SYSTEM REQUIREMENT SPECIFICATION

GROUP SUCH

CARPOOL SYSTEM

TABLE OF CONTENT

TABLE OF CONTENT

TABLE OF CONTENT

TABLE OF CONTENT

LIST OF FIGURES

SYSTEM REQUIREMENT SPECİFİCATION OF SUCH

1. Introduction

1.1. Problem Definition

Since inappropriate planning of the cities, there has been a big problem of traffic in most cities of Turkey. People waste very long times in traffic every day. In Addition, because of so many vehicles in traffic, there has been an increasing problem of air pollution.

Oil supplies are very limited all over the world and oil prices are extremely expensive in our country. Therefore, most of the people have to take buses and since number of the public transportation vehicle are not sufficient, they travel under uncomfortable conditions.

There are some attempts to solve these problems. The most effective one is blablacar.com which is widely used in Europe. However, they focus only on intercity transportations. Also, in Turkey, there is yolyola.com and varmigelen.com. Both of them allows only users in Istanbul. However, our project will be used for both intercity and urban transportations all over Turkey.

As a result, our system will be designed to solve these problems and deficiencies of other systems.

1.2. Purpose

Aim of this software specification requirements document is to provide a complete description of all of the features that are planned to implement to system and define the expectations from the Carpool project. It also describes how the system operates and how users interact with the application. Besides external systems and interfaces which the application depends, are specified in this SRS document

The potential audiences for this document are design and development team of the Carpool Project in order to specify software designs. Test team utilizes this software specification requirements document to define test scenarios according

to the mentioned requirements. Besides project manager, quality manager and acquirer use this SRS document for reviewing purposes.

1.3. Scope

The Carpool Project is an MVC (Model-View-Controller) web based application which includes user interaction. Our project is going to be a web portal. It is going to provide communication environment for users (drivers and hitchhikers). Every user has their own profiles and they can have access with given password to the system.

The drivers can draw their routes from map in our web site. And hitchhikers can communicate with the driver via the messaging system and pick their path. After mutual agreement with each other, they record the transportation information to the system. At the end, users can assess each other via feedback system.

The system will bring many advantages. For instance, the drivers and hitchhikers spend less money on traffic. Moreover, traffic jam and air pollution will be decreased. And everyone benefits from these advantages.

In high level details, system will use Google Map API for retrieve location information, MySQL DBMS to store and manipulate the data, PHP for server side management, and GUI to interact with users.

1.4. Definition, Acronyms and Abbreviations

The definitions of the terms, which are used in this SRS document, are shown below:

Terms	**Definitions**
Acquirer	Project Consultant - Attila Özgit
Customer	Drivers and hitchhikers
GUI	Graphical User Interface
DBMS	Database Management System
IEEE	Institute of Electrical and Electronics Engineers

SDD	Software Design Description
SRS	System Requirements Specification
API	Application Programming Interface
PHP	Hypertext Preprocessor
Hitchhiker	User accompanies the driver during the transportation
Driver	User who owns the car
Route	Transportation path
SMS	Short Message Service
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart

1.5. References

[1] IEEE STD 1233-1998, IEEE Guide for Developing System Requirements Specifications

[2] IEEE STD 830-1998, IEEE Recommended Practice for Software Requirements Specifications

1.6. Overview

The rest of the document contains overall description of the system which includes interface properties, product functions and dependencies. In addition, it contains system specific requirements which composed of functional and non-functional requirements. Moreover, there will be data and description models of the system and these models are specified with diagrams such as use cases. And finally at the end of the document, there is conclusion part which explains the overall description about the system. SRS is organized according to the table of content.

2. Overall Description

This section gives background information about specific requirements of the web based carpool environment to be developed in brief. Although we will not describe every requirement in detail, this section will describe the factors that affect the final product.

2.1. Product Perspective

Carpool project is independent and self-contained. The constraints which describe how the software operates are listed below:

2.1.1. System Interfaces

The system has Google Map API as a subsystem. Google Map subsystem has their own web based interface which is a map consists of roads and locations in a desired area and user can easily intersect with this system.

2.1.2. User Interfaces

This software product is developed for drivers and hitchhikers. Product will be deployed to web site and all users of the system will access the system through the web interface which includes multiple pages according to the system functionality for example for login functionality there will be login page. To access the system, every user has unique user name and password. In addition, there will be a database who stores and manipulate all the data about the users. Website will only be the interface for all the user data which stored by database and the execution of provided functionalities.

After the sign up, user information will be transferred to database. In the sign up process, there will be e-mail verification to verify user information. After that point, users can register through the web interface. After log in, user will be able to log out whenever he or she wants.

2.1.3. Hardware Interfaces

Because carpool system is web based, it is compatible with all the browsers and can be run on any operating system and processor.

2.1.4. Software Interfaces

Database management system is required software product for Carpool system because all data about system for example user and route information must be stored in database for later use and system functionality.

MySQL database management system is used for that purpose and it has nice open source user interface which displays table and rows in well formatted form for developers to create and manage the whole database.

Another server that will be used is Google Map Server to provide geographical service and to visualize transportations.

In terms of user interface, HTML and Bootstrap library will be used to illustrate the system attractively.

These client and server sides attraction will be handled with Http Requests by JavaScript and PHP Languages.

2.1.5. Communication Interfaces

The system shall send automatic verification e-mail to the user who wants to register to the system. Moreover, in communication between driver and hitchhiker, users shall send and receive an e-mail through the e-mail interface.

For communication between users, system shall support SMS functionality and users can be able to send and receive SMS through the remote mobile devices.

2.1.6. Memory

64-bit, 4 cores processor. 8 GB RAM. 80 GB for system drive.

2.1.7. Operations

This part is explained in the user interfaces section.

2.1.8. Site Adaptation Requirements

Carpool system is able to perform every platform for example any browser that can be run on any operating system so it does not need any adaptation to a particular platform.

2.2. Product Functions

All use cases are explained below:

- **Sign Up:** Users need to sign up to use the web site. The users should have a username and password. After filling their name, surname, e-mail, age, job, phone and gender information, they register the system.
- **Sign In:** If a user is signed up, s/he can sign in the system by filling username and password boxes.
- **Sign Out:** A user may need to sign out the system. S/he can do it by clicking the sign out button which is placed in every page.

- **Add Transportation Route:** Users may add transportations by specifying a route, time/time period and number of empty seats. The user can select the route by two different way. The first way is entering start and end locations. Thus, the route is drawn on the map. The other way is selecting start and end locations on the map. Also, he/she can select at most 8 waypoints.
- **Delete Transportation Route:** A user may delete his/her transportation route. After deleting route, other passengers in that transportation will be informed by the system.
- **Request Transportation Route:** A user may use a transportation by sending transportation request to the driver of the transportation.
- **Search Transportation Route:** A user can search for transportations that the user can see suitable routes to his/her route by specifying time and route.
- **Send Message:** The users can have communicate each other by sending message.
- **Reply To A Message:** After receiving a message, the user can read the message.
- **Block User:** When a user receive disturbing message, s/he can block the user who send that message.
- **Rate User:** After having a transportation, the users in the same transportation can rate each other on the web site. Thus, other users can see the user rates and they can decide which transportation is better.
- **Change Language:** The user can change the web site language by clicking small flags that is available in all pages. There are two suitable languages which are Turkish and English.

