FMWay Işık

Requirements Specification and Analysis

V1.0
08.03.2020

Ersin ÇEBİ
Umutcan Neşet ÖLÇER
Çağatay DEMİRCAN
Emre Can RUA

Prepared for
SE301 Software Engineering

IŞIK UNIVERSITY
COMPUTER
SCIENCE AND
ENGINEERING

Table of Contents

1. Introduction 1

1.1. Purpose of the System 1
1.2. Scope of the System 1
1.3. Objectives and Success Criteria of the Project 1
1.4. Definitions, Acronyms, and Abbreviations 1
1.5. Overview 1

2. Current System 2
3. Proposed System 2

3.1. Overview 2
3.2. Functional Requirements 2
3.3. Nonfunctional Requirements 3

Usability ..... **Error! Bookmark not defined.**
Reliability ..... **Error! Bookmark not defined.**
Performance ..... **Error! Bookmark not defined.**
Supportability ..... **Error! Bookmark not defined.**
Implementation ..... **Error! Bookmark not defined.**
Interface ..... **Error! Bookmark not defined.**
Packaging ..... **Error! Bookmark not defined.**
Legal ..... **Error! Bookmark not defined.**

3.4. System Models 3

Scenarios ..... **Error! Bookmark not defined.**
Use case model ..... **Error! Bookmark not defined.**
Object model ..... **Error! Bookmark not defined.**
Dynamic model ..... **Error! Bookmark not defined.**
User interface—navigational paths and screen mock-ups. **Error! Bookmark not defined.**

3.5. Project Schedule 34

4. Glossary 35
5. References 35

<FMWay Işık>

REQUIREMENTS ANALYSIS DOCUMENT[1]

1. Introduction

1.1. Purpose of the System

FMWay Işık project is a transportation application exclusively for Işık University students. The main goal of the project is to provide a mobile application for the students who tries to reach to the campus or tries to travel to the city center from the campus. Users who get approved from the system can create trips as drivers, and the passengers can join them from the application, hence they can share the trip together.

1.2. Scope of the System

FMWay Işık application targets Işık University students. The users can only sign up with their Işık e-mails. The application has passenger and driver sides for the users. Every user signs up as passenger in the beginning, the users who have appropriate files can upload the files to the system and after the confirmation, they can be drivers as well.

FMWay Işık is a ride-sharing application. Drivers can add, update,delete trips, give feedback of the passenger.

Passengers can search, select trips, cancel the participation of a specific trip, give feedback of the driver, make the payment and apply to be a driver.

Admin can ban(delete) user's profiles, approve users' driver roles, and provide support for inconvenients of users.

1.3. Objectives and Success Criteria of the Project

- To provide reliable, efficient, lossless data.
- Well association between platform and database design.
- The general design of system in order to have fast, efficient application.
- The system has implementations that are understandable, clear and efficient.
- The system should be used by the people who are related with the application.
- The system should guarantee every user's protection of information of their data.

1.4. Definitions, Acronyms, and Abbreviations

- View is a visual representation of a model.
- GUI is graphical user interface.
- DB is short version of database term.
- Instructor is an actor an FMWay Işık and approved by admin
- Passenger, driver, visitor and registered user are an actor in a system.

1.5. Overview

- Rest of the RAD contains non-functional (includes usability, reliability, performance, supportability, implementation, interface, operational, packaging, and legal

requirements) and functional requirements (includes high-level functionality of the system).

- System models are given. Scenarios are inside of system model section. Scenarios are telling us about details of functional requirements. Use case models, object model, dynamic model and user interface view (mockup) are the parts of system model section.

2. Current System

The application is providing travel opportunity exchange of money to registered users. In addition, the system should guarantee every user's protection of information of their data.

3. Proposed System

We are going to try developing an hitchhiking application, exchange of money between users. This application is a user friendly application and useful for everyone.

3.1. Overview

The system will be including three actors which are admin, passenger, driver. Every driver is also can be passenger.

3.2. Functional Requirements

Functional requirements for the passenger:

- Sign up
- Login
- Edit profile
- Logout
- Search Trip
- Apply for to be a driver
- Confirm trip
- Cancel the participation
- Finish the trip
- Payment
- Driver feedback

Functional requirements for the driver(Except the passenger features):

- Add trip
- Edit trip
- Delete trip
- Passenger feedback

Functional requirements for the admin:

<FMWay Işık>

- Edit user profiles
- Ban(Delete) user profiles
- Approve drive role for users
- Provide support for users

Nonfunctional Requirements

Usability

Application should be easily available for all users. The admin panel must also be available on phone. The home page must be the same for all users. Each page must have the same body structure. Actors' pages should be understandable.

Reliability

All members must have secure access to the application. Visitors should not do anything. Logging to the application should be provided with unique e-mails and passwords that are appropriate for password criteria.

Performance

Application should be shown in mobile phones and tablets. Application should be running in more than one device simultaneously and access should be guaranteed. The application is going to be a dynamic content, so there should not be complicated queries in back end to not decrease performance.

Supportability

The system should be managed by admin. Developer will be responsible to provide continuance, compatibility and testability for the created program.

Implementation

Design of database will be implemented by using Parse Database(Back4App).

Android studio will be used as coding IDE.

Java 8 will be used.

Interface

The system should not interact with any existing system. The system should be able to be used by a user. The user should be connected to the network to use the features of the system.

Packaging

Admin should install the system. Also, the system is an application so the application should be uploaded on the server. There should be no time constraints on the installation. System's all steps as a package are given within GitHub.

Legal

Project's all contents are protected by the law of copyright.

<FMWay Işık>

3.3. System Models

Scenario name: Sign Up
Participant Actor Instances: Ersin: User
Flow of events: 1.) Ersin is at login page. He wants to login to the system but he has not any account for FMWay Işık. 2.) He clicks register button. He enters information of his name, e-mail and password for this application. He clicks register button and; i.) If he enters wrong/used e-mail address, he sees an error message and he returns to the register page. ii.) If he enters a used user name or selects bad user name, user sees error message and returns to the register page. iii.) If he enters short password or does not enter any passwords, he sees an error message and returns to the register page. **3.)** He registers.

Scenario name: Login
Participating Actor : Ersin : User
Flow of Events: 1.) He is at login page screen. He enters e-mail and password and he clicks on the login button. 2.) User can see 3 options; i.) If user leaves any blank area that is email/user name or password, user sees an error message that is saying “Username or password is wrong”. ii.) If user enters wrong email/user name or password, user sees a message saying “Username or password is wrong”. iii.) If user doesn’t make any mistakes, user logins to system. 3.) He clicks log out button and exits his account and he returns to login page.

Scenario name: Edit Trip
Participant actor instances: Ersin: User
Flow of events: 1.) He logins at login page screen and He goes to home page. 2.) He clicks the profile button. 3.) He sees the current trips button and opens the trips page. He clicks the edit button on list, so he makes the desired changes. 4.) He clicks on “Save” button and sees changes are saved, then he returns to profile page. 5.) He clicks home page button, so he can click on log out button. She logs out of system.

Scenario name: Delete Trip
Participant actor instances: Ersin: User
Flow of events: 1) He logins at login page screen and He goes to home page. 2) He clicks the profile button. 3) He sees the current trips button and opens the trips page. 4) He clicks on “Delete” button and sees dialog box clicks ‘confirm’, then he returns to profile page. 5) He clicks home page button, so he can click on log out button. She logs out of system.

<FMWay Işık>

Scenario Name: Search Trip
Participant Actor instances: Deniz(User)
Flow of events: 1) She logins to the application and sees the homepage with some trips listed. 2) On the list, she doesn't see the right trip, so she clicked on the search button. 3) She moves to the search page, enters the needed information such as starting point or destination. 4)She clicks "Show" button, a) If the wanted trips exist, she sees the matching trips. b) If the wanted trips don't exist, the application says " No match found!" 5)She chooses one and continues to approve that trip.

Scenario Name: Adding Driver Role
Participant Actor instances: İlkay(User)
Flow of events: 1)He logins to the application and sees the homepage. 2)Because of every user sign up as passengers in the beginning, his profile is still a passenger. He wants to be a driver also. 3)He clicks to his profile, moves to the profile page. Then he clicks edit profile. 4)On edit page, he moves to "Driver" section, when he moves that, he sees the needed information for to be a driver. 5)He fills the information and upload driver-car licenses. a)If he doesn't fill the information, the application sends a feedback "This area cannot be empty!" 6)He clicks to "Send" button, then he waits for the approval. 7)After approval, he becomes a driver.

<FMWay Işık>

Scenario Name: Approving Driver Role
Participant Actor instances: Umut(Admin)
Flow of events: 1)Umut logins to the application and sees the homepage. 2)Umut clicks “Driver requests” button and moves to that page. 3)On that page, he sees the names who requests to be a driver. 4)He clicks one name, license pictures appear. 5)Approve and decline options appear. a) If he clicks “Decline” nothing changes on passenger’s profile and the passenger cannot be a driver. b) If he clikcs “Approve”, user profile gets edited on database and the selected user becomes driver also. 6)He returns to the driver requests list.

Scenario Name: Cancelling Trip
Participant Actor instances: Simay(User)
Flow of events: 1)Simay enters the application and sees the homepage. 2)She selected and confirmed a trip earlier, so the trip appears on the homepage as “Upcoming trips”. 3)She clicks on the trip and moves to the “Trip Details” page. 4)She cannot joint to the trip, hence she wants to cancel her participation. Therefore she clicks “Cancel the trip” button. 5)There is a feedback says “Are you sure?”, she clicks “Yes” button. 6)Application cancels her participation to the trip and notifies the driver.

