

Glass Booking

Requirements Specification and Analysis

Version 1.0

11.11.2018

Mert Pekdüzgün

Abdulwahab A. Sani

Alper Çevik

Burak Sönmez

Prepared for SE301

Software Engineering

!



Table of Contents

Table of Contents

REQUIREMENTS ANALYSIS DOCUMENT[1]
3	
1. INTRODUCTION
3	
1.1.PURPOSE OF THE SYSTEM
3	
1.2.SCOPE OF THE SYSTEM
3	
1.3.OBJECTIVES AND SUCCESS CRITERIA OF THE PROJECT
4	
1.4.DEFINITIONS, ACRONYMS, AND ABBREVIATIONS
4	
1.5.OVERVIEW
5	
2. CURRENT SYSTEM
6	
3. PROPOSED SYSTEM
7	
3.1.OVERVIEW
7	
3.2.FUNCTIONAL REQUIREMENTS
8	
3.3.NONFUNCTIONAL REQUIREMENTS
9	
3.4.SYSTEM MODEL
10	
PARTICIPANT ACTOR INSTANCES: MARIANNE: MANAGER
27	
PARTICIPANT ACTOR INSTANCES: MBEMBA: MANAGER
27	
3.5.PROJECT SCHEDULE
61	
4.GLOSSARY
61	
5. REFERENCES
63	

REQUIREMENTS ANALYSIS DOCUMENT[1]

1. Introduction

The purpose of this system is to provide users with an easy way of being able to reserve space in a restaurant right from their mobile phone so that they are assured of being able to get a place at any restaurant of their choice.

1.1.Purpose of the System

The driving reason behind developing this system is to give restaurant goers assurance of having space/table when they go out to eat. More times than not, people go to a restaurant usually in a group and unfortunately there is no space to accommodate them all as such, our system will let them know for sure whether they will get a table or not which also saves their time and energy. It also gives the users valuable information about the restaurant and also importantly- choices. There is an expansive list that the user can choose from and also see the ratings.

1.2.Scope of the System

Our system will allow the users to reserve a place at a restaurant of their choosing prior to their visit to the restaurant. The user first needs to sign up, then after logging in they can view the variety of restaurants available/open and select the number of people that will be coming. This service is very versatile and as such it can be used by a people ranging from government officials down to students at a college.

Our system will have four users each with a different level of authority. They are the admin, the manager, the visitor and the customer. The admin is the one with the most authority and he is responsible for adding new restaurants (or disabling them) from the system.

He can also add new managers and customers to the system and just as well disable their accounts. Maintaining the system is also his responsibility.

The manager is the one who will send a request to the admin to add a new restaurant and can also send a disable restaurant request. The manager can update restaurant info such as location (if the restaurant moved elsewhere), restaurant name, available capacity etc..

Visitor is a potential customer and they have significantly less functionalities available to them than the customer. Tasks of a visitor are done without a need to login to the system. These tasks include searching a restaurant, viewing the restaurant profile etc.. To make tasks like making reservation or rating restaurant, visitor needs to sign up and become a customer.

Finally, the customer is the one with least authority but has the most functionalities. The customer can make a reservation and select the number of people that they want to reserve for. In addition to making the reservation, a customer can of course have a change of mind and cancel their reservation. The customer can also leave an optional rating which will be useful in giving other users insight about the restaurant's quality. They can also update their information such as email address, favourite restaurant etc. The customer will be able to search for a restaurant and view it which is part of the process for making their reservation.

1.3.Objectives and Success Criteria of the Project

Our objective is to give our users the peace of mind and assurance of knowing they are guaranteed a place at the restaurant they want to visit. By doing this we also want to save their time so they don't go to a restaurant and its full and they have to choose another place again which could be full again. This can get frustrating. We believe that our system can help assist with this problem. We shall gauge the success of the project by:

- positive reviews
- number of users/downloads
- improving people's time efficiency
- system popularity
- improved planning for large groups of people

1.4.Definitions, Acronyms, and Abbreviations

- RAD: Requirement Analysis Document.

- Admin: System administrator who manages users and managers and also maintains and updates the system.
- System: the software that is to be developed.
- Disable: freezes activity for the account or restaurant that is requesting to be disabled. Action cannot be undone.

1.5.Overview

This RAD shall compare the current system that is being used which shall be replaced with the to-be developed system. It shall look at the upsides and benefits of replacing the old system. It also contains the system models which include functional and nonfunctional requirements which shall be described in detail and explained, there is also a use case description and scenarios that were created. Also included is the Gantt Chart which a schedule for when the project is expected to be fully developed along with milestones. We included a glossary as well so as to define words that may seem foreign to those outside of the project team.

The organizational structure of the RAD is designed from the ground up so, we describe the things included from the basic descriptions of our system to the more complex system models and functionalities of the system. So we started with Introduction where we specify and familiarize the reader with what the project is all about and what it will do. The purpose of the system and the objectives of the system and what it is expected to achieve. Furthermore, we give an overview of the project which tells about what this RAD contains.

Next up is the current system section where we talk about the system that is currently being used by a majority number of people and this is the system we intend to replace and ultimately improve. We mention the things that make this current system inefficient and how is it time consuming and leaves room for uncertainty such as visiting a restaurant and it is full and so people have to search for another place.

In proposed system section we speak on how the to-be developed system shall be an improvement on the old one. We discuss the advantages and why it is the right step forward. We mentioned that the current system is time consuming and uncertain, in the proposed system, this problem is solved by allowing the customer to reserve a table/ space for a number of people. If you reserve the space, you are guaranteed a table and this saves time and makes it easier for people to choose where to go. The

proposed system will also be made to be as user friendly as possible so that customers of all ages and computer literacy can use it with minimal difficulty.

In overview section, we discuss the functionalities of the system which is what the system is going to be capable of and what actions it supports.

In functional requirements section, we discussed how the different users of our system -the manager, admin and customer- interact with the system and their different levels of authority. We make mention of what the Restaurant Booking system is capable of doing and its features how the users have different but sometimes similar functions.

In non-functional requirements section, we talk about functions that are not necessarily defining the behaviour of our system but rather, they are criteria for how the system should be judged or rated. These criteria include: usability, performance, reliability or supportability, implementation, interface, legal and packaging. We shall discuss these in more detail in later sections of the RAD.

In system models section, we talk about the scenarios, use cases, object model and dynamic model of the system. We created scenarios for manager, admin and the customer as our actors. Each scenario discusses how the actor shall partake in the use case and achieve the task at hand. For the use cases we gave full descriptions including, entry conditions, exit conditions, flow of events etc. The use case is an action to be performed by the specified actor within the system.

Lastly, in the glossary section we defined terms and other words or phrases needed to fully understand the RAD with minimal fuss. It is a guide for the reader who gets lost or misunderstands something.

2. Current System

The reservation system currently being used is by making a phone call to the restaurant a person desires to go and provides details to get the reservation, the correspondent then lets the caller know if the reservation is available. Alternatively the person can go to the restaurant physically and make the reservation at the reception. Using this method, if the person has no idea about the restaurant (quality, food taste, location etc) then it doesn't provide insight about the restaurant you are going to. In addition to this, because it is via network, the caller can be made to wait

because of a busy line or, being even unable to reach the restaurant due to some network problems. In this case the caller is being made to wait. If the person goes physically to make the reservation, he goes without knowledge of whether there is available space or not, and he may not know the location.

3. Proposed System

The system to be designed is going to be made for making online reservations for restaurants. A restaurant manager signs up and sends a request to the admin to registers the restaurant. For the registration, the manager needs to provide information about the restaurant which include it's location, food type, menus etc.. The system admin accepts the restaurant registration request which then makes the new restaurant visible to customers who may want to reserve. Then the customer also signs up and logs into their account and from there, they browse through the list of available restaurants and view the restaurant of their choice. The rating of a restaurant helps a customer decide where they want to make the reservation. So each restaurant has a rating which is given by previous customers based on their personal experiences. Because this is done online, a customer doesn't need to call to make a reservation so this avoids the inconvenience of waiting. Another benefit of this proposed online reservation system is that the menus and type of restaurant are displayed on screen before making a decision, so it helps to make up people's minds. Restaurants have ratings, so our customers have a general idea of knowing what kind of restaurant they are walking into. The uncertainty we spoke of earlier about restaurant location wont be a problem either because restaurant managers provide the location address before registering to the system. The customer can search for the address and pinpoint the location. After a customer completes their reservation, they are assured of being able to get a place in that restaurant and it avoids the frustration of going to a restaurant and it is unfortunately full, prompting another search for another place which may be full as well. We make all this unnecessary by making it possible online where there is a variety of different restaurants.

3.1.Overview

In this section, we describe the functional requirements of our system and how they are implemented and who implements them. We have four users in our system. First, let's start with the Customer. The customer has access to the most functionalities and he is the one who is able to: Make Reservation, Rate Restaurant, Edit Profile, Change Reservation, Cancel Reservation. These steps all require a login action to be

performed first meaning, a customer must be registered on the system before having these functionalities available. Our customer is able to reserve a restaurant through our system and to this he logs in, he chooses the restaurant of this choice and he selects the number of guests he wants to reserve for. He is shown the available times and he chooses the one that suits him the most. After this he completes his reservation.

Next up we have the Manager. He is the one who is responsible for adding restaurants to the system but first, he needs to send a request to the admin who has to approve the add request. The manager has the following functionalities available to them: Send Add Restaurant Request, Send Disable Restaurant Request, Edit Restaurant, Edit Profile, Show Reservation List, SendRestaurantProfileEdit Request. As with the customer, the manager needs to be registered to the system before these functionalities become available to them.

We also have the Admin who is overseer of all operations on the system which includes accepting requests and declining requests. His functionalities include: Add Restaurant, Disable Restaurant and EnableRestaurantProfileEdit. However, before he is able to perform these actions, he obviously needs to receive requests from the manager.

As mentioned earlier, visitor is a potential customer and they have significantly less functionalities available to them than the customer. Because they don't need to register to the system, their tasks are limited. These tasks include searching a restaurant, viewing the restaurant profile etc.. To make tasks like making reservation or rating restaurant, visitor needs to sign up and become a customer.