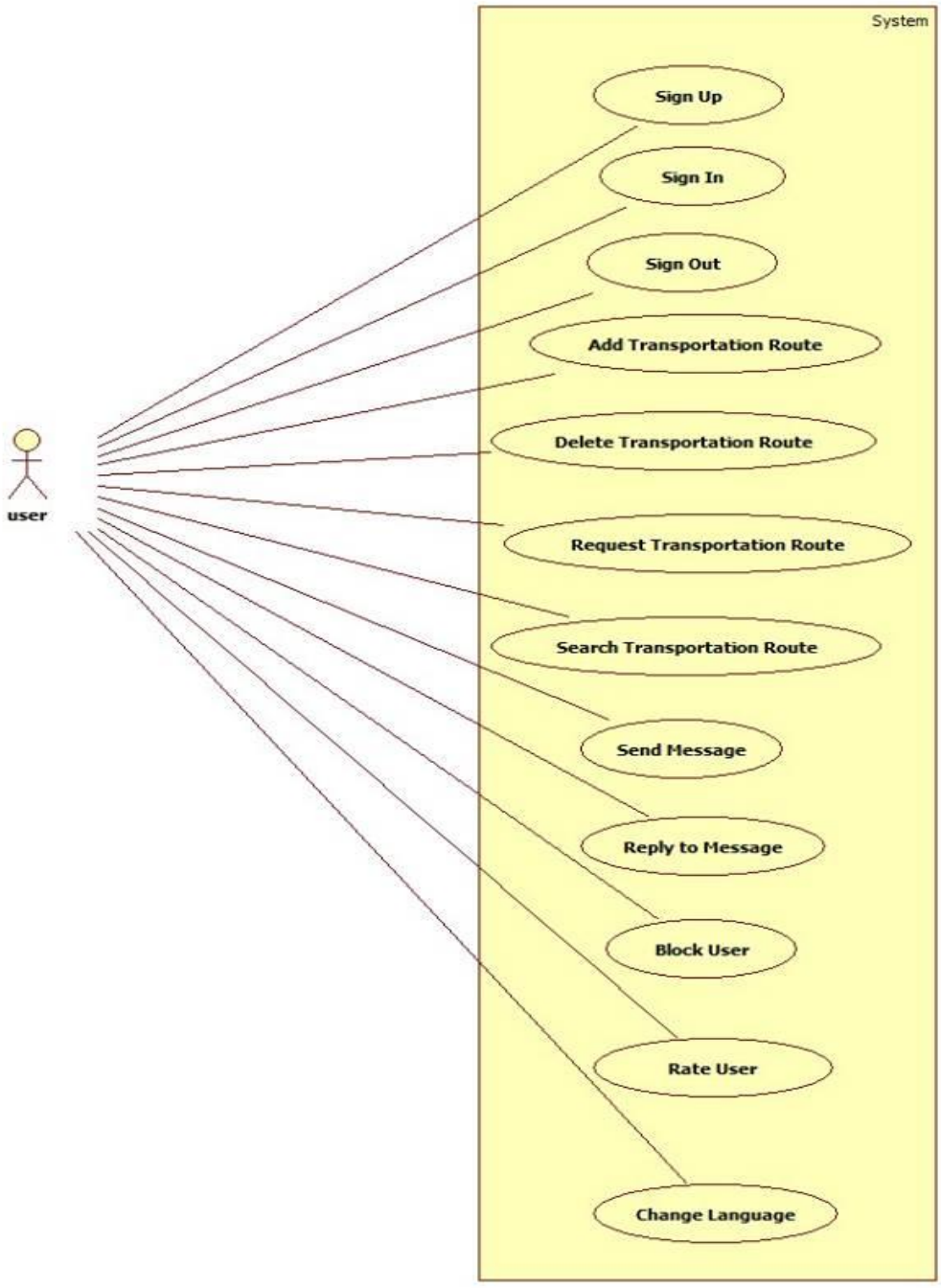


Figure 1. Use Case Activity Diagram

2.3. Constraints

Carpool system is required mutual trust for example user's security of life must be protected by the government's law system but there is no legal infrastructure about this driver and hitchhiker relation in our country. So, this is an important constraint for Carpool system. Another constraint is that the system requires remote server which enables the system functionality and data storage. Because of this situation, when the server crashes the system will not be able to its operations until the server become available to respond system requests. In addition to these, since the user information is stored in a database and this database can be hacked and user information will be no longer private to the user. To sum up, Carpool system has constraints in terms of regulatory, reliability, safety and security but these constraints can be manageable.

2.4. Assumption and Dependencies

User interface and some functionalities can change during the development process of project. And also new functionalities can be added which is able to change the dependent system requirements.

3. Specific Requirements

3.1. Interface Requirements

The system shall consist of user friendly web based user interfaces which are explained in 2.2. Product Functions section. Also, visualized version of the interfaces are explained in 5.1. Description for Software Behavior section.

3.2. Functional Requirements

3.2.1. Sign Up

Use Case ID	UC1
Actor(s)	User
Description	User Sign In (Register)
Preconditions	No precondition
Post conditions	User will be able to log in to the system
Precedence	Mandatory
Normal flow of event	1. The user opens the browser and types the address of the system. 2. User presses the sig up button. 3. User enters her or his user name, surname, password and e-mail information. 4. User checks his or her e-mail account to verify his or her user information.
Alternative Flow(s)	Flow 1: 1. If one of the required fields (user name, user surname, password, e-mail) in sign up page are not filled properly, the warning message will be shown by the system. Flow 2: 1. If all the required fields are properly filled, the user will be redirected to the main page of the system.

3.2.2. Sign In

Use Case ID	UC2
Actor(s)	User
Description	User Log In
Preconditions	The user shall be able to sign in to the system.
Post conditions	User will be able to use the system.
Precedence	Mandatory
Normal flow of event	1. The user opens the browser and types the address of the system. 2. User presses the log in button. 3. User enters her or his user name and password. 4. If the user forget his or her account information, he or she get account information via the “forgot your password?” panel under the log in page.
Alternative Flow(s)	Flow 1: 1. If the user enter wrong username or password information, the warning message for example ”Wrong username and password information” will be shown to the user. Flow 2: 2. If the user enters his or her user name and password information correctly, user will be redirected to the application relevant page of the system.

3.2.3. Sign Out

Use Case ID	UC3
Actor(s)	User
Description	User Log Out
Preconditions	The user shall be able to log in to the system.
Post conditions	User will be able to leave the system.
Precedence	Not mandatory
Normal flow of event	1. User presses the log out button. 2. User leaves the system. 3. The system's main page will be loaded.

3.2.4. Add Transportation Route

Use Case ID	UC4
Actor(s)	User
Description	User shall be able to add route from the map.
Preconditions	The user shall be able to sign in to the system.
Post conditions	User shall be retrieve transportation requests from the other users.
Precedence	Mandatory
Normal flow of event	1. User shall enter her or his profile page. 2. User shall press add new transportation button. 3. Add transportation page will be loaded. 4. User enters departure time, available seats and iteration

	of transportation like “one time” or “periodic”. 5. User draws a route on the map panel.
Alternative Flow(s)	Flow 1: 1. The user clicks the radio button of “one time”. 2. User enters date information in terms of day, month and year. Flow 2: 1. The user clicks the radio button of “periodic”. 2. User selects days from the week day check boxes.

3.2.5. Delete Transportation Route

Use Case ID	UC5
Actor(s)	User
Description	User shall be able to delete route.
Preconditions	The user shall add transportation route before.
Post conditions	User cannot see the route which is deleted by the user.
Precedence	No mandatory
Normal flow of event	1. User shall presses my transportations button. 2. My transportations page will be loaded. 3. User selects the route he or she wants to delete. 4. Delete button is clicked. 5. The user deletes the route.