Scenario Name: User Feedback
Participant Actor instances: Hera(Passenger), Venüs(Driver)
Flow of events: 1)Hera and Venüs completes the trip. 2)Hera enters the application and sees the specific trip that she attended to, on “Upcoming Trips” section. 3)She enters “Trip Details” page and then clicks “Trip completed” button. 4) The driver feedback page appears, she grades Venüs out of 5. 1)Venüs enters the application and sees the specific trip. 2) She enters “Trip Details” page and then clicks “Trip completed” button. 3)The passenger feedback page appears, she grades Hera with “Tick” or “Cross” shapes.

<FMWay Işık>

Scenario name: Edit Profile
Participant actor instances: Emre: User
Flow of events: 6.) He logins at login page screen and he goes to home page. 7.) He clicks the profile button. 8.) He sees the information of his own. He clicks the edit button, so he can change his user e-mail, password, first name, last name and gender. 9.) He clicks on “Save” button and sees changes are saved (“It is edited successfully”), then he returns to profile page. 10.) He clicks home page button, so he can click on log out button. He logs out of system.

Scenario name: Log Out
Participant actor instances: Emre: User
Flow of events: 1.) He is at login page screen and he enters email/username and password and clicks “Log In” button. 2.) He clicks log out button, and he exits system in addition he returns to the login page.

Scenario name: Accepting Trip
Participant actor instances: Emre: Passenger
Flow of events: 1.) Emre is on the login page screen, he has entered the program by filling in the e-mail / username and password section. 2.) Driver / trip search is completed from the trip search section. He matches the driver. He sees the information of the drive it matches. 3.) If Emre wants to accept the trip, he presses the ‘Accept’ button. 4.) The passenger with the trip is directed to the trip tracking page containing the map on which he appears.

Scenario name: Altering trip (EDITLEME OLAYI DEDİK GÜZERGAH VEYA SAAT)
Participant actor instances: Emre: Passenger
Flow of events: 1.) The passenger has accepted the trip. 2.) Emre can change the position of the trip if the vehicle is cruising. In addition, while waiting for the vehicle, it can change the location or time of the destination. 3.) If Emre wants to accept the question 'Are you sure' about this, he approves the change by pressing the 'Accept' button. 4.) Emre is directed to the homepage. // The passenger with the trip is directed to the trip tracking page containing the map on which he appears.

Scenario name: User Banning
Participant actor instances: Emre: Admin
Flow of events: 11.) The user who is admin, logins at login page screen and goes to admin's home page // control panel. 12.) He clicks the page where he can see all passengers and drivers' names // information. 13.) He chooses the user he wants to be banned and see it's information. 14.) He clicks on "Banned" button on the page he is using. 15.) He sees the question "Are you sure to ban this user?" 16.) Admin clicks the "Yes" button to ban the user he chose. 17.) He returns to admin's home page where he can click on log out button. He logs out of system.

<FMWay Işık>

Scenario name: Payment
Participant actor instances: Emre: User
Flow of events: 3.) The passenger logs in and sees the home page. 4.) S/He completes the trip section successfully. 5.) S/He approves the trip information and goes to the payment section. 6.) Passenger decides to make the payment with “Google Pay” 7.) After the passenger approves it, the money goes directly to the payment page // wallet. 8.) The payment page // wallet sends the payment to the driver.

<FMWay Işık>

Scenario name: Application Support Center
Participant actor instances: Ersin: User
Flow of events: 18.) He logins at login page screen and He goes to home page. 19.) He clicks the profile button. 20.) He sees the “support center” button and clicks. 21.) He fills the required fields. 22.) He clicks on “Send” button and sees “Your message sent”, then he returns to profile page. 23.) He clicks home page button, so he can click on log out button. He logs out of system.

Scenario name: Send Message
Participant actor instances: Ersin: User
Flow of events: 1.) He logins at login page screen and He goes to home page. 2.) He clicks the profile button. 3.) He opens his trips page. 4.) Clicks the desired trip(next available trip of his) and opens it. 5.) He clicks “send message” button and fill the required fields. 6.) He clicks on “Send” button and sees “Your message sent”, then he returns to profile page. 7.) He clicks home page button, so he can click on log out button. He logs out of system.

<FMWay Işık>

Scenario name: Payment
Participant actor instances: Emre: User
Flow of events: 9.) Emre logins the system. 10.) Emre clicks payment pages. 11.) Emre clicks add money to balance button. 12.) Emre enters how much money add. 13.) Emre enters his credit card information. 14.) Emre accepts the information and payment. 15.) System send notification to Emre.

Use case name: Payment
Participant actors: The Passenger
Flow of events: 1.) User clicks at Payment Page. 2.) User clicks add money to balance button. 3.) User enters information of how much money that s/he wants to add and credit card. 4.) User accepts the information 5.) System sends notification to user.
Entry Condition: The passenger wants to add money to balance.
Exit Condition: Payment success and money add to balance.
Quality Requirements: 3D secure option.

Use case name: Sing up
Participant actors: Potential User
Flow of events: 1.) Visitor opens application. 2.) Visitor wants to registered in FMWay, so Registered User clicks Register

button.
3.) Visitor enters e-mail, first name, last name, password.
4.) Visitor clicks Register button and returns login page.

Entry Condition: Visitor is not login.

Exit Condition: Visitor wants to exit in the system.

Quality Requirements:
If there is login, visiter dont acces Registered Page.
If This e-mail used before, no permission to register.
If user name used before, no permision to register.

Use case name: Login

Participant actors: User

Flow of events:
1. User opens application.
2. System responds by displaying the login screen of the application.
3. User enters his/hers login info e-mail and password to the login fields.
4. User clicks "Login" button.
5. If the username and password match with the e-mail and password which is stored in database, users logins.
6. System fetches the dashboard of user.

Entry Condition: User visits the application.

Exit Condition: USer clicks "Login" button and entered information should be correct.

Quality Requirements:
If users leaves one or more fields empty, system displays a warning message, like "This area cannot be empty."
If the information's checked from database are not true, system displays a warning message, like "Wrong username or password, please retry."

Use case name: Edit Trip
Participant actors: User
Flow of events: 1.) User clicks edit trip button. 2.) User makes desired changes on a desired trip.
Entry Condition: User must be login**.**
Quality Requirements: User cant leave break any area.

Use case name: Delete Trip
Participant actors: User
Flow of events: **1)** User clicks delete trip button on desired trip . **2)** System respondes with a dialog box. **3)** According to dialog box system deletes/keeps the trip.
Entry Condition: User must be login**.**
Exit Condition: User declines the dialog box.
Quality Requirements: User cant leave break any area.

Use Case Name	**Search Trip**
Participating Actor	Registered User
Entry Condition	User must be logged in.
Flow of Events	1)User clicks “Search Trip” Button. 2)User moves to “Search Trip” Page. 3)User fills the needed information about wanted trip. 4)User clicks “Show” button. 5)The application shows the filtered result.
Exit Condition	User sees the filtered result with related to his/her needs.
Quality Requirements	• If the user leaves one or more fields empty on needed areas, system displays a warning message such as "This area cannot be empty.” • If there isn’t any matched trip, the application displays a warning message such as “No match found!”

Use Case Name	**Adding Driver Role**
Participating Actor	Registered User
Entry Condition	User must be logged in.
Flow of Events	1)User clicks “Profile” button. 2)User sees the details of his/her profile. 3)User clicks “ Edit Profile” button. 4)User moves to “Driver” section. 5)User fills the needed information ,uploads his/her licenses. 6)User clicks “Send” button. 7)User waits for the approval. 8)After approval, user gets logged out and gets log in again. 9)User has the option of “Add Trip”
Exit Condition	User becomes a driver.
Quality Requirements	• If the user leaves one or more fields empty on needed areas, system displays a warning message such as "This area cannot be empty.” • If the needed information is not accurate or true, the user cannot be a driver on the system.

Use Case Name	**Approving Driver Role**
Participating Actor	Admin
Entry Condition	Admin must be logged in.
Flow of Events	1)Admin enters the homepage. 2)Admin clicks “Driver requests” button. 3)Admin sees the list of users who uploads their licenses. 4)Admin chooses one and checks the licenses. 5)Admin clicks approve or decline.
Exit Condition	User becomes a driver.
Quality Requirements	• If admin clicks approve button, the user profile gets edited and passenger also becomes driver. • If admin clicks decline button, the user profile doesn’t change and the user cannot be a driver.

Use Case Name	**Cancelling Trip**
Participating Actor	Registered User(as Passenger and as Driver)
Entry Condition	User must have chosen a trip earlier.

<FMWay Işık>

Flow of Events	1)User clicks on the specific trip below the section “Upcoming Trips”. 2)User sees the details of the trip. 3)User clicks “Cancel the Trip” button. 4)User clicks “Yes” on the warning. 5)The driver gets notified.
Exit Condition	User cancels the trip.

<FMWay Işık>

Use Case Name	**User Feedback**
Participating Actor	Registered User(as Passenger and as Driver)
Entry Condition	The trip must be completed.
Flow of Events	1)After clicking "Trip Completed" button on trip details, the feedback of driver appears for the passenger and the feedback of passenger for the driver . 2)Passenger rates the driver out of 5. 3)Driver rates the passenger with the signs "Tick" or "Cross".
Exit Condition	The rates of both users reaches to the application database and the rates appear on both profiles.
Quality Requirements	• If the users don't want to rate, they can skip the feedback option.

