

SOFTWARE DESIGN DOCUMENT

GROUP SUCH

CARPOOL SYSTEM

TABLE OF CONTENT

1.	OVERVIEW	7
1.1.	SCOPE	7
1.2.	PURPOSE	7
1.3.	INTENDED AUDIENCE.....	7
2.	DEFINITIONS.....	8
3.	CONCEPTUAL MODEL FOR SOFTWARE DESIGN DESCRIPTION.....	9
3.1.	SOFTWARE DESIGN IN CONTEXT	9
3.2.	SOFTWARE DESIGN DESCRIPTIONS WITHIN THE LIFE CYCLE.....	9
3.2.1.	INFLUENCES ON SDD PREPARATION.....	9
3.2.2.	INFLUENCES ON SOFTWARE LIFE CYCLE PRODUCTS.....	9
3.2.3.	DESIGN VERIFICATION AND DESIGN ROLE IN VALIDATION	9
4.	DESIGN DESCRIPTION INFORMATION CONTENT	10
4.1.	INTRODUCTION	10
4.2.	SDD IDENTIFICATION	10
4.3.	DESIGN STAKEHOLDERS AND THEIR CONCERNS.....	10
4.4.	DESIGN VIEWS.....	10
4.5.	DESIGN VIEWPOINTS	10
4.6.	DESIGN ELEMENTS	11
4.6.1.	DESIGN ENTITIES.....	11
4.6.2.	DESIGN ATTRIBUTES.....	11
4.6.3.	DESIGN RELATIONSHIPS	12
4.6.4.	DESIGN CONSTRAINTS.....	12
4.7.	DESIGN RATIONALE.....	12
4.8.	DESIGN LANGUAGES.....	13
5.	DESIGN VIEWPOINTS.....	14
5.1.	INTRODUCTION	14
5.2.	CONTEXT VIEWPOINT	14
5.2.1.	SIGN UP	16
5.2.2.	SIGN IN	16

OVERVIEW

5.2.3.	SIGN OUT	16
5.2.4.	ADD TRANSPORTATION ROUTE.....	17
5.2.5.	DELETE TRANSPORTATION ROUTE	17
5.2.6.	REQUEST TRANSPORTATION ROUTE.....	18
5.2.7.	SEARCH TRANSPORTATION ROUTE.....	18
5.2.8.	SEND MESSAGE	19
5.2.9.	REPLY TO A MESSAGE.....	19
5.2.10.	BLOCK USER	20
5.2.11.	RATE USER	20
5.2.12.	CHANGE LANGUAGE	21
5.3.	LOGICAL VIEWPOINT	22
5.3.1.	USER CLASS.....	22
5.3.1.1.	FIELDS	23
5.3.1.2.	FUNCTIONS	24
5.3.2.	TRANSPORTATION CLASS.....	24
5.3.2.1.	FIELDS	24
5.3.3.	MESSAGE CLASS.....	24
5.3.3.1.	FIELDS	25
5.3.4.	RECEIVED MESSAGE CLASS.....	25
5.3.4.1.	FIELDS	25
5.3.4.2.	FUNCTIONS	25
5.3.5.	RATING CLASS	26
5.3.5.1.	FIELDS	26
5.3.6.	DRIVER RATING CLASS.....	26
5.3.6.1.	FIELDS	26
5.3.7.	HITCHHIKER RATING CLASS.....	27
5.3.7.1.	FIELDS	27
5.3.8.	LOGIN USER CLASS	27
5.3.8.1.	FIELDS	27
5.4.	INFORMATION VIEWPOINTS	27
5.5.	INTERFACE VIEWPOINTS.....	28

OVERVIEW

5.5.1.	SYSTEM INTERFACE.....	28
5.5.1.1.	USER COMPONENT	29
5.5.1.2.	SERVER COMPONENT.....	30
5.5.2.	USER INTERFACE	31
5.5.2.1.	SIGN UP USER INTERFACE	31
5.5.2.2.	SIGN IN USER INTERFACE	31
5.5.2.3.	ADD TRANSPORTATION USER INTERFACE.....	32
5.6.	INTERACTION VIEWPOINT.....	34
5.6.1.	SING UP	34
5.6.2.	SING IN	35
5.6.3.	SING OUT.....	36
5.6.4.	ADD TRANSPORTATION ROUTE.....	37
5.6.5.	DELETE TRANSPORTATION ROUTE	38
5.6.6.	REQUEST TRANSPORTATION ROUTE.....	39
5.6.7.	SEARCH TRANSPORTATION ROUTE.....	40
5.6.8.	SEND MESSAGE	41
5.6.9.	REPLY TO MESSAGE.....	42
5.6.10.	BLOCK USER	43
5.6.11.	RATE USER	44
5.6.12.	CHANGE LANGUAGE	45
5.7.	STATE DYNAMIC VIEWPOINT	45
5.7.1.	MAIN PAGE.....	46
5.7.2.	SIGN UP PAGE.....	46
5.7.3.	USER PROFILE PAGE.....	47
5.7.4.	TRANSPORTATION ADDING PAGE	47
5.7.5.	SEARCH TRANSPORTATION PAGE	47
5.7.6.	TRANSPORTATION PAGE.....	47
5.7.7.	TRANSPORTATION REQUEST PAGE	47

TABLE OF FIGURES

Figure 1. Visualization of Use Case Diagram	15
Figure 2. Visualization of Sign Up Use Case Diagram	16
Figure 3. Visualization of Sign In Use Case Diagram	16
Figure 4. Visualization of Sign Out Use Case Diagram.....	17
Figure 5. Visualization of Add Transportation Route Use Case Diagram.....	17
Figure 6. Visualization of Delete Transportation Route Use Case Diagram.....	18
Figure 7. Visualization of Request Transportation Route Use Case Diagram	18
Figure 8. Visualization of Search Transportation Route Use Case Diagram	19
Figure 9. Visualization of Send Message Use Case Diagram	19
Figure 10. Visualization of Reply to Message Use Case Diagram.....	20
Figure 11. Visualization of Block User Use Case Diagram	20
Figure 12. Visualization of Rate User Use Case Diagram	21
Figure 13. Visualization of Change Language Use Case Diagram	21
Figure 14. Visualization of Class Diagram	22
Figure 15. Visualization of User Class Diagram.....	23
Figure 16. Visualization of Transportation Class Diagram.....	24
Figure 17. Visualization of Message Class Diagram	25
Figure 18. Visualization of ReceivedMessage Class Diagram.....	25
Figure 19. Visualization of Rating Class Diagram.....	26
Figure 20. Visualization of DriverRating Class Diagram	26
Figure 21. Visualization of HitchhikerRating Class Diagram.....	27
Figure 22. Visualization of LoginUser Class Diagram.....	27
Figure 23. Visualization of ER Diagram	28
Figure 24. Visualization of System Component Diagram.....	29
Figure 25. Visualization of User Component Diagram	29
Figure 26. Visualization of Server Component Diagram	30
Figure 27. Visualization of Sign Up User Interface	31
Figure 28. Visualization of Sign In User Interface	32
Figure 29. Visualization of Add Transportation User Interface	33
Figure 30. Visualization of Sign Up Sequence Diagram	34
Figure 31. Visualization of Sign In Sequence Diagram.....	35
Figure 32. Visualization of Sign Out Sequence Diagram	36
Figure 33. Visualization of Add Transportation Route Sequence Diagram.....	37
Figure 34. Visualization of Delete Transportation Route Sequence Diagram.....	38
Figure 35. Visualization of Request Transportation Route Sequence Diagram	39

OVERVIEW

Figure 36. Visualization of Search Transportation Route Sequence Diagram	40
Figure 37. Visualization of Send Message Sequence Diagram	41
Figure 38. Visualization of Reply To Message Sequence Diagram	42
Figure 39. Visualization of Block User Sequence Diagram	43
Figure 40. Visualization of Rate User Sequence Diagram	44
Figure 41. Visualization of Change Language Sequence Diagram	45
Figure 42. Visualization of State Chart Diagram	46

1. OVERVIEW

Software Design Document (SDD) of Carpool provides necessary definitions to conceptualize and further formalize design of the software, whose requirements and functionalities were summarized in Software Requirements Specifications (SRS) Report. Aim is to provide guidance to a design which could be easily implemented by any programmer reading this report. The document complies with the IEEE standards (IEEE Std 1016 – 2009).

1.1. SCOPE

This complete SDD will contain the general definition and features of the project, design constraints, the overall system architecture and data architecture, a brief explanation about our current progress and schedule of the project. With the help of UML diagrams, design of the system and subsystems/modules will be explained visually in order to help the programmer to understand all information stated in this document correctly and easily.

1.2. PURPOSE

This SDD is intended to provide a software system design which will satisfy functional and nonfunctional requirements stated in SRS Document of Carpool. Purpose of this document is serving as a guideline throughout development phase of the project for developers.

1.3. INTENDED AUDIENCE

Audience is the staff who will develop the application described in this document and advisors of this staff.

DEFINITIONS

2. DEFINITIONS

DEFINITIONS TABLE

TERMS	DEFINITIONS
Acquirer	Project Consultant – Atilla 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
ER	Entity Relation
HTML	Hyper Text Markup Language
UML	Unified Modeling Language

CONCEPTUAL MODEL FOR SOFTWARE DESIGN DESCRIPTION

3. CONCEPTUAL MODEL FOR SOFTWARE DESIGN DESCRIPTION

Basic terms, concepts and context of SDD will be given in this part.

3.1. SOFTWARE DESIGN IN CONTEXT

The task of the project is to provide communication environment between drivers and hitchhikers to optimize use of private transportation vehicles to reduce air pollution and fuel usage via a web portal. This web portal is aimed to help people who use transportation by supplying necessary services.

This project will be implemented with JavaScript and PHP as scripting languages and CSS and bootstrap as style sheet language. In addition, Google Map Api will be used to get location information.

3.2. SOFTWARE DESIGN DESCRIPTIONS WITHIN THE LIFE CYCLE

3.2.1. INFLUENCES ON SDD PREPARATION

The key software life cycle product that drives a software design is typically the software requirements specification. The requirements in the SRS (product perspective, functional and non-functional requirements and interface requirements) and also the demands of the stakeholders specify the design of the project.

3.2.2. INFLUENCES ON SOFTWARE LIFE CYCLE PRODUCTS

During the preparation phase of SDD and/or during the implementation stage of the project some requirements may change. Besides, SDD influences test plans and test documentation of the Carpool System.

3.2.3. DESIGN VERIFICATION AND DESIGN ROLE IN VALIDATION

Verification and validation will be tested after preparation of the test cases. All system parts will be tested against these cases. It will be checked for whether the requirements fulfilled or not.

4. DESIGN DESCRIPTION INFORMATION CONTENT

4.1. INTRODUCTION

Software Design Description of Carpool Web Portal identifies how this web application will be designed and implemented. Throughout the document identification, diagrams, user views and user viewpoints are provided.

4.2. SDD IDENTIFICATION

After testing for the verification and validation Carpool will be deployed to the real world environment. Users can use this application as both driver and hitchhiker. 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.

4.3. DESIGN STAKEHOLDERS AND THEIR CONCERNS

Stakeholders include developers, testers and end users of “Carpool”. Since it will be a web application, usage should be easy and interfaces should be practical and appealing. In addition, it must be a trustworthy environment for stakeholders that their profile information should be kept in private.

4.4. DESIGN VIEWS

This project will be implemented as modular structure. MVC architecture will be used as architecture pattern. Object-oriented design is chosen for this project. This give us an advantage to integrate new features to our project and remove and replace the components which we want to.

For instance, we are about to integrate an activity module to our system, MVC architecture and object-oriented design provide us an environment to adapt our web application to this new feature easily. Context view gives us structured analysis context diagram and UML use case diagram of systems services and users. Interface views clearly specify which inputs give which outputs.

4.5. DESIGN VIEWPOINTS

Context viewpoint shows what is expected from the user actor in the system. The roles of the users and stakeholders are clearly specified. System boundary should be designated. Design entities will be user, and the information will flow between this user and system. Input-output relations will be explained in context viewpoint-design elements. Analysis technique and evaluation technique can be applied to this viewpoint.

DESIGN DESCRIPTION INFORMATION CONTENT

In addition, information view provides us persistent information with help of UML class diagram and entity relation diagram. With composition view we made team organization. This view also shows estimated cost, staffing and scheduling. Relationships of the classes are easily perceived. Interaction views gives service definition and service access. Interface viewpoint is needed for the test cases. It includes the details of the external and internal interfaces. State dynamic view shows the state transitions with diagrams.

