BTE 533 – Software Engineering

Taxi Booking Application (Uber)

Section 3 – Requirements Specification - 2

Student Name: Harun Murat Bulut
Student Number: 0701201015
Submission Date: 11.07.2021

CHANGE HISTORY

Date	Version	Description	Updated by
15.06.2021	1	Initial Draft	Bulut, Harun Murat
27.06.2021	2	Added Requirement Specification	Bulut, Harun Murat
07.10.2021	3	Added User interface model and Flow diagrams	Bulut, Harun Murat

1. INTRODUCTION

Requirements Specification documents includes information about system requirements, use cases, user interface model and flow diagram. There is information about functional and non-functional requirements.

2. SYSTEM REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS

MAIN PURPOSES

1. Passenger can book a car.
2. Booking can be done instantly or scheduled.
3. Driver can answer the booking request.
4. Driver should have option to accept or reject the booking.
5. If a driver refuses a booking, then the next available driver may approve the request.
6. Driver can search for nearest passenger.
7. Ticket would produce after successful booking.

RATING & COMMENT

8. Passenger can rate drivers.
9. Drivers would have average rating in their profile.
10. Passenger can make a comment for drivers.
11. Passenger can make favorite drivers list.
12. Passenger can give some tag's for drivers. For example: friendly, funny.

SEARCHING/TRACKING

13. Passenger should track the car and check out the estimated time of arrival.
14. Position of car can be determined by third party application that uses GPS.
15. There should be a map page and, in this page, passenger could be able to see their own position and also all driver positions.
16. Passenger can search available driver from a certain area that he/she specified.

PAYMENT

17. Payment can be done with credit card.
18. Passenger can save credit card information in the application.
19. Payment would be done before travel.
20. Cost should be evaluated according to km that traveled.
21. For payment calculator, last position of travel must be selected by passenger.

REGISTRATION

22. There should be a Passenger and Driver section in the application.
23. There should be an authentication page.
24. Each passenger and driver should have unique id in database.
25. Personal nationality number can be unique id.
26. Driver and passenger login information should be saved in this application.

COMMUNICATION

27. There should be options for communication between passenger and driver.
28. Messaging through application is one option.
29. Driver and passenger can also communicate by their personal phone.
30. In sign in page phone and e-mail information should be asked.
31. Without phone information sign in cannot be done.
32. All messages between driver and passenger should save in database.

2.2 NON-FUNCTIONAL REQUIREMENTS

Usability	Friendly GUI's
Accessibility	Security
Performance	Maintainability
Speed	Extensibility and Stability
Efficiency	Disaster Recovery
Availability	Error Handling

2.2.1 Usability

The Application should be easy to use by every user. In order to accomplish this objective, the system should have a simple and well-designed interface.

2.2.2 Accessibility

The Application should be made accessible to the high number of people. anyone can use the system regardless of the location and can get the information they acquire.

2.2.3 Performance

The performance of the application should be fast and efficient to make the short total travel time. The system should be available for user in real time and always up to date.

2.2.4 Speed
The application response time is a significant requirement because the action cannot be postponed or delayed. The application should be fast enough to satisfy the passenger needs and should not waste their time.

2.2.5 Efficiency
Efficiency of any system is concerned with the minimum processing time as well as the optimal use of system resources in designing the proposed systems. Our android application will be efficient in using processing resources. It can be efficiently run on all android devices.

2.2.6 Availability
The Application should operate 24 hours a day.

2.2.7 Friendly GUI's
The users of this application have different types of people and different levels of technical skills, therefore the application should be understandable by all the users. Consequently, the Application should provide an easy to use, friendly Graphical User Interface (GUI).

2.2.8 Security
Data inserted by user is secured and saved by this application, and will be redundant, in order to perform the exact action in specified situation. The payment security is also so important and must have been take consideration.

2.2.9 Maintainability
The application should be maintained in order to perform the best of its ability.

2.2.10 Extensibility and Stability
The application should be flexible enough to allow improvements for the future and should be able to adapt any additional future change in activities; the application components can be modified for more changes and features allow the addition of new features without disturbing the main functionalities of the application.

2.2.11 Disaster Recovery
The application should be able to recover from an unsuitable problem and should back up data.

2.2.12 Error Handling
The application should be able to handle unexpected errors quickly and easily.

3. USE CASES

3.1 USER TYPES

There would be 3 types of users in this application:

1. Passenger

   Passenger could be able to book a car.
   Passenger can track all drivers with map feature.
   Passenger can login to application.
   Passenger can rate and make a comment for drivers.
   Passenger can make a payment.
   Passenger can communicate with drivers.

2. Driver

   Driver can login to application.
   Driver can answer the booking request of passengers.
   Driver can communicate with passengers.

3. Admin

   Admin has all rights for fixing issues in application.
   Admin can login to application.

3.2 USER SCENARIOS

LOGIN TO APPLICATION

I am Murat **(Passenger/Driver/Admin)**. I want to login to application. My login information is not saved in the application. I enter my information and select the box that save my information for next login. After success login main page would appear.

BOOKING A CAR (INSTANTLY)

I am Murat **(passenger)**. I want to make a booking from application. I login to application with my e-mail and password. I enter where I want to go and select instantly option from application. Application take my current position from its GPS service. Fee is calculated by mobile application automatically. After that I am looking for nearest driver from map feature of application and select one of them. Ahmet **(driver)** accept my booking request.