Use case name: Edit Profile
Participant actors: Registered User
Flow of events: 3.) User clicks Edit Profile button. 4.) User wants to change first name, last name and e-mail. If user don't enter anything any area, Edit profile button is not active. 5.) User can change information and clicks Edit Profile button, so user information saved and redirect Profile Page.
Entry Condition: User must be login**.+**
Exit Condition: User approves the changes.
Quality Requirements: User cannot leave break any area.

Use case name: Log Out
Participant actors: Registered User
Flow of events: 6.) Registered User wants to exit in the system. 7.) Registered User clicks log out button. 8.) Registered User returns Login Page.
Entry Condition: Registered User must be login in the system.
Exit Condition: Registered User wants to exit system.
Quality Requirements: If there is not login, Registered User will does not do log out.

Use case name: Accepting Trip
Participant actors: Passenger
Flow of events: 1.) After the passenger has entered the required route and time information, he presses the trip search button. 2.) A list of trip suitable for the passenger's screen is displayed. The passenger chooses the one that suits him. 3.) The passenger with the trip is directed to the trip tracking page containing the map on which s/he appears.
Entry Condition: Passengers must have completed the required travel hours and logged in.
Exit Condition: Passenger accepts the suitable trip.
Quality Requirements: After the passenger starts the journey, if he / she wants to search for the choice of the

destination, the approval is taken.

Use case name: User Banning

Participant actors: Registered Admin

Flow of events:

6.) Admin is in admin's home page // control panel and s/he clicks on the list where all passengers and drivers' names//information are shown.
7.) Admin chooses on the user s/he wants to be banned.
8.) Admin clicks on the "ban" button that became active when s/he chose the user.
9.) The question "Are you sure?" appears on the control panel.
10.) Admin clicks on the "Yes" button to ban the user the chose from the list.
11.) The user gets banned. Admin returns to his/her home page // control panel.
12.) He clicks on the list of the passengers and drivers' name and s/he sees the new column of banned users as well.
13.) After s/he checks the list, he clicks on the home page // control panel button on the page.
14.) Admin returns to his home page // control panel.

10.) S/He logs out.

Entry Condition: User must be logged as an admin.

Exit Condition: Admin wants to ban a user.

Quality Requirements:
Admin can see the list of passengers and drivers' list to select the right user to ban.
Admin sees the "Are you sure?" warning to be sure.
S/He returns to the control panel to see detailed information.

Use case name: Altering trip
Participant actors: Passenger
Flow of events: 1.) The passenger has accepted the trip. 2.) While the passenger waits for the car, he wants to change the location of his destination. 3.) He changes his destination 4.) He confirms the changes by pressing the passenger 'Accept' section.
Entry Condition: Passenger must have confirmed the trip
Exit Condition: The passenger wants to change the time or route of the trip.
Quality Requirements: Passenger after making changes to the trip, he confirms his transactions by pressing 'Accept' under the question 'Are you sure?'

<FMWay Işık>

Use case name: Payment

Participant actors: The Passenger

Flow of events:

9.) The passenger logs in and sees the home page in the application.
10.) S/He fills the trip information and competes the section successfully
11.) S/He approves the information s/he has filled and goes directly to the payment section.
12.) The passenger decides to make the payment with “Google Pay”
13.) After approving it, the money s/he paid goes to the ‘fmway’s payment page // wallet.’ Which they record the payment details for the good of passengers.
14.) The payment page // wallet send the payment to the driver passenger needs.

Entry Condition: The passenger has to approve a trip.

Exit Condition: The passenger wants to take a trip and has to pay for it.

Quality Requirements:

The payment system is Google Pay which means the payments are protected by 256SSL, it’s safe for the passenger the use the application.

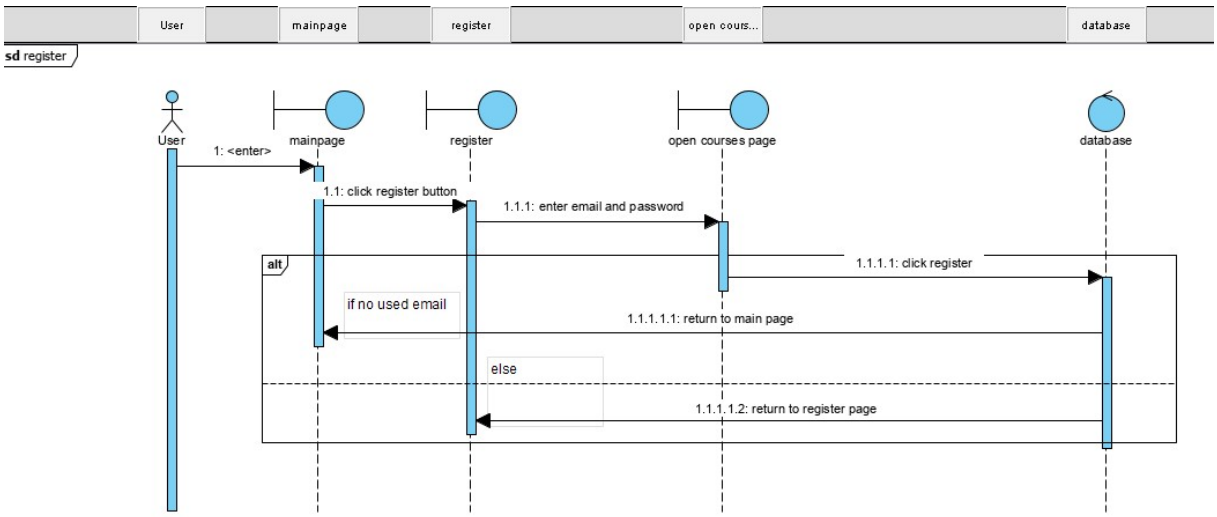
The payments are held by payment page // wallet.

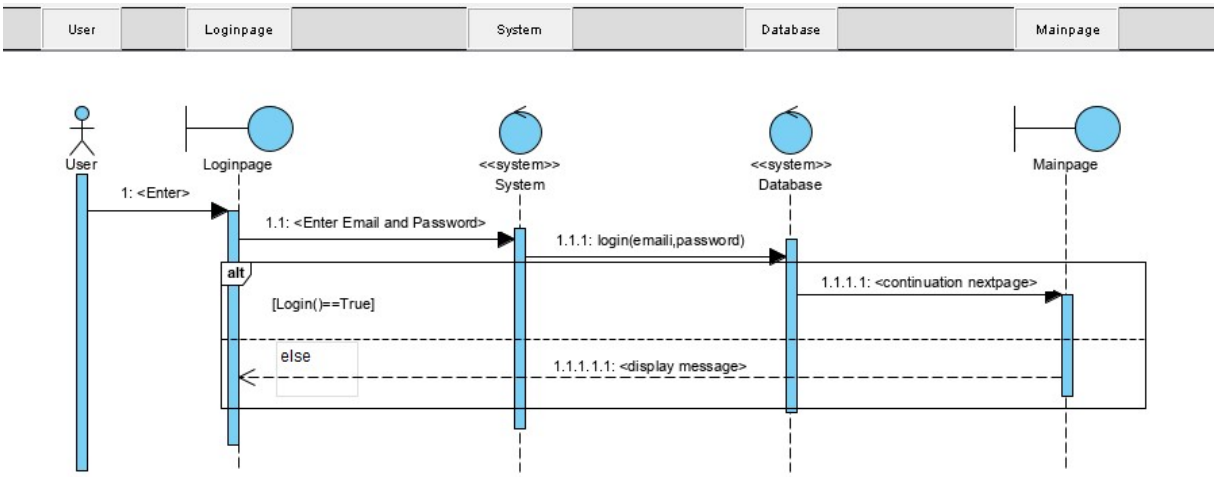
<FMWay Işık>

Use case name: Application Support Center
Participant actors: User
Flow of events: 1. User opens application. 2. User clicks the profile button. 3. User opens “support center” page. 4. User clicks on “Send” button. 5. Application responds with confirmation dialog.
Entry Condition: User visits the application.
Exit Condition: User clicks "Send" button all fields should be filled.