4.6. DESIGN ELEMENTS

4.6.1. DESIGN ENTITIES

- User
- Contactinfo
- Blacklist
- Trash
- Message
- Inbox
- Outbox
- Transportation
- Request
- periodicTransportation
- transportationInfo
- Schedule
- Passengers
- Rating

4.6.2. DESIGN ATTRIBUTES

Attributes and attributes types of the entities can be seen at entity-relationship diagram in section 5.4. Information Viewpoint.

DEFINITIONS TABLE

TABLE NAME	DESCRIPTION
User	Holds the information of the user.
ContactInfo	Holds contact information of the user.
Blacklist	Holds the ids of other users blocked by the user.
Trash	Holds the messages deleted by the users.
Messages	Holds the entire messages.

DESIGN DESCRIPTION INFORMATION CONTENT

Inbox	Holds the messages send to user.
Outbox	Holds the messages send by the user.
Transportation	Holds the some of the transportation information.
TransportationInfo	Holds some information about the transportation table.
PeriodicTransformation	Holds the periodic transportation data.
Passengers	Holds the passengers ids and transportation ids.
Schedule	Holds the schedule information of periodic transportation.
Request	Holds the request information of hitchhiker.
generalRating	Holds general rating information
driverRating	Holds driver's rating information
hitchhikerRating	Holds hitchhiker's rating information

4.6.3. DESIGN RELATIONSHIPS

The relationship of all tables is illustrated in section 5.4. Information Viewpoint

4.6.4. DESIGN CONSTRAINTS

Carpool system requires 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.

4.7. DESIGN RATIONALE

We chose MVC as architecture design pattern and object oriented design pattern because MVC architectural pattern separates an application into three main components: the model, the view, and the controller. We control our data which are in model, and views via controller. That makes easy to change information in tables.

Moreover, with object-oriented design we can easily maintain our project as we mentioned before we can easily add some extra features and identify the source of error and fixed it immediately. Finally, each function in the software will be commented so that it can be

understandable for the other developers and also they can change the code by help of these comments.

We choose six viewpoints which are Context Viewpoint, Logical Viewpoint, Information Viewpoint, Interface Viewpoint, Interaction Viewpoint, State Dynamic Viewpoint.

Context Viewpoint is chosen because our project is an event-based application so it is very useful showing user events on the system in detail.

Logical Viewpoint is chosen because object-oriented systems such as our system can be illustrated with class diagrams very easily and effectively.

Information Viewpoint is chosen because database management is crucial for our system and it should be handled in more detail in terms of ER Diagrams.

Interface Viewpoint is chosen because our system is a web-based application and user behavior is very significant on our system. Therefore, we should provide some user interfaces and component diagram to handle over this issue.

System reactions to user events, such as sign up and sign in events, should be clear for both developers and stakeholders. For that reason, Interaction Viewpoint is chosen.

In our system and almost all web-based applications, every single web page can be considered as a state and transitions between states are very crucial. Therefore, we chose to use State Dynamic Viewpoint.

4.8. DESIGN LANGUAGES

Unified Modeling Language (UML) is selected as a part of design viewpoint specification.

5. DESIGN VIEWPOINTS

5.1. INTRODUCTION

In this document, five viewpoint are designed for the system as listed below;

- Context Viewpoint
- Logical Viewpoint
- Information Viewpoint
- Interface Viewpoint
- Interaction Viewpoint
- State Dynamic Viewpoint

5.2. CONTEXT VIEWPOINT

There is only one kind of user in our system. The system supplies twelve services to them, namely sign up, sign in, sign out, add transportation route, delete transportation route, request transportation route, search transportation route, send message, reply to message, block user, rate user and change language.

A more detailed information about use cases can be found in system requirement specification document.

DESIGN VIEWPOINTS

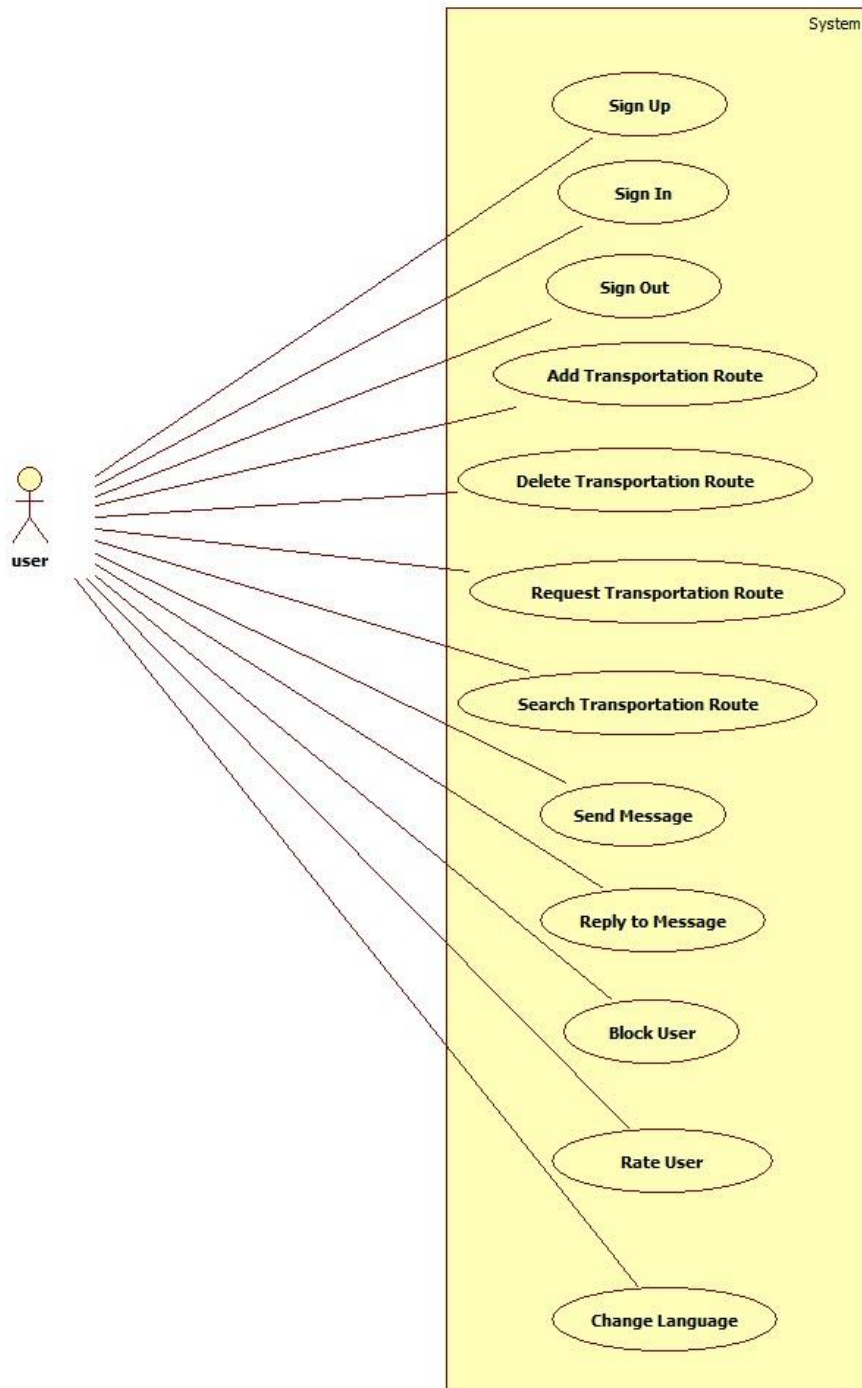


Figure 1. Visualization of Use Case Diagram

DESIGN VIEWPOINTS

5.2.1. SIGN UP

Users need to be signed up in order to benefit from the website. They need to specify their unique username and password. They are required to enter their name, surname, e-mail, age, job, phone and gender.

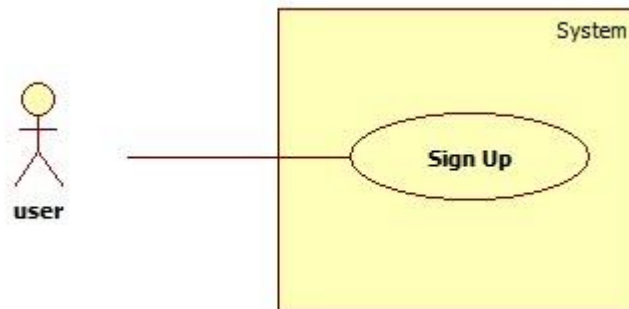


Figure 2. Visualization of Sign Up Use Case Diagram

5.2.2. SIGN IN

Users may sign in to the system by entering their unique username and password in main page. He/She needs to be signed up first.

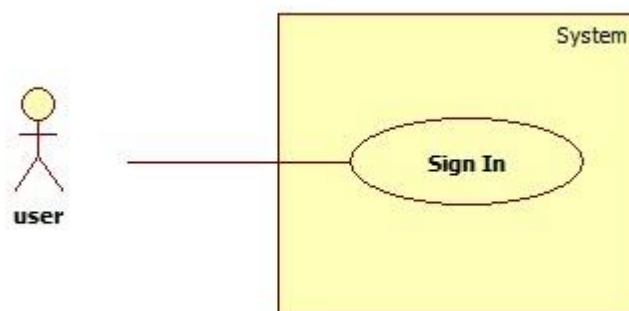


Figure 3. Visualization of Sign In Use Case Diagram

5.2.3. SIGN OUT

Users who have been signed may sign out by clicking sign out button that exists in every page.

DESIGN VIEWPOINTS

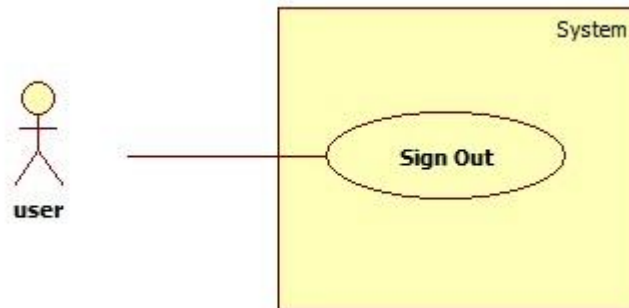


Figure 4. Visualization of Sign Out Use Case Diagram

5.2.4. ADD TRANSPORTATION ROUTE

Users may add transportations by specifying a route, time/time period and number of empty seats. Route can be selected by two ways. First one is that, user writes start and end locations and corresponding route is drawn in the map. The other one is that, user can select start and end locations on the map. Also, he/she can select waypoints (up to 8).

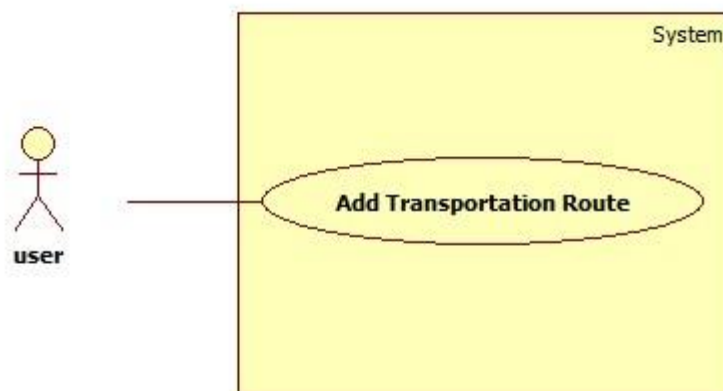


Figure 5. Visualization of Add Transportation Route Use Case Diagram

5.2.5. DELETE TRANSPORTATION ROUTE

When a user will no longer use a transportation, he/she may delete it. Passengers in that transportation will be informed.

DESIGN VIEWPOINTS

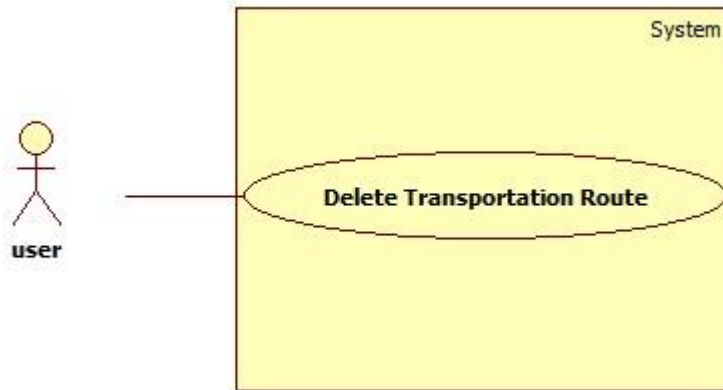


Figure 6. Visualization of Delete Transportation Route Use Case Diagram

5.2.6. REQUEST TRANSPORTATION ROUTE

When a user wants to be involved in a transportation, he/she can send a transportation request to the driver of the transportation.

