Bus ticket/seat reservation

Requirements Specification and Analysis

Version 0.1

11.11.2018

Gamze Saraç

Recep İlkay Depe

Can Utku

Umutcan Neşet Ölçer

Prepared for

SE301 Software Engineering



Table Of Content

[1. Introduction 3](#_Toc529730990)

[1.1. Purpose of the System 3](#_Toc529730991)

[1.2. Scope of the System 3](#_Toc529730992)

[1.3. Objectives and Success Criteria of the Project 3](#_Toc529730993)

[1.4. Definitions, Acronyms, and Abbreviations 4](#_Toc529730994)

[1.5. Overview 4](#_Toc529730995)

[2. Current System 5](#_Toc529730996)

[3. Proposed System 5](#_Toc529730997)

[3.1. Overview 6](#_Toc529730998)

[3.2. Functional Requirements 6](#_Toc529730999)

[3.3. Nonfunctional Requirements 7](#_Toc529731000)

[3.4. System Models 7](#_Toc529731001)

[3.5. Project Schedule 33](#_Toc529731002)

[4. Glossary 34](#_Toc529731003)

[5. References 34](#_Toc529731004)

# Introduction

Today, technology is used for nearly everything in people’s daily lives. Technology is now a part of people that they can never get rid of. It makes people’s lives a lot better and easier. One of the products that people use actively every day is smartphone. People can travel, connect with each other, shop, and do lots of other things with using it. For making these things, mobile applications are being implemented.

In today’s tech era people often use their smart phones for online shopping. Through online shopping, buying an airplane ticket or getting a bus ticket by using a mobile application are commonly use. Many of the domestic bus travel companies started to take reservation on online platform which is more efficient, easy and effective way.

The crossing part that includes travelling and shopping defines our application: ‘’TrustBus’’. “TrustBus” is a mobile bus ticket application located in Istanbul, Turkey.

## Purpose of the System

The aim of the system is providing an efficient platform for users so they can search convenient trips to the places that want to travel and buy their tickets easily. Mobile-based application creates an ease for the customers, therefore they can search and buy their bus tickets quickly and without confused with the lots of processes they have to make.

## Scope of the System

There are 4 parts of usage in this system: Customer, Guest, Admin, and Manager.

On customer part, users can continue using the program either logging in or not. The main function that system provides is buying bus tickets. Application starts with the page that includes choosing functions of destination and date. Users can search their desired trip with this function. Customers who make login to the system, can get a free ticket if they buy 10 tickets before. Basically, system provides customers to search the wanted trips and buy tickets on these trips. Customers can also see their previous tickets, edit their profile information and freeze their account if they want.

Guests can use the system without login. They can search and see the trips that they want and buy tickets with their personal information. However, they can’t benefit on free ticket campaign.

On admin part, users have to login. These users can be defined as the head of the system. They have the authority to do every change on the system. Admin can update users’-both managers and customers- info, delete manager accounts, edit bus schedules such as add, update or delete trips, edit bookings such as update and delete.

On manager part, users have to login. These users can manage booking such as edit and delete them and edit bus schedules such as add, update and delete trips.

## Objectives and Success Criteria of the Project

Our objective is to create a system which has understandable, easy to use functions, fast interface which customers can search their targeted trips and buy their tickets quickly, irreproachable payment function which gives customer no suspicion on their safety. Lastly, creating a system which works for 24/7 so the customers can make their operations anytime they want. The success criteria for the project would be:

\*User’s satisfaction

\*The rate of usage

\*The popularity of the system

\*Usefulness

## Definitions, Acronyms, and Abbreviations

RAD: Requirements Analysis Document

BusTrust: Android Application which is going to be implemented.

MockUp: First design of application. Includes screenshots.

## Overview

This document contains the sections “Current System”, “Proposed System”, “Overview”, “Functional Requirements”, “Nonfunctional Requirements”, “System Models”, “Project Schedule”, “Glossary”, and “References”.

In the section “Current System”, we talked about the features and existing situation of the current bus ticket system which is based on buying the ticket on ticket shops. We mentioned the disadvantages of the system such as the wasted time on getting ticket and the disorders on setting up the passenger information and travel plans. We also explained that current system doesn’t satisfy the necessities of today’s world of technological possibilities. We mentioned that the current system stays very primitive with the situation of not using technology. In addition, we represented that current bus ticket system is not futuristic, not fast and requires a lot of man-power.

In the section “Proposed System”, we talked about our new bus ticket system’s features and advantages. We talked about the matters which the technology and the system bring us. For instance, using the system on buying bus ticket makes the procedure very fast and reduced the man-power to the minimal level.

In the section “Overview”, we mentioned the features, functions of the system and the system itself briefly. In addition, we made the explanation of usage on user’s view.

In the section “Functional Requirements”, we mentioned the functions and properties of our new system. We described the functions of our system in point of our actors: Customer, guest, admin, manager. We lastly advert on the relationship between our actors and the system which includes their interaction with system and their environment.

In the section “Nonfunctional Requirements”, we talked about nonfunctional parts of our new system such as usability, performance, reliability or availability and so on. We described all nonfunctional requirements of our project.

In the section “System Models”, we published scenarios and use cases of our project. We defined scenarios, actors, and use cases, flow events, object and dynamic models and other relevant subjects in this part.

In the section “Project Schedule”, we published our planned schedule of developing the project. We prepared a Gantt chart and showed our planned timeline.

In the section “Glossary”, we wrote the terminology of our project and identify the participating objects.

In the section “References”, we published the list of documents we used during the project development.

# Current System

As we mentioned on the introduction part, the current system is insufficient on many ways and has some disadvantages.

Firstly, the old system allows passengers to buy their tickets only on the ticket stores. This situation creates a waste of time for the customers. People must go to the store, get in ticket line if the store is the main one on the city, choose the appropriate trip and the seat, get the printed ticket and leave. This process needs lots of effort from both customers and the staff.

Secondly, when the customer wants to change his/her trip on trip’s hour or destination, he/she has to go to the store and repeat the above process again. This situation also creates a waste of time and the ticket changes may create problems on the system since the old (current) system is manual, runs slow and unobstructed to the errors.

Thirdly, the current system needs lots of workers on work field. At basic level, the store needs workers to use ticket system, communicate with customers, provide security and so on. Hence the worker means money spending for the company.

Fourthly, on manual system it is hard to keep in touch with the customers when the company needs to make proclamation about the trips or the campaign about prices. The company has to make advertisements on related environments or reach the customers one by one.

Lastly, the currents system doesn’t use the technology sufficiently. In the era of technology, the companies have to use the technology nearly on their every function and process in order to get rid of extra costs and speed up their system. In general, the current system is functional but not optimal.

# Proposed System

The new system designed as mobile application. This system brings lots of ease and advantages with itself.

Firstly, customers can buy their tickets on application in a short time. At basic level, they just have to enter the application, search their targeted trip and make the payment. This short procedure saves people from the necessity of visiting the ticket store. The procedure creates a save of time for customers.

Secondly, customers easily can make a request about their trips such as changing the trip time or the destination. They can use the request function, also the admin or the manager can easily see the request and make the alterations.

Thirdly, since the system turns into to the mobile application, it doesn’t need many workers in field. Surely the system needs programmers, designers and other related workers and paying for the system may be costly in the beginning, however in long term it will be economically proper than the old system and it needs much less worker once the system starts to run.

Lastly, the company can use the system efficiently when they want to make some announcements and campaigns just making notifications. This way, the company can reach the customers in a second. In addition, the mobile application is one of the most efficient ways to use technology. Transferring the manual system to the mobile application can be interpreted as the thriving usage of technology in the job sector.

## Overview

Trustbus is an online bus ticket system which can be used by everyone who lives in Turkey. The system includes four types of users: Customer, Guest, Admin and Manager. Although the customer and the guest nearly have the same specialties, customers sign up to the system and this sign up procedure gives the customers extra possibilities such as free ticket opportunity.

Customers use the system to search the appropriate trip to the destination which they want to travel and buy the ticket. They have the privilege of one free ticket when they buy 10 tickets. They can make updates on their profile information.

Guests have the same functions with customers, however they don’t have to register or login to the system,they can just search and buy their tickets. Unfortunately they can’t benefit from the campaigns like free ticket.

Admins use the system to make backdoor adjustments, repairs such as editing managers’ and customers’ profiles, deleting managers’ profiles who don’t work on the company anymore, managing bookings and so on. Admins have all the abilities on the system.

Managers use the system to manage bookings and edit bus schedules. They fix the inconsistencies and discrepancies of the bookings and trips, make updates on the system.