3.2.Functional Requirements

In our system, one of our main functionalities is making a reservation which is done by the customer. As we mentioned earlier the customer is a user who is registered to the system. To make a reservation he logs in and chooses from a list of restaurant which one he wants to go to. He selects the number of people going and then choose the available times. He confirms the reservation and he is displayed with a receipt. Our system will also support changing reservation. This happens when the customer has to change change/edit his reservation. For example, he wants to increase the number of guests to reserve for in the same restaurant, so he changes his

reservation to edit this. The system also allows the customer to rate a restaurant. This will be done by going to previous reservations and choose the restaurant. There are five stars and the customer can give from one to five stars with five being the best. Customer can also search for a restaurant by name. We have included a search bar functionality and when the customer presses it, the keyboard pops up and he types in the name. Results matching the name are returned.

Our Manager is able to send requests to add their restaurant to the system. He goes to the add restaurant section and he is presented with a form. He fills in the fields which are required to register the restaurant to the system. The fields include: restaurant's name, address, phone number, features. He then sends the request which shall be delivered to the admin's inbox. Our system also allows the manager to be able to take down their restaurant from the system. To do this he has to send another request to the admin. He goes to the "show my restaurants" section and he selects one restaurant at a time (if he has multiple ones). He then presses on the send disable request which shall be delivered to the admin inbox.

The admin's main functionalities will be to add and disable restaurants on the system. To add a restaurant he opens his inbox and he has some waiting requests sent by managers, he opens it and if he chooses to accept the request, he adds the restaurant to the system's database so customers can now make reservation to this restaurant. To disable restaurant, he again opens inbox and there are some requests, he opens it and if he chooses to accept the request he disables it from the system and customers cannot make reservations to the restaurant anymore.

3.3. Nonfunctional Requirements

Usability- We made our system very usable and to measure this, making a reservation only takes five clicks. And this is the main functionality of our system. So just

about anyone with basic knowledge of technology can use our system.

Reliability- Our system is expected to run 100% of the time so that our users can be able to reserve at any time.

Performance- System performance needs to be optimal, this includes search results times, loading restaurant pages etc.. So our users wont need to wait too long for system to load. We also want to support at least 300 parallel users.

Supportability- System needs to be modified easily without difficulties.

Implementation- System shall be developed on Android App Studio. This means the programming language will be mainly based on Java. Our web service will be supported by Spring MVC Framework. Our database will be managed via MySQLWorkbench.

Interface- Our interface will be an android application interface.

Packaging- No packaging requirements. *Legal-* The software is provided "as is", without warranty of any kind, express or

implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and no infringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

3.4.System Model

Here, we show the high level functionalities of our system and how the users of our system interact with the system. For example, one of the functionalities our system is designed to support is to make a reservation. To do this, firstly the customer needs to create an account. The customer subsequently logs in after successfully creating the account. After this he browses through the list of available restaurants and views the restaurant. He then chooses the number of people to reserve for and finalises his reservation. For the object models we depict our classes using UML diagrams and show the relation between them. In the dynamic model, we showed interactions between objects within a use case using sequence diagrams and showed how state machines are used to show the behaviour of that single object.

Scenarios

Scenario 1:**Scenario name: Log-in****Participant actor instances: Ali: Customer, Admin Or Manager**

The flow of events:

1. When Ali open the restaurant booking app, System starts and brings Login screen
2. The username and password fields are filled. He presses the login button.
3. Both password and username match, there is no error so logging in process is successfully completed.
4. Ali touch logout button and logout.

Scenario 1 Extensions:

- 2a. Ali inputs wrong username or password so logging in the process fails. He is asked to recheck the information he provided.
- 2b. Ali inputs wrong username or password so logging in the process fails. He is asked to recheck the information he provided. He realized he has forgotten his password. The system guides him to reset the password by sending a reset link to his attached email address.
5. Ali chooses no so next time he tries logging he has to manually input all his information.

Scenario 2:**Scenario name: Make Reservation****Participant actor instances: Aysel: Customer**

The flow of events:

1. Aysel logs into the system using her information. She wants to make reservation.
2. She chooses the restaurant.
3. After the choosing restaurant, reservation screen is opened by system.
4. She wants to make reservation for 2 person. She choose it.
5. She choose the time like: 21.00 23.11.2018
6. Then she touch reservation button and applied.
7. Aysel touch logout button and logout.

Scenario 2 Extensions:

4. If this restaurant don't have an available table for 2 persons in this time. She can not reservation. She should change time or persons number.
5. Aysel can not choose this time if it is not available. She should choose another date
6. If she has network connection problem she can not apply reservation. She should connect network.

Scenario 3:**Scenario name: Cancel Reservation****Participant actor instances: Aysel: Customer Or Manager**

The flow of events:

1. Aysel logs into the system using his information. She made reservation.
2. If she has a problem and can not go this place so she wants to cancel reservation.
3. She clicks my reservation button and see her reservation list.
4. She choose the reservation which she can not go to touch the reservation
5. Then she touch cancel reservation button and applied.
6. Aysel touch logout button and logout.

Scenario 3 Extensions:

5. If she does not have reservation she can not see her reservation list.

6. If she has network connection problem she can not cancel reservation. She should connect network.

Scenario 4:**Scenario name: Choose Restaurant****Participant actor instances: Aysel: Customer Or visiter**

The flow of events:

1. She touches the search button and restaurant screen is opened by the system and available restaurants listed.
2. She have two options;
 - a) if she wants reservation which knows the name of the restaurant, she use the search field.
 - b) or use the scroll down in this screen and she can find restaurant which she wants to choose.
3. When she touch the restaurant, she chose the restaurant.

Scenario 4 Extensions:

1. If any restaurants are not available in this time, she can not see any restaurant in list.

Scenario 5:**Scenario name: Logout****Participant actor instances: Aysel: Admin, Customer Or Manager**

Flow of events:

1. Aysel has just completed reservation. She checks her profile and makes a few changes.
2. She has finished doing all she wants to do and decides to log off.
3. She looks for the log out button in main page and touch it.

4. System immediately logs him out.
5. Aysel touches logout button and logout.

Scenario 6:

Scenario name: Help

Participant actor instances: Aysel: Customer Or Manager

Flow of events:

1. Aysel open the app.
2. She try to be register fill all fields email, phone number, username, password, ConfirmPassword

3. She received error, "help" tells her how can solve problem.
4. She apply it and solve the problem she is registered anymore.

Scenario 7:

Scenario Name: Sign Up as a Customer

Participant actor instances: Alper: Visitor

The flow of events:

- 1-) Alper wants to reserve a table from a restaurant with using restaurant reservation application.

For doing this, firstly he has to sign up the application. When he opening the application the app shows him Login and Sign-Up button. When he chose the sign-up button, system redirect him to the sign up page.

- 2-) The sign-up page consists of required information. So Alper fills the username and password information about himself. He must to select manager or Customer options.

3-) After pressing the sign-up button if any error occurs the system give him an information about it and want to fix them.

4-) If any error occurs event #3 is repeated. If everything is fine, the new page informs Alper to activate his membership via e-mail address. The system sends an activation email to Alper's given e-mail.

5-) Alper opens corresponding e-mail and opens the activation link given. The new page informs him that his account is activated.

6-) Then he logs out from the application.

Scenario 8:

Scenario Name: Sign Up as a Manager

Participant actor instances: Mert: Visitor

The flow of events:

1-) Mert wants to add his managing restaurant to the restaurant reservation application.

For doing this, firstly he has to sign up the application. When he opening the application the app shows him Login and Sign-Up button. When he chose the sign-up button, system redirect him to the sign up page.

2-) The sign-up page consists of required information. So Mert fills the username and password information about himself. He must to select manager or Customer options.

3-) After pressing the sign-up button if any error occurs the system give him an information about it and want to fix them.

4-) If any error occurs event #3 is repeated. If everything is fine, the new page informs Mert to activate his membership via e-mail address. The system sends an activation email to Mert's given e-mail.

5-) Mert opens corresponding e-mail and opens the activation link given. The new page informs him that his account is activated.

6-) Then he logs out from the application.

Scenario Sign Up Extensions:

3-) Mert may enter invalid username or password or he may enter a username already taken.

Scenario 9:

Scenario Name: Disable Restaurant

Participant actor instances: Burak: Admin

The flow of events:

1-) Burak's main job in the application is managing the application, so he must answer the requests. He wants to check waiting request. Burak opens the Restaurant Booking Application for log into his account.

2-) He fills username and password fields. He presses the log-in button.

3-) Both password and email match, there is no error so logging in process is successfully completed.

4-) When Burak presses the show waiting disable requests button the system shows him the waiting requests.

5-) When Burak's examination finish about the restaurant deleting request he choose a button accept or reject.

6-) If he chooses accept button the system disable it from database.

7-) The system sends a feed back to the manager about the deletion is successful or not.

8-) Then he logs out from the application.

Scenario Disable Restaurant Extensions:

3-) Burak may fill the blanks wrong and if username and password don't match, the system can send an error message. Repeat step 2.

Scenario 10:

Scenario Name: View Profile

Participant actor instances: Abdul: Customer

1-) Abdul wants to see his own profile for controlling his information. Abdul opens the Restaurant Booking Application for log into his account.

2-) He fills username and password fields. He presses the log-in button.

3-) Both password and email match, there is no error so logging in process is successfully completed.

4-) When he opened the Restaurant booking application he presses the view profile button.

5-) The opening page shows him his own profile page and his information.

6-) He completes his examination about his own profile.

7-) Then he logs out from the application.

Scenario View Profile Extensions:

3-) Abdul may fill the blanks wrong and if username and password don't match, the system can send an error message. Repeat step 2.

Scenario 11:

Scenario Name: View Restaurant

Participant actor instances: Alper: Customer

1-) Alper wants to reserve a table for meeting with his colleagues and before doing this he wants to examine the restaurant profile. For looking a suitable place he opens the Restaurant Booking Application for log into his account.