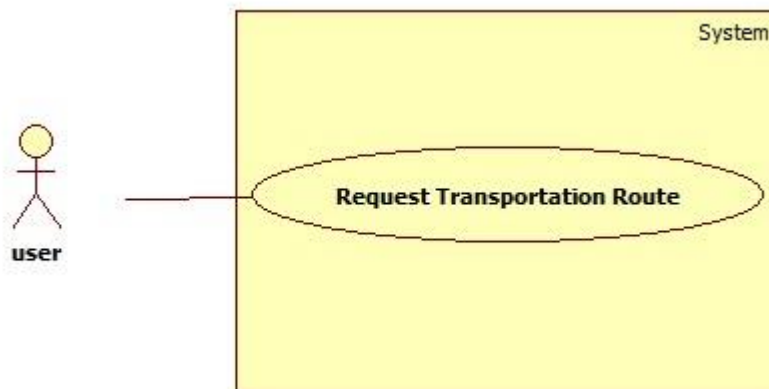


Figure 7. Visualization of Request Transportation Route Use Case Diagram

5.2.7. SEARCH TRANSPORTATION ROUTE

Users can search for transportations that are proper for them. They need to specify a route and a time to see related transportations.

DESIGN VIEWPOINTS

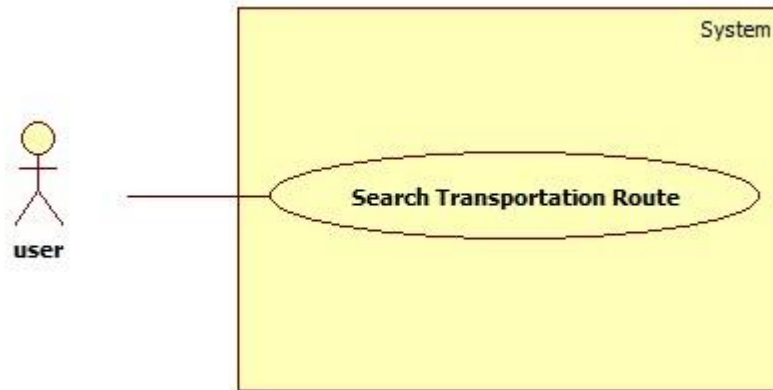


Figure 8. Visualization of Search Transportation Route Use Case Diagram

5.2.8. SEND MESSAGE

Users may need to ask questions to each other. In such cases, they can send messages to each other.

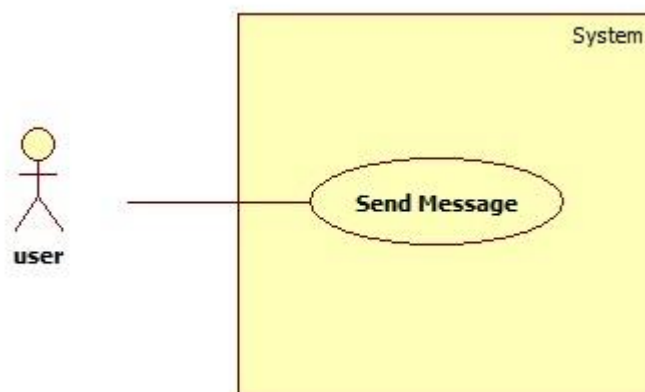


Figure 9. Visualization of Send Message Use Case Diagram

5.2.9. REPLY TO A MESSAGE

When a message is sent to a user that user is informed and he/she can read the message.

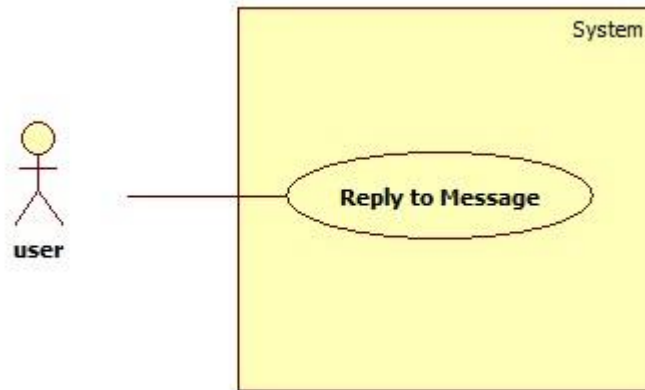


Figure 10. Visualization of Reply to Message Use Case Diagram

5.2.10. BLOCK USER

When a user is bothered from messages that are coming from a user, he/she can block that user and never gets messages from him/her.

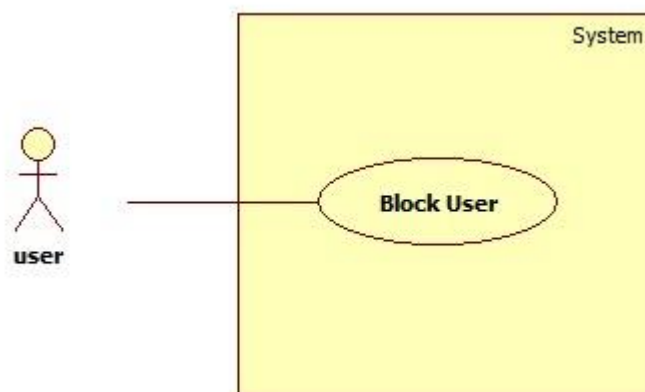


Figure 11. Visualization of Block User Use Case Diagram

5.2.11. RATE USER

People want to know about people they travel with. Therefore, users are allowed to rate each other in our webpage. Hereby, they can look at ratings about a person and decide whether to travel with him/her or not.

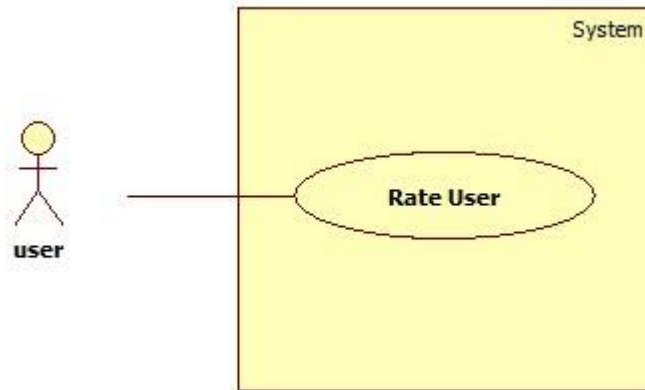


Figure 12. Visualization of Rate User Use Case Diagram

5.2.12. CHANGE LANGUAGE

The system has two languages, namely Turkish and English. Users may switch between the languages by clicking on the small flags that exists in all pages.

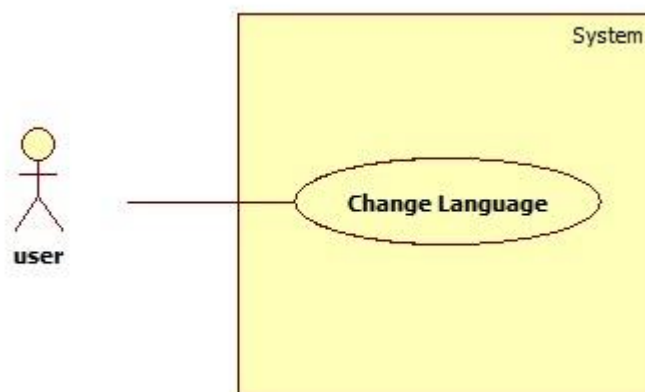


Figure 13. Visualization of Change Language Use Case Diagram

DESIGN VIEWPOINTS

5.3. LOGICAL VIEWPOINT

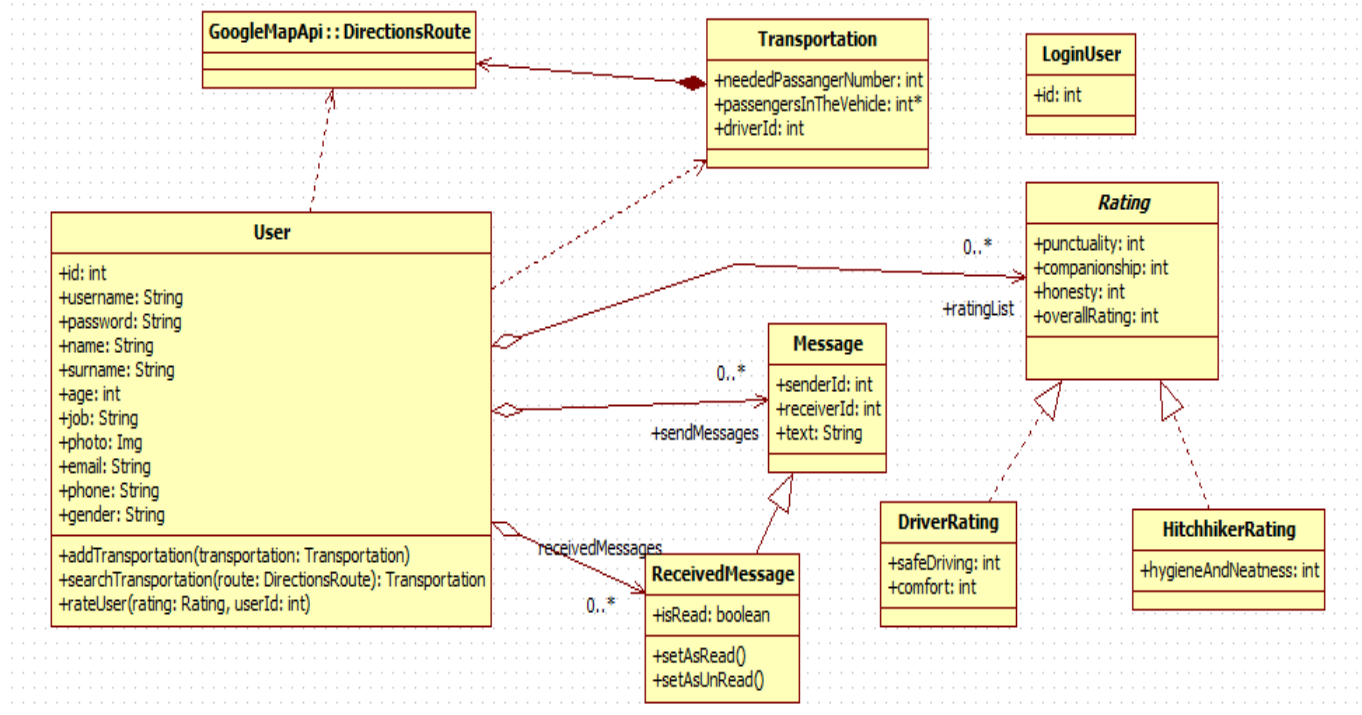


Figure 14. Visualization of Class Diagram

5.3.1. USER CLASS

This class 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.

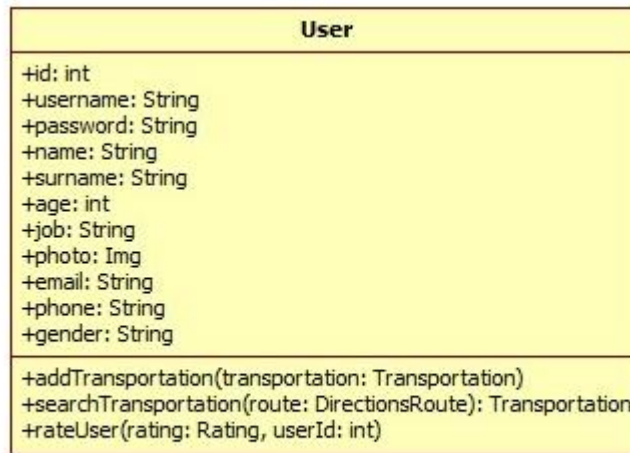


Figure 15. Visualization of User Class Diagram

5.3.1.1. 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.

DESIGN VIEWPOINTS

5.3.1.2. 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.

5.3.2. TRANSPORTATION CLASS

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

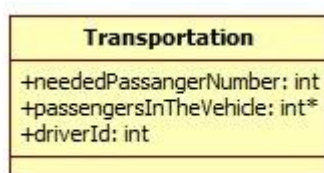


Figure 16. Visualization of Transportation Class Diagram

5.3.2.1. 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.

5.3.3. MESSAGE CLASS

This class is a simple message class. It has sender and receiver ids and a text. Also, this class is base class for two other classes.

