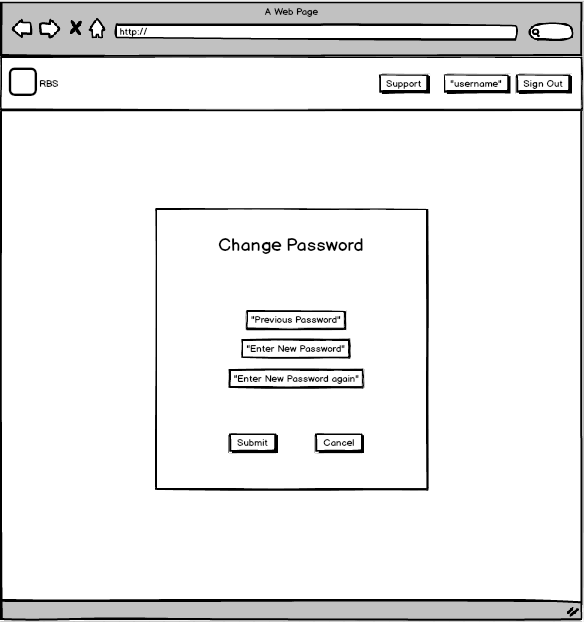
EditBooking



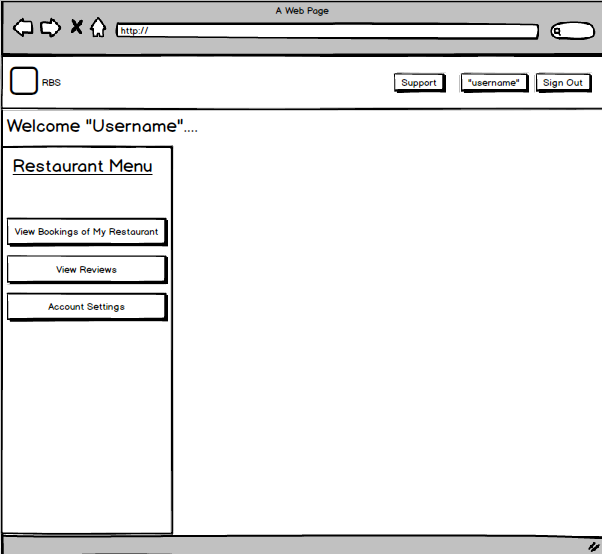
CancelBooking



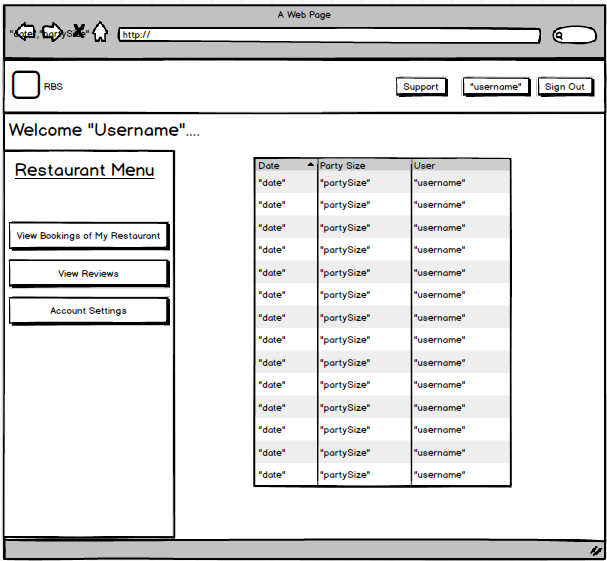
ChangePassword



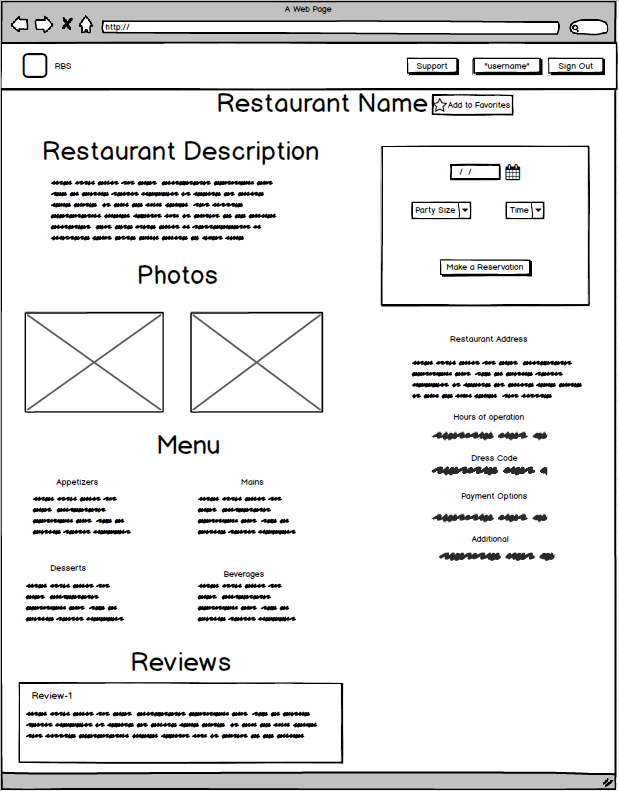
Homepage(RestaurantOwner)