2-) The username and password fields are filled. He presses the log-in button.

3-) Both password and email match, there is no error so logging in process is successfully completed.

4-) When he opened the Restaurant booking application he is searching a specific restaurant or he can look at all restaurants and when he finds the restaurant he is pressing the restaurant name.

5-) After this steps the system presents the restaurant profile into the screen.

6-) Then he logs out from the application.

Scenario View Restaurant Extensions:

3-) Alper may fill the blanks wrong and if username and password don't match, the system can send an error message. Repeat step 2.

7-) If he searches a specific restaurant name for viewing it may not registered the system or it may not exist.

Scenario 12:

Scenario Name: Change Reservation

Participant actor instances: Alper: Customer

1-) Alper wants to check his reservation list to check if there is a problem or not.

2-) He is login to his restaurant booking application account.

3-) Then he presses the show my reservations button and the system presents the reservation page.

4-) He sees a time issue about one of his reservations.

5-) He presses the change reservation button and the system wants to choose a reservation which he wants to change from him.

6-) Then the system wants a new reservation time and place from him.

7-) When he enters his new reservation date and place the system update this information and presents it into my reservations section.

8-) Then he logs out from the application.

Scenario Change Reservation Extensions:

2-) Alper may fill the blanks wrong and if username and password don't match, the system can send an error message. Repeat step 2.

7-) The system may occur a problem about same time reservation if this happens the system turns a warning message "The place just filled." and the system opens reservation screen again.

Scenario 13:

Scenario Name: Forgot Password

Participant actor instances: Alper: Customer

1-) Alper wants to login his restaurant reservation application.

2-) He opens the application and when he tries to enter his own username and password he figured out he forgot his password and wants a new password.

3-) He presses forgot password button the opening page is enter new password page. The required fields are e-mail and new password and repeat new password.

4-) He fills the fields and create new password. The system updates his password to new one.

Scenario Forgot Password Extensions:

4-) Invalid password may be entered.

Scenario 14

Scenario name: Edit Profile

Participant actor instances: Berkay: Customer

Flow of events:

1. Berkay logs into the system by filling his username and password informations.
2. He taps the Edit Profile button on his Profile screen.
3. He can choose to modify:
 - Name
 - Surname
 - Email
 - Password
 - Phone number
4. He makes the change what he wants to make and taps to Save Changes button.
5. He finally logs out to the system.

Scenario 14 Extensions:

- 1a. Username or password not true so the system send notification message to customer.
- 3a. Password matched the old password so it's rejected.
- 3b. Password is too short.
- 3c. Password does not match the confirm password so verification failed.
- 3d. Email is already exist so it's rejected.
- 3e. Inappropriate phone number so it's rejected.

Scenario 15

Scenario name: Edit Restaurant

Participant actor instances: Ceren: Manager

Flow of events:

1. Ceren logs into the system by filling username and password fields.
2. She choose restaurant what she wants to edit in My Restaurants screen.
3. She can choose to modify:

-Restaurant's Name

-Address

-Phone number

-Features

-Table capacity

-Seat capacity

4. She makes the change what she wants to make and taps for save changes.
5. She finally logs out the system.

Scenario 15 Extensions:

- 1a. Username or password not true so the system send notification message to customer.
- 3a. Inappropriate phone number so it's rejected.

Scenario 16

Scenario name: ShowMyReservations

Participant actor instances: Ertan: Customer

Flow of events:

1. Ertan logins to the application by filling username and password informations.

2. Ertan see his all reservations by using the My Reservations button.
3. He checks his reservations.
4. He logs out to the system.

Scenario 16 Extensions:

- 1a. Username or password not true so the system send notification message to customer.

Scenario 17

Scenario name: AddRestaurant

Participant actor instances: Alper: Admin

Flow of events:

1. Alper logins to the application by filling username and password informations.
2. He opens his inbox for see the waiting add restaurant requests.
3. He choose a waiting add request for answering.
4. He examine the request and accept or decline the request.
5. System send a notification message to manager.
6. He finally logs out the system.

Scenario 17 Extensions:

- 1a. Username or password not true so the system send notification message to customer.

Scenario 18

Scenario name: SendAddRestaurantRequest

Participant actor instances: Gizem: Manager

Flow of events:

1. Gizem logins to the system by filling her username and password informations.
2. She enters add restaurant section using Add Restaurant button in the My Restaurants screen.
3. She fills the all fields about restaurant informations:
 - Restaurant's name
 - Address
 - Phone number
 - Features
 - Table capacity
 - Seat capacity
4. She sends to request to the admin of the application using Send Request button.
5. She finally logs out the system.

Scenario 18 Extensions:

- 1a. Username or password not true so the system send notification message to customer.
- 3a. Not all fields are filled so it's rejected.
- 3b. Inappropriate phone number so it's rejected.

Scenario 19

Scenario name: ShowReservationList

Participant actor instances: Burak: Manager

1. Burak logins to the system by filling his username and password informations.

2. He chooses the restaurant where he wants to see the reservation list in the My Restaurants screen.
3. He taps the Show Reservation List button in the restaurant's screen
4. He chooses the date what he wants to see.
5. He chooses the time and he see all reservations.
6. Finally he logs out the system.

Scenario 19 Extensions:

- 1a. Username or password not true so the system send notification message to customer.

Scenario 20

Scenario name: Search for a Restaurant

Participant actor instances: John: Customer

Flow of events:

1. John is already registered as a user and he wants to search a particular restaurant by name. He logs in using his username and password.
2. He navigates to the search bar which is at the top of his screen.
3. He taps on it and the keyboard pops up.
4. John then types in the name of the restaurant he has in mind.
5. The restaurant(s) matching the name he entered a displayed.
6. He logs out successfully.

Scenario Extensions:

- 1.1. John enters the wrong username or password. He is asked to re-enter the password.

5.1. No restaurants match the name entered by John, as such, no result is displayed.

Scenario 21

Scenario name: Rate a Restaurant

Participant actor instances: Wayne: Customer

Flow of events:

1. Wayne just finished eating at a restaurant he reserved via Glass Booking. After getting home he decided to rate the restaurant based on what he experienced.
2. He logs in to the app using his username and password.
3. He searches for the restaurant that he reserved for by name.
4. Wayne is able to rate it between 1 star and 5 stars (5 being the best).
5. He rates it a high 4 star because he enjoyed his time at the restaurant.
6. He logs out successfully.

Scenario Extensions:

- 2.1. Wayne enters the wrong username or password. He is asked to re-enter the password.

3.1. The name of the restaurant doesn't match any existing ones.

Scenario 22

Scenario name: Disable Account as a Customer

Participant actor instances: Blake: Customer

Flow of events:

1. Blake has decided he wants to disable his account. He logs in using his username and password.
2. He goes to view his profile where he can make changes and updates.
3. He located the disable account button and taps it.
4. He is asked to enter his password for confirmation.
5. Finally, his account gets disabled making it inactive.
6. He logs out successfully.

Scenario Extensions:

1.1 Blake enters the wrong username or password. He is asked to re-enter the password.

4.1. Password confirmation fails due to mistake and account fails to disable.

5.1 Account is not disabled due to system error. He is asked to try again later.

Scenario 23

Scenario name: Disable Account as a Manager

Participant actor instances: Faisal: Manager

Flow of events:

1. Faisal has decided he wants to disable his account. He logs in using his username and password.
2. He goes to view his profile where he can make changes and updates.
3. He located the disable account button and taps it.
4. A confirmation message is displayed letting Blake know that all the restaurants that are displayed on the system will be lost.
5. He confirms and taps ok.
6. The system ask for his password to confirm the deletion.
7. He enters the password and all his information is hidden from system.
8. He logs out successfully.

Scenario Extensions:

- 1.1 Faisal enters the wrong username or password. He is asked to re-enter the password.
- 5.1. He doesn't press ok after getting the confirmation message and the disable action is aborted.
- 6.1 Password confirmation fails due to mistake and account fails to disable.
- 7.1 Account is not disabled due to system error. He is asked to try again later.

Scenario 24

Scenario name: Sending Disable Restaurant Request

Participant actor instances: Marianne: Manager

Flow of events:

1. Marianne is a restaurant manager who wants to disable her restaurant from the system. She needs to send a request to the admin firstly.
2. She is already logged in to her account.
3. She goes to view her profile.
4. On that menu she taps on show my restaurants. She is displayed with the restaurant(s) that she has registered on the system.
5. She taps on disable restaurant. She is allowed to select multiple ones (if she has) using a check box.
6. She confirms her selection and continues.
7. She is prompted to enter a password for confirmation.
8. Her request is sent to the admin.
9. She logs out successfully.

Scenario Extensions:

4.1 Upon viewing, the manager has no registered restaurants currently on the system and so no result is displayed.

6.1 No restaurant was selected so selection cannot be confirmed. The system asks to choose at least one restaurant to disable.

7.1 She entered the wrong password so confirmation fails. She is asked to try again.

8.1 Request fails to send due to system error.

Scenario 25

Scenario name: ViewMyRestaurants

Participant Actor instances: Mbemba: Manager

Flow of events:

1. Mbemba is already signed into his account. He wishes to view the restaurants he has registered with the system.
2. He taps on view profile to see further information about his account.
3. On that menu he location the show my restaurants button and he taps on it.
4. The registered restaurants he has available are displayed to him.

Scenario Extensions:

4.1 Upon viewing, the manager has no registered restaurants currently on the system and so no result is displayed.

Scenario 26:

Scenario Name: Edit Restaurant Profile

Participant actor instances: Mert: Manager

- 1-) Mert wants to do some changes about his restaurant information.
- 2-) He enters restaurant booking application with using his username and password and he presses my restaurants page.
- 3-) Then he presses edit my restaurant and then he chooses a restaurant from the list.
- 4-) After he fills the blanks which he wants to change he press send request button.
- 5-) System transmit the request to the admins inbox.

Scenario Edit Restaurant Profile Extensions:

- 2-) Username and password don't match.
- 3-) He may enter invalid information. Like string for table number.