DESIGN VIEWPOINTS

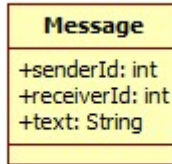


Figure 17. Visualization of Message Class Diagram

5.3.3.1. 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.

5.3.4. RECEIVED MESSAGE CLASS

This class inherits Message class and adds isRead attribute. Also, it adds two functions for marking the message as read and unread.

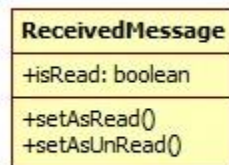


Figure 18. Visualization of ReceivedMessage Class Diagram

5.3.4.1. FIELDS

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

5.3.4.2. FUNCTIONS

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

DESIGN VIEWPOINTS

5.3.5. RATING CLASS

This class is an abstract base class for DriverRating and HitchhikerRating classes. It keeps common attributes.

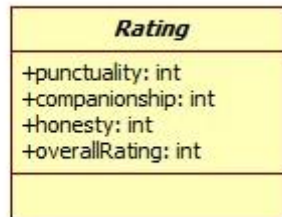


Figure 19. Visualization of Rating Class Diagram

5.3.5.1. 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.

5.3.6. DRIVER RATING CLASS

This class inherits Rating class and adds two fields.



Figure 20. Visualization of DriverRating Class Diagram

5.3.6.1. FIELDS

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

DESIGN VIEWPOINTS

5.3.7. HITCHHIKER RATING CLASS

This class inherits Rating class and adds one field.

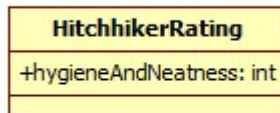


Figure 21. Visualization of HitchhikerRating Class Diagram

5.3.7.1. FIELDS

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

5.3.8. LOGIN USER CLASS

This class keeps track of which user is currently logged in.

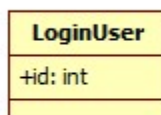


Figure 22. Visualization of LoginUser Class Diagram

5.3.8.1. FIELDS

- id : This is the id of current logged in user.

5.4. INFORMATION VIEWPOINTS

ER Diagram of the system is illustrated as below and elements and attributes of the diagram is explained in section 4.6.1. Design Entities and 4.6.2. Design Attributes.

DESIGN VIEWPOINTS

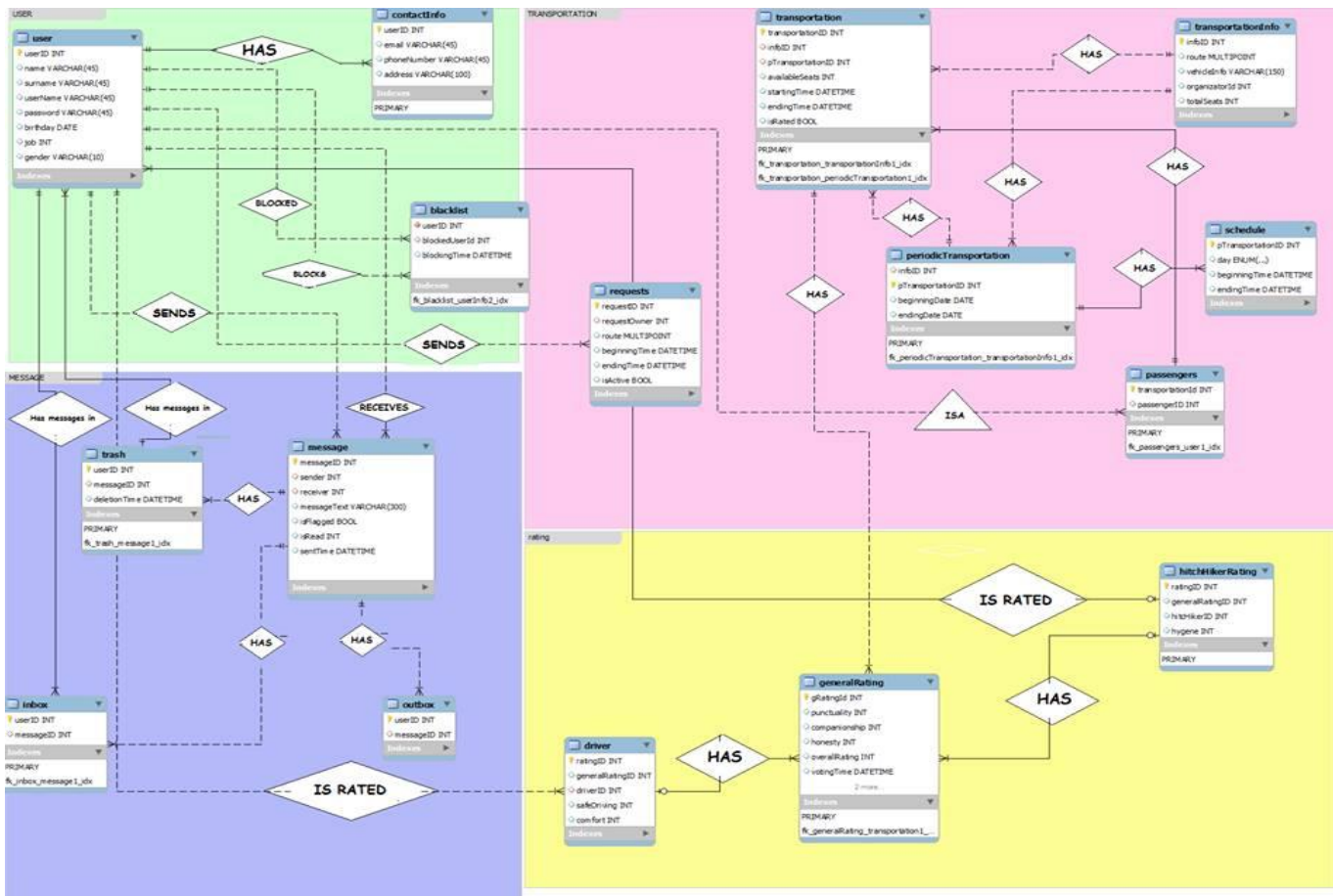


Figure 23. Visualization of ER Diagram

5.5. INTERFACE VIEWPOINTS

5.5.1. SYSTEM INTERFACE

This viewpoint gives information about the correct usage of services provided by a design object. The design contains three main components in terms of User, Server and Google Map Server. Server and Google Map Server are connected with User Management Component of User Component. As a result of this connection, the project arises as a structure of the system.

The component diagram for interfaces can be seen as below;

DESIGN VIEWPOINTS

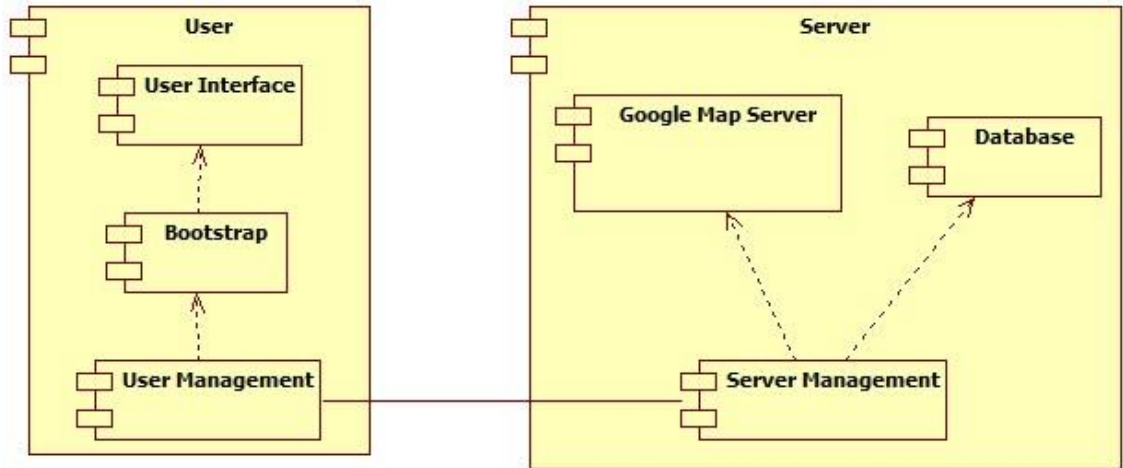


Figure 24. Visualization of System Component Diagram

5.5.1.1. USER COMPONENT

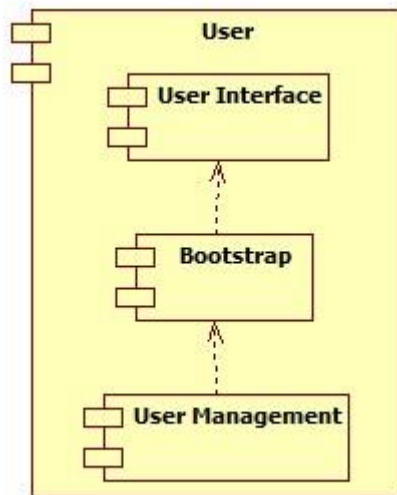


Figure 25. Visualization of User Component Diagram

User Component aims to visualize events from the system and react the user event to the system via Web browsers. User Component has three inside components in terms of User Management, Bootstrap and User Interface Component.

User Management Component is connected with Server and Google Map Component to handle events in the system.

DESIGN VIEWPOINTS

Bootstrap Component helps User Interface Component to show elegant design to the users.

User Interface Component aims to show interface of the system to users via HTML.

5.5.1.2. SERVER COMPONENT

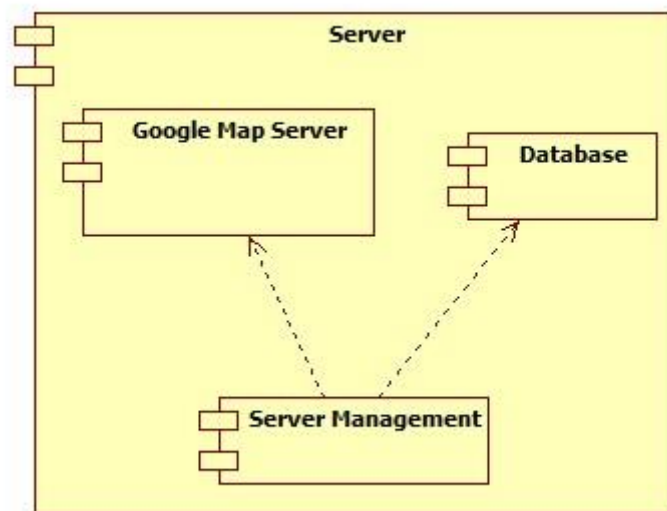


Figure 26. Visualization of Server Component Diagram

Server Component aims to handle control of data flow over the system as managing the server side. Server Component has two components in terms of Server Management and Database Component.

Server Management Component aims to handle event listening responding with User Management Component.

Database Component is responsible from holding the user, transportation and event information in database.

Google Map Database Component is responsible from provide geographical service from Google Map API.

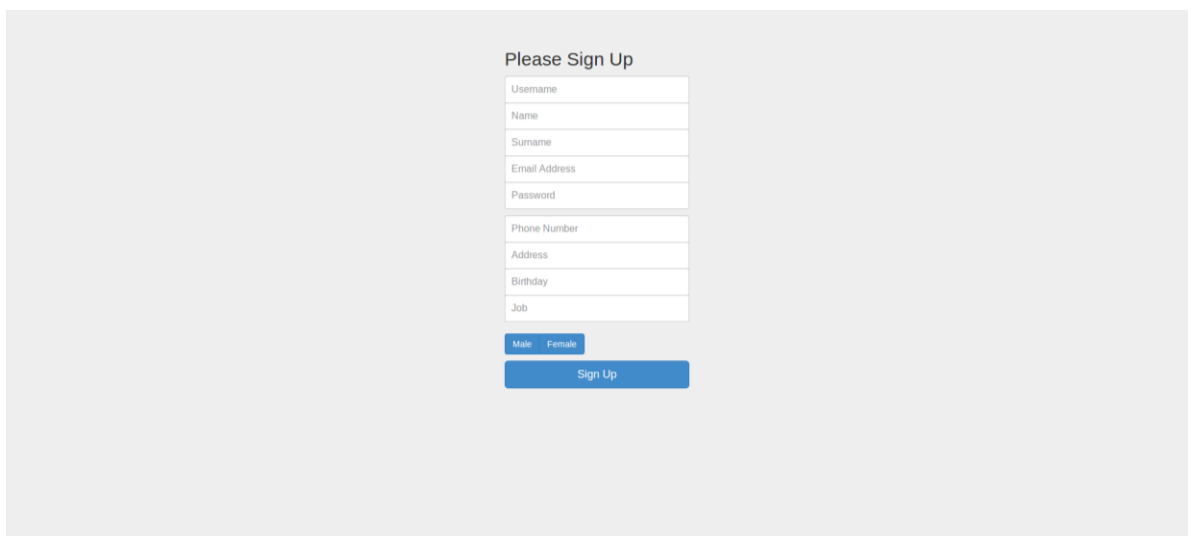
DESIGN VIEWPOINTS

5.5.2. USER INTERFACE

All of the user interface layouts are mentioned in section 5.1. in SRS document . Some of these are shown below with Bootstrap facilities.