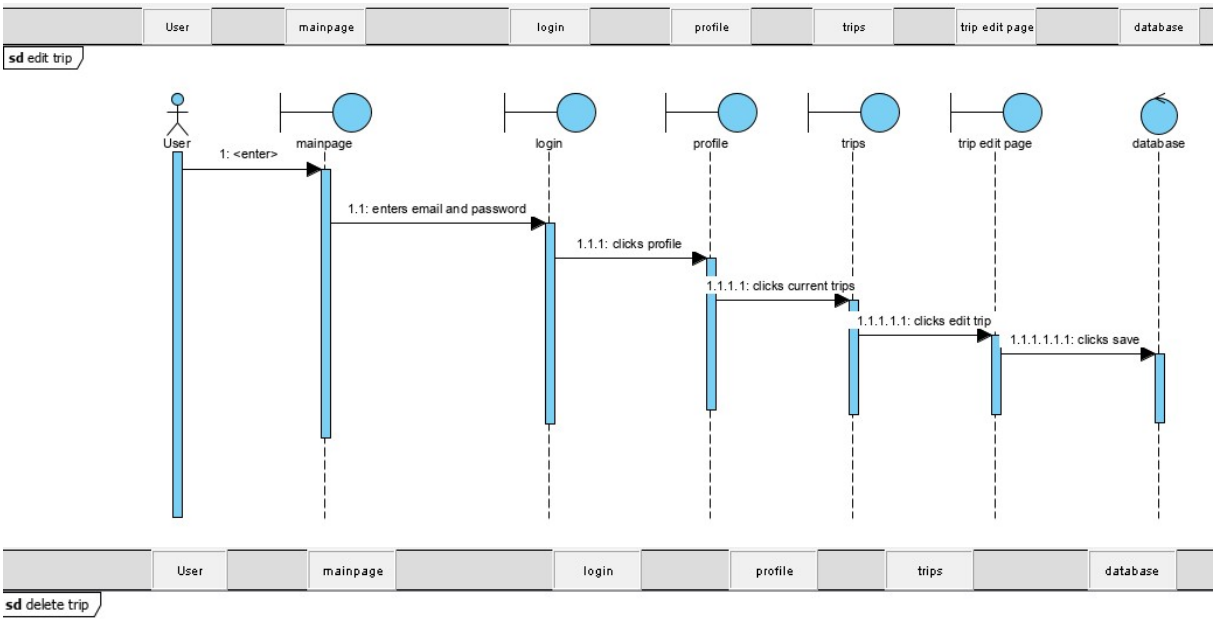
Use case name: Send Message
Participant actors: User
Flow of events: 1. User opens application. 2. User clicks the profile button. 3. User opens his/hers trips list page. 4. User opens a trip and clicks message dialog. 5. User clicks on “Send” button. 6. Application responds with confirmation dialog.
Entry Condition: User visits the application.
Exit Condition: User clicks "Send" button all fields should be filled.

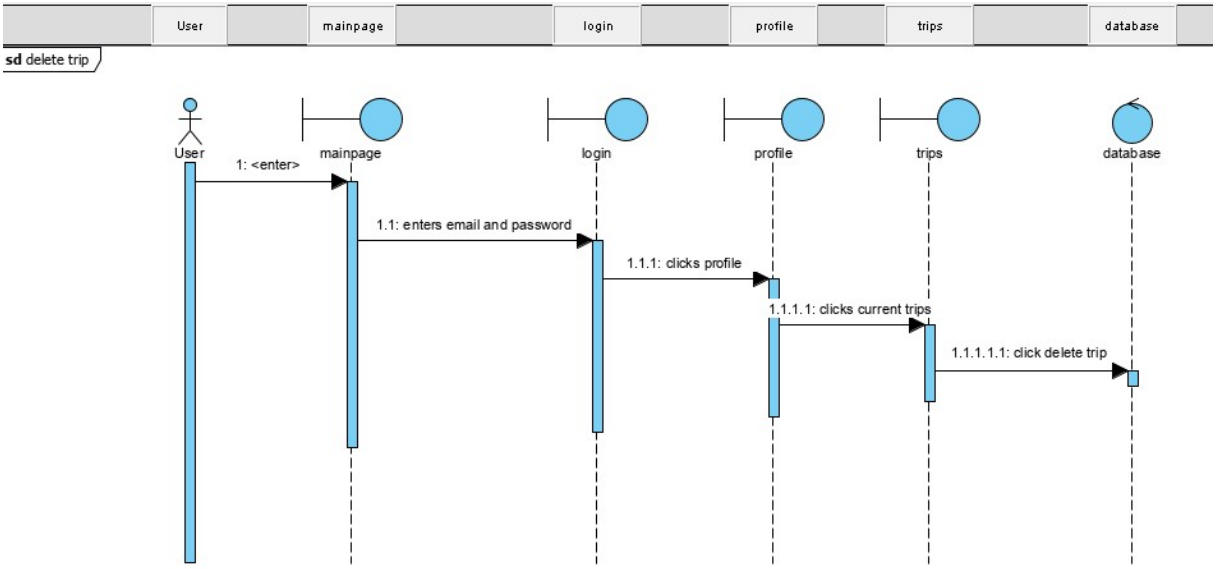
<FMWay Işık>


User
mainpage
register
open cours...
database
sd register
User
mainpage
register
open courses page
database
1: <enter>
1.1: click register button
1.1.1: enter email and password
alt
1.1.1.1: click register
if no used email
1.1.1.1.1: return to main page
else
1.1.1.1.2: return to register page

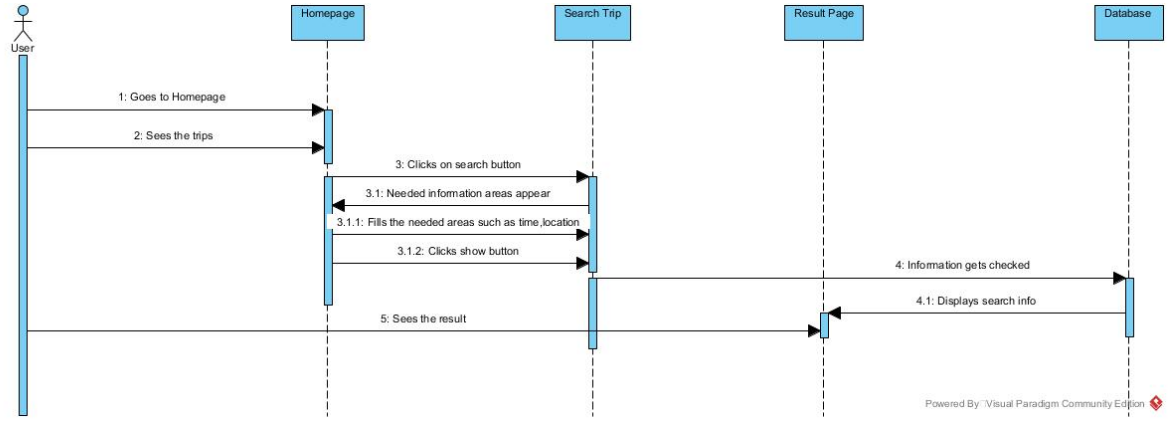

User
Loginpage
System
Database
Mainpage
User
Loginpage
<<system>>
System
<<system>>
Database
Mainpage
1: <Enter>
1.1: <Enter Email and Password>
1.1.1: login(emaili,password)
alt
[Login()==True]
1.1.1.1: <continuation nextpage>
else
1.1.1.1.1: <display message>