Scenario 27

Scenario Name: Enable Restaurant for Profile

Edit

Participant actor instances: Hobbs: Admin

Flow of Events:

1. Hobbs opens his inbox to see the requests waiting for being accepted to allow for restaurant editing.
2. He opens the first request and he sees the restaurant manager making the request and he sees the restaurant the edit is going to be made for.
3. Request is accepted by Hobbs so edit access has been enabled for the manager.

Scenario Extensions:

- 1.1 There are no waiting requests in inbox.
- 3.1 Request is not accepted so access for edit is not enabled.

Use case model

Use Case 1:

Use case name: Login (Low Priority)

Participant actors: Customer, Admin Or Manager

Flow of events:

- 1) The person first enters to Online Restaurant Booking system.
- 2) The system presents the “Login Form” to the user.
- 3) The Person enters username into username text field on the screen, also enters own password into password text field on the screen. Lastly, the user sends a request to the system by using login button on the screen to be logged in.
4. The system checks the username and password from the “Person” table so that the system allows the user to login. Then it determines role of the user by retrieving the role data from the “User” and redirects to proper screen.(Manager or Admin)

Entry Condition: The user enters to login screen.

Exit Condition: The user is logged in, OR, The user has received an explanation indicating why he/she could not login.

Use Case 2:

Use case name: Make Reservation (High Priority)

Participant actors: Customer

Flow of events:

- 1) The customer chooses how many people to book.
- 2) The system brings up the times when there are as many spaces as the number of people to book.
- 3) The customer touch the time. System redirects to reserve screen.
- 4) The customer reservation completes touching the reserve button.

Entry Condition: Admin is logged into restaurant booking system.

The person choosed which restaurant wants to make reservation.

Exit Condition: The person is made Reservation, OR, if restaurant was full in this time, the person can not make reservation. The person received an info about this time not available.

Use Case 3:

Use case name: Cancel Reservation (Low Priority))

Participant actors: Customer

Flow of events:

1) It sends a request to the system to see own reservation screen.

2) The system presents the “Own Reservation Screen” to the person.

The Customer touch the reservation which made reserve.

3) The system present reservation information

4) The Customer touch the cancel button

5) The system delete it in reservation table

Entry Condition: The Customer who logged in the system and has a reservation.

Exit Condition: The Customer is cancelled the reservation.

Use Case 4:

Use case name: Choose Restaurant (Low Priority))

Participant actors: Customer, Visitor

Flow of events:

1) The Customer search restaurant

2) The system bring online restaurants

3) The Customer touchthe resraurant which the person wants to reservation.

Entry Condition:

The person searched a restaurant.

Exit Condition: The person can see restaurant view screen.

Use case 5:

Use case name: Logout (High Priority)

Participant actors: Customer, Admin Or Manager

Flow of events:

1)The person touchown profile button

2)The system bring profile screen

3)The person touch logout button

Entry Condition: The person logged in the app.

Exit Condition: The Person logout the restaurant booking system.

Use case 6:

Use case name: Help (Low Priority)

Participant actors: Customer or Manager

Flow of events:

1)The person fill in the blank: Email, TelNo,Username,Passoword,ConfirmPassword

Name, Surname.

2)The person received error,

3)"Help" tells to the person why he/she taken this error.

Entry Condition: Open register screen.

Exit Condition: The person is registered.

Use Case 7:

Use Case Name: Sign Up(Low Priority)

Participant Actors: Initiated by Visitor

Flow of Events:

1-) The person is firstly open the application and see the sign up button and presses it.

2-) The application opens the sign up page.

3-) The person is filling username, e-mail, phone number, confirm password fields about himself/herself. And choose his/her role for this application there is a radio button for this selection. (The role of person must be Customer or manager)

4-) The person presses the sign up button and it will send a request to the system.

Entry Condition: The person enters to the sign up screen.

Exit Condition: The Customer signed up OR The system has send an error message about why he/she couldn't signed up(can be invalid username/password).

Use Case 8:

Use Case Name: Disable Restaurant(Low Priority)

Participant Actors: Initiated by ADMIN

Flow of Events:

1-) The admin of the system wants to look waiting requests for disable restaurant.

2-) He enters the application with using his username and password.

3-) He touch his inbox and looks for waiting disable restaurant requests.

4-) Admin analyze the request and accept or reject the disable request. With using accept or disable buttons.

Entry Condition:

The manager must be logged into the system.

The manager must sends a Disable Restaurant Request.

Exit Condition:

The System returns an information about whether the Restaurant Profile disabled or not to the Admin.

The system sends an information about whether the admin accept the deletion or not.

Use Case 9:

Use Case Name: View Profile(Low Priority)

Participant Actors: Initiated by CUSTOMER

Flow of Events:

1-) The Customer firstly opens the application and sign in with username and password.

2-) The Customer presses my profile button and the system presents the his/her own profile page.

Entry Condition:

The Customer must logged in the his/her account already.

Exit Condition:

The system shows his own profile.

Use Case 10:

Use Case Name: View Restaurant (Low Priority)

Participant Actors: Initiated by CUSTOMER

Flow of Events:

- 1-) The Customer presses the search button on the screen and write a restaurant name or he/she can see all restaurants also.
- 2-) The Customer choose a restaurant and presses its name
- 3-) The system opens the Restaurants own page and presents it to the Customer.

Entry Condition:

The Customer must be logged in.

The Customer must use search button for finding restaurant

Exit Condition:

The system presents the Restaurant page.

Use Case 11:

Use Case Name: Change Reservation (Low Priority)

Participant Actors: Initiated by CUSTOMER

Flow of Events:

- 1-) The Customer presses enters the restaurant booking application and sign in.
- 2-) The Customer opens his/her reservations list and presses the reservation which he wants to change.
- 3-) The system opens the reservation page and requests to select a new booking date.
- 4-) When the changes done, the reservations information update into the system.
- 5-) The system shows new reservation in the Customer's reservations page.

Entry Condition:

The Customer must be logged in.

The Customer must already has a reservation.

Exit Condition:

Reservation date and information updated.

Use Case 12:

Use Case Name: Forgot Password (Low Priority)

Participant Actors: Initiated by CUSTOMER or Manager

Flow of Events:

1-) The Customer presses enters the restaurant booking application and sign in. The user enters his/her own username and password

2-) The system returns a warning about his/her password is wrong.

3-) The user see forgot password button and presses it because he/she doesn't remember his/her password.

4-) The system opens forgot password screen and wants him/her enters his/her username. Then he/she has to enter new password.

5-) When the changes done, the system updates password information about user.

Entry Condition:

The Customer must be logged in.

The Customer must already has a reservation.

Exit Condition:

The user's password information updated. OR

The system returns warning about invalid password text.

Use Case 13

Use case name: EditProfile (Low Priority)

Participant actors: Customer, Manager

Flow of events:

1. The customer sends a request to the system to edit his/her information which are his/her name, surname, email and phone number.
2. The customer edits the fields he/she wants, then he/she submits the form by using the Edit Profile button to be updated.
3. The system updates the information into the “Customer” table.

Entry Condition: The customer is logged into system.

The customer chooses Edit Profile option on the screen.

Exit Condition: Information of the customer has been updated. OR,

The customer has received an explanation indicating why he/she could not login.

Use Case 14

Use case name: EditRestaurantProfile (Low Priority)

Participant actors: Manager

Flow of events:

1. The manager sends a request to the system to edit restaurant information which are restaurant's name, address, phone number and features
2. The manager edits the fields he/she wants, then he/she submits the form by using the Edit Restaurant Profile button to be updated.
3. The system updates the information into the “Manager” table.

Entry Condition: The manager is logged into system.

The customer chooses Edit Restaurant Profile option on the screen.

Exit Condition: Information of the restaurant has been updated and the manager has received an acknowledgement message. OR,

The manager has received an explanation indicating why he/she could not login.

Use Case 15

Use case name: ShowMyReservations (Low Priority)

Participant actors: Customer

Flow of events:

1. The customer sees his/her reservations by using the Show My Reservations button.
2. The customer checks his/her reservations.

Entry Condition: The customer is logged into system.

The customer chooses Show My Reservations option on the screen.

Exit Condition: The customer see his/her own reservation

Use Case 16

Use case name: AddRestaurant (Low Priority)

Participant actors: Admin

Flow of events:

1. The admin opens his inbox and he see all the waiting add restaurant request from managers.
2. The admin examines the requests and he can accept or decline the requests.
3. The manager taps Send Request button to admin.
4. Admin takes request and admin can accept or decline the request.
5. The system sends a “Notification Message” to Manager.
6. The system updates the information into the “Restaurant” table.

Entry Condition: The Admin is logged into system.

Exit Condition: The admin accept or decline the waiting request in his/her inbox. OR,

The admin has received an explanation indicating why he/she could not login.

Use Case 17

Use case name: SendAddRestaurantRequest (Low Priority)

Participant actors: Manager

Flow of events:

1. The manager sends a request to the system to add restaurant information which are restaurant's name, address, phone number, capacity and features
2. The manager fills the restaurant name, address, phone number, capacity and features fields on the Add Restaurant screen
3. The manager taps Send Request button to admin.
4. The system sends request to the admin

Entry Condition: The manager is logged into system.

The customer chooses Add Restaurant option on the screen.

Exit Condition: The manager add his/her restaurant successfully. OR,

The manager has received an explanation indicating why he/she could not login.

Use Case 18

Use case name: ShowReservationList (Low Priority)

Participant actors: Manager

Flow of events:

1. The manager has to choose the restaurant in the My Restaurant screen.
2. The manager taps the Show Reservation List button.
3. The manager has to choose the date and time.
4. The manager sees all the reservations in the chosen date and time.

Entry Condition: The manager is logged into system.

Exit Condition: The manager see the reservation list in chosen date and time.

Use Case 19

Use Case Name: Edit Restaurant Profile(Low Priority)
Participant Actors: Initiated by MANAGER
Flow of Events:
1-) The manager opens the application and sign in with using his/her username and password.
2-) Then he/she opens his/her own restaurants page. This page has his/her all restaurants list.
3-) Then he presses edit my restaurant and then he chooses a restaurant from the list.
4-) He/she edits the changes and send request to the admin.
5-) Then he logs out from the system.
Entry Condition: The manager must enter the application.
Must choose a restaurant and do changes.
Exit Condition: The request can send seamlessly or the system can returns an error like invalid table number, invalid number of people.