3.2.6. Request Transportation Route

Use Case ID	UC6
Actor(s)	User
Description	User shall be able to request route.
Preconditions	The user shall search the route.
Post conditions	User will be able to contact the driver who owns the route.
Precedence	No mandatory
Normal flow of event	1. User presses the request route button. 2. The driver receives the notification e-mail and SMS by the system.

3.2.7. Search Transportation Route

Use Case ID	UC7
Actor(s)	User
Description	User shall be able to search route.
Preconditions	The user shall sign in to the system.
Post conditions	User will be able to select route from the available route list.
Precedence	No mandatory
Normal flow of event	1. User fills “from” input field. 2. User fills “to” input field. 3. User presses the search button.
Alternative Flow(s)	Flow 1: 1. User forgets to fill “from” or “to” input field. 2. The related warning message is shown to the user to fill the input fields properly. Flow 2:

	1. User fills the input fields properly. 2. The available routes will be listed.

3.2.8. Send Message

Use Case ID	UC8
Actor(s)	User
Description	User shall be able to send message through the system.
Preconditions	The user shall sign in to the system.
Post conditions	Users will be able to communicate with each other.
Precedence	No mandatory
Normal flow of event	1. User enters the profile page of the user who is intended to be communicated. 2. User presses the send message button. 3. The message page will be loaded. 4. User types the content of the message. 5. User presses the send button to send the message content. 6. The message content will be stored and viewed in the message panel.

3.2.9. Reply to Message

Use Case ID	UC9
Actor(s)	User

Description	User shall be able to reply to incoming message through the system.
Preconditions	The user shall receive the message from other user.
Post conditions	Users will be able to communicate with each other.
Precedence	No mandatory
Normal flow of event	1. User presses the message box button. 2. The message box page will be loaded. 3. User clicks the intended row from the message list. 4. The message page will be loaded. 5. User types the content of the message. 6. User presses the send button to send the message content. 7. The message content will be stored and viewed in the message panel.

3.2.10. Block User

Use Case ID	UC10
Actor(s)	User
Description	User shall be able to block the user through the system.
Preconditions	The user shall enter the intended to blocked user's profile page.
Post conditions	Users will be able to prevent communication with the user.
Precedence	No mandatory

Normal flow of event	1. User enters the profile page of the user who is intended to be blocked. 2. User presses the block user button. 3. The same page will be reloaded as marked as blocked.

3.2.11. Rate User

Use Case ID	UC11
Actor(s)	User
Description	User shall be able to rate the driver through the system.
Preconditions	The transportation route shall be completed with driver and hitchhiker.
Post conditions	The driver's rating will be updated.
Precedence	No mandatory
Normal flow of event	1. After the transportation, the hitchhiker login to the system. 2. Popup window is opened.
Alternative Flow(s)	Flow 1: 1. Hitchhiker clicks the star icon to rate driver's related attitude. Flow 2: 1. Hitchhiker clicks close icon. 2. The popup window will be closed.

3.2.12. Change Language

Use Case ID	UC12
Actor(s)	User
Description	User shall be able to change the language
Preconditions	No precondition.
Post conditions	User shall be able to see the system in selected language.
Precedence	No mandatory
Normal flow of event	1. User clicks the change language button. 2. The system converts its content in the selected language.
Alternative Flow(s)	Flow 1: 1. The user clicks the “Türkçe” from the navigation bar. 2. The page content will be converted to English language. Flow 2: 1. The user clicks the “English” from the navigation bar. 2. The page content will be converted to Turkish language.

3.3. Non-Functional Requirements

3.3.1. Performance Requirements

The user who has 8Mbits internet connection speed, shall be able to enter a page of the system in less than 1 second.

The system shall be able to respond more than one thousand users simultaneously.

The system shall be able to keep user information of more than one hundred thousand users.

3.3.2. Design Constraints

Passwords of the user shall be encrypted in DBMS for security purposes.

To prevent spam robots, the system has verification CAPTCHA module for security purposes.

When the system crashes it will return back at most one hour in maintainability purposes.

The system shall run on every browsers (Chrome, Safari, Mozilla Firefox etc.) and operating system (Windows, Linux, Mac Mavericks etc.).

4. Data Model and Description

4.1. Data Description

4.1.1. Data Objects

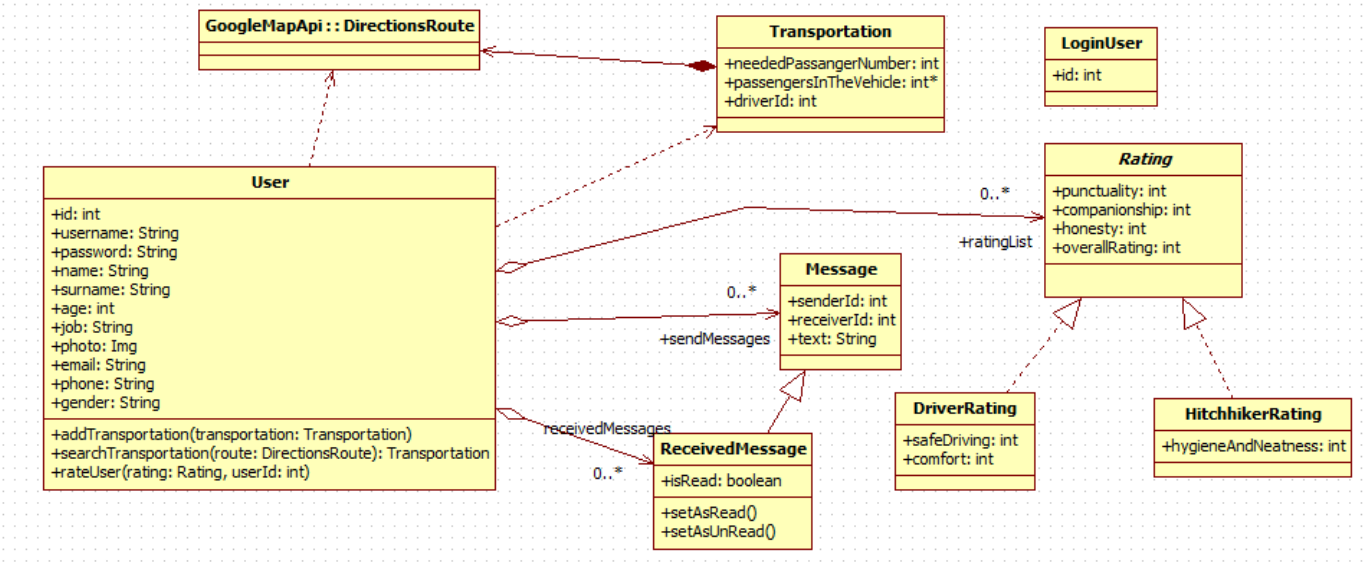


Figure 2. Class Diagram

User: It holds information of a registered user, such as name and surname. Also, it has the main functions a user can do such as adding and searching transportations and rating users.

DirectionsRoute: This class is from Google Map API. It has the information about a route, start and end points and all path information.

Transportation: A transportation has a route, a driver and passengers. Also, number of passengers that can be added are held.

LoginUser: This class is for knowing which user is logged in currently.

Message: This class is a simple message class. It has sender and receiver ids and a text. ReceivedMessage class inherits this class and add isRead attribute. Also, it adds two functions for marking the message as read and unread.

Rating: This class is a abstract base class for DriverRating and HitchhikerRating classes. It keeps common attributes

4.1.2. Data Dictionary

User Class:

Fields :

- id : This is an integer value that is unique for every user. This id is chosen by the system.
- username : This is a string value that is unique for every user. This is selected by the user.
- password : Every user has his/her own password that is selected by him/her.
- name : This is first name of the user.
- surname : This is family name of the user.
  age : This is age of the user
- job : This is job of the user
- photo : This is image of the user.
- email : This is e-mail of the user. Every user has to have unique e-mails.