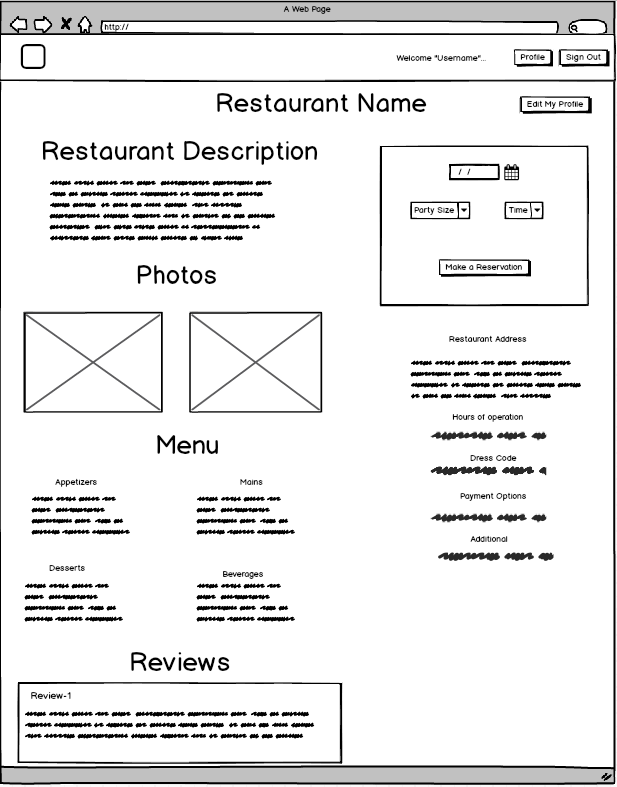
Bookings of My Restaurant



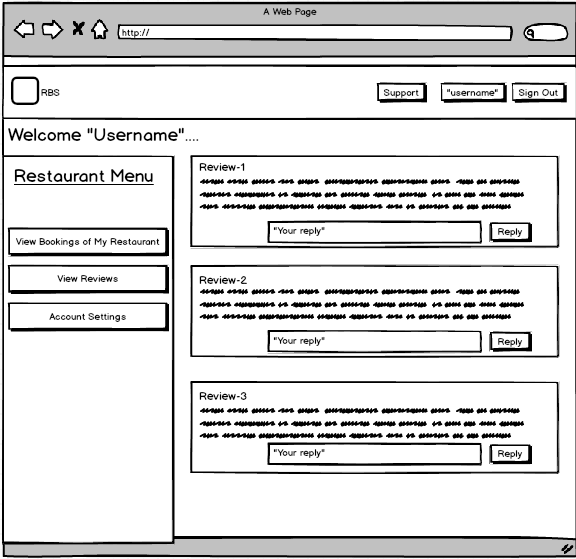
RestaurantProfile(OthersView)