Use Case 20

<i>Use case name:</i> EnableRestaurantProfileEdit
<i>Participant actors:</i> Initiated by Admin
<i>Flow of events:</i>
1. Admin presses the inbox icon to view requests for editing restaurant profile.
2. Glass Booking opens the inbox and displayed the queue that has the request messages waiting.
3. Admin selects a request to enable/accept.
4. Glass Booking gets the request details which include the manager ID requesting for the restaurant profile edit access and the restaurant he is requesting to edit.
5. Admin enables access for the requesting manager to be able to edit the restaurant so they can make changes for the restaurant on the system.
6. Glass Booking system opens access for the manager to make edits to the restaurant profile thus, updating the restaurant information on the system.

Use Case 21

Use case name: SearchRestaurants

Participant actors: Initiated by Customer

Flow of events:

1. Customer taps search bar which is at the top of screen.
2. Glass Booking system responds by popping up the keyboard for typing.
3. Customer then types in the *name* of the restaurant they wish to search for. Once this is done they hit the “search” button.
4. Glass Booking takes this name that customer typed as the input and makes a search in the registered restaurants table. If there are any matches with the same name as the input, the system returns all these rows.

Entry Condition: Customer taps on the search bar to access keyboard.

Exit Condition: Glass Booking has completed a search for matching results for the input given by user.

Use Case 22

Use case name: RateRestaurant

Participant actors: Initiated by Customer

Flow of events:

1. Customer searches for restaurant by name by making use of search functionality or customer can view previously reserved restaurants.
2. Glass Booking system returns restaurant that was previously by the account that is being used to make this search.
3. Customer taps on the rate (star icon) button. There are five stars that they can use to give the rating. The customer taps the right number of stars they want.
4. Glass Booking takes this rating (number of stars) as the input and updates the restaurant's average rating.

Entry Condition:

- Customer has previously reserved from the restaurant before.
- The customer must be logged into the system.
- Customer has only one rating per reservation.

Exit Condition:

- Customer gets a confirmation that rating has been completed.
- Customer get an explanation explaining why rating was not completed.

Use Case 23

Use case name: SendRestaurantDisableRequest

Participant actors: Initiated by Manager

Flow of events:

1. Manager goes to view profile.
2. Glass Booking system returns a menu and there is a “View My Restaurants” function on that menu.
3. Manager opens the View My Restaurants which is accessible from the view profile menu.
4. Glass Booking searches for the restaurants that are registered to that manager.
5. Manager taps on disable restaurant button and selects which restaurant they want to send the disable request for. Manager then confirms selection and continues.
6. Glass Booking system takes the restaurant ID of the restaurant(s) that manager selected. It sends a password confirmation request to the manager.
7. Manager enters password to confirm request disable restaurant action.
8. Glass Booking sends the request for disabling restaurant to the admin inbox.

Entry Condition:

- Manager has at least one registered restaurant on the system.
- The manager must be logged into the system.
- Manager activates sending request for disable restaurant action using the disable button.

Exit Condition:

- Manager gets a confirmation that request has been sent.
- Manager has made selection of restaurants and entered password correctly to complete this action.

Use Case 24

Use case name: EnableRestaurantProfileEdit

Participant actors: Initiated by Admin

Flow of events:

1. Admin presses the inbox icon to view requests for editing restaurant profile.
2. Glass Booking opens the inbox and displayed the queue that has the request messages waiting.
3. Admin selects a request to enable/accept.
4. Glass Booking gets the request details which include the manager ID requesting for the restaurant profile edit access and the restaurant he is requesting to edit.
5. Admin enables access for the requesting manager to be able to edit the restaurant so they can make changes for the restaurant on the system.
6. Glass Booking system opens access for the manager to make edits to the restaurant profile thus, updating the restaurant information on the system.

Entry Condition:

- Admin opens request from the inbox.
- Admin has received edit restaurant profile request from manager.

Exit Condition:

- Admin enables/accepts manager request for editing restaurant profile.
- Admin rejects manager request for editing restaurant profile.
- Admin receives a message explaining why accepting/rejecting request was not successful.

Use Case 25

Use case name: DisableAccount

Participant actors: Initiated by Customer, Manager

Flow of events:

1. Customer or manager open “View Profile” from home page.
2. Glass Booking opens the profile menu where changes can be made.
3. There is a “Disable Account” button and they tap this button.
4. Glass Booking system sends a confirmation action by prompting a password request.
5. They input their password and press ok.
6. Glass Booking verifies the password and makes all information pertaining to the account are hidden.

Entry Condition:

- Customer or manager have pressed the disable button.

Exit Condition:

- Account has been disabled successfully.
- Customer or manager receive a message explaining why disabling account was not successful.
- Password verification failed.

Use Case 26

Use case name: ViewMyRestaurants

Participant actors: Initiated by Manager

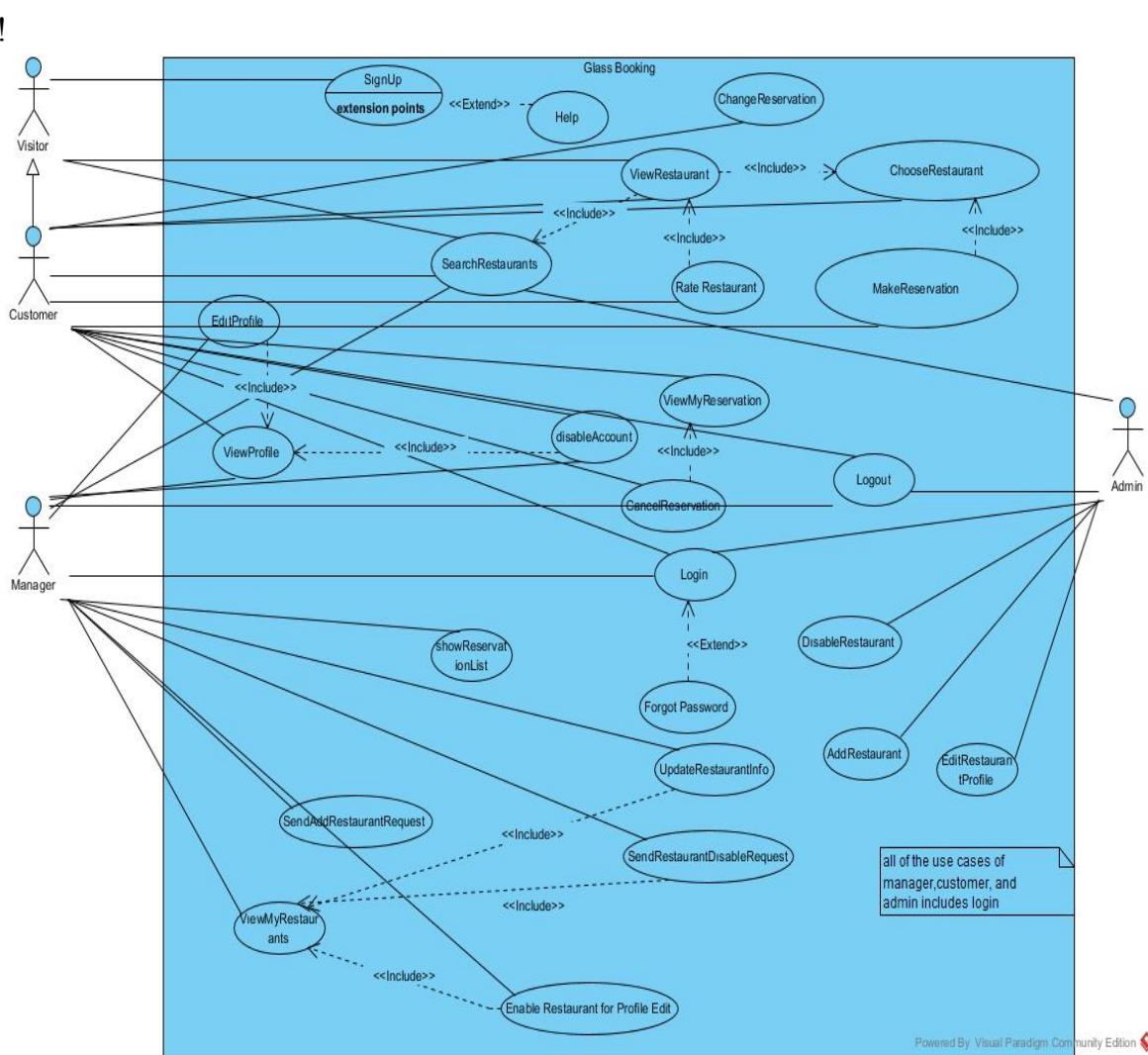
Flow of events:

1. Manager sends a request to the system to display the registered restaurants.
2. Glass Booking system searches the restaurants table and matches the restaurants registered to that manager ID.
3. The system returns the rows and displays the registered restaurants to the manager.