In all, this will be a system which reduces the man-work, hours of process and problems on previous system.

## Functional Requirements

TrustBus android app is a mobile application which user can buy online ticket using their credit card. It contains SLL security for safety buying. System is stabilized and fast.  Bus ticketing system deals with maintenance records of each passenger who had reserved a seat for a journey.  Ticket system contains schedule, fare and detailed. However, application hide consumer details from other consumers. System never down also at payment never stops or causes error. System give gift when 10 ticked bought from one consumer. Guest can list trips without any login and can buy ticket with few information.

On admin side, TrustBus application has many different functions. Admin can do some changes on user and manager. Control trips but manager can only access with trips and booking. Update, delete and add trip on the application.

## Nonfunctional Requirements

*Usability*: For a consumer buying a ticket will be easy and every person can use the application without high knowledge.

*Reliability*: The system must be running %100 of the time when buying and also when listing trips. The system never crash.

*Performance*: The system must allow more than 1000 parallel user. 1000 user can use the system at same time. System must return response immediately.

*Supportability*: The system must be changeable and easily to maintenance.

*Implementation*: The system will be implemented on Android Studio. In addition, java will be used as the programing language. For Database the system uses SQLite, Firebase . User interface will be android mobile.

*Interface*: There is no external systems or legacy system work with. Thus, there is no interface requirements.

*Packaging*: No constraints on the actual delivery of the system are determined. Thus, packaging requirements will be decided in the future.

*Legal*: Despite,the system supports online payment , non of the card information are stored. During payment proccess VISA/MASTERCARD are responsible for caused issues.For additional information of payment contact with your bank. The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability. TrustBus does not guarantee any action based upon the receipt of such information.

## System Models

**Scenarios:**

**Scenario 1:**

**Scenario Name: Sign-Up**

**Participant actor instances: Darwin: Consumer**

The flow of events:

1. Darwin wants to sign-up the system. In order to do that he has to sign-up to the system. When he

Entered the site he touch green sign-up button which is located to the center of the screen.

2. The upcoming page consists of required information. So, Darwin fills the information which are:

Name, surname, date of birth, gender, e-mail address, phone number, social security number. Darwin filled blanks and agrees on terms & conditions then press the sign-up button.

3. If done, the new page informs the user to activate his membership via. e-mail address. Random generated password is sent to the user's given mail address.

4. The new page informs the user to enter with random generated password which sent to e-mail address. After log in with random generated password user fill the new password blank. System change user’s password from database.

**Scenario 1 Extensions:**

2a) After touching the sign-up button if any error occurs user is informed and wrong information are shown with red color. After re-filling, Darwin clicks the sign-up button again.

3a) If any error occurs event #3 is followed.

**Scenario 2:**

**Scenario Name: Log-in**

**Participant actor instances: Darwin: Member of the site**

The flow of events:

1. Darwin opens the TrustBus mobile application by ios device in order to log in.

2. The TrustBus e-mail address and password fields are filled. He touch the sign-in button.

4. Both password and email match, there is no error so logging in process is successfully

completed.

5. System briefly asks Darwin if he wants to be remembered next time he attempts logging in.

6. He chooses yes.

**Scenario 2 Extensions:**

2a. Darwin inputs wrong email or password so logging in the process fails and system give alert which written “wrong email or password”. He is asked to recheck the information he provided.

2b. Darwin inputs wrong email 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 new random generated password to his attached email address. After he enters email which comes email, he change with new password.

5. He chooses no so next time he tries logging he has to manually input all his information.

**Scenario 3:**

**Scenario Name: Edit Profile**

**Participant actor instances: Gamze: Member of the site**

Flow of events:

1. Gamze is an existing user and she wants to make some modifications to her profile.

2. She logs into the system.

3. She can choose to My Account—Edit Profile:

- profile picture

- information

4. She makes the change she wants to make and clicks save changes.

**Scenario 4:**

**Scenario name: Log-out**

**Participant actor instances: Aden: Member of the site**

Flow of events:

1. Aden decides to log out from account.

2. He touch My Account menu on the bottom tab bar and touch it.

3. Now, he click the logout button at right top of screen.

4. System immediately logs him out.

**Scenario 5:**

**Scenario name: Update User Info**

**Participant actor instances: Larry: Admin, Matt: Manager**

1. Larry enters the application to log in.

2. He chooses the administration log in to get in to the system.

3. He types his e-mail and password to log in and then clicks to continue.

4. Then, he finds the “Manager lists” option and clicks it.

5. In that list, he finds the Matt, who is a manager.

6. He clicks Matt’s profile and then clicks edit.

7. He changes Matt’s password and clicks “save”.

8. A warning appears to the screen which says “Are you sure?”

9. He taps “Continue” .After he clicks continue, new password is being saved in the database and the system shows a message which says “Operation successfully done”.

10. After the change, he finishes his job and clicks “log out.

**Scenario 6:**

**Scenario Name: Delete User Account**

**Participant Actor Instances: Neşet: Admin, Kenan: Manager**

One of the managers-Kenan- which was working at the company left his job. The company wants to remove his account from the system.

1. Neşet enters the application to log in.

2. He chooses the administration login to get in to the system.

3. He types his e-mail and password to login and then taps to continue.

4. He finds the “Manager Lists” option and taps it. In that list, he finds Kenan.

5.He taps Kenan’s profile and then taps delete.

6. A warning appears to the screen which says “Are you sure?” .

7. He taps “Continue”, Kenan’s profile is being deleted from the database and the system shows a message which says “Operation successfully done”.

8. After the deletion, he taps “log out” and exits.

**Scenario 7:**

**Scenario Name: Edit Booking**

**Participant Actor Instances: Neşet: Admin, Ayyüce: Customer**

A customer, whose name is Ayyüce, wants to change her ticket destination from İstanbul (departure 19.45) to Antalya (22.00).

1. Neşet enters the application to log in.

2. He chooses the administration login to get in to the system.

3. He types his e-mail and password to login and then taps to continue.

4. He chooses “Manage Booking” function.

5. He chooses “Edit Booking” function.

6. on searching bar, he searchs Ayyüce, the customer.

7. He taps Ayyüce’s profile, then taps travel information.

8. He chooses her İstanbul ticket and taps “Change”.

9. He chooses the 22.00 departure to Antalya from the departure list and taps confirm.

10. A warning appears to the screen which says “Are you sure?”.

11. He taps “Continue”, the destination changes as Antalya.

12. After the change, he logs out.

**Scenario 8:**

**Scenario Name: Delete Booking**

**Participant Actor Instances: Neşet: Admin, Ayten: Customer**

A customer, whose name is Ayten, wants to delete his travel to Amasya (17.00)

1. Neşet enters the application to log in.

2. He can login through opening page.

3. He types his e-mail and password to login and then taps to continue.

4. He chooses “Manage Booking” function.

5. He chooses “Delete Booking” function.

6. On searching bar, he searchs Ayten,the customer.

7. He taps Ayten’s profile, then taps travel information.

8. He chooses her Amasya ticket and taps “Delete”.

9. A warning appears to the screen which says “Are you sure?”.

10. He taps “Continue”, the departure is deleted.

11. After the deletion, he logs out.

**Scenario 9:**

**Scenario Name: Freeze Account**

**Participant Actor Instances: Kerem: Customer**

A customer, whose name is Kerem, wants to freeze his account.

1. Kerem enters the application to log in.

2. He chooses the customer login to get in to the system.

3. He types his e-mail and password to login and then taps to continue.

4. He taps the function “Manage Account”.

5. On the new page, he taps the function “Freeze Account.”

6. A warning message is seen to the Kerem which says “Are you sure?”

7. He taps “Continue”.

8. A message appears which says “Process is successful”.

9. System logs him out to the system.

**Scenario 10:**

**Scenario Name: Buy Ticket**

**Participant Actor Instances: Ekin: Customer**

A customer, whose name is Ekin, wants to buy a ticket to İzmir, 18.45 departure.

1. Ekin enters the application to log in.

2. She chooses the customer login to get in to the system.

3. She types her e-mail and password to login and then taps to continue.

4. She chooses the function “Buy ticket”.

5. She checks the list and chooses the İzmir travel at 18.45 on her planned date.

6. The system shows the information about the customer and the trip.

7. She checks the information, then taps to “Check out”.

8. The system shows the ticket cost to her and leads her to the payment part.

9. She writes her credit card info and taps the “Buy now” button.

10. The system gets the payment.

11. The system shows a message which says “Process successfully done.”

**Scenario 10 Extensions:**

10. The payment fails. System wants customer to enter her credit card information again.

**Scenario 11:**

**Scenario Name: Delete Trip**

**Participant Actor Instances: Deniz: Admin**

The company wants to remove the trip to Eskisehir at 18.00

1. Deniz enters the application to log in.

2.She chooses the administration login to get in to the system.