- Phone : This is phone number of the user.
- gender : This is sex of the user.
- sendMessages : This is the messages that are sent by the user.
- receivedMessages : This is the messages that are sent to the user.
- ratingList : This is the list of rating that are done to the user.

Functions :

- addTransportation : This function is called when user hits add button in add transportation page. And add the specified transportation to transportation list.
- searchTransportation : This function is called when user hits search button in search transportation page. And finds all transportations that includes the specified route.
- rateUser : This function is called when a user wants to rate a user that he/she transported with.

Transportation Class :

Fields:

- route : This is the specific route for the transportation. This is an object of DirectionsRoute class of Google Map API.
- neededPassengerNumber : This is the number of people that might enter the transportation.
- driverId : That is the id of the user that created the transportation and is the driver of the transportation.

Message Class :

Fields:

- senderId : Id of the user who sent the message.
- receiverId : Id of the user to whom the message was sent.
- text : Body of the message.

ReceivedMessage Class :

Fields :

- isRead : This fields is about whether the message has been read by the user or not.

Functions :

- setAsRead : This functions marks the message as read.
- setAsUnRead : This function marks the message as unread.

Rating Class :

Fields :

- punctuality : This is the rating criteria about users attending transportation on time.
- companionship : This is the rating criteria about friendship of the user during the transportation.
- honesty : This is the rating criteria about users being honest.
- overallrating : This is the rating criteria about general attitudes of the user.

DriverRating Class :

Fields:

- safeDriving : This criteria is about carefulness of the driver during the transportation.
- comfort : This criteria is about comfort of the transportation vehicle

HitchhikerRating Class :

Fields:

- hygieneAndNeatness : This criteria is about users being clean and not griming the car.

5. Behavioral Model and Description

5.1. Description for Software Behavior

When the users enter the website, a home page is displayed. On the home page, users see the links which are “About Us”, “FAQ”, “How To”, “Contact”, “Login” and “Sign Up”. And also there is a video which is about how our application works. Clicking “About Us” link, users can see who we are and why we created this

application. When they click the “FAQ” link, they can see the answers of the frequently ask questions. On “How To” page, the users can be familiar with carpooling system. By clicking “contact us” link, they can contact with admin by filling the textbox.

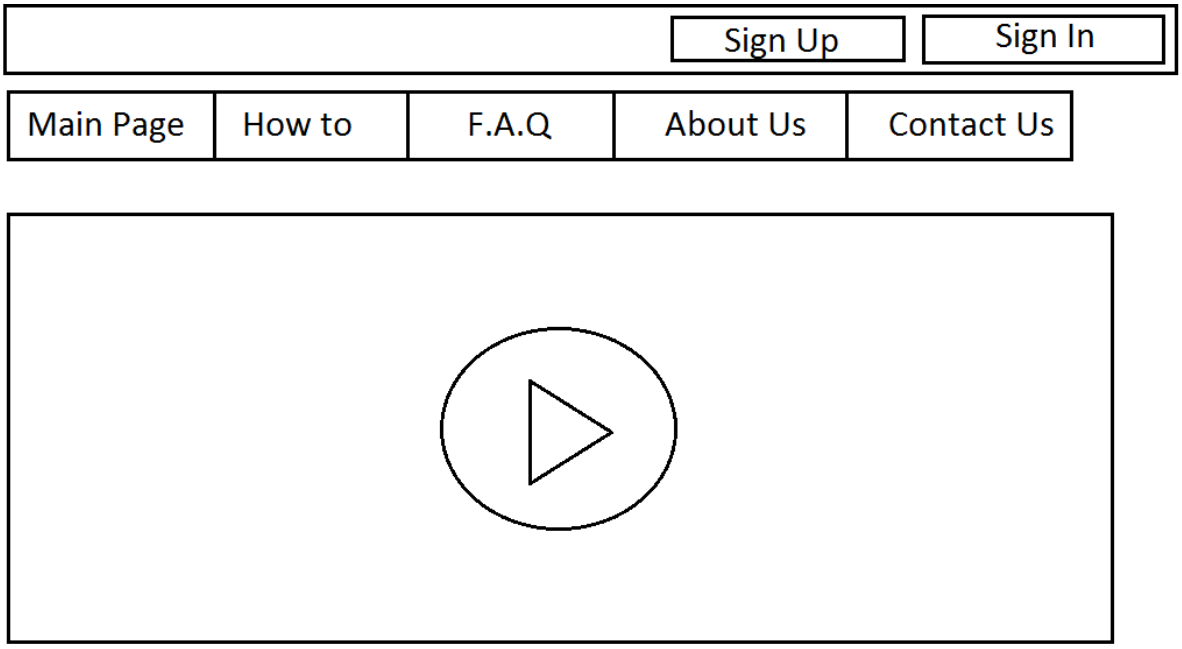


Figure 3. Visualization of Main Page

Users can display the login page via “Login” button. The users can enter the system by entering user name and password on this page.

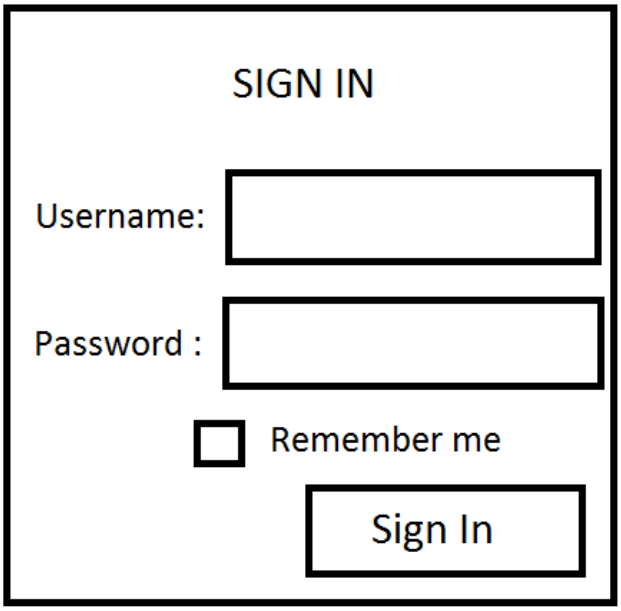


Figure 4. Visualization of Sign In Page

If they do not have an account, they can register the system by clicking "Sign Up" button. In this page, users have to fill the form which is about their name, surname, username, password, phone number, address information, gender, birth day, job, e-mail and photo.

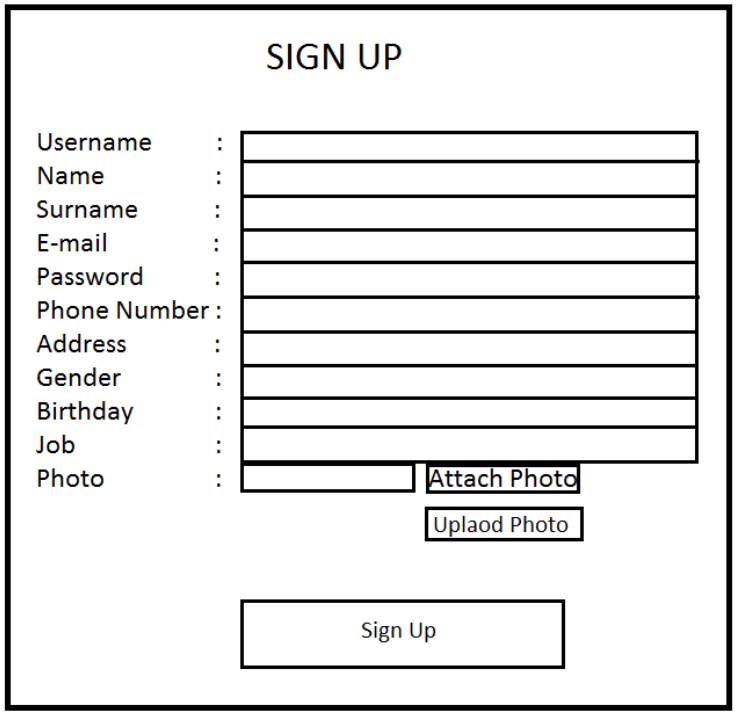


Figure 5. Visualization of Sign Up Page

When they enter the system, they see their profile information such as name, surname, birthday, user avatar and contact information. They can add or change their user information by clicking update button. And also they can see their rating info and previous actions. On the top pane, there some buttons such as search, my profile, massages and logout. These buttons are always visible in the top pane until user logs out.