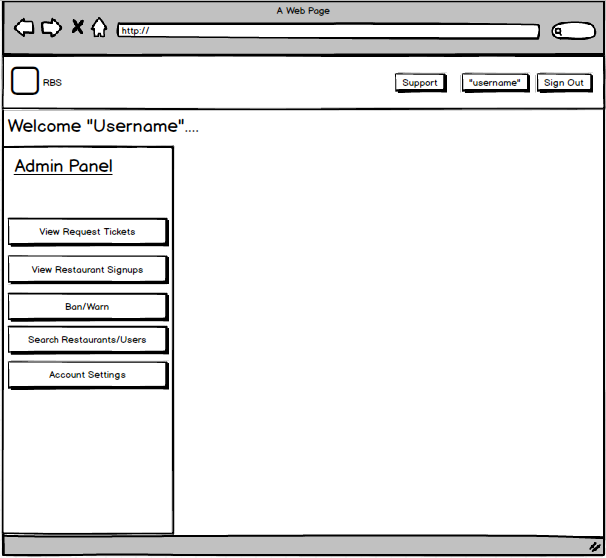
MyRestaurantProfile



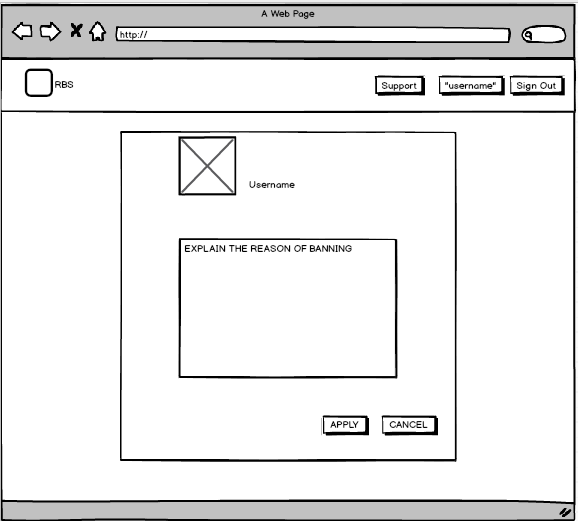
ViewMyReviews(RestaurantOwner)



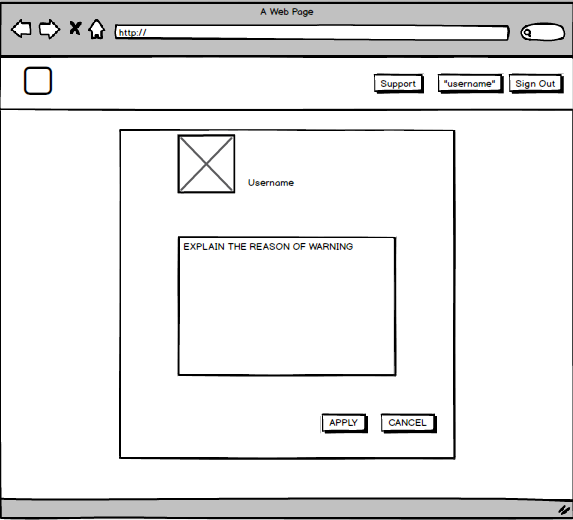
Homepage(Admin)