3. She types her e-mail and password to login and then taps to continue.

4. She chooses the function “Edit Bus Schedule”.

5. She chooses the function “Delete Trip”.

6. She chooses the targeted date ,18.00 and the targeted trip to Eskişehir.

7. She deletes the information of the trip and taps done button.

8. System shows the message “Are you sure?”. She taps to continue.

9. System completes deleting the trip.

**Scenario 12:**

**Scenario Name: Update Trip**

**Participant Actor Instances: Nur: Manager**

The company wants to update the trip to Ankara from 16.00 to 16.45

1. Nur enters the application to log in.

2. She chooses the administration login to get in to the system.

3. She types her e-mail and password to login and then taps to continue.

4. She chooses the function “Edit Bus Schedule”.

5. She chooses the function “Update Trip”.

6. She chooses the targeted date, 16.00 and the targeted trip to Ankara.

7. She edits the information of the departure time as 18.00 and taps done button.

8. System shows the message “Are you sure?”. She taps to continue.

9. System completes editing the trip.

**Scnerio 13:**

**Scenario Name: Buy Ticket**

**Participant actor instances: Newton: Member of the App**

1. Newton decided to buy round ticket to Eskisehir. He opens the app log in his account.

2. After he logged in, he selects the dates of trip then touches the ‘’search’’ button.

3. System lists all possible trips with related to given date. He also can see the time and cost of the trip.

4. He first choose the departure trip then system lists the arrival tickets. Same as before he can see all the details of trip. When he chooses arrival ticket system calculates total cost and guide him to make payment.

5. After he touches ‘’buy’’ system directs him to payment step.

6. He enters his credit card information and agrees the contract. A message and E-mail sent automatically.

7. Know he can see his ticket through his account by using left menu.

8. Sent messages includes e-ticket information in order to get the trip.

**Scenario 13 Extensions:**

1.a. Newton enters wrong password or invalid mail address.

b. He realized that he forgot the password. To reset password he selects reset password which is located under login screen.

c. He fills his mail address and system sends new generated password to his mail address.

d. He enters given passwords then login.

e. When he login system automatically wants new password. He enters new password then click save.

**Scnerio 14:**

**Scenario Name: Add Trip**

**Participant actor instances: Walter: Manager of the system**

1.Walter, wants to add additional trips to Antalya due to popular demand on Ramadan Feast. He login through app and selects add trip button.

2.He fills the date and time to add a new trip. System checks the time and want some more details as capacity, bus model. He enter related info and then touches red ‘’add trip’’ button.

3. When addition is complete system give feedback” trip number XXXX successfully added’’.

**Scenario 14 Extensions:**

1.a. He entered his e-mail or password incorrect.

b. These flow of events and conditions are all same for the actors of ‘customer’s.

**Scnerio 15:**

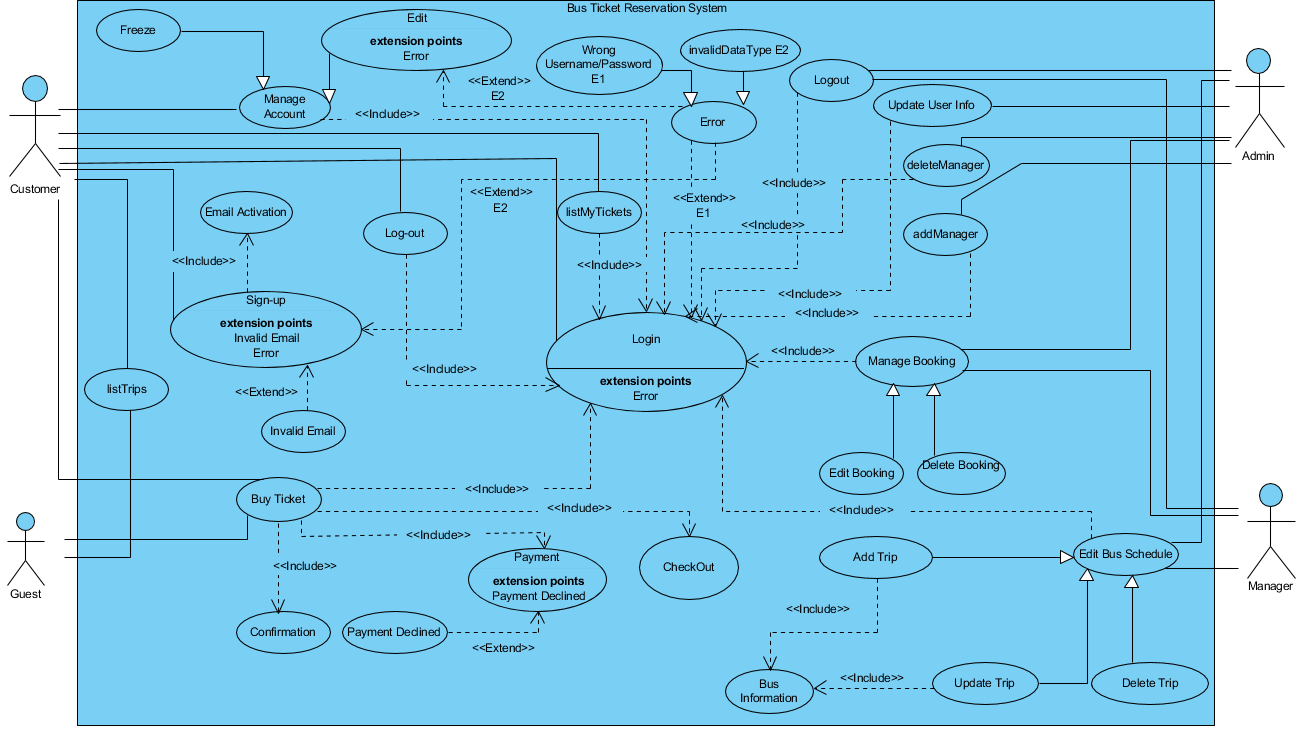
**Scenario Name: Delete User**

**Participant actor instances: Jesse: Admin**

1.One of manager of the company quit. Jesse wants to delete his manager account. He log in to app and click list managers button.

2.All managers of the company are listed in name order. He fined the manager and click the delete account button.

3. System warns him ‘’Do you want to delete a User?’’, admin clicks red yes button then system deletes the manager successfully.



|  |  |
| --- | --- |
| **Use Case Name** | Sign-up |
| **Participating Actor** | Consumer |
| **Entry Condition** | Consumer opens application |
| **Flow of Events** | 1. Consumer touch the ‘Sign Up’ button.  2. The application go to sign up page. Give the signup form to user.  3. Consumer fill the blanks about information ‘Sign Up’.  4. Consumer touch the ‘Sign up’ button  5. The application checks the e-mail address is valid or not.  6. The application checks the other required blanks are filled.  7. Application check is true then save the information to database. |
| **Exit Condition** | Consumer fills correctly and touch sign up button. |

|  |  |
| --- | --- |
| **Use Case Name** | Log-in |
| **Participating Actor** | Consumer |
| **Entry Condition** | Consumer opens application |
| **Flow of Events** | 1. Consumer touch the ‘Log-in’ button.  2. The application go to log-in page. Give the login form to user.  3. The consumer enters his/her e-mail address into e-mail address text field on the screen, also enters his/her password into password text field on the screen.  4. Consumer touch the ‘log-in’ button and sends a request to the system.  5. The application checks the e-mail and password is included in database. If it is not true application gives eror41 if is true application logs in. |
| **Exit Condition** | Consumer fills correctly and touch log-in button. |

|  |  |
| --- | --- |
| **Use Case Name** | Edit Profile |
| **Participating Actor** | Consumer |
| **Entry Condition** | Consumer needs to login to the system. |
| **Flow of Events** | 1. Consumer touch the edit profile logins to the system.  2. The application go to edit profile page and give the consumer list of information which loaded from database.  3. Consumer fill the blanks about information which he/she want to change.  4. Consumer save the changes  5. New changes are saved in database.  6. System is updated with new changes.  7. System gives feedback about completing the changes. |
| **Exit Condition** | Consumer touch ‘Save’ button |

|  |  |
| --- | --- |
| **Use Case Name** | Log-Out |
| **Participating Actor** | Consumer |
| **Entry Condition** | Consumer needs to login to the system. |
| **Flow of Events** | 1.Consumer go to profile menu.  2.Consumer touch log-out button for exit from account on phone. |
| **Exit Condition** | Consumer touch log-out button. |