5.5.2.1. SIGN UP USER INTERFACE

Sign up interface of the system has ten attributes in terms of username, name, surname, e-mail address, password, phone number, address, birthday, job and gender fields.



The image shows a web form titled "Please Sign Up". It contains ten input fields: Username, Name, Surname, Email Address, Password, Phone Number, Address, Birthday, and Job. Below the Job field are two radio buttons for "Male" and "Female". At the bottom of the form is a blue "Sign Up" button.

Figure 27. Visualization of Sign Up User Interface

5.5.2.2. SIGN IN USER INTERFACE

Sign in interface of the system has three attributes in terms of e-mail address, password and forgot password fields.

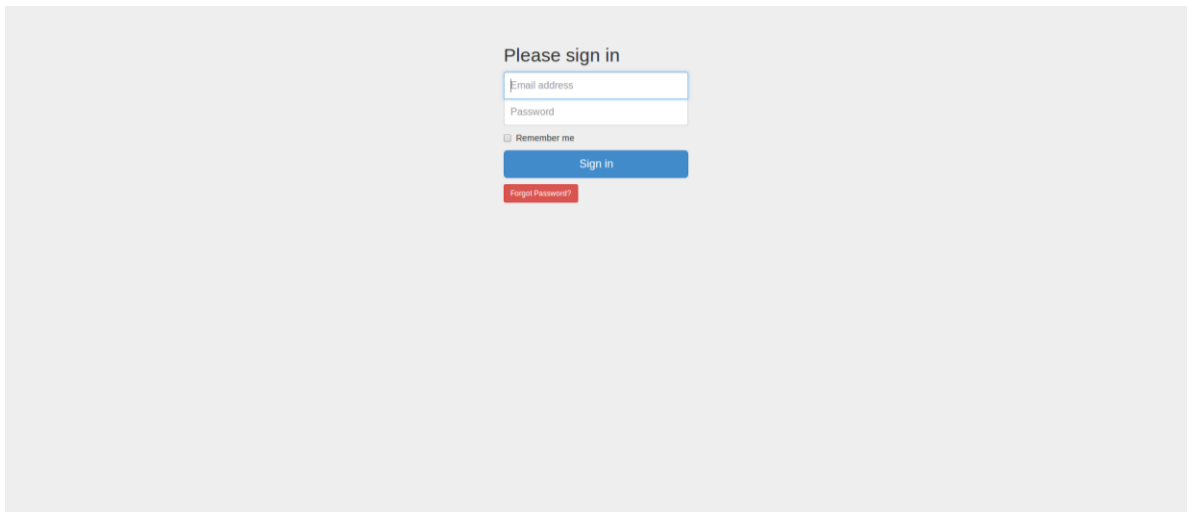


Figure 28. Visualization of Sign In User Interface

5.5.2.3. ADD TRANSPORTATION USER INTERFACE

Add transportation interface of the system has date and time field for transportation. Also for this transformation, one time and periodic, which may be selected as days of weeks, facilities.

Google map API interface is shown in the middle of the page to pin directions of the route on the map.

The other alternative of determine the route via filling the textbox of start location and end location.

DESIGN VIEWPOINTS

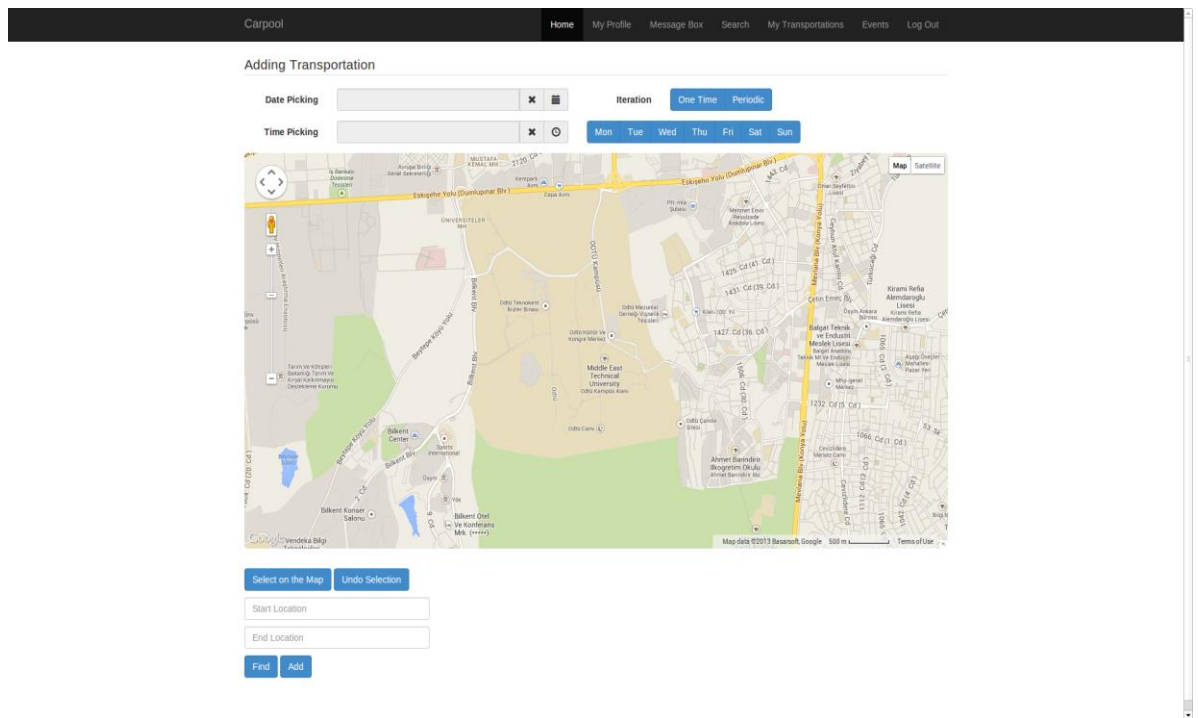


Figure 29. Visualization of Add Transportation User Interface

DESIGN VIEWPOINTS

5.6. INTERACTION VIEWPOINTS

5.6.1. SING UP

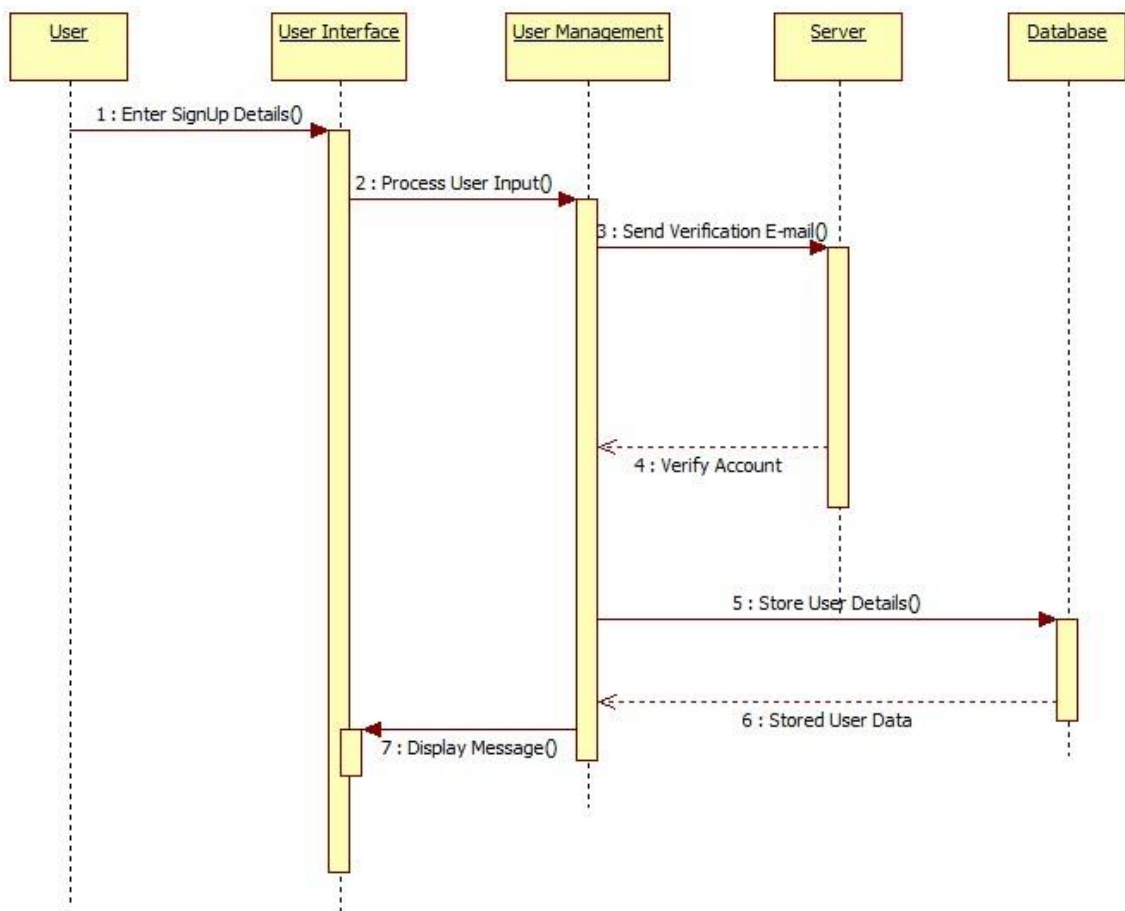


Figure 30. Visualization of Sign Up Sequence Diagram

User performs register operation by this functionality. User enters his or her name, surname, password and e-mail information and receives verification e-mail send by the system. By confirming register information, user successfully finish the register operation.

DESIGN VIEWPOINTS

5.6.2. SIGN IN

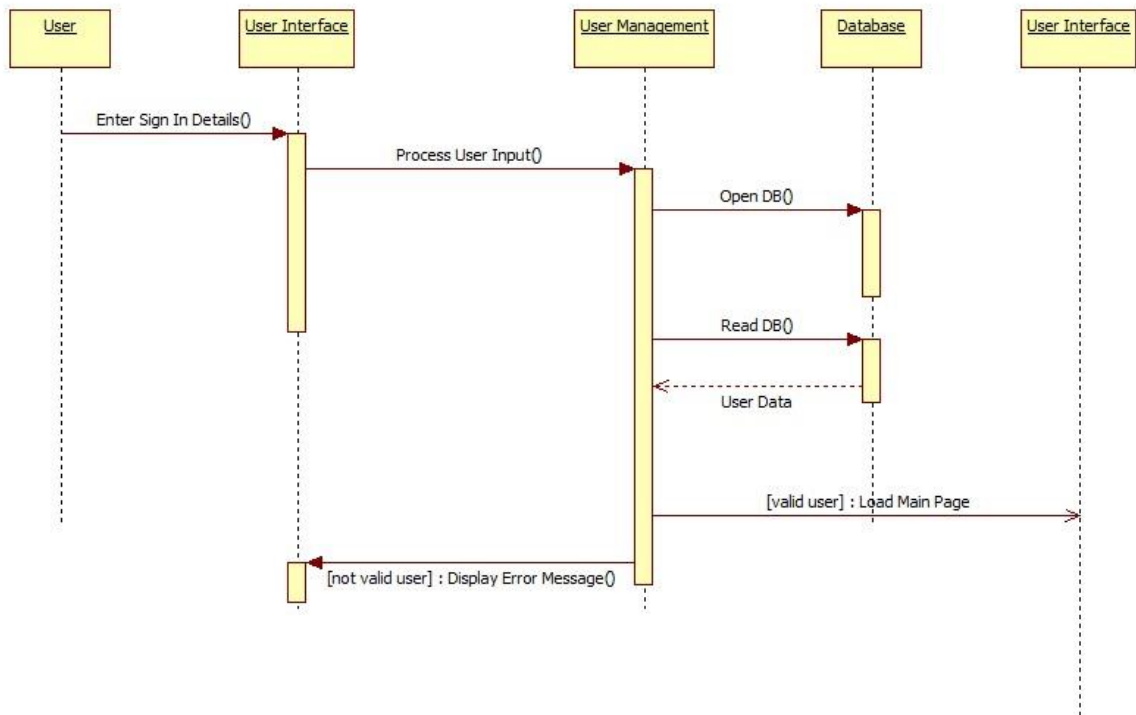


Figure 31. Visualization of Sign In Sequence Diagram

After sign up process, user can sign in to the system by entering his or her account information. Then, user can benefit from system functionalities.