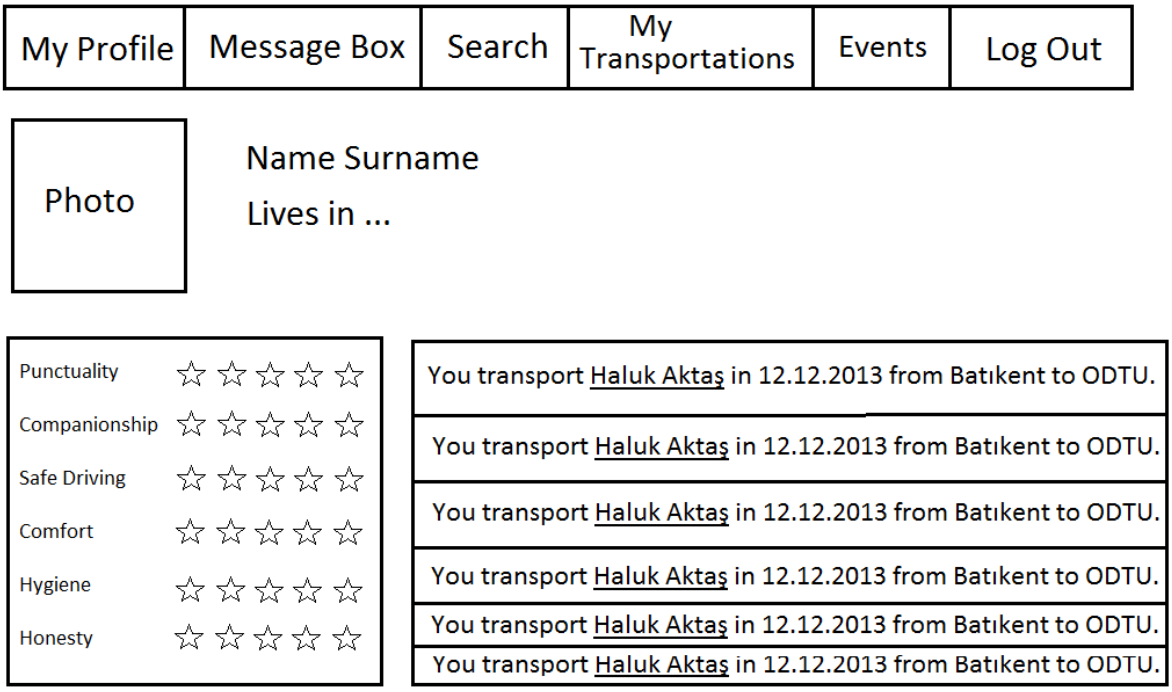


Figure 6. Visualization of Profile (Owner) Page

User can search transportations in the search page. Users can access this page via search button. On the search page, users can choose start and end points

that they want to use for their route. After clicking the search button, available transportations are listed. User can choose any one of them.

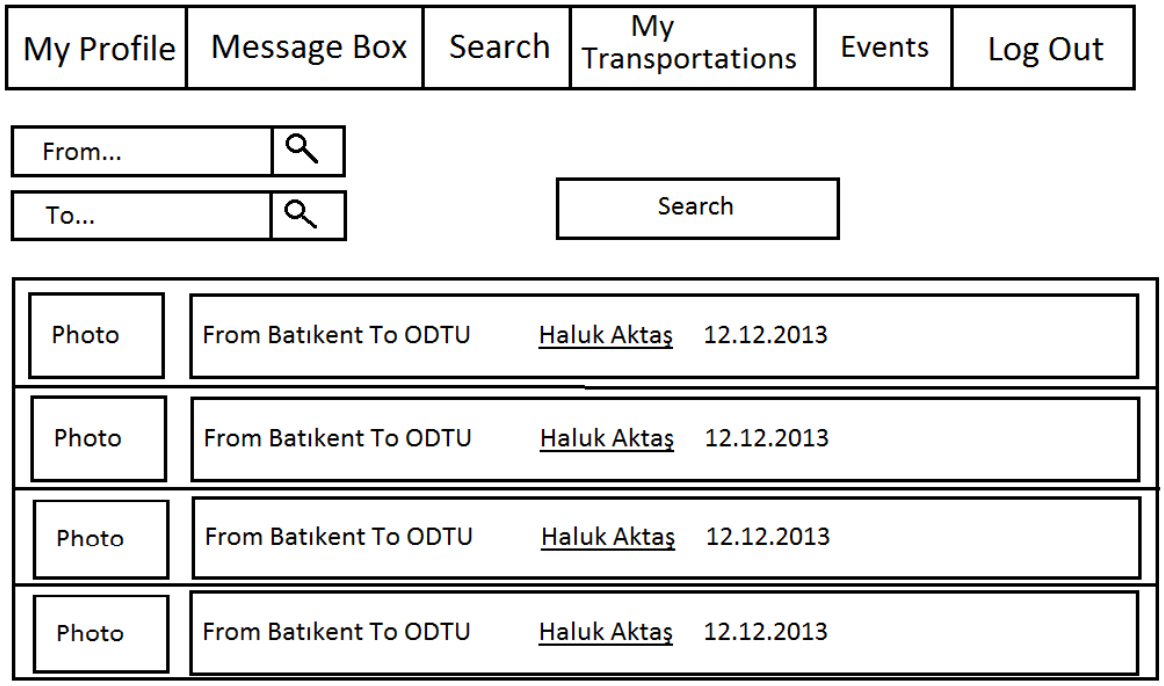


Figure 7. Visualization of Search Page

Then, transportation info page is displayed. On this page, route is detailed by Google Map. So, the user can send a request to driver via request button. In addition to this, user can see drivers profile information by clicking driver avatar.