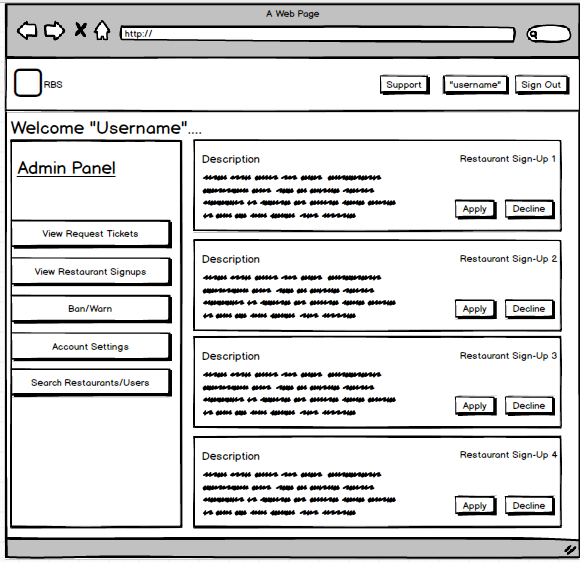
Ban



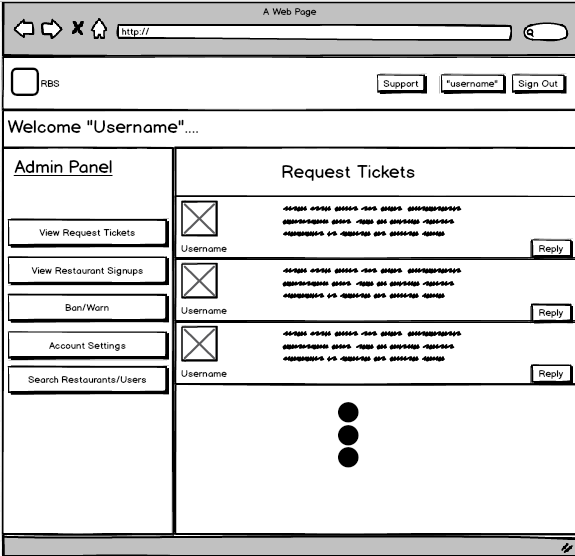
Warn



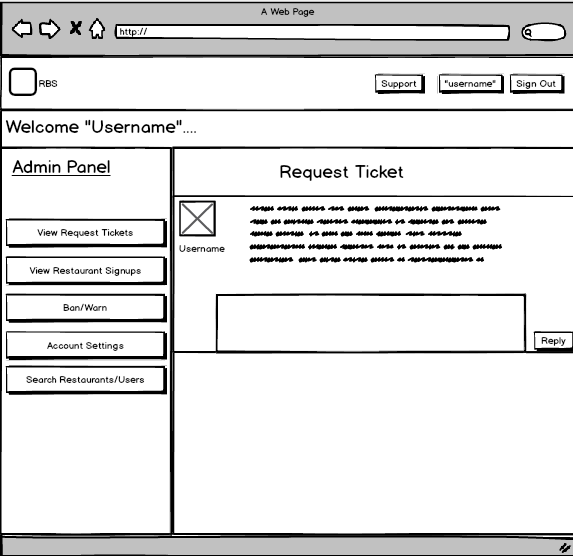
ViewRestaurantSignups(Admin)



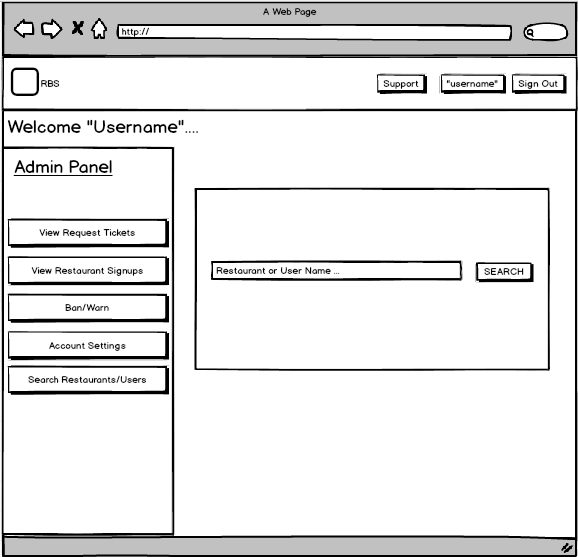
ViewTickets(Admin)