|  |  |
| --- | --- |
| **Use Case Name** | Update User Info |
| **Participating Actor** | Admin |
| **Entry Condition** | The admin needs to login to the system. |
| **Flow of Events** | 1.Admin logins to the system.  2. Admin enters the manager accounts list.  3.Admin chooses the manager’s profile to edit the information.  4.Admin makes the changes which have been planned. (Username, password changes)  5. New changes are saved in database.  6. System is updated with new changes.  7. System gives feedback about completing the changes. |
| **Exit Condition** | The admin updates the information on manager accounts. |

|  |  |
| --- | --- |
| **Use Case Name** | Delete User Account |
| **Participating Actor** | Admin |
| **Entry Condition** | The admin needs to log in to the system. |
| **Flow of Events** | 1. Admin logins to the system. 2. Admin enters the manager accounts list. 3. Admin selects the manager account. 4. Admin selects the option “Delete Account”. 5. The information about the selected manager is deleted from database. 6. System gives feedback about deleting the account. |
| **Exit Condition** | The admin deletes the selected manager account. |

|  |  |
| --- | --- |
| **Use Case Name** | Delete Booking |
| **Participating Actor** | Admin & Manager |
| **Entry Condition** | The admin or the manager need to log in to the system. |
| **Flow of Events** | 1.Admin(manager) logins to the system.  2. Admin(manager) chooses the function “Delete Booking”.  3. Admin(manager) chooses the person from customer list.  4. Admin(manager) sees the travel information of customer.  5. Admin(manager) chooses the targeted departure and click “Delete”.  6. System gives feedback about deleting the booking. |
| **Exit Condition** | The admin(manager) deletes the selected booking. |

|  |  |
| --- | --- |
| **Use Case Name** | Edit Booking |
| **Participating Actors** | Admin & Manager |
| **Entry Condition** | The admin or the manager need to log in to the system |
| **Flow of Events** | 1. Admin(manager) logins to the system.  2.Admin(manager) chooses the function “Edit Booking”  3.Admin(manager) chooses the person from customer list.  4. Admin(manager) sees the travel information of the customer.  5. Admin(manager) chooses the targeted departure and click “Change”.  6. Admin(manager) chooses the new departure from the departure list and clicks confirm.  7. System gives feedback about editing the booking. |
| **Exit Condition** | The admin(manager) edits the selected booking. |

|  |  |
| --- | --- |
| **Use Case Name** | Add trip |
| **Participating Actor(s)** | Admin & Manager |
| **Entry Condition** | Admin/Manager needs to login to system. |
| **Flow Of Events** | 1. Admin/Manager get access to add trip function through the system.  2. Admin/Manager choose the related date to add trip.  3. The system lists the trips that are in given date for information.  4. Admin/Manager fill the info blanks of trip as capacity,bus model then clicks add button.  5. System checks the info and confirm show as successfully added. |
| **Exit Condition** | System adds the trip successfully. |

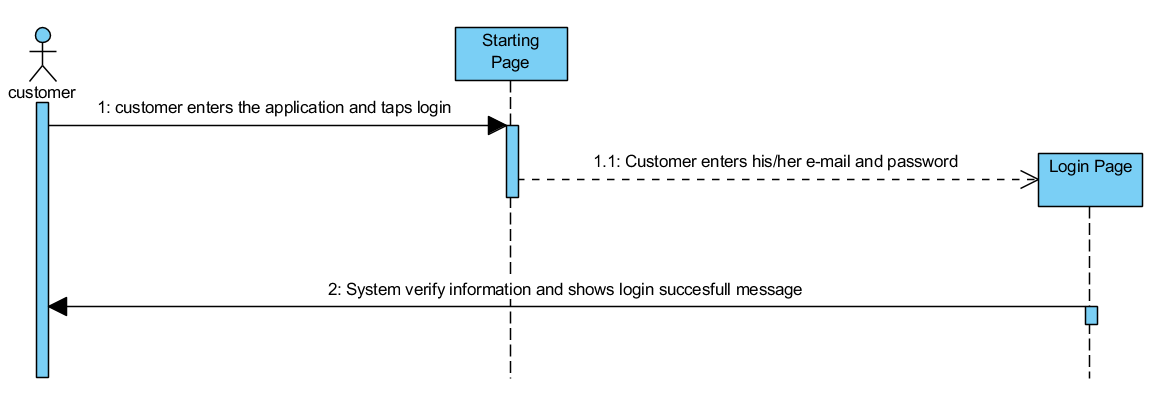
|  |  |
| --- | --- |
| **Use Case Name** | Update Trip |
| **Participating Actor(s)** | Admin & Manager |
| **Entry Condition** | Admin/Manager needs to login to the system. |
| **Flow Of Events** | 1. Admin/Manager get access to update trip function through the system.  2. Admin/Manager choose the related date to update the trip.  3. The system lists the trips that are in given date.  4.Admin/Manager selects the trip needs to be updated.  5.Admin/Manager edits the information of trip then clicks done button.  6.System checks and successfully updates the info of trip. |
| **Exit Condition** | System successfully updates the trip. |

|  |  |
| --- | --- |
| **Use Case Name** | Buy Ticket |
| **Participating Actor(s)** | Customer |
| **Entry Condition** | Customer needs to open the app and needs to sign in. |
| **Flow Of Events** | 1. Customer selects dates and lists the trips, the system list the trips from database.  2.Customer checks the list and selects a trip and click next.  3. Customer checks the trip info and add additional info if needed (must be another person).  4. The system list the all information about trip and person(s) that are in trip. Consumer clicks check-out button.  4.The system calculate total cost and guide consumer to make payment.  5.Consumer fills his/her credit card info and clicks buy now button.  6.The system gets the cost of the trip given credit card information.  7.The system send feedback if process is done. |
| **Exit Condition** | Successfull payment and check-out. |

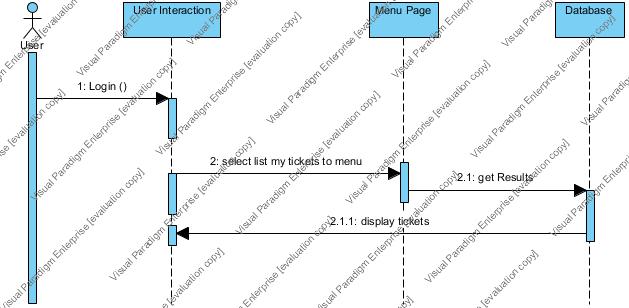
|  |  |
| --- | --- |
| **Use Case Name** | Delete Trip |
| **Participating Actor(s)** | Admin/ Manager |
| **Entry Condition** | Admin/Manager needs to login to system. |
| **Flow Of Events** | 1. Admin/Manager get access to delete trip function through the system.  2. Admin/Manager choose the related date.  3. The system lists the trips that are in given date.  4.Admin/Manager selects the trip needs to be deleted.  5.Admin/Manager deletes the information of trip then clicks done button.  6. System shows a message says “Are you sure?”  7. Admin/Manager taps “continue.”  6.System checks and successfully deletes the info of trip. |
| **Exit Condition** | Admin/Manager deletes the selected trip. |