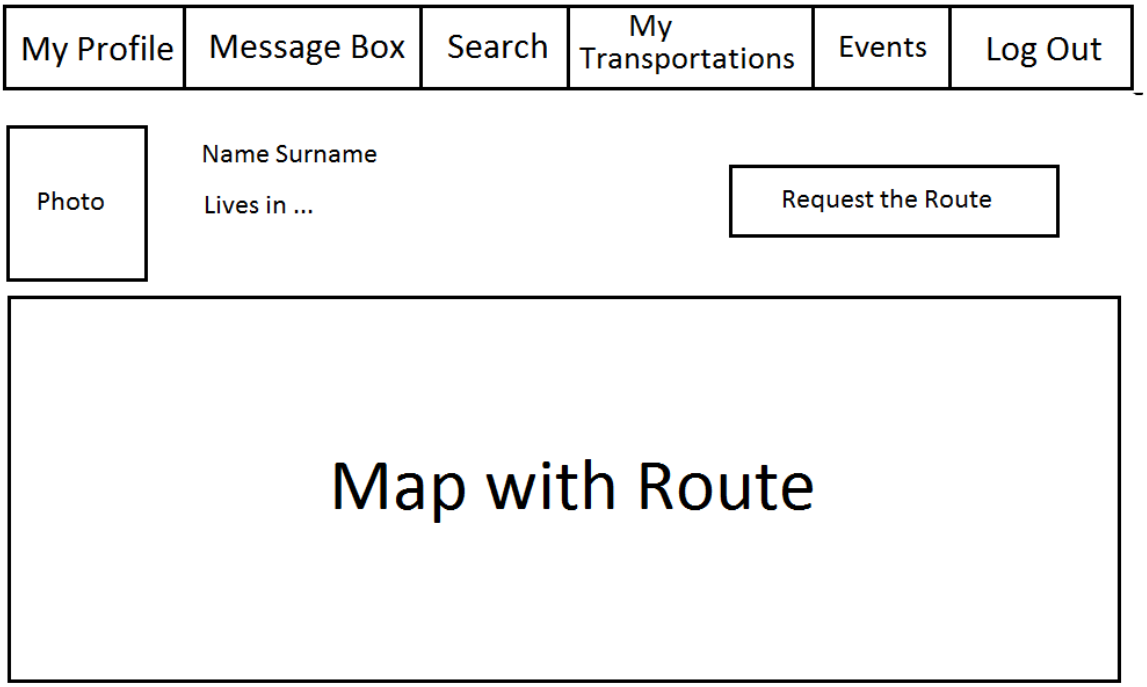


Figure 8. Visualization of Route Page

Moreover, users can see received messages via messages button. On messages page, users have actions on their messages as deleting message, remarking important message and selecting multiple message.

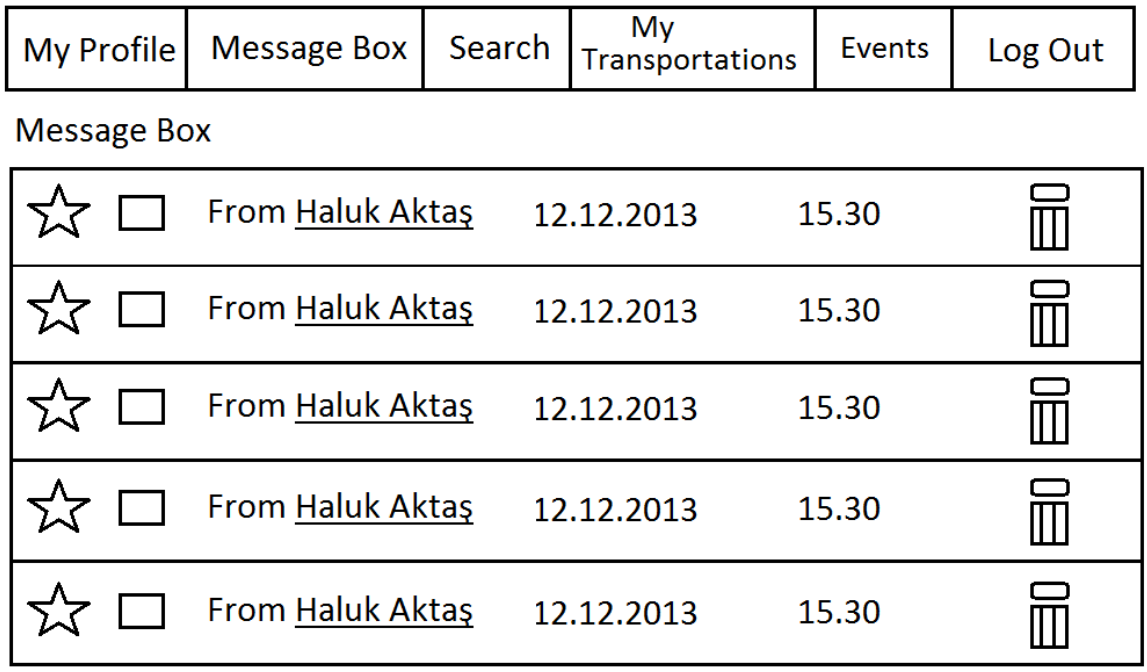


Figure 9. Visualization of Message Box Page

If users want to send message to a driver, they have to view the driver profile to click send message button.

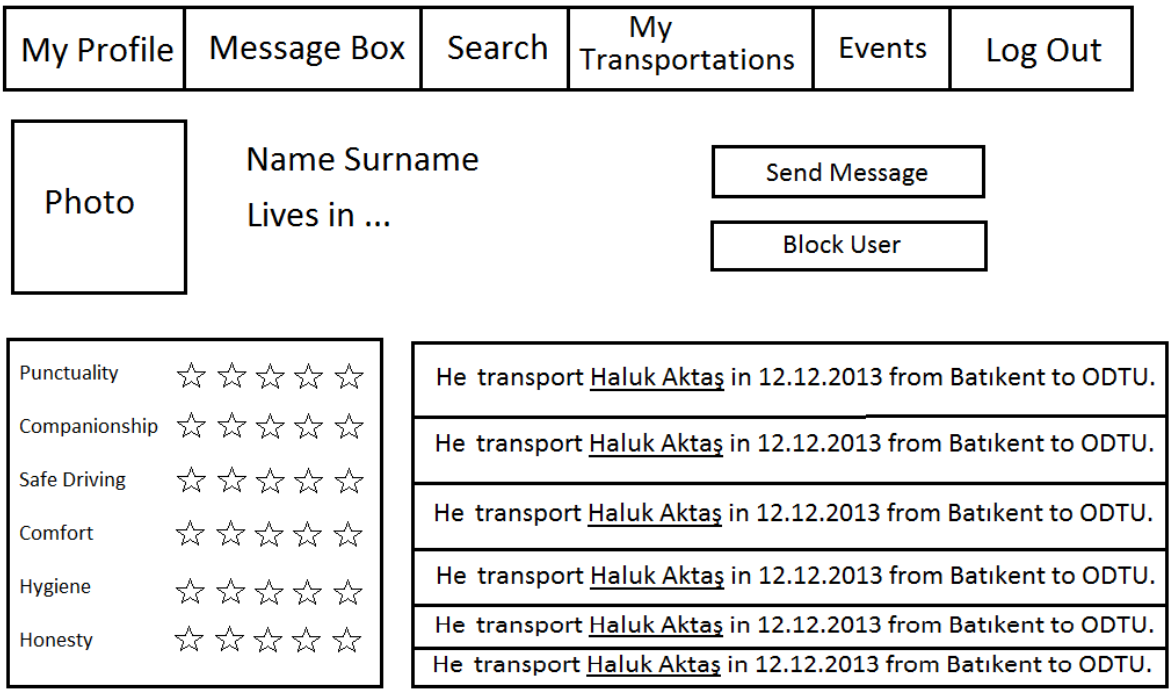


Figure 10. Visualization of Profile Page

When user want to send a message or reply a message, Message Page will display. They can also see the previous messages in this page.