<FMWay Işık>



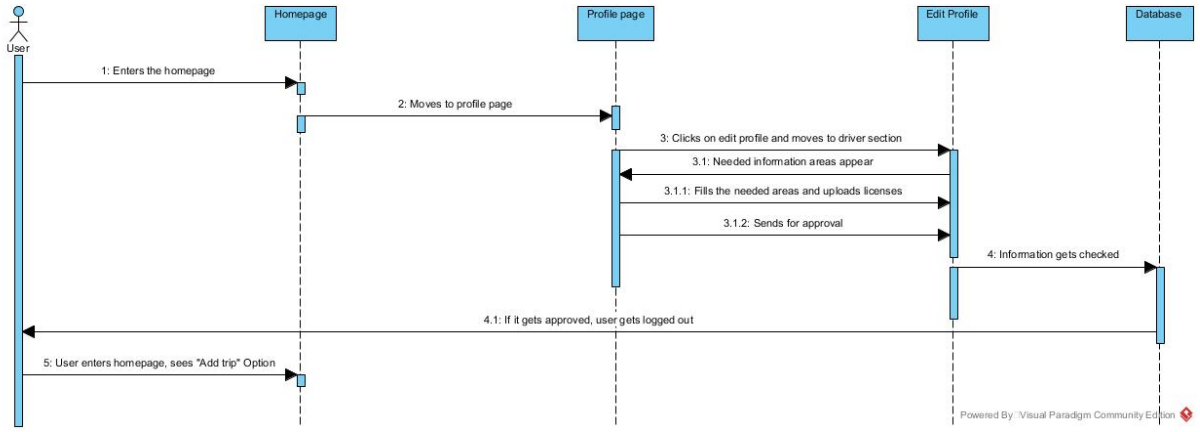


SEARCH TRIP

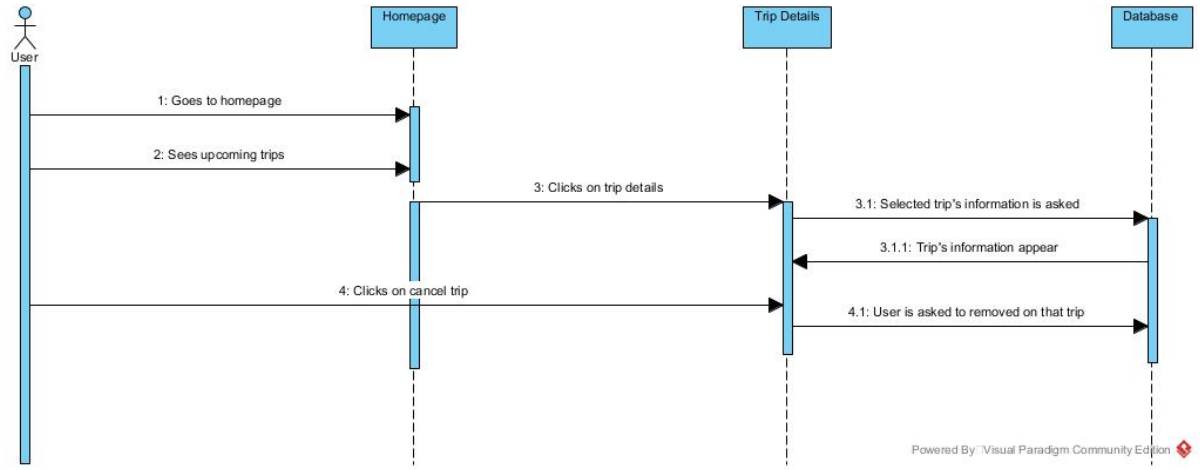


<FMWay Işık>

ADDING DRIVER ROLE

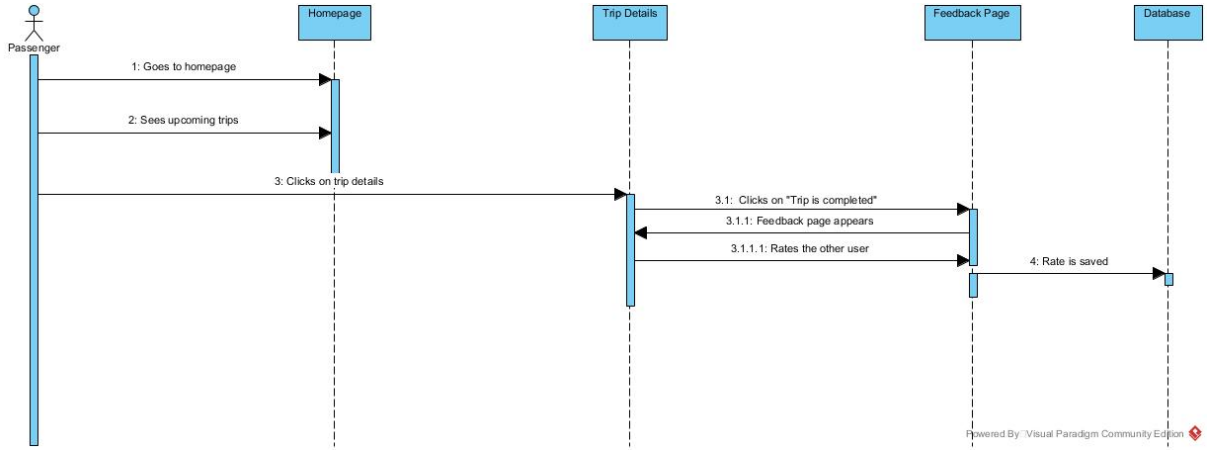


CANCELLING TRIP

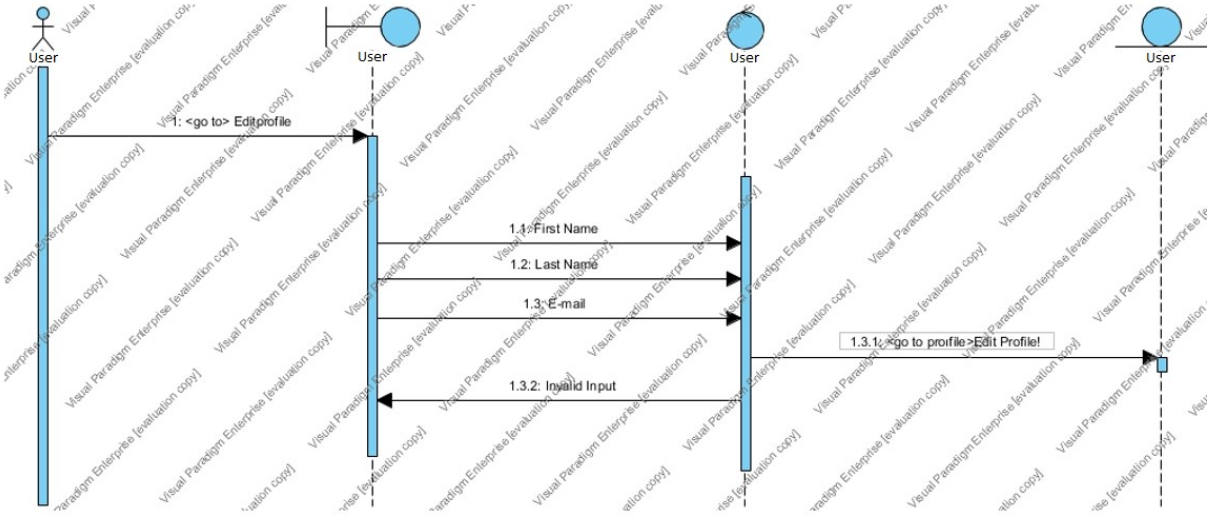


<FMWay Işık>

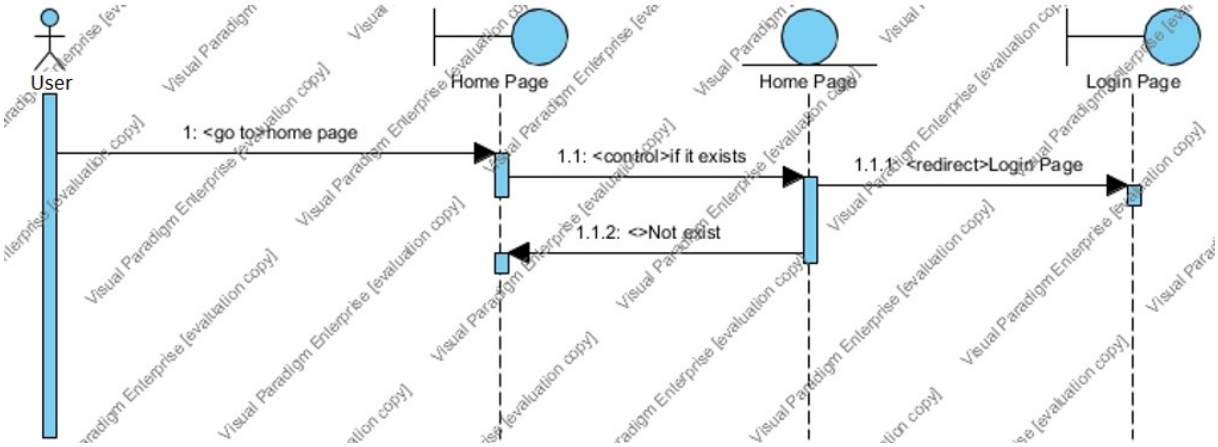
USER FEEDBACK



User Edit Profile

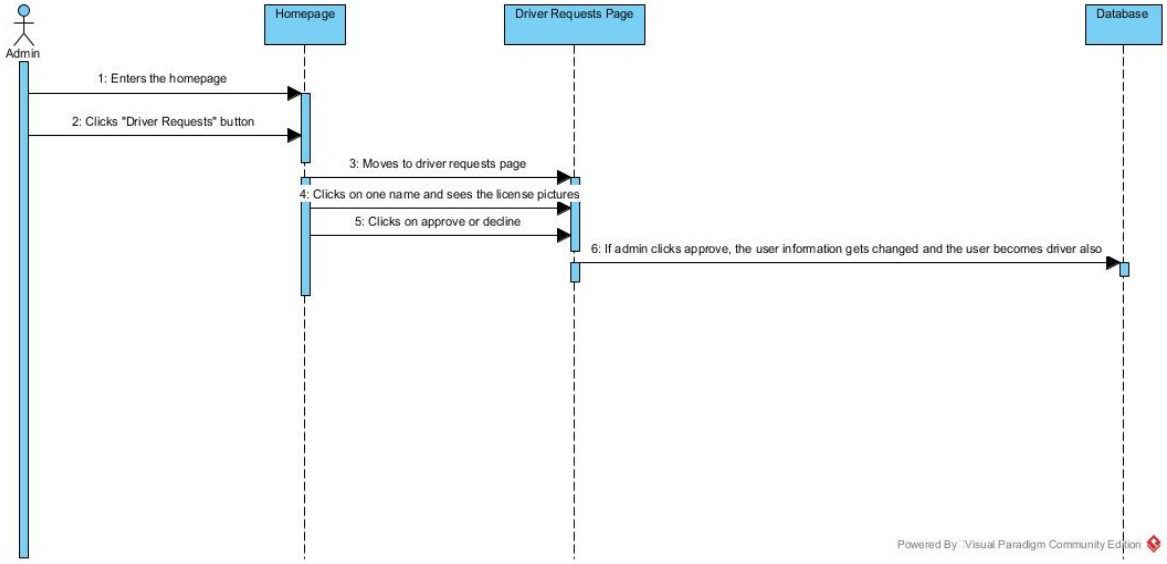


User Log Out

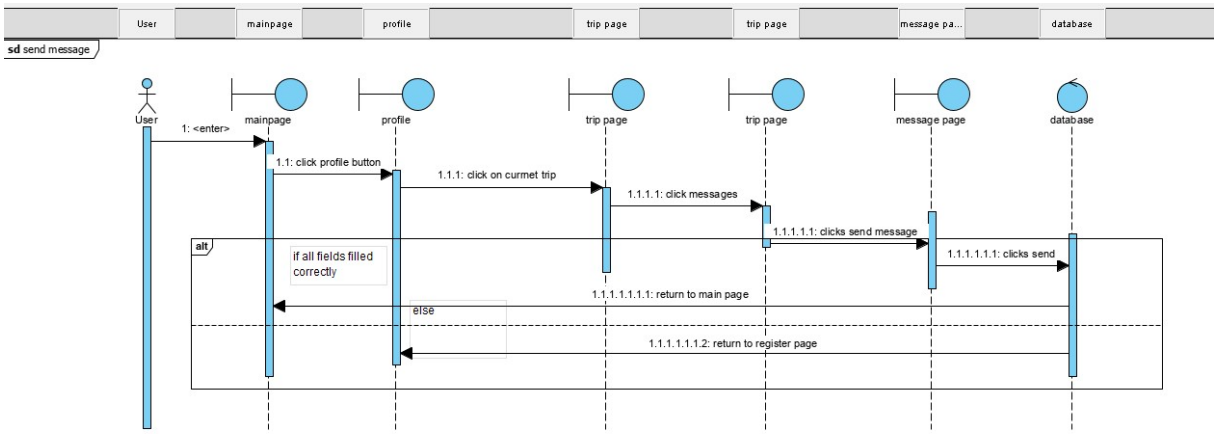


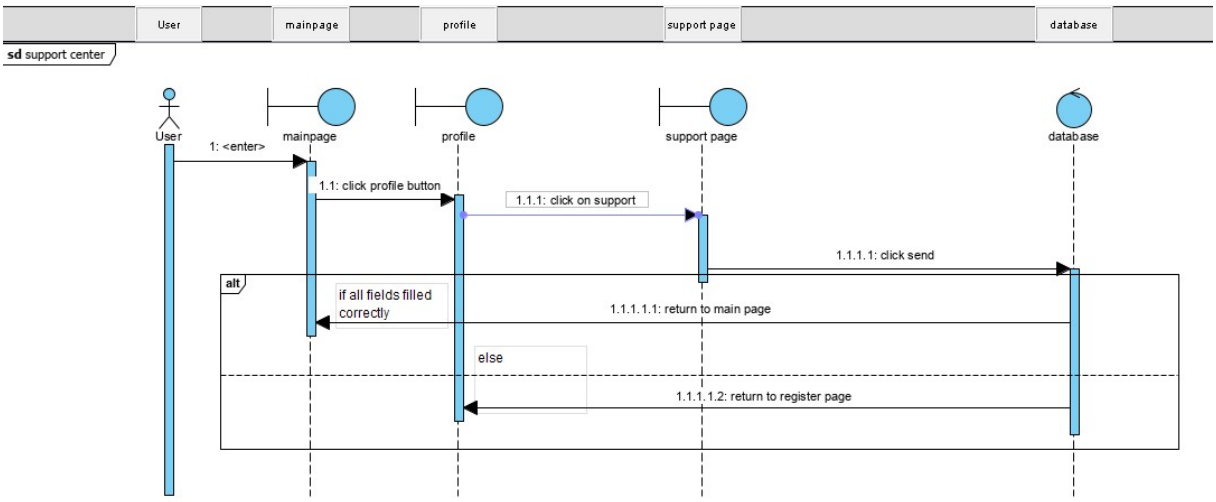
<FMWay Işık>

APPROVING DRIVER ROLE