**SEQUENCE DIAGRAMS:**

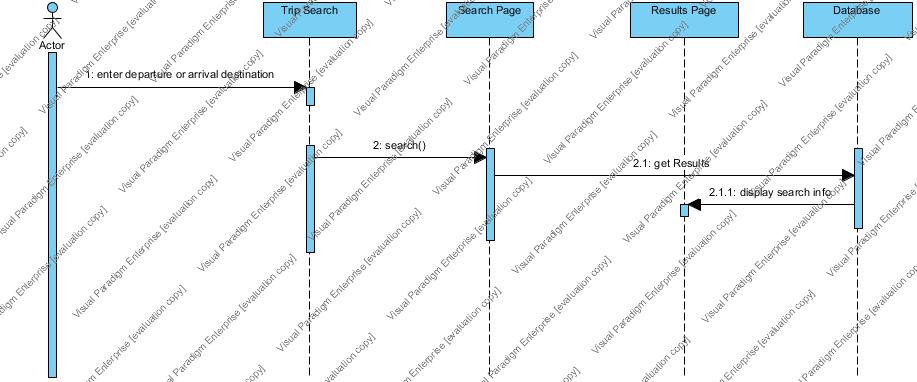
**Login**



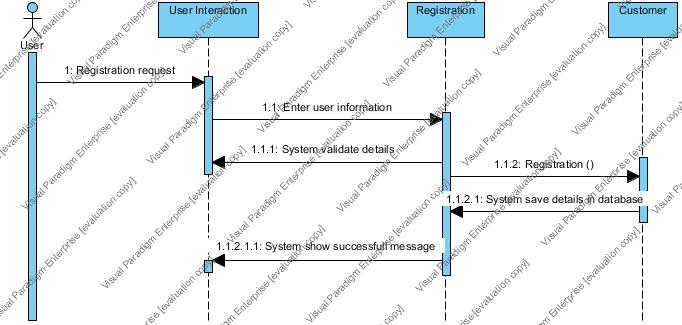
**List my tickets**



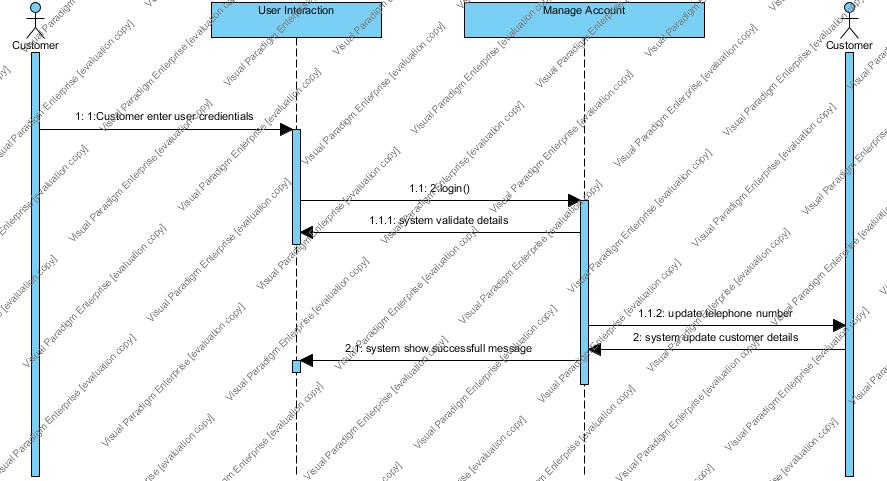
**Search Trip**



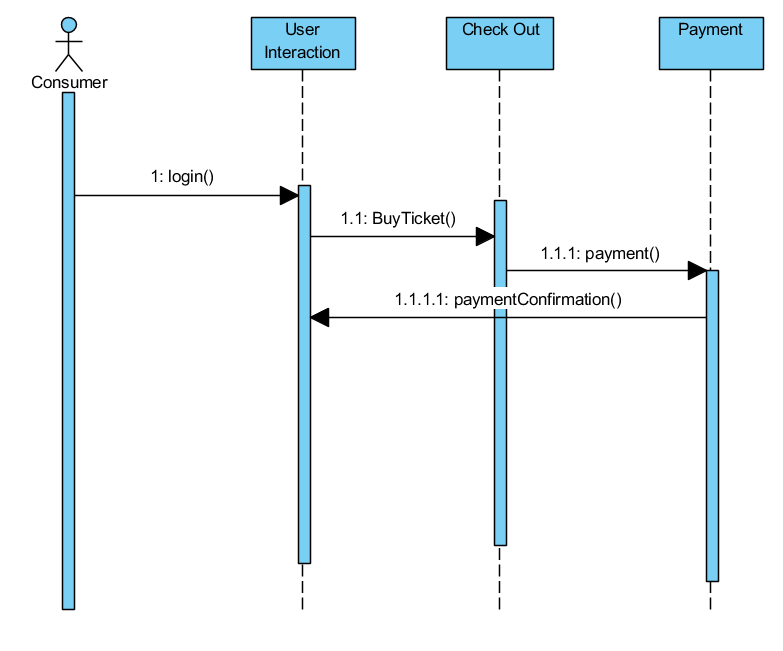
**Registration**

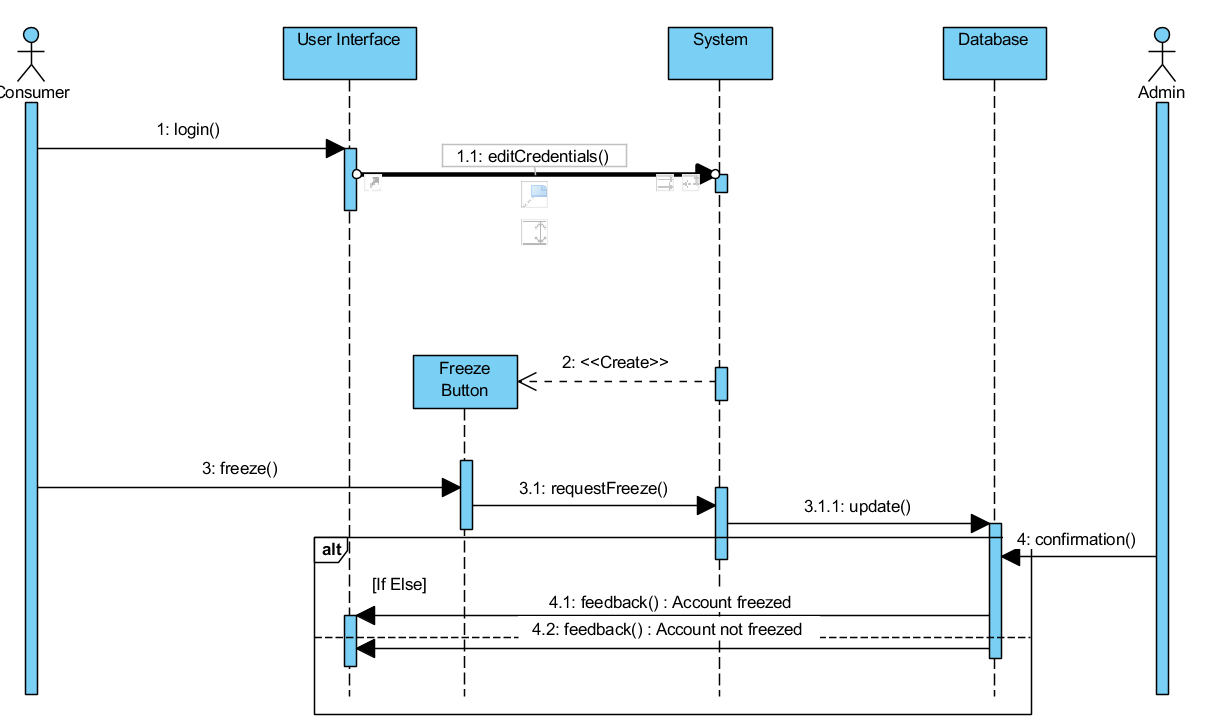


**Manage Account**

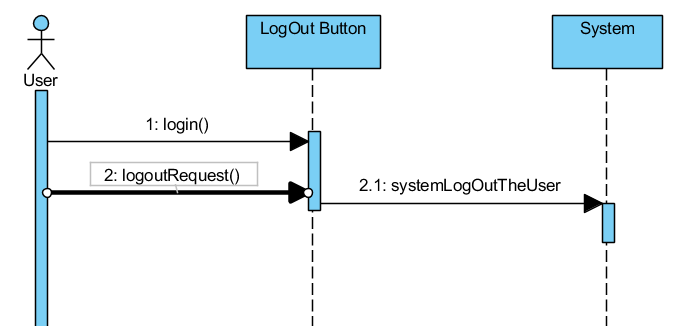