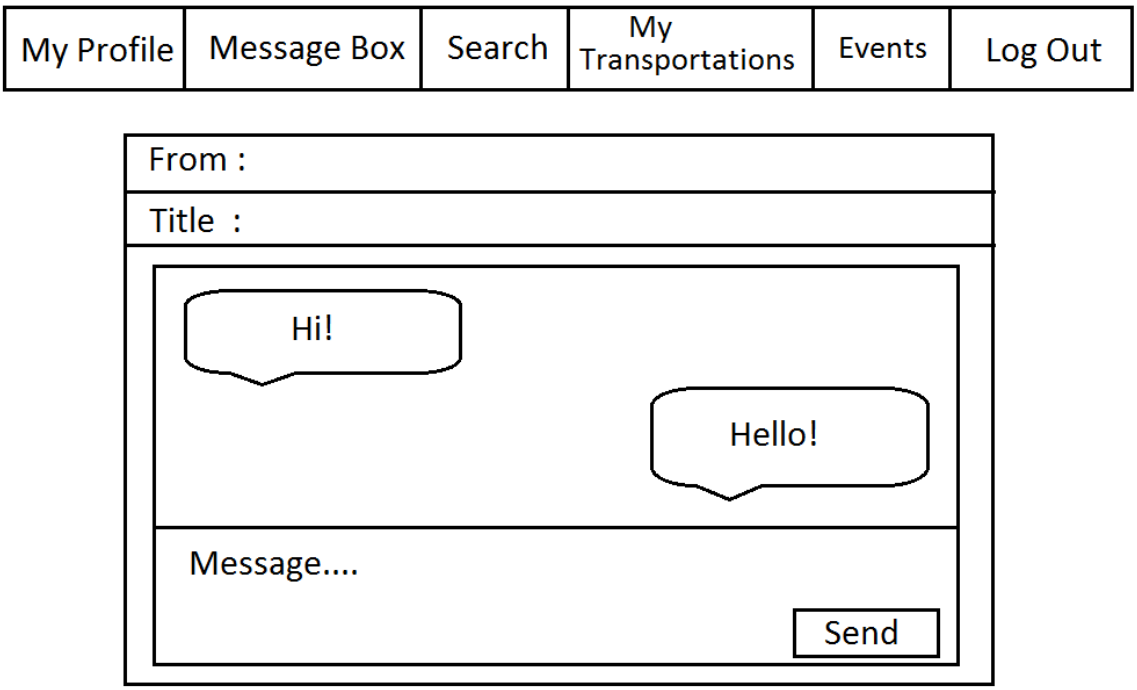


Figure 11. Visualization of Message Page

When user click My Transportations link, there will be displayed list of current their transportations as a driver and a hitchhiker.

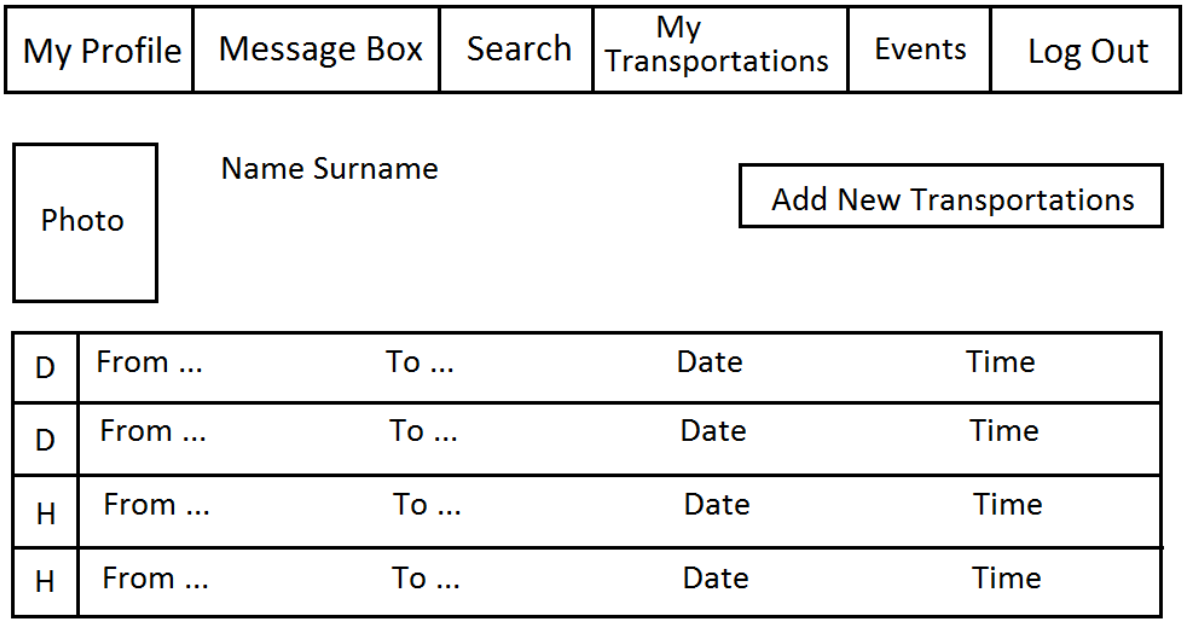


Figure 12. Visualization of My Transportation Page

When users click Add New Transportation link, there will be displayed Add Transportation Page.

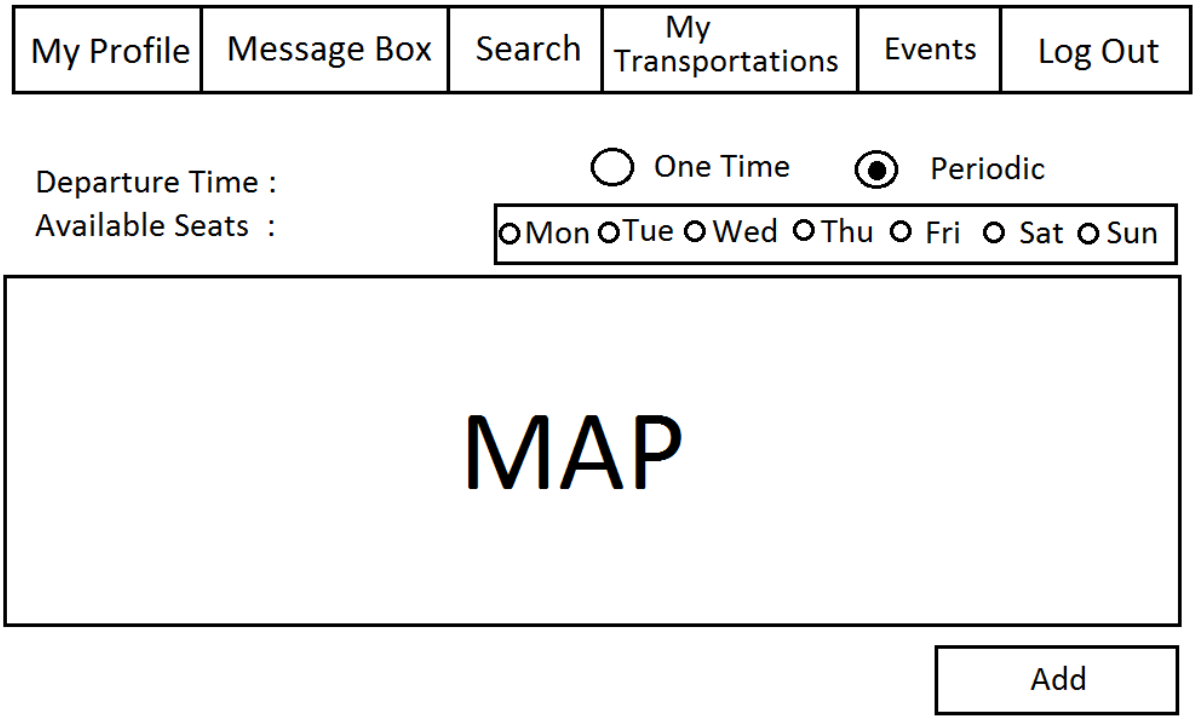


Figure 13. Visualization of Add New Transportation Page

When users click Events link, there will be displayed list of special events in that city.