DESIGN VIEWPOINTS

5.6.3. SIGN OUT

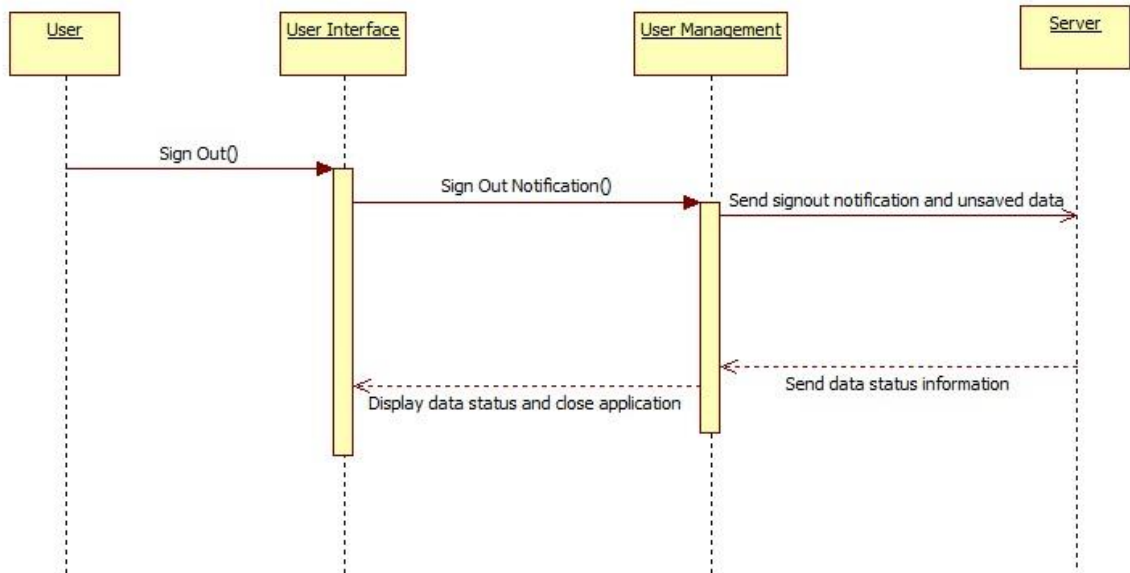


Figure 32. Visualization of Sign Out Sequence Diagram

When the user click sign out button through the system interface, he or she can perform secure exit from the system. All cookies and sessions will be cleared by the system.

DESIGN VIEWPOINTS

5.6.4. ADD TRANSPORTATION ROUTE

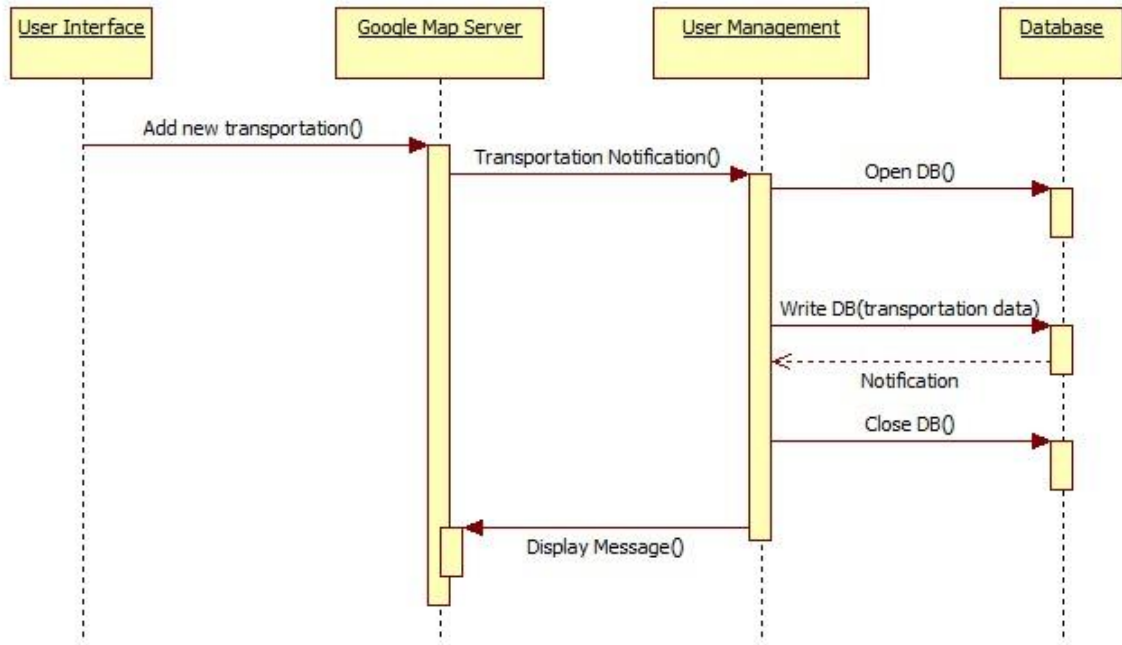


Figure 33. Visualization of Add Transportation Route Sequence Diagram

User selects transportation route from Google Map and fills other necessary information required for the route. After this step, system stores user's route data through database service.

DESIGN VIEWPOINTS

5.6.5. DELETE TRANSPORTATION ROUTE

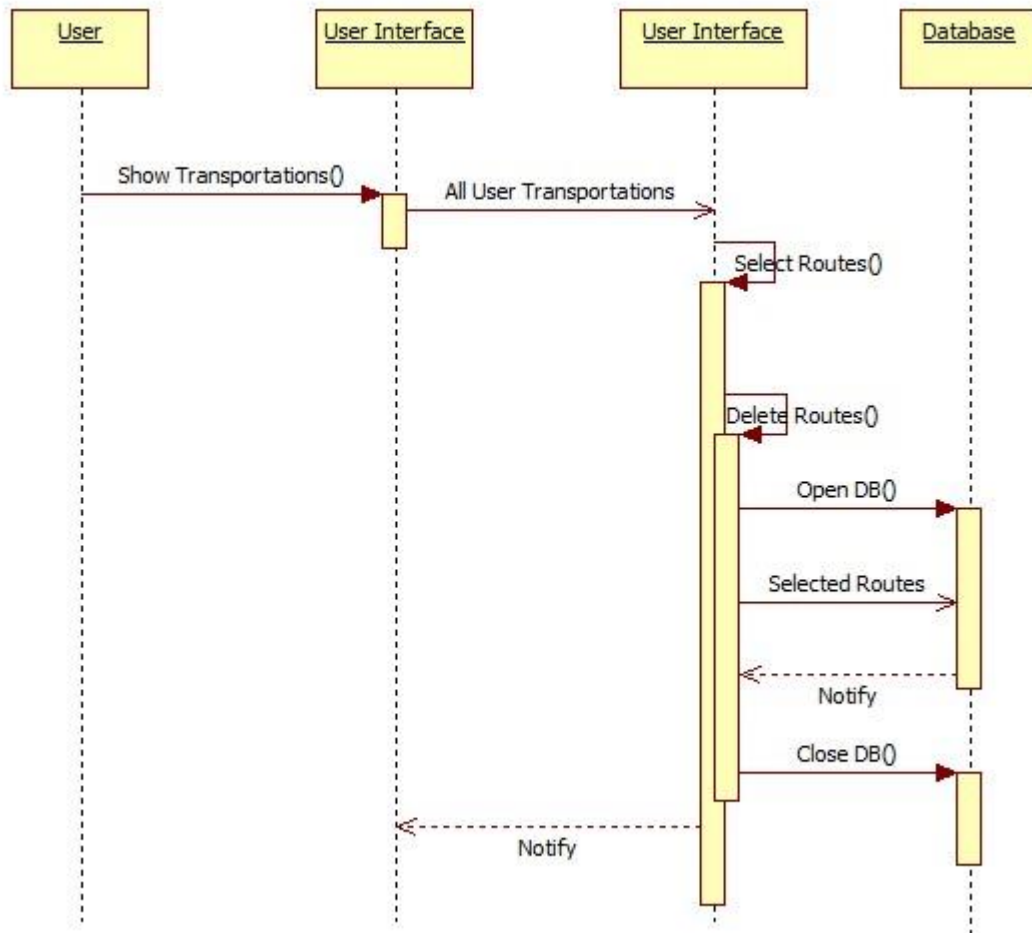


Figure 34. Visualization of Delete Transportation Route Sequence Diagram

This functionality is used for deletion of a selected route from route list of a user. When the deletion is successful, the deleted route will no longer be available to the user.

DESIGN VIEWPOINTS

5.6.6. REQUEST TRANSPORTATION ROUTE

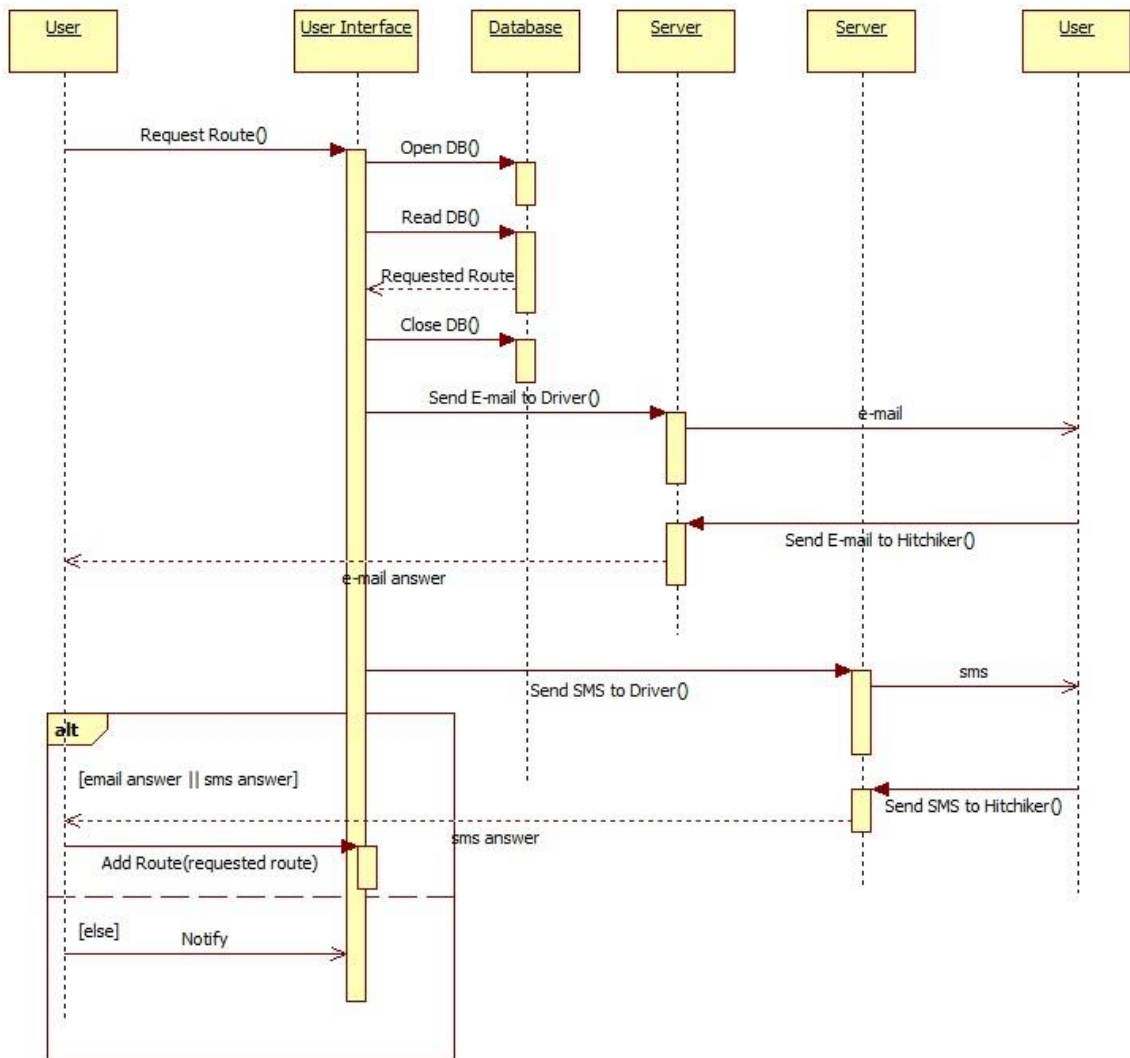


Figure 35. Visualization of Request Transportation Route Sequence Diagram

When user logs in to the system, he or she can request transportation route from the driver. After the request, system sends e-mail and sms to driver about requested route. From that point, if hitchhiker and driver settle on the route, system adds this route related to both user.