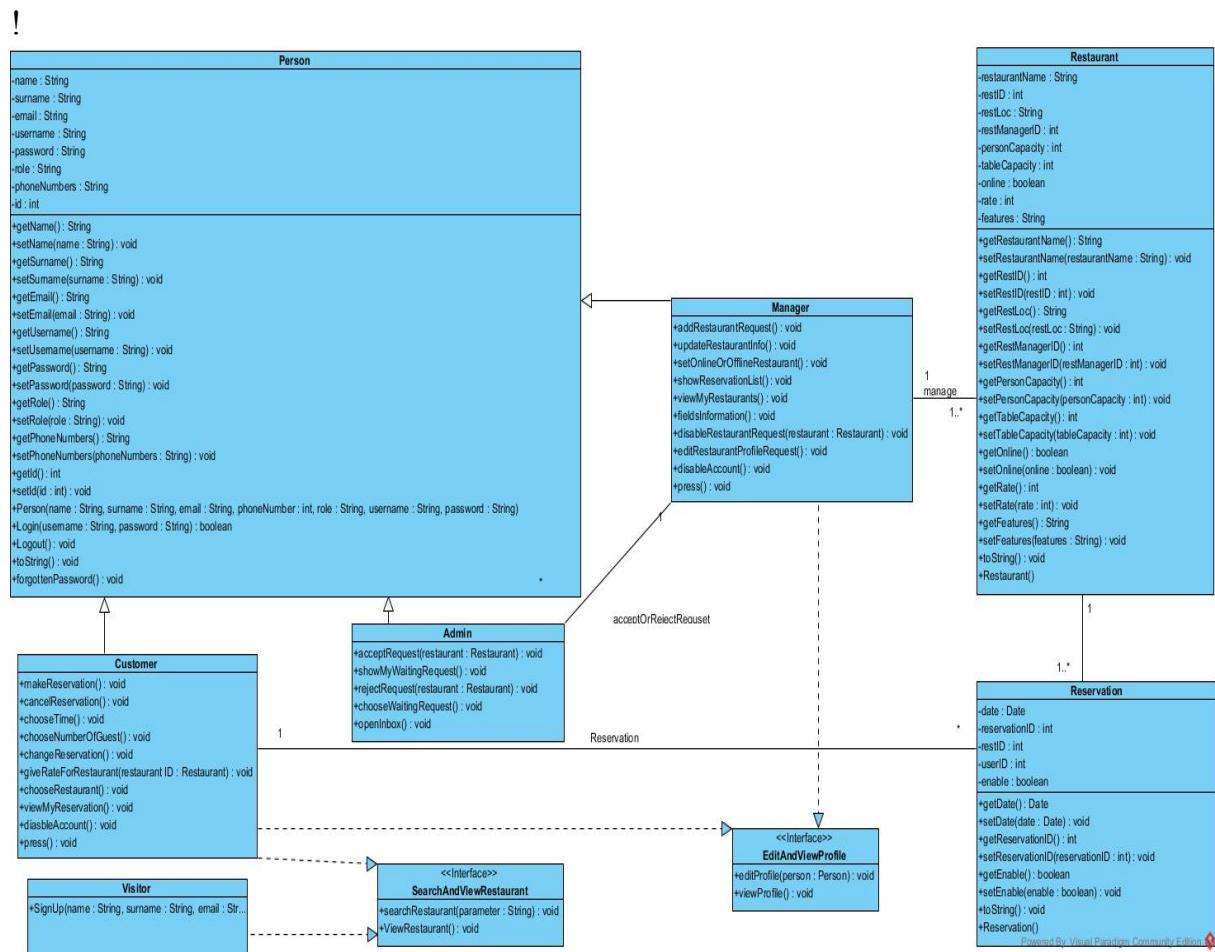
Entry Condition: The manager is logged in the system.

The manager selects view profile and taps display my restaurants button.

Exit Condition: The manager closes the page by either exiting the app OR going to previous/home page.

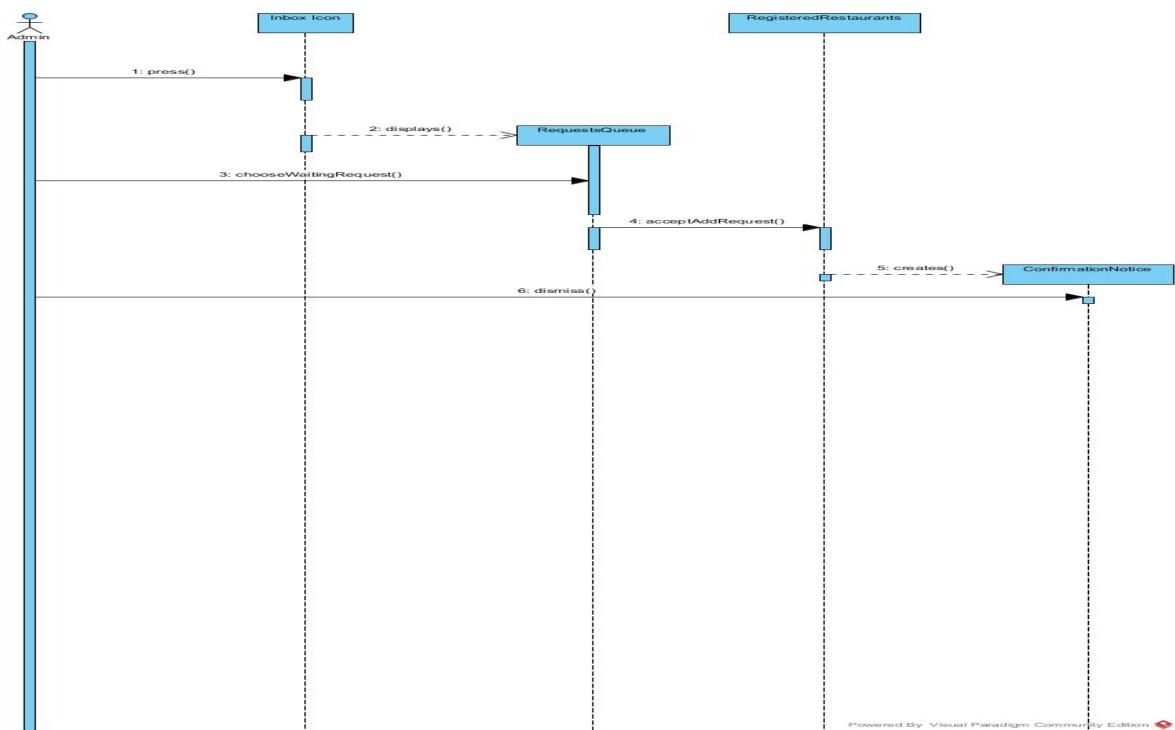


Object model



Dynamic model

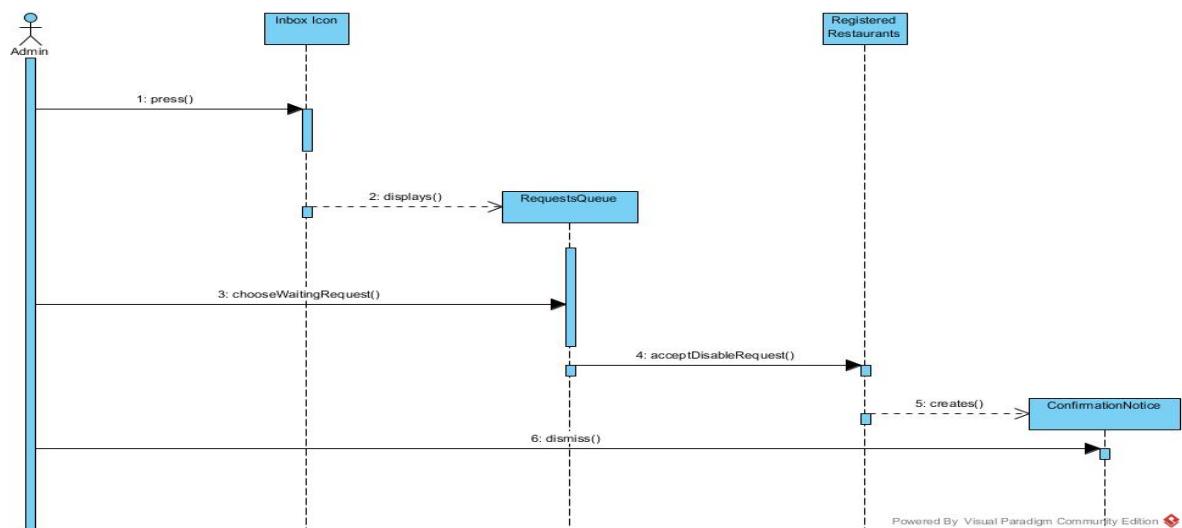
Add Restaurant (ADMIN)



Powered By Visual Paradigm Community Edition

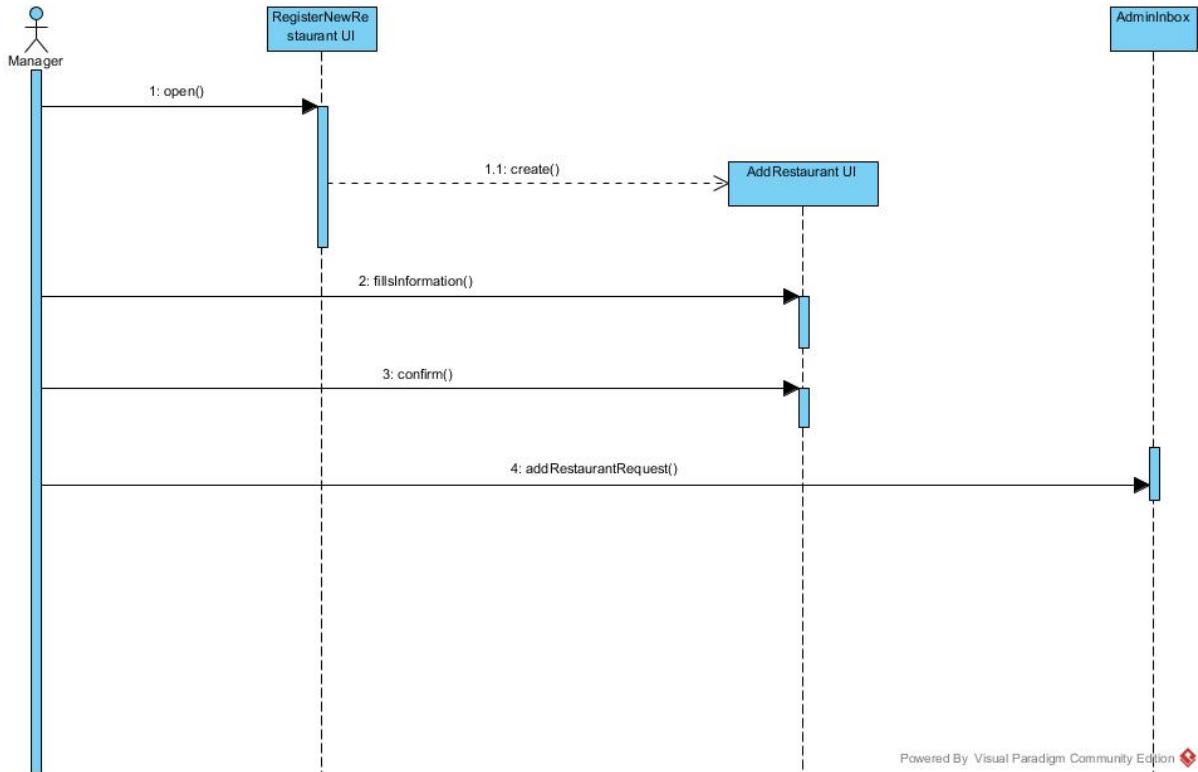
!