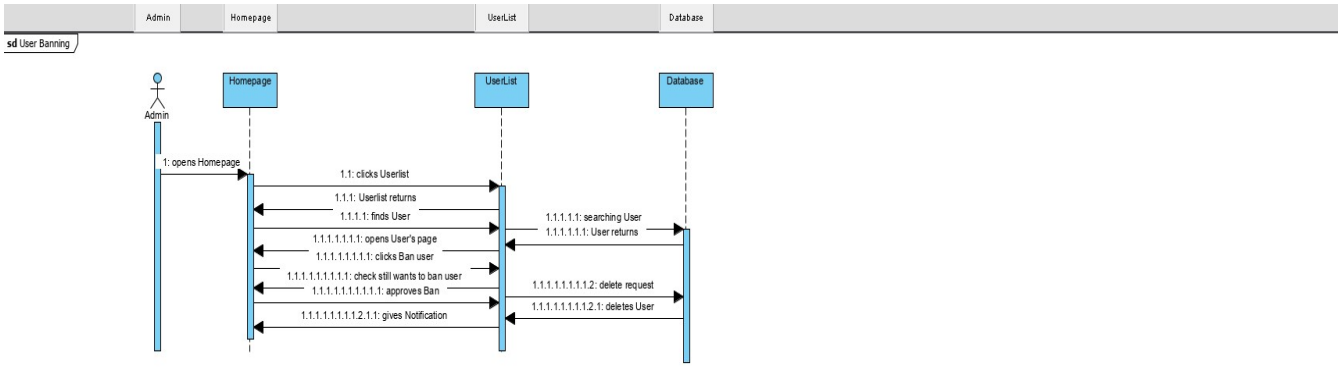


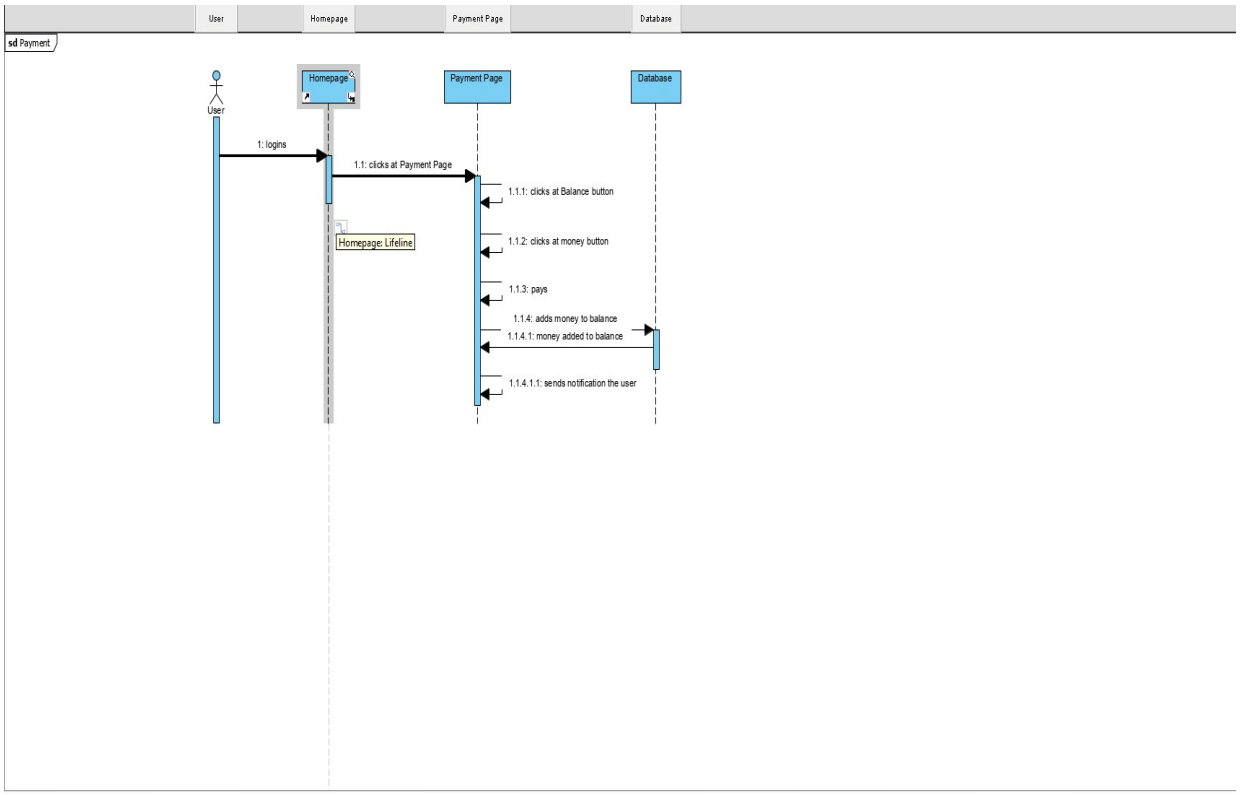
<FMWay Işık>


User
mainpage
profile
trip page
trip page
message pa...
database
sd send message
User
mainpage
profile
trip page
trip page
message page
database
1: <enter>
1.1: click profile button
1.1.1: click on curmet trip
1.1.1.1: click messages
1.1.1.1.1: clicks send message
1.1.1.1.1.1: clicks send
alt
if all fields filled correctly
1.1.1.1.1.1.1: return to main page
else
1.1.1.1.1.1.2: return to register page