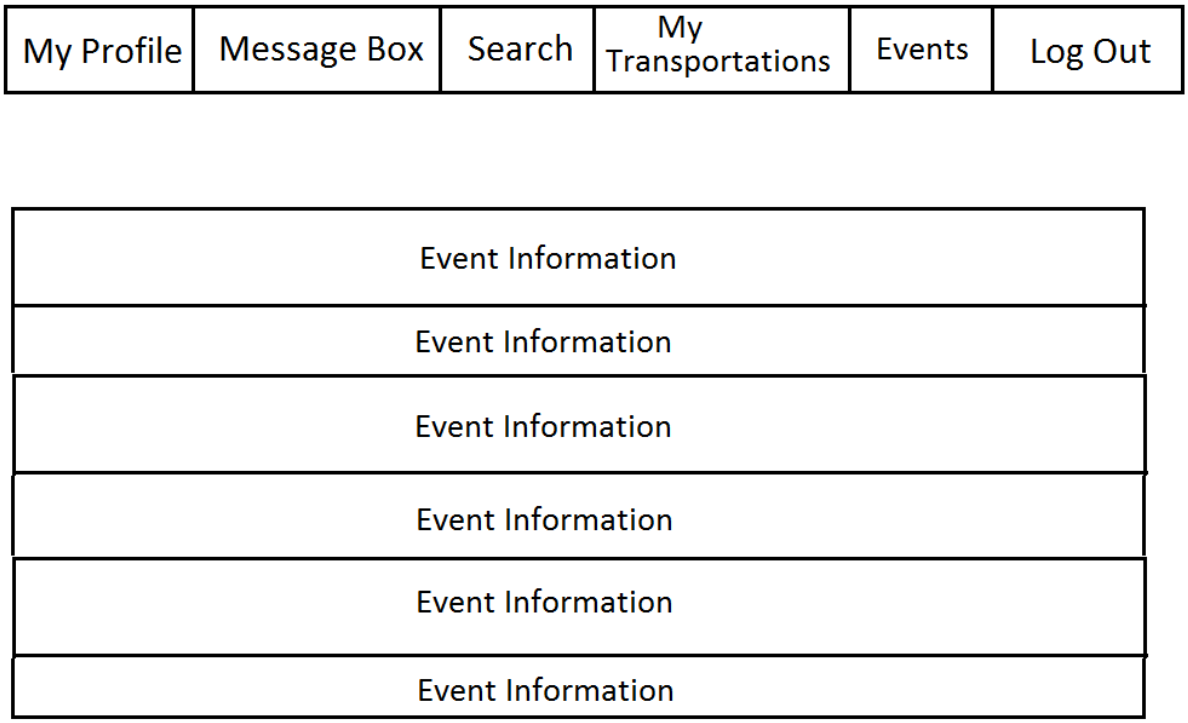
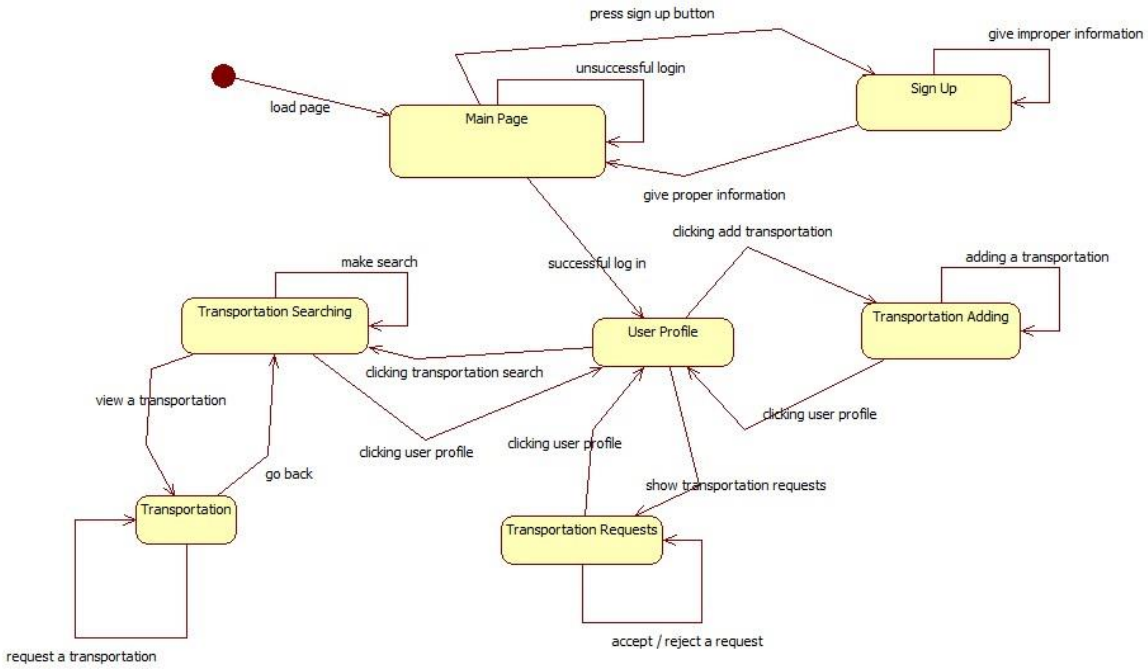


Figure 14. Visualization of Events Page

Finally, users can log out with logout button. By logout button, the home page is displayed.

5.2. State Transition Diagrams



When a person enters the website, main page is loaded. In this page, a registered user can log in and go to the user profile page. If he/she enters a wrong username-password match, he/she stays at the same page and can enter username-password again. If the person is unregistered, he/she can click sign up button and go to sign up page. There, he/she can give his information and go to the main page. If information is improper, he/she stays at the page and can try again. In the main page, users can log in and be directed to his/her profile page.

In user profile page, users have three main options. The first one is that users may click transportation add button and go to transportation adding page. There, they can add transportations or return to the user profile page. The second one is that users may click transportation search button and go to transportation searching page. There, they can search specific transportations. By clicking one, they can go to transportation page and see properties of the transportation. The last one is that users can see the request that have been done to their added transportations. They can accept or reject them depending on their will.

6. Planning

6.1. Team Structure

The structure of team is arranged as work-based. Everyone may have different task. However, the arrangement of team works is assigned to every person. The arrangement is listed below,

- Design of the system: Selahattin Muhammet Dülgeroğlu, Ferit Altay
- Documentation of the project: Mustafa Haluk Aktaş, Can Mehteroğlu
- Server Side of the system: Mustafa Haluk Aktaş, Ferit Altay
- Client Side of the system: Uğur Temiz, Can Mehteroğlu
- Testing of the system: Selahattin Muhammet Dülgeroğlu, Uğur Temiz

As a result of this arrangement, everyone has major fields.

6.2. Estimation

The estimation of basic schedule is listed below,

- System Requirement Specification should be finalized in 17.11.2013.
- System Design Document should be finalized in 01.12.2013.
- A simple prototype should be developed until 25.01.2013.
- System Level Test Design should be finalized until 15.03.2014.
- System Level Test Report should be finalized until 15.04.2014.
- After testing the whole system should be finalized until 01.05.2014.

6.3. Process Model

The project will be processed by Waterfall Model. Waterfall Model is plan driven process which is listed below consecutively.

- Cleared requirements,
- System and software design,
- Implementation and unit testing,
- Integration and system testing,
- Operation and Maintenance.

7. Conclusion

In this document, the functional and other requirements of the system are described. Furthermore, the needs of the user are stated through the document. However, all requirements are not defined and some of the requirements needs to be clarified in this document.

To sum up, this document is the primary document which upon all of the subsequent design, implementation, test and validation processes will be based.

8. Supporting Information

At the beginning of the document, table of contents and list of figures are included.