DESIGN VIEWPOINTS

5.6.7. SEARCH TRANSPORTATION ROUTE

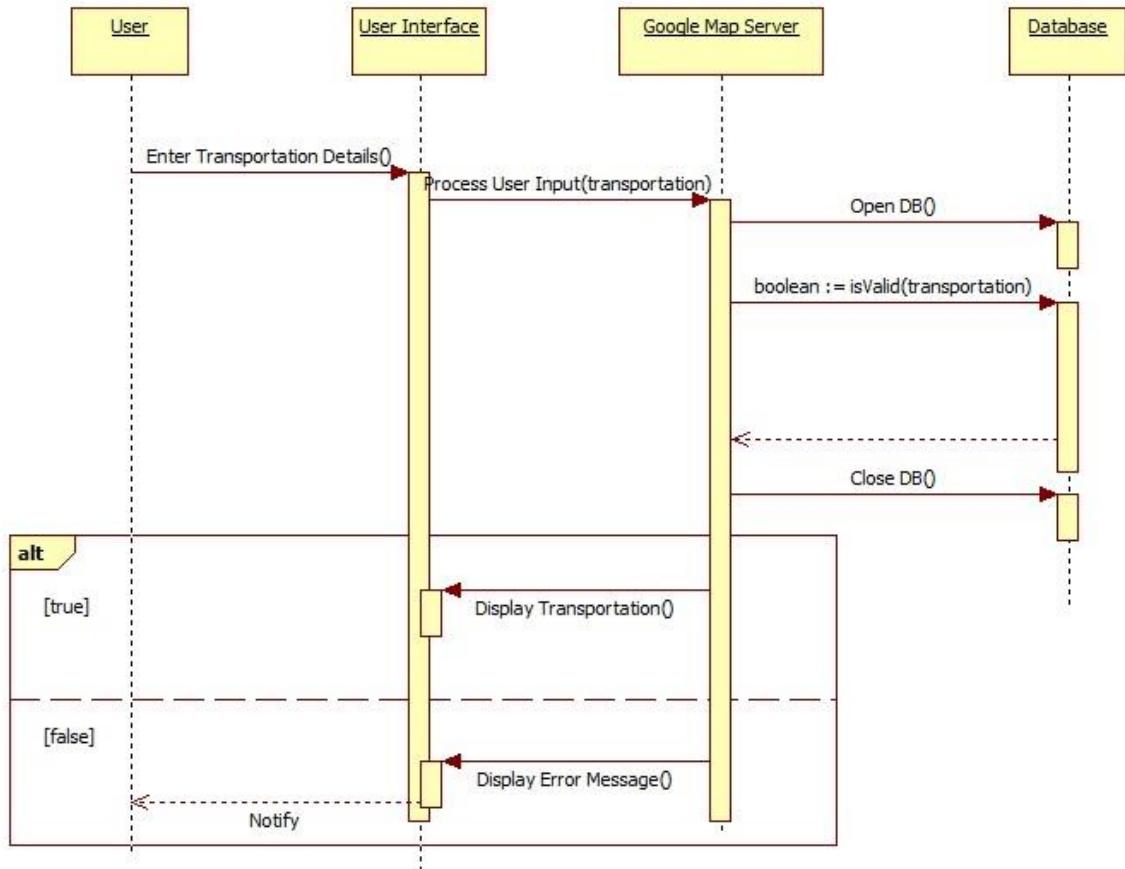


Figure 36. Visualization of Search Transportation Route Sequence Diagram

The system provides search about available routes to its users. To accomplish this functionality, user shall enter his or her starting and destination point of a route. After that, system lists all available transportations related to these points.

DESIGN VIEWPOINTS

5.6.8. SEND MESSAGE

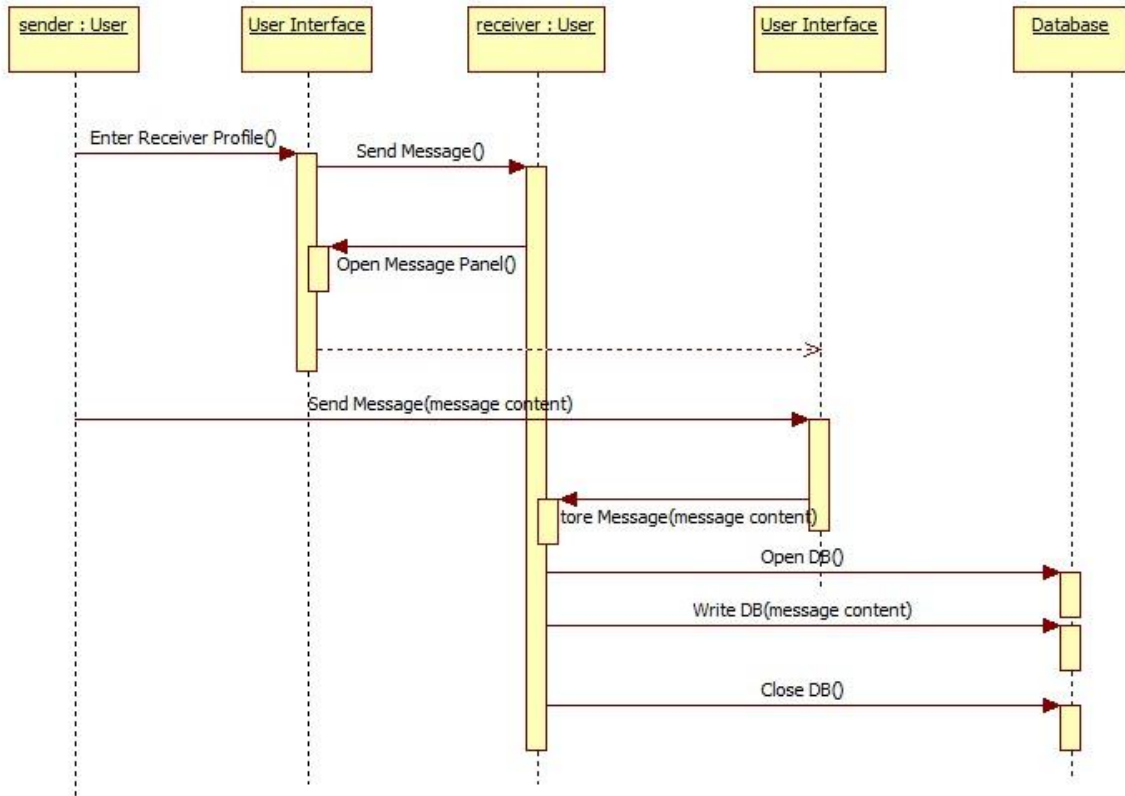


Figure 37. Visualization of Send Message Sequence Diagram

The system has capability of messaging between its users. To accomplish this, sender user shall enter receiver's profile page and click the send message button. After that point, message panel will be loaded and send message will be stored to database by the system.

DESIGN VIEWPOINTS

5.6.9. REPLY TO MESSAGE

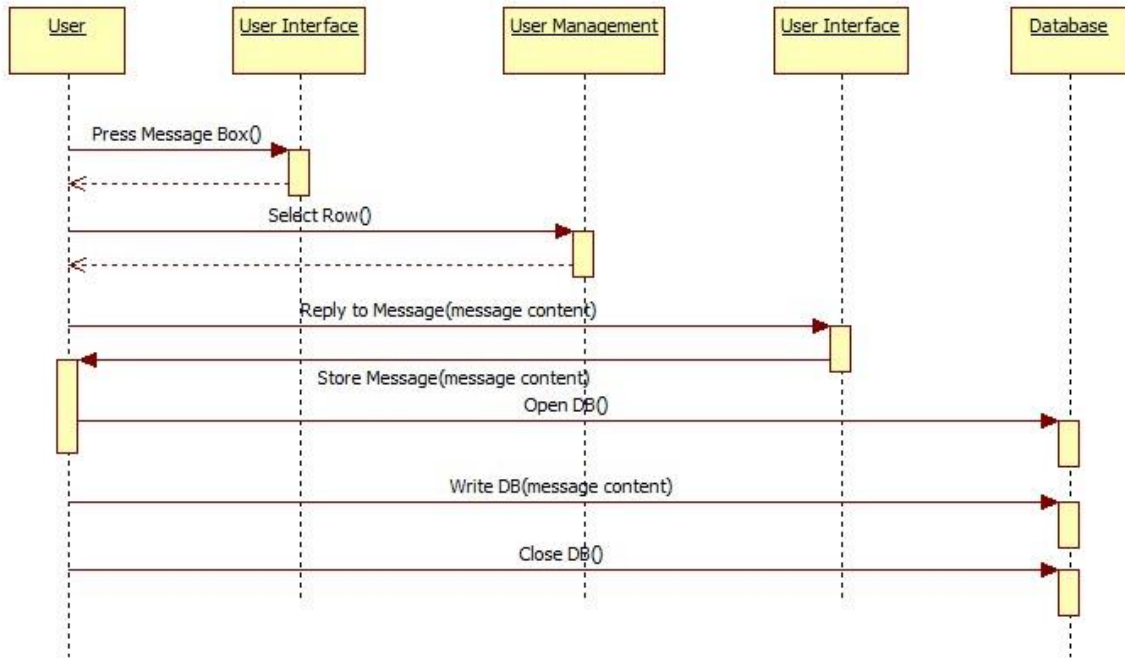


Figure 38. Visualization of Reply To Message Sequence Diagram

User can list his or her messages by clicking message box in profile page. And he or she selects row which is intended to reply. After that, user can click send message button and message panel will be loaded by the system.

DESIGN VIEWPOINTS

5.6.10. BLOCK USER

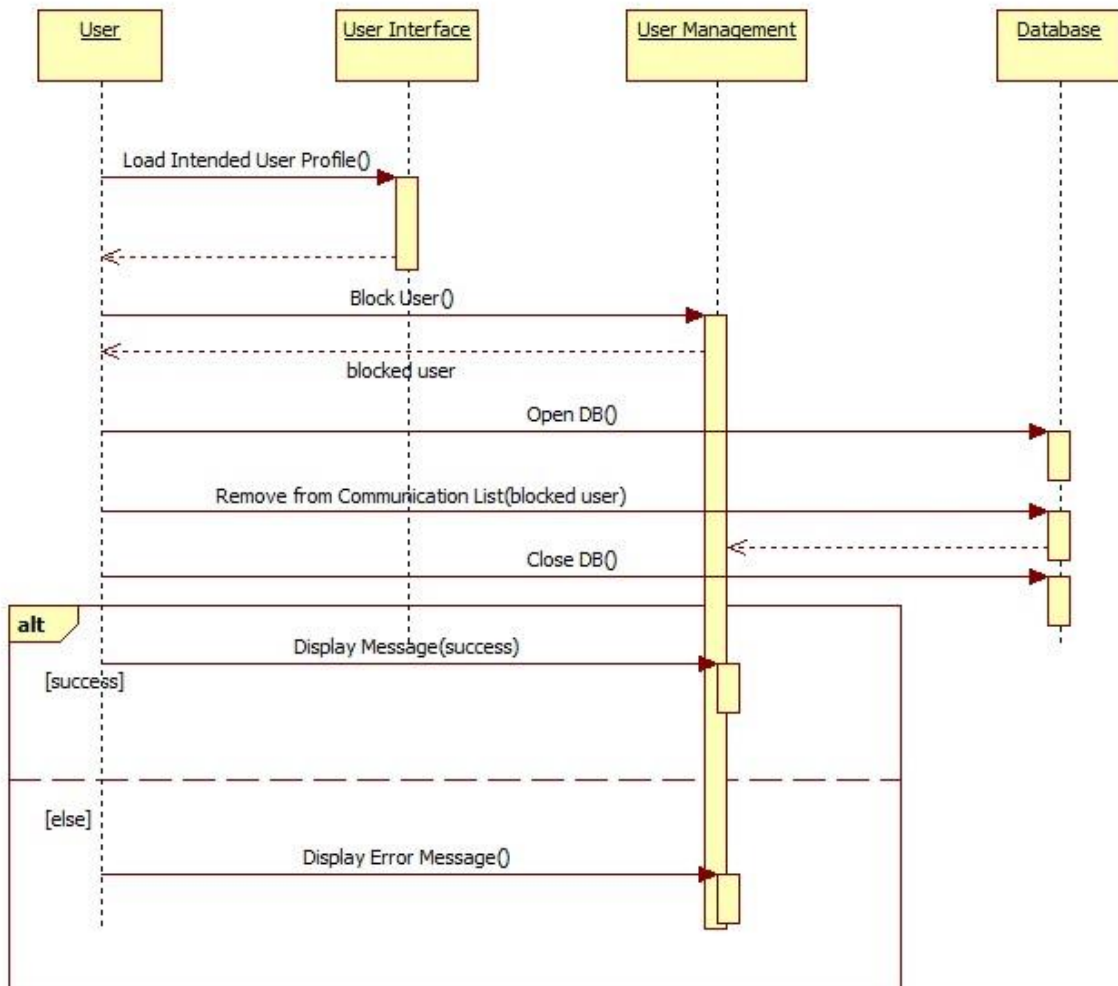


Figure 39. Visualization of Block User Sequence Diagram

User can block another user by the system. To accomplish this, first user shall enter the intended user's profile page and click block user button. After this point, system removes the blocked user from communication list of a user. When the operation is successfully executed, user will lose contact with the blocked user.