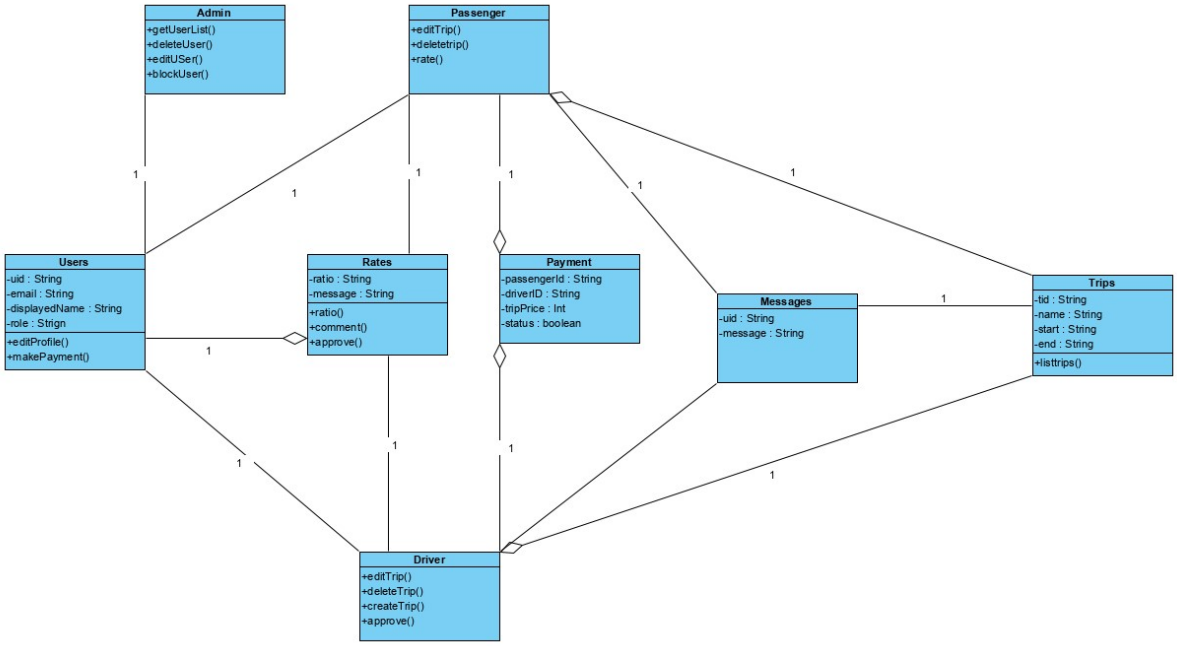

User
mainpage
profile
support page
database
sd support center
User
mainpage
profile
support page
database
1: <enter>
1.1: click profile button
1.1.1: click on support
1.1.1.1: click send
alt
if all fields filled correctly
1.1.1.1.1: return to main page
else
1.1.1.1.2: return to register page

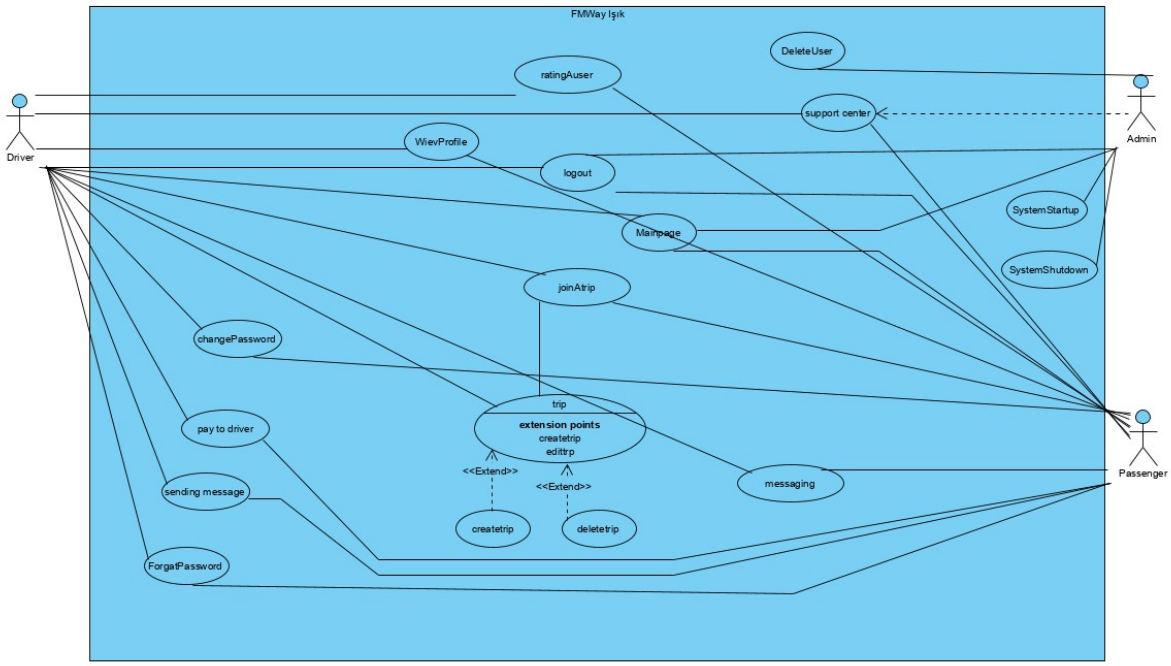
<FMWay Işık>


Admin
Homepage
UserList
Database
sd User Banning
Admin
Homepage
UserList
Database
1: opens Homepage
1.1: clicks Userlist
1.1.1: Userlist returns
1.1.1.1: finds User
1.1.1.1.1: searching User
1.1.1.1.1: User returns
1.1.1.1.1.1: opens User's page
1.1.1.1.1.1.1: clicks Ban user
1.1.1.1.1.1.1.1: check still wants to ban user
1.1.1.1.1.1.1.1.1: approves Ban
1.1.1.1.1.1.1.1.2: delete request
1.1.1.1.1.1.1.1.2.1: deletes User
1.1.1.1.1.1.1.2.1.1: gives Notification

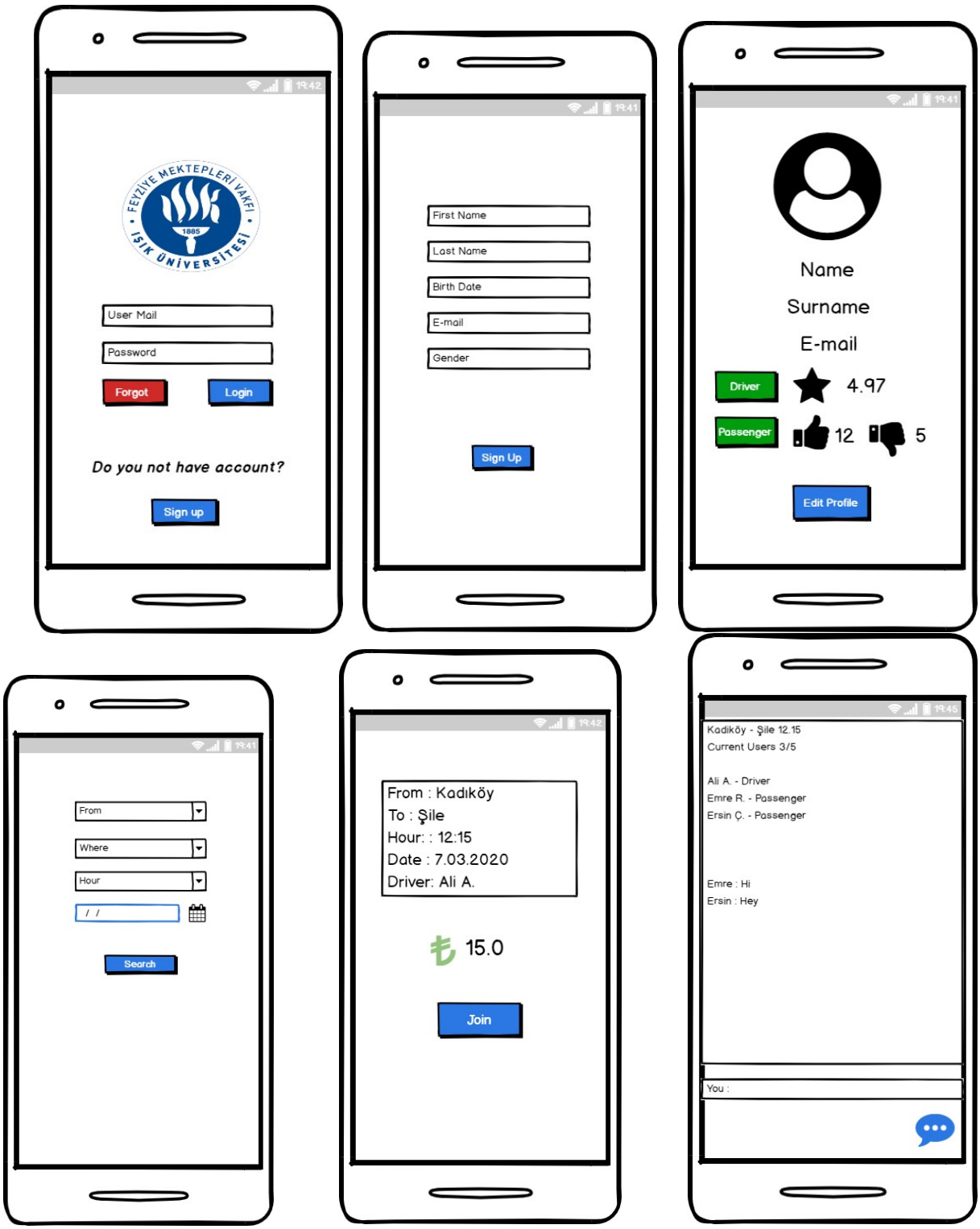

User
Homepage
Payment Page
Database
sd Payment
User
Homepage
Payment Page
Database
1: logins
1.1: clicks at Payment Page
1.1.1: clicks at Balance button
Homepage: Lifeline
1.1.2: clicks at money button
1.1.3: pays
1.1.4: adds money to balance
1.1.4.1: money added to balance
1.1.4.1.1: sends notification the user