**Payment**

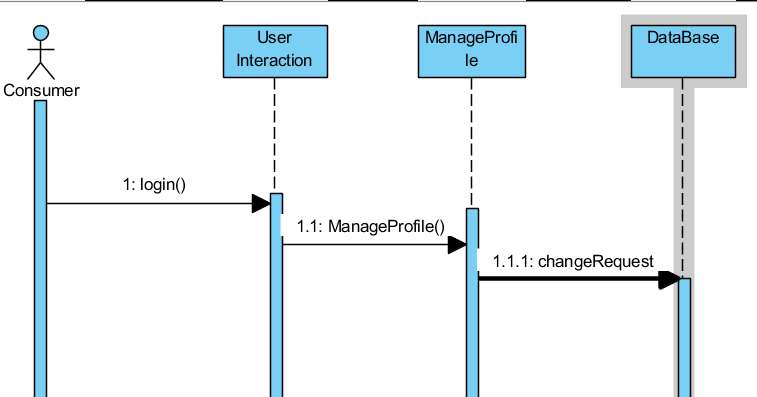


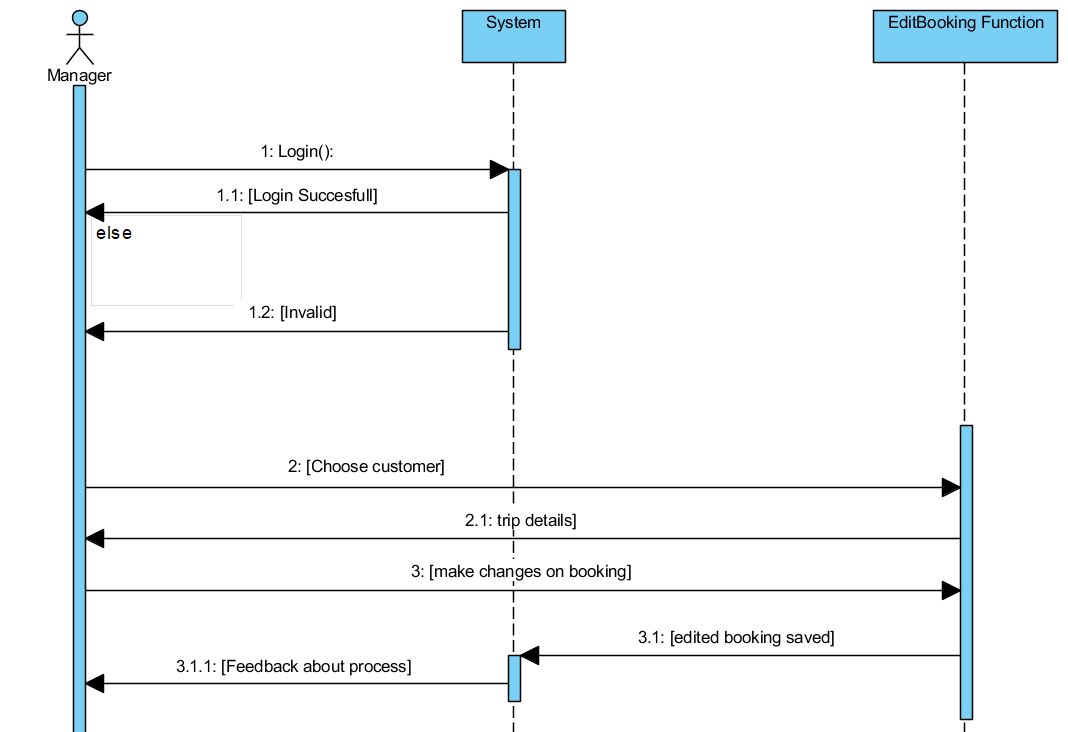
**Freeze**

**LogOut**

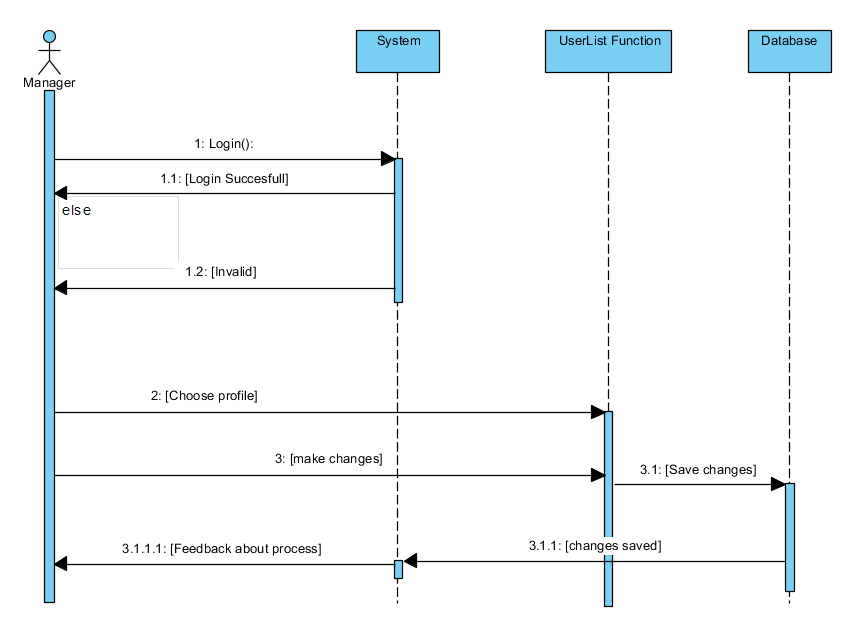


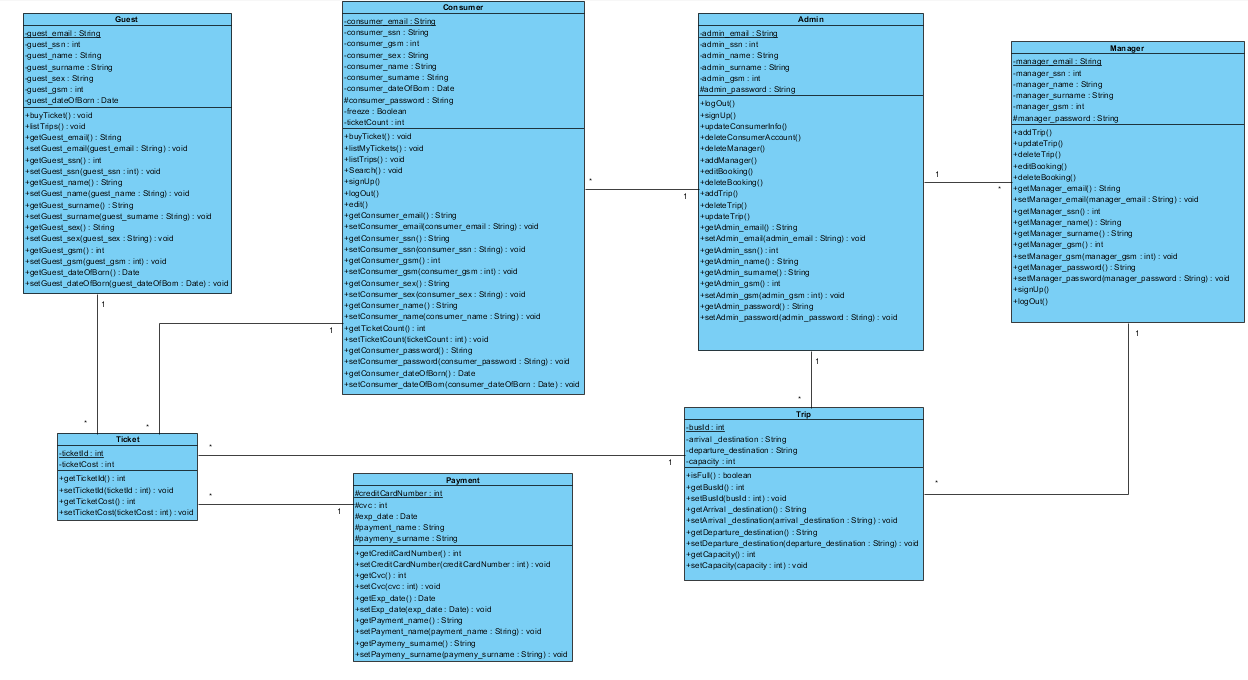
**ManageProfile**



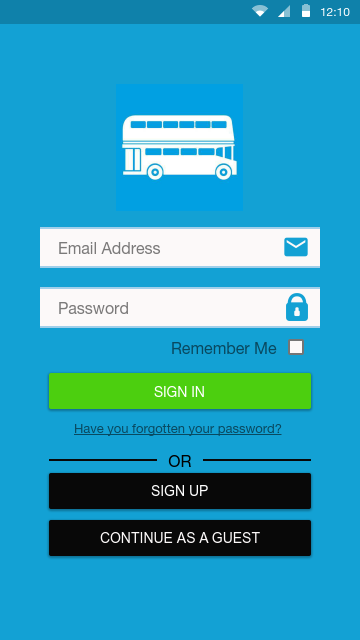
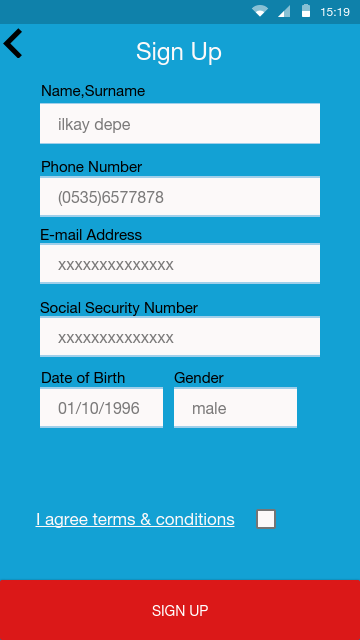
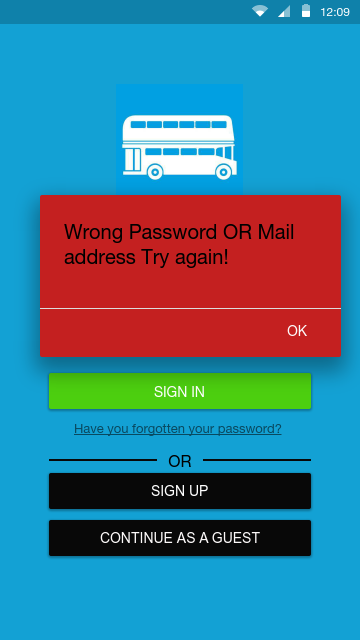
**Delete Booking** 