Disable Restaurant(ADMIN)

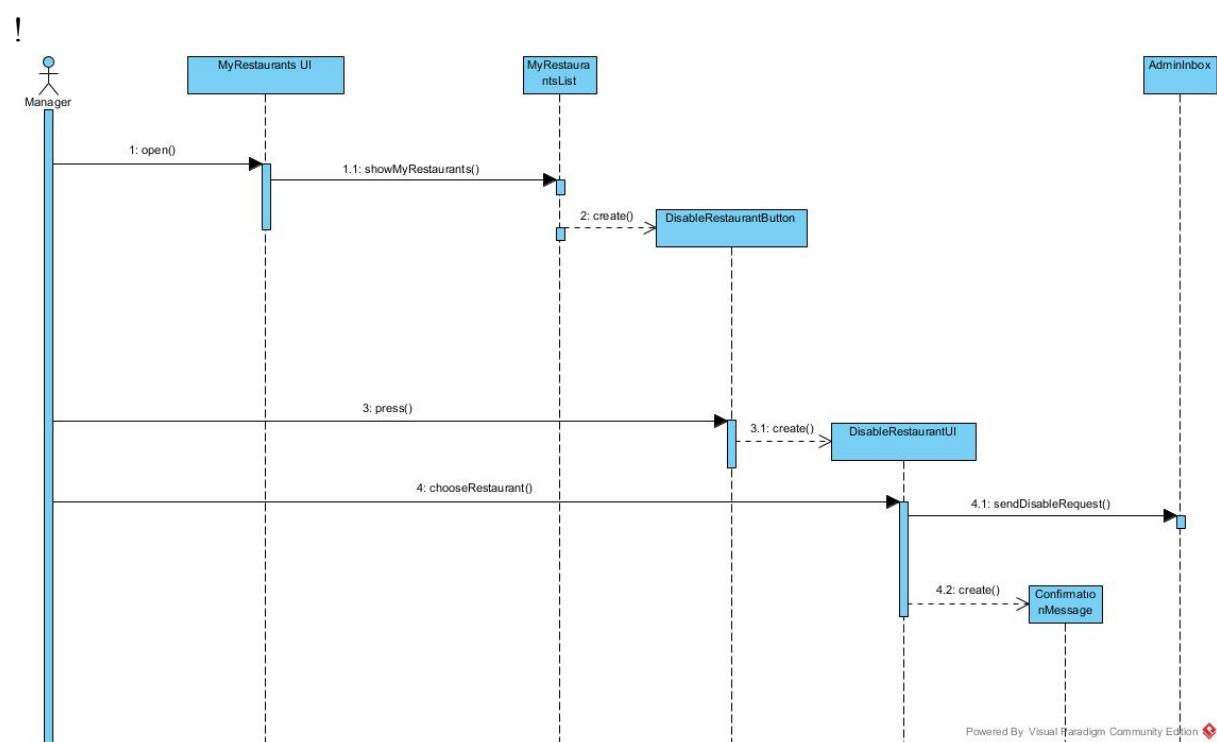


!

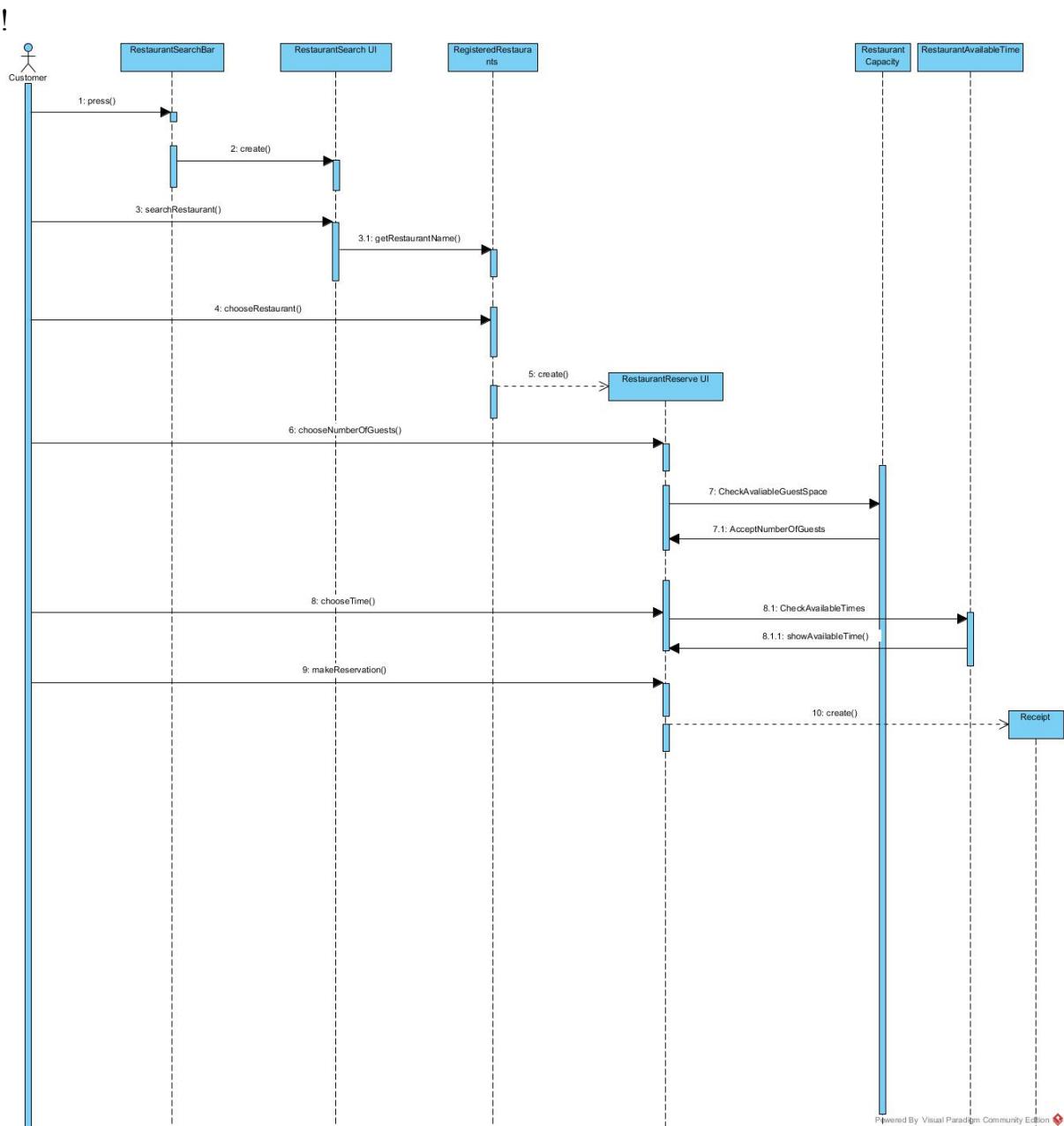
Send Add Restaurant Request(Manager)



Send Disable Restaurant Request(Manager)



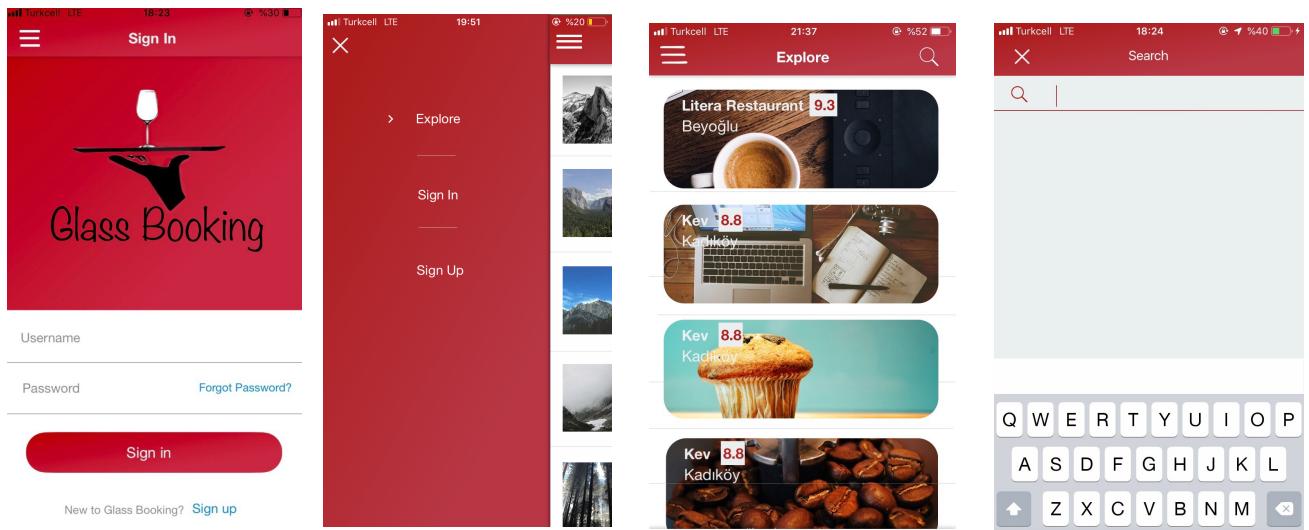
[Make Reservation](#)



Powered By Visual Paradigm Community Edition

Mock-Ups

Visitor's screens



Sign In
Sidebar

Explore

Search

The image shows two detailed screenshots of the app's features:

- Restaurant Detail Screen:** Shows a card for "Litera Restaurant" with a rating of 9.3. It includes a "Rate" button, a "Make Reservation" button, an "Address" section with the address "İstiklal Caddesi, Yeni Çarşı Caddesi, Beyoğlu, İstanbul", a "Phone Number" section with the number "0(535) 678 27 13", and a "Features" section.
- Sign Up Screen:** Shows a "Sign Up" button at the top. Below it are input fields for "Username", "Password", "Confirm", "E-mail", and "Phone Number". At the bottom is a "SIGN UP" button with "Customer" and "Manager" radio buttons.