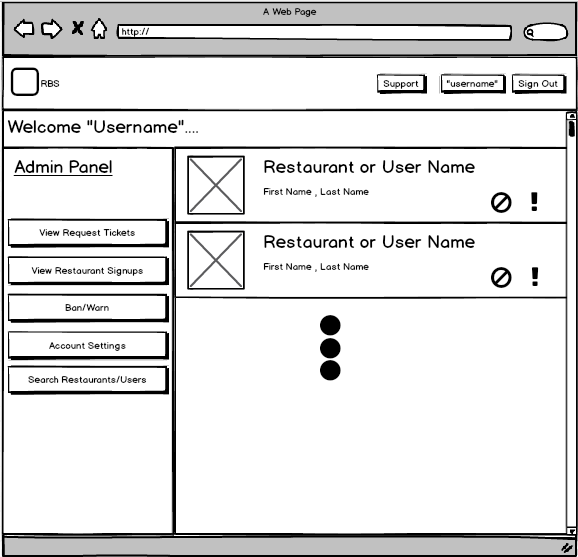
RespondToRequestTicket



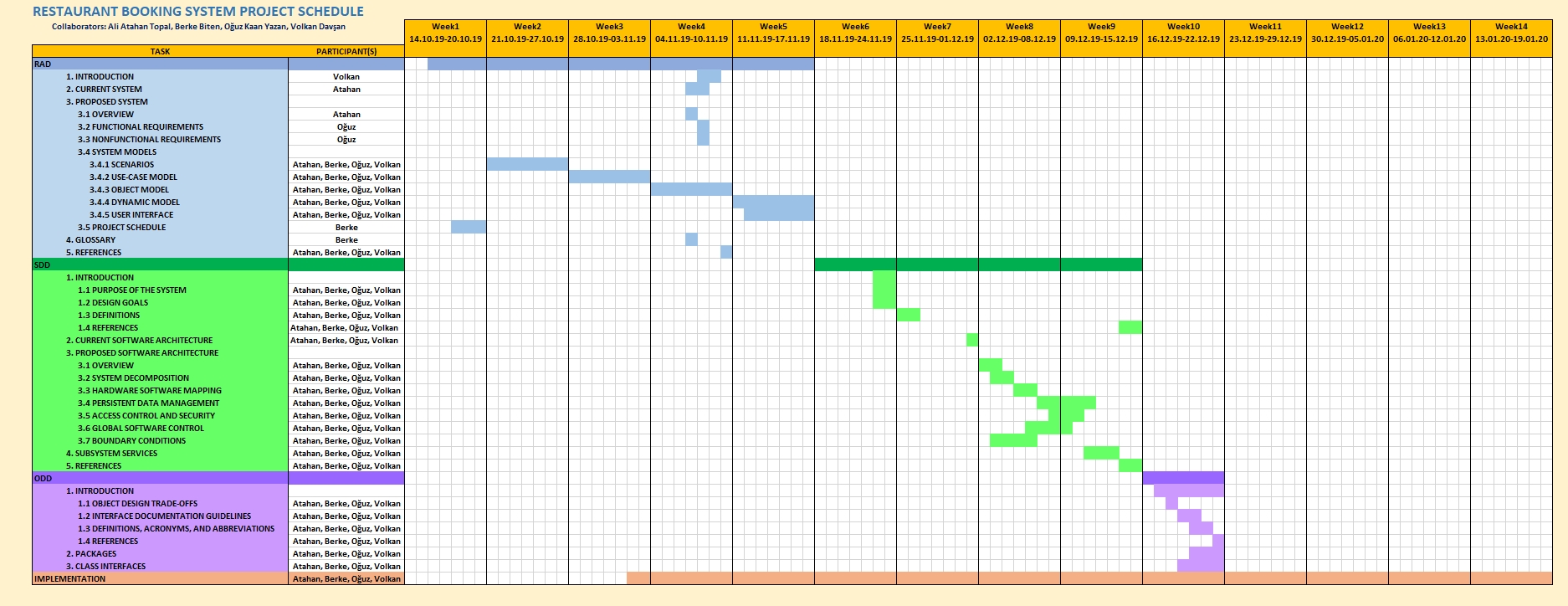
Search(Admin)



SearchResult(Admin)



## Project Schedule



# Glossary

# Use Case: A specific situation in which a function could be used.

**Entry Condition:** The condition needed to activate a function.

**Exit Condition:** The condition needed to consider a function successfully done.

**Guest:** A user who is viewing the but didn’t log in to an account yet.

**User:** A user who is viewing the RBS and logged into an account.

**Restaurant Owner:** A user who is viewing the RBS and logged into a Restaurant Account.

**Admin/Administrator:** A person who manages all content about the RBS (users, database, restaurants etc.)

**Admin Panel:** A panel which can be seen only by admins and contains the operations that admin can do.

**System:** A controller which operates all of the functions in the background.

**Sign in Form:** A form that contains “username” and “password” fields which will be filled by the user who wants to log into an account.

**Sign up Form:** A form that contains “first name”, “last name”, “username”, “e-mail”, “password” and “re-enter password” fields which will be filled by the user who wants to register.

**Homepage:** The initial screen of the RBS.

**Profile:** A page that contains user’s information.

**Booking:** A reservation for a restaurant.

**Support Page:** Page that users can see F.A.Q and submit a request ticket.

**Request Ticket:** A question/suggestion/help requests that can be sent by users to admins.

**Recovery Code:** A unique code which is necessary to recover a forgotten password.

**Ban:** Temporarily or permanently forbidding an account to use the RBS. A banned account can’t be seen by other users.

**Review/Comment:** A text that has been written by a user about his/her thoughts about a restaurant.

**Forgot Password Form:** A form contains a text field for the recovery code and two password fields for the new password which need to be filled by the user who forgot his/her password in order to set a new password.

**Change Password Form:**  A form contains three password fields (one for the current password, two for the new password) which need to be filled by the user who wants to change password.

**Restaurant Search Form:** A form that contains a text field for restaurant name or location, a combo box for selecting the seat number and a date selector.

**Restaurant Sign Up Form:** A form that contains “first name”, “last name”, “Restaurant Name”, “e-mail address”, “password”, “re-enter password”, “phone number” and “restaurant address” fields for user to fill. This form is being used by users who wants to register his/her restaurant to the website.

Notes

: Restaurant Booking System

# 

# References

This subsection should:

* Provide a complete list of all documents referenced elsewhere in the RAD, or in a separate, specified document.
* Identify each document by title, report number - if applicable - date, and publishing organization.
* Specify the sources from which the references can be obtained.