<FMWay Işık>


Admin
+getUserList()
+deleteUser()
+editUSer()
+blockUser()
Passenger
+editTrip()
+deletetrip()
+rate()
Users
-uid : String
-email : String
-displayedName : String
-role : Strign
+editProfile()
+makePayment()
Rates
-ratio : String
-message : String
+ratio()
+comment()
+approve()
Payment
-passengerId : String
-driverID : String
-tripPrice : Int
-status : boolean
Messages
-uid : String
-message : String
Trips
-tid : String
-name : String
-start : String
-end : String
+listtrips()
Driver
+editTrip()
+deleteTrip()
+createTrip()
+approve()
1


FMWay Işık
DeleteUser
ratingAuser
support center
Admin
Driver
WievProfile
logout
SystemStartup
Mainpage
SystemShutdown
joinAtrip
changePassword
trip
extension points
createtrip
edittrp
pay to driver
<<Extend>>
<<Extend>>
messaging
Passenger
sending message
createtrip
deletetrip
ForgatPassword

<FMWay Işık>


User Mail
Password
Forgot
Login
Do you not have account?
Sign up
First Name
Last Name
Birth Date
E-mail
Gender
Sign Up
Name
Surname
E-mail
Driver
4.97
Passenger
12
5
Edit Profile
From
Where
Hour
Search
From : Kadıköy
To : Şile
Hour: : 12:15
Date : 7.03.2020
Driver: Ali A.
15.0
Join
Kadıköy - Şile 12.15
Current Users 3/5
Ali A. - Driver
Emre R. - Passenger
Ersin Ç. - Passenger
Emre : Hi
Ersin : Hey
You :

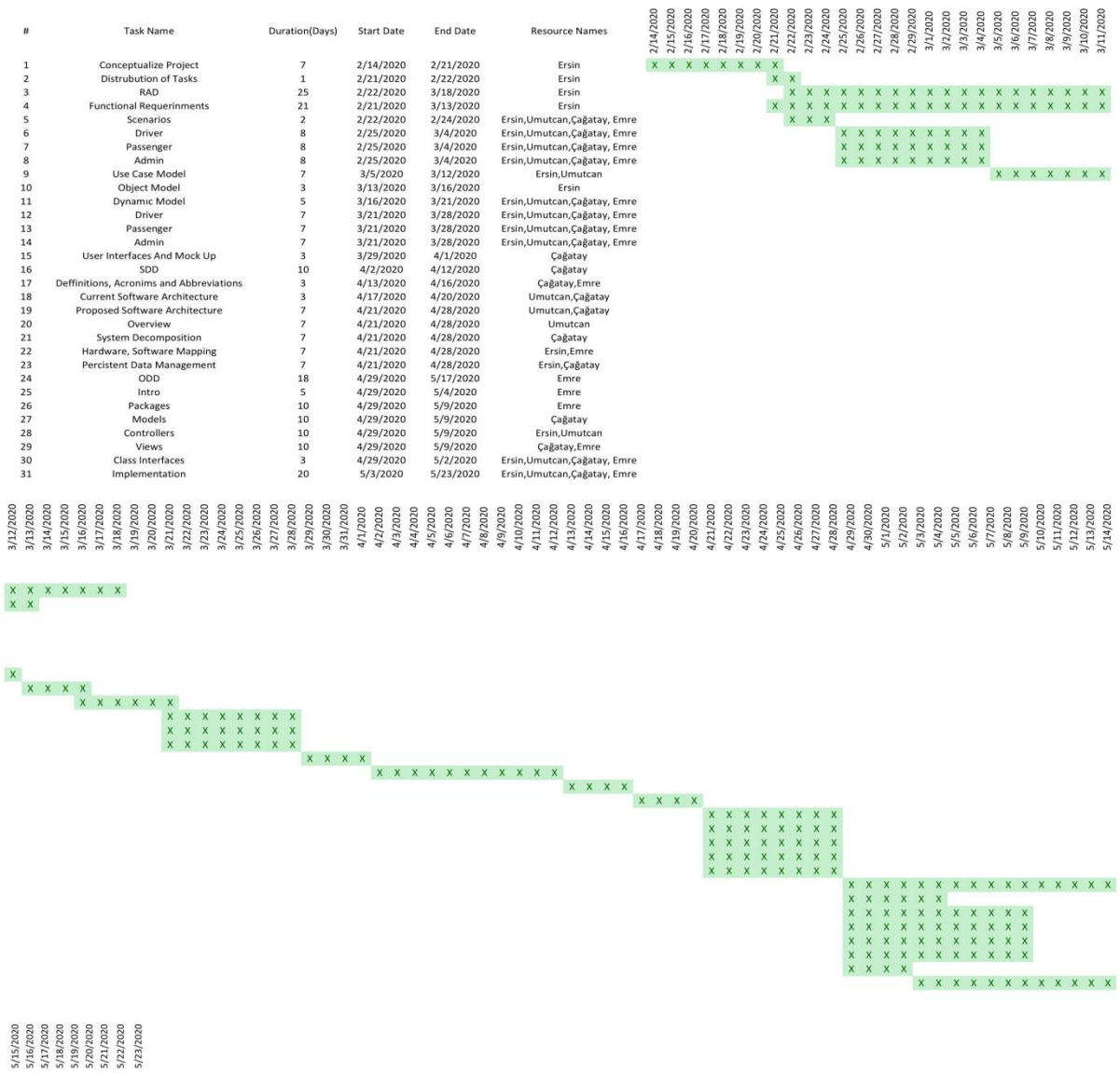
<FMWay Işık>

Total Price : 20.00 TL
Card Holder
Credit Card No
CCV
3D Secure
Pay
From (job
Wher
Hou
Driver
Price^
Kadıköy
Şile
12:15
Ahmet(M
12.00
Kartal
Şile
14.00
Deniz(F)
20.00T

3.4. Project Schedule

Project Name	Project Duration	Start Date	End Date
When The Ring Bells	84	2/14/2020	5/23/2020

#	Task Name	Duration(Days)	Start Date	End Date	Resource Names
1	Conceptualize Project	7	2/14/2020	2/21/2020	Ersin
2	Distrubution of Tasks	1	2/21/2020	2/22/2020	Ersin
3	RAD	25	2/22/2020	3/18/2020	Ersin
4	Functional Requerinments	21	2/21/2020	3/13/2020	Ersin
5	Scenarios	2	2/22/2020	2/24/2020	Ersin,Umutcan,Çağatay, Emre
6	Driver	8	2/25/2020	3/4/2020	Ersin,Umutcan,Çağatay, Emre
7	Passenger	8	2/25/2020	3/4/2020	Ersin,Umutcan,Çağatay, Emre
8	Admin	8	2/25/2020	3/4/2020	Ersin,Umutcan,Çağatay, Emre
9	Use Case Model	7	3/5/2020	3/12/2020	Ersin,Umutcan
10	Object Model	3	3/13/2020	3/16/2020	Ersin
11	Dynamic Model	5	3/16/2020	3/21/2020	Ersin,Umutcan,Çağatay, Emre
12	Driver	7	3/21/2020	3/28/2020	Ersin,Umutcan,Çağatay, Emre
13	Passenger	7	3/21/2020	3/28/2020	Ersin,Umutcan,Çağatay, Emre
14	Admin	7	3/21/2020	3/28/2020	Ersin,Umutcan,Çağatay, Emre
15	User Interfaces And Mock Up	3	3/29/2020	4/1/2020	Çağatay
16	SDD	10	4/2/2020	4/12/2020	Çağatay
17	Deffinitions, Acronims and Abbreviations	3	4/13/2020	4/16/2020	Çağatay,Emre
18	Current Software Architecture	3	4/17/2020	4/20/2020	Umutcan,Çağatay
19	Proposed Software Architecture	7	4/21/2020	4/28/2020	Umutcan,Çağatay
20	Overview	7	4/21/2020	4/28/2020	Umutcan
21	System Decomposition	7	4/21/2020	4/28/2020	Çağatay
22	Hardware, Software Mapping	7	4/21/2020	4/28/2020	Ersin,Emre
23	Percistent Data Management	7	4/21/2020	4/28/2020	Ersin,Çağatay
24	ODD	18	4/29/2020	5/17/2020	Emre
25	Intro	5	4/29/2020	5/4/2020	Emre
26	Packages	10	4/29/2020	5/9/2020	Emre
27	Models	10	4/29/2020	5/9/2020	Çağatay
28	Controllers	10	4/29/2020	5/9/2020	Ersin,Umutcan
29	Views	10	4/29/2020	5/9/2020	Çağatay,Emre
30	Class Interfaces	3	4/29/2020	5/2/2020	Ersin,Umutcan,Çağatay, Emre
31	Implementation	20	5/3/2020	5/23/2020	Ersin,Umutcan,Çağatay, Emre



<FMWay Işık>

4. Glossary

Admin: Actor of FMWay Işık System.
Database: The collection of large data set/stack.
Feedback: The answer of the system for the operation.
GitHub: A web-based hosting service, mostly used for computer code.
Passenger: Actor of Quiz Maker System.
Driver: Actor of Quiz Maker System.
Trips: The specific travels that drivers create and passengers attend to.
Parse Server: Open source Backend-as-a-Service(BaaS) framework initially developed by Facebook.
Java 8: A programming language.
Template: A guide of making patterns.
Android Studio: An Integrated Development Environment (IDE) developed by IntelliJ.

5. References

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
2. https://developer.android.com/guide/
3. https://www.blablacar.com.tr/
4.