Restaurant

Sign Up

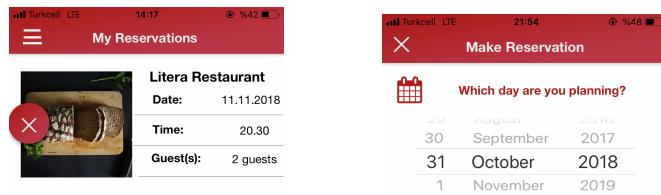
Customer's Screens

The image displays eight screenshots of the Glass Booking mobile application interface, arranged in two rows of four. The top row shows:

- Sign In**: A red screen with a logo of a hand holding a wine glass, and fields for 'Username' and 'Password'.
- Explore**: A screen showing restaurant cards for 'Litera Restaurant' (Beyoğlu, 9.3), 'Kev' (Kadıköy, 8.8), and 'Kev' (Kadıköy, 8.8). The background features a blurred image of a coffee cup.
- Search**: A search screen with a search bar and a keyboard at the bottom.
- Litera Restaurant**: A detailed view of the 'Litera Restaurant' card, showing a star rating of 9.3, address (İstiklal Caddesi, Yeni Çarşı Caddesi, Beyoğlu, İstanbul), phone number (0(532) 678 27 13), and features.

The bottom row shows:

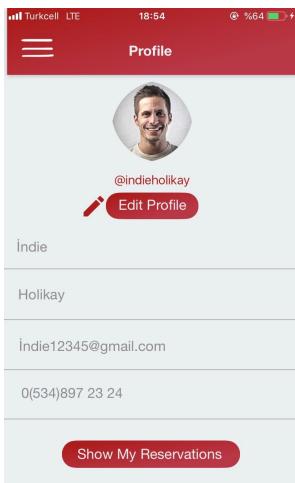
- Make Reservation**: A screen for selecting a date, showing a calendar from August 2017 to December 2020. It includes fields for time (19.00) and guests (2 guests).
- Explore Restaurant**: A screen titled 'Explore Restaurant' with a sub-section 'My Reservations' showing a reservation for 'Litera Restaurant' on 11.11.2018 at 20.30 for 2 guests.
- Search**: A navigation menu with options: Profile, Explore, My Reservations, and Logout.
- Profile**: A user profile screen for 'indieholikay' with fields for name (@indieholikay), email (indie12345@gmail.com), and phone number (0(534)897 23 24). It also shows a profile picture and a 'Save Changes' button.



Make



Reservation
Reservation
Profile



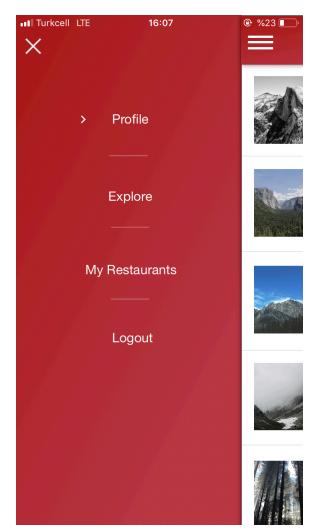
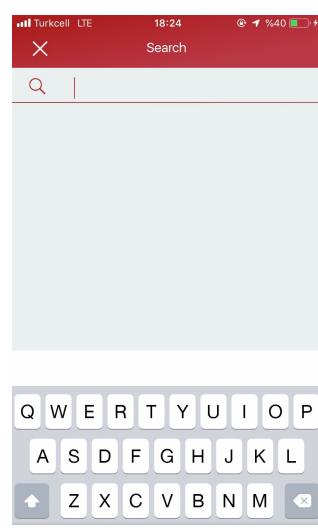
List Sidebar

Profile

Reservation

Reservation

Make Reservation



[Edit Profile](#)[Cancel Reservation](#)[Change Reservation](#)

Manager's Screens

[Login](#)[Explore](#)[Search](#)[Sidebar](#)

Restaurant name

Address

Phone Number

Features
 +

Table Capacity

Seat Capacity

[Save Changes](#)


Midpoint
Split

[Delete Restaurant](#)

Restaurant name

Date

Price

Guests

Date

[Delete Restaurant](#)

Profile


@indieholikay
[Edit Profile](#)

Indie

Holikay

indie12345@gmail.com

0(534)897 23 24

Phone Number

Features

Table Capacity

Seat Capacity

[Send Request](#)

Profile


@indieholikay
[Save Changes](#)

Indie

Holikay

indie12345@gmail.com

0(534)897 23 24

Phone Number

Features

Table Capacity 70 Tables

Seat Capacity 280 Seats

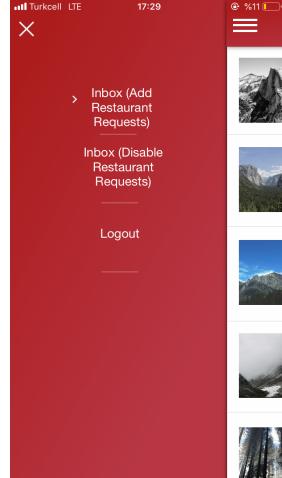
[Delete Restaurant](#)[Add Restaurant](#)

My Restaurants

Restaurant

[Edit Restaurant](#)[Reservation List](#)[Profile](#)[Edit Profile](#)

Admin's Screens

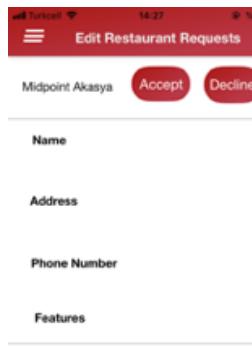


New to Glass Booking? [Sign up](#)

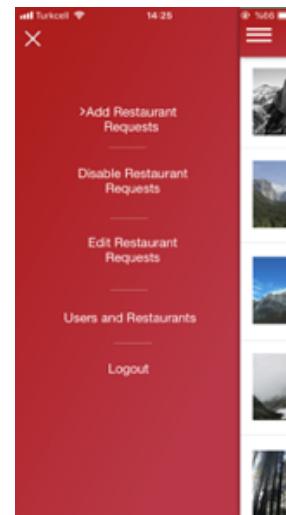
Login



Sidebar



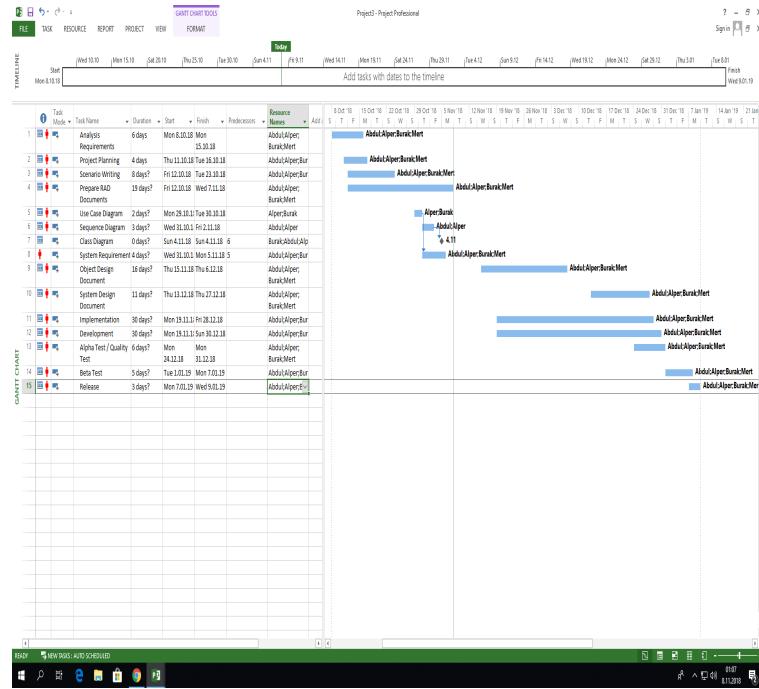
Add Restaurant Request



Disable Restaurant Request



3.5.Project Schedule



4.Glossary

Admin: A person who manages all the content (customers, managers, database etc.) of the Glass Booking System.

Customer: A person who uses the Glass Booking System for searching restaurant and making reservations.

Database: A collection of specific datas.

Guest: A customer at a restaurant.

Log in: Being entered to the Glass Booking System with a specific registered account.

Log out: Being exited from the Glass Booking System with a specific registered account.

Functionality: An action or control supported by our system for any of the users to carry out a task.

Manager: A person who can add, edit, delete restaurants and see all reservations in the Glass Booking System.

Password: A secret word or phrase that must be choose of all customers and managers when they are register.

Profile: A section where the customer's and manager's informations.

Rating: An action for customer to give a number to a restaurant according to experience in that restaurant.

Registered User: A user who has an account on the Glass Booking System

Reservation: An action for customers to have a reserving table and to eat meals in the specific date, time and a restaurant.

Sign Up: A section for visitors to be a customer or manager with entering their specific informations.

Use Case: A methodology used in system analysis to identify, clarify, and organize system requirements.

Visitor: A person who is an unregistered user in the system.

5. References

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
2. Reztoran Application - https://www.reztoran.com/?gclid=CjwKCAiAt4rfBRBKEiwAC678KeT-2nQLU1G5_JGZZoXuinQblP7TKPQN5B-0YsC3AiqfHo1nUyTr3hoCNfYQAvD_BwE
3. Our Restaurant logo - https://www.freepik.com/free-vector/restaurant-logo-design_1101126.htm#term=restaurant&page=1&position=20