DESIGN VIEWPOINTS

5.6.11. RATE USER

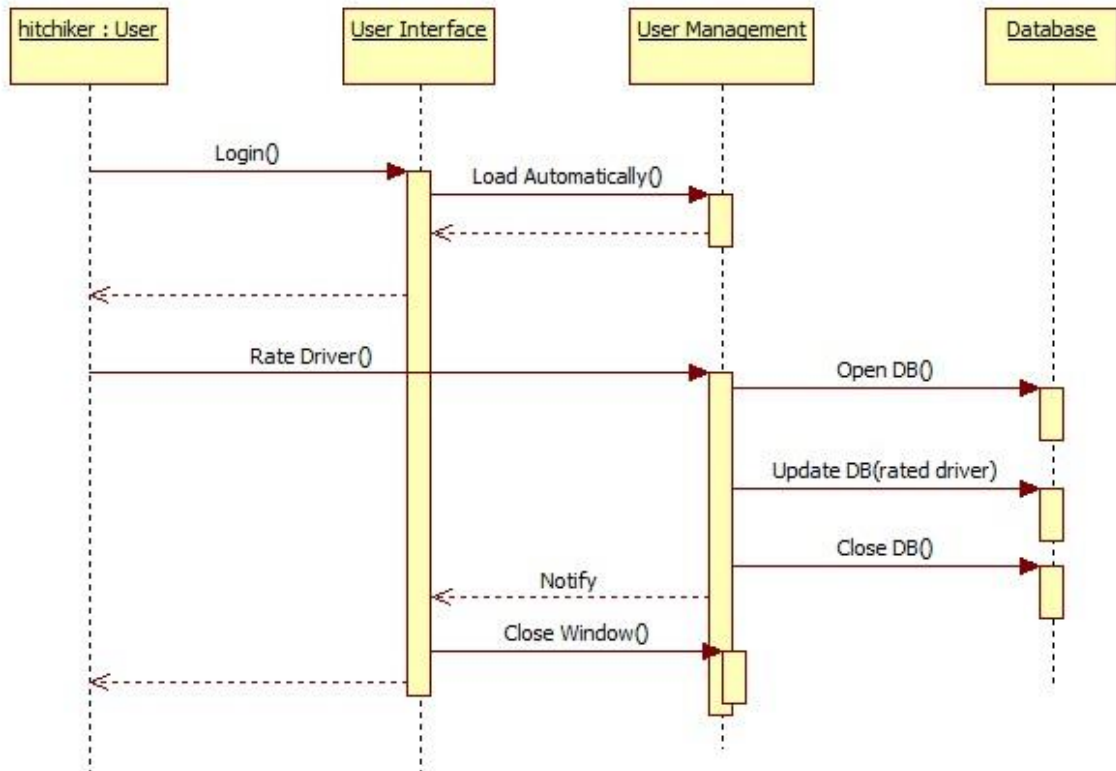


Figure 40. Visualization of Rate User Sequence Diagram

Hitchhikers can rate drivers by the system. When the hitchhiker logs in to the system after the transportation, system loads a popup window related to driver specific attributes. And hitchhiker can rate driver by clicking the star icon on the popup window.

DESIGN VIEWPOINTS

5.6.12. CHANGE LANGUAGE

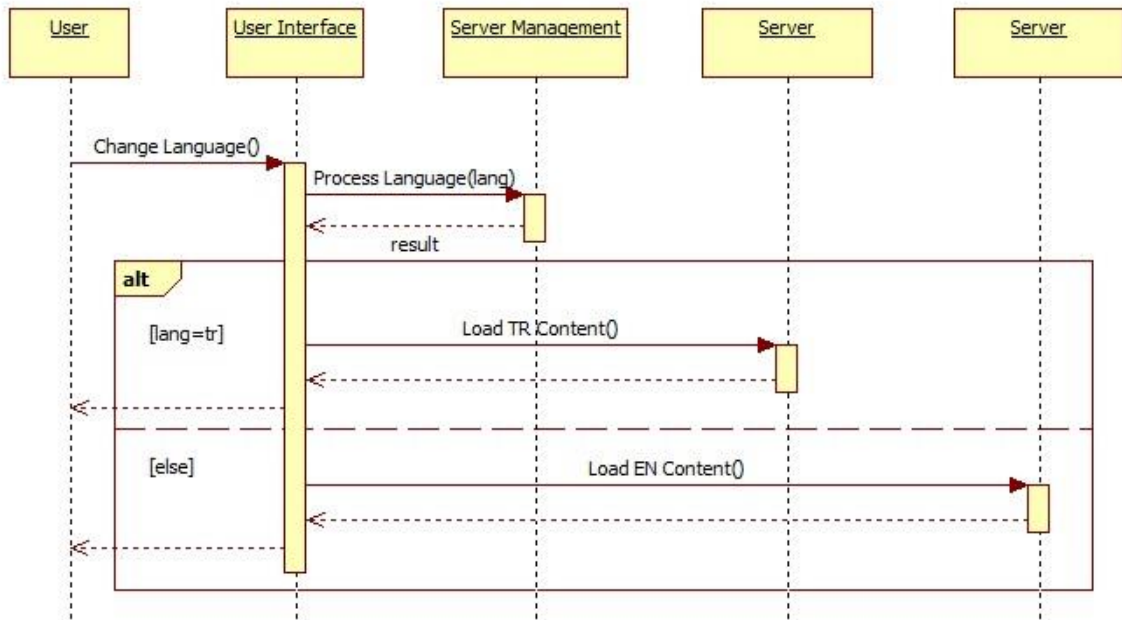


Figure 41. Visualization of Change Language Sequence Diagram

System supports Turkish and English languages. There is a button top of the main page interface. By clicking this button, user can change language of a system and system loads related page content from the server and user can view system with desired language.

5.7. STATE DYNAMIC VIEWPOINTS

There are seven states in our software. Each of them is a webpage actually. These are main page, sign up page, transportation searching page, user profile page, transportation adding page, transportation page and transportation request page.

DESIGN VIEWPOINTS

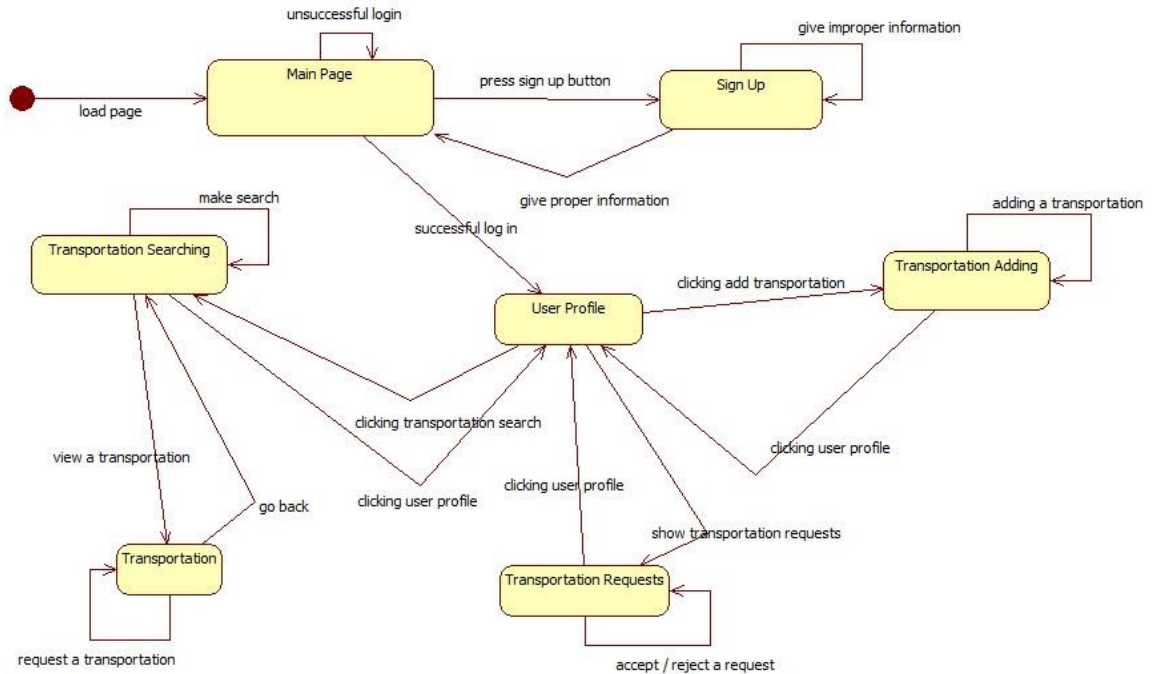


Figure 42. Visualization of State Chart Diagram

5.7.1. MAIN PAGE

When a user enters the website, main page will be displayed. Some visual information about the website is included in the main page. However, main importance is that this page is visible to everyone and allows users to register themselves. When the user clicks the sign up button, he/she is directed to the sign up page. If the user is already registered, he/she can log in by entering his/her username and password to the input widgets in the page.

5.7.2. SIGN UP PAGE

Every user has to be logged in in order to add/find transportations. Because, when someone takes the other in his/her car, or vice versa, he/she needs to know who he/she is, what his/her job is, how old he/she is etc. Therefore, everyone needs to sign up once in order to use the page. They are required to enter their name, surname, e-mail, age, job, phone and gender. Also, they need to specify their usernames and passwords. All usernames must be unique. When a user enters valid information, he/she is redirected to the main page and an activation code is sent to his/her e-mail address. If the information user entered is not valid, sign up page is reloaded and proper warnings are given to user.

5.7.3. USER PROFILE PAGE

User profile shows the user his user information and let him/her to switch to other pages. When user clicks to add transportation button, he/she is redirected to that page. Also, the user can click search transportation button in order to go that page. Moreover, the user will be directed to transportation requests page if he/she clicks show transportation request(s) button.

5.7.4. TRANSPORTATION ADDING PAGE

This page allows users to add transportations of their choice. They specify route, time and empty seat numbers. When they click add button in that page, entered transportation is added to the system and the page is reloaded. When the user press go to user profile button, he/she is redirected to user profile page.

5.7.5. SEARCH TRANSPORTATION PAGE

In this page, users search for transportations that are most suitable for them. They specify the criteria, and then, information about related transformations are displayed. If the user wants to see detailed information about a transportation, he/she may click to the transportation and he/she will be directed to transportation page.

5.7.6. TRANSPORTATION PAGE

In this page, detailed information about the transportation that user picked is displayed. If the user decides that he/she wants to be involved in the transportation, he/she can press request transportation button. Then, a request is sent to the driver of that transportation and the page is reloaded.

5.7.7. TRANSPORTATION REQUEST PAGE

When a user wants to get involved in a transportations, a request is sent to the driver of the transportation. In transportation requests page, he/she can press accept or reject buttons in order to accept/reject the request and stay in the page. If he/she presses user profile button, he/she is redirected to user profile page.

PLANNING OF THE PROJECT

6. PLANNING OF THE PROJECT

The Planning of the project is listed below:

PLANNING TABLE

DATE	TASK
16.12.2013	Sign Up and Sign In Modules should be completed.
06.01.2014	Add Transportation Module should be completed.
27.01.2014	Search Transportation and User Profile Modules should be completed.
17.03.2014	Rating Module should be completed.
21.04.2014	Messaging Module should be completed.
28.04.2014	Compatibility of whole system should be completed.
12.05.2014	Unit Testing should be done.
26.05.2014	System Testing should be done.
02.06.2014	System Release.