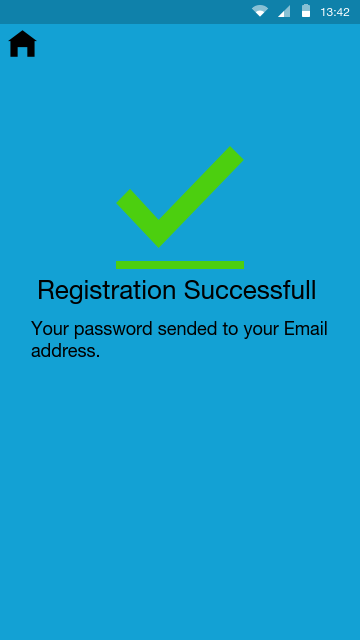
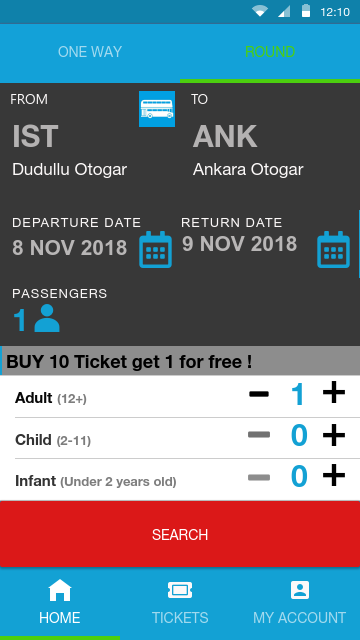
**Update Consumer**

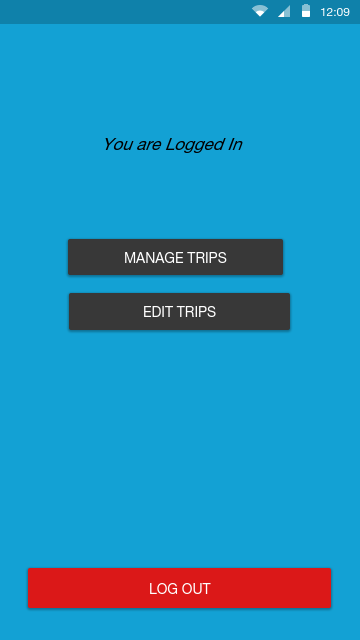
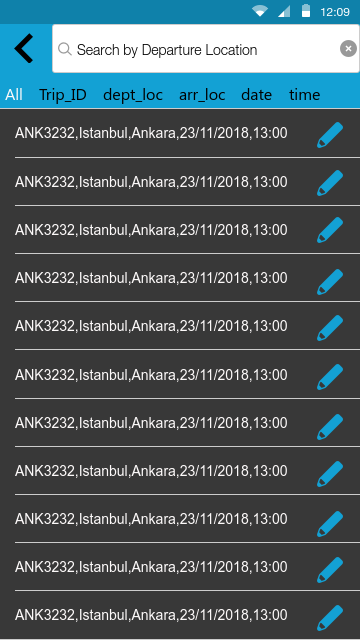
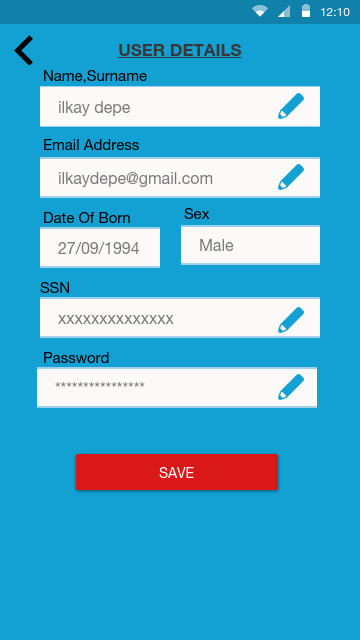
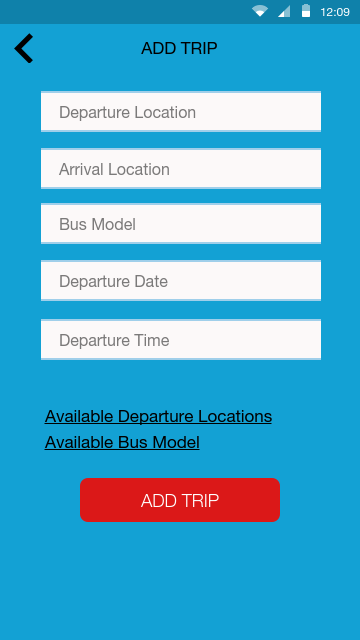
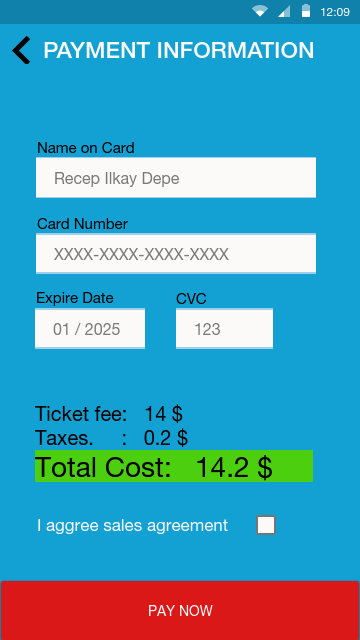
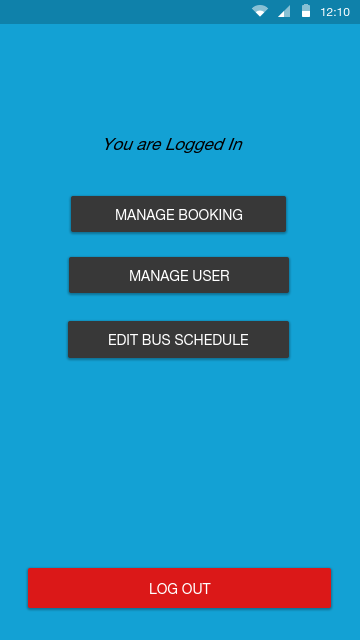
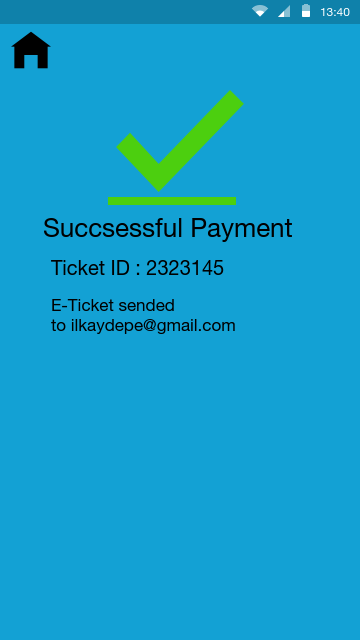
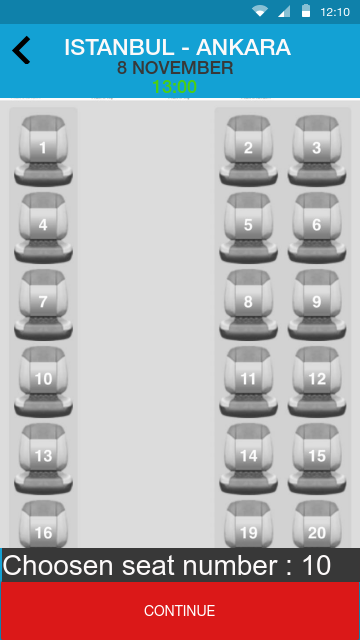
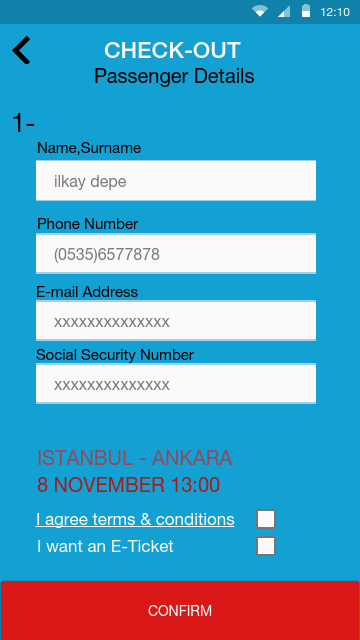
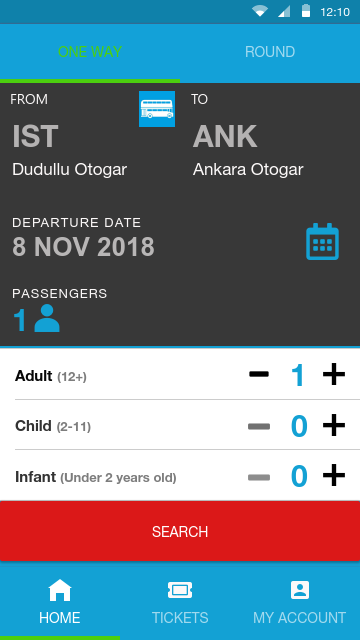




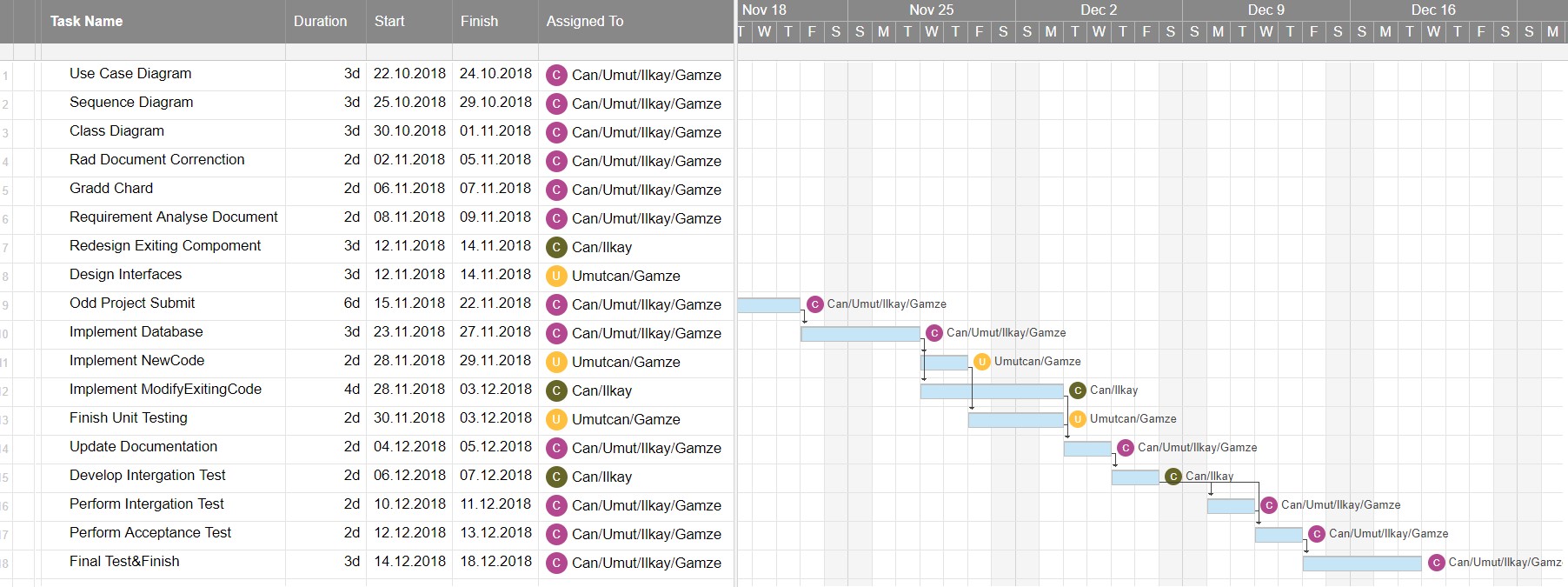
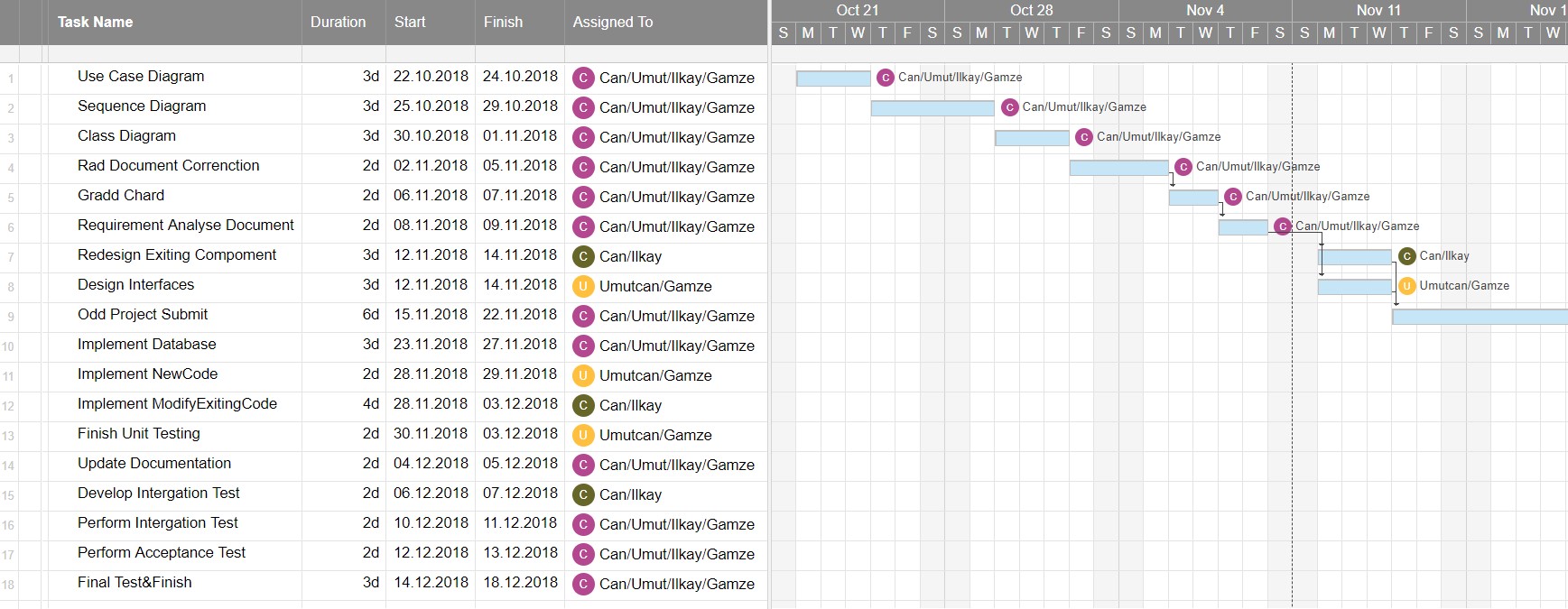
**Screen Mock-ups:**







## Project Schedule



# Glossary

*Consumer table*: consumer contains email, ssn, gsm, sex, name, surname, dateOfBorn, consumer password, ticket count and freeze which shows account is active or disactive.

*Admin table*: admin table contains email, ssn, name, surname, gsm and password.

*Guest table*: guest contains email, ssn, name, surname, sex, gsm and dateOfBorn.

*Manager table*: manager contains email, ssn, name, surname, gsm and password.

*Trip table*: Trip table contains busId, arrivalDestination, departureDestination and capacity which coming from bus id.

*Payment table*: Payment is protected it can't reachable from other clas. Payment contains creditCardNumber, cvc, expDate, paymentName, paymentSurname which same as credit card information.

*Payment rejected*: consumer enters his credit card credentials wrong or card limit out of balance.

*Sign Up Form:* In order to complete registeration Email addres,name,surname,ssn,phone number,sex and date of born needed.

# References

*1. Bruegge B. & Dutoit A.H.. (2010). Object-Oriented Software Engineering Using UML, Patterns, and Java, Prentice Hall, 3rd ed.*

*2. Lecture presentations of the course (the presentations were provided by the Instructor who is Emine Ekin).*

*3. About application design:* <https://www.kamilkoc.com.tr/> ; Android App, https://www.pamukkale.com.tr/ ; Android App