BOOKING A CAR (SCHEDULED)

I am Murat **(passenger)**. I want to make a booking from application. I login to application with my e-mail and password. I enter where I want to go and select scheduled option from application. Application take my current position from its GPS service. Fee is calculated by mobile application automatically. Before 30 minute of scheduled booking application is assign me a nearest driver automatically and Ahmet **(driver)** accept my booking request.

MAKE A PAYMENT

I am Murat **(passenger)**. I want to make a payment for a booking. My credit card information is not saved to application. I enter my credit card information and select a box that save my credit card information for next payment. After that if payment is succeed application give me notification and make a booking.

SEARCHING/TRACKING A CAR

For searching:

I am Murat **(passenger)**. I switch to map page of application. I select specific area to search for car. I arrange the size of this area. Then I am searching for available driver.

For tracking:

I am Murat **(passenger)**. I made a successful booking. From map page I can track the related driver.

RATING/COMMENT

I am Murat **(passenger)**. I finished my travel. After travel I go to driver's page and from rating section I give the star out of 5. And I can also make a comment.

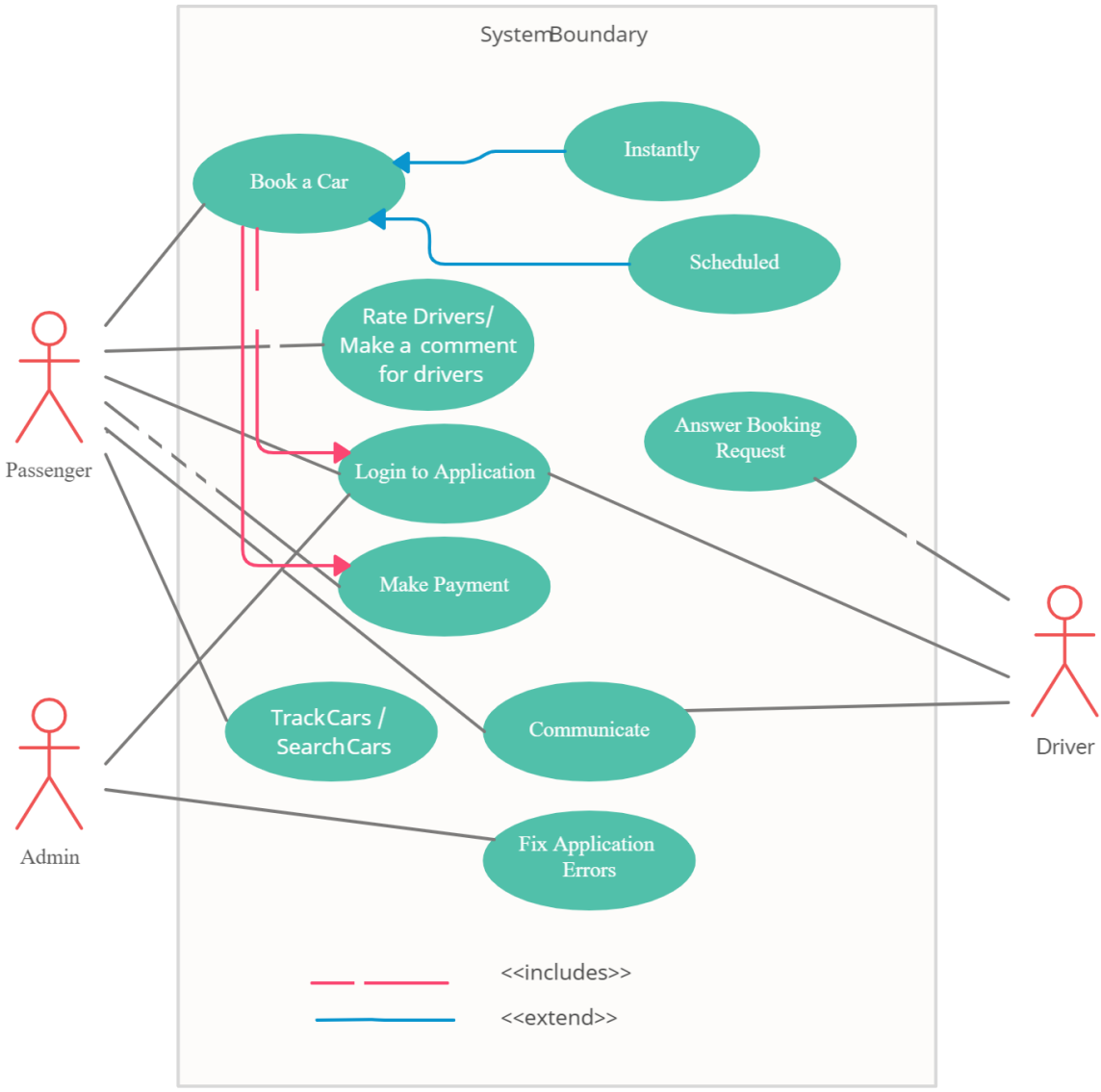
ACCEPT OR REFUSE BOOKING OFFER

I am Ahmet **(driver).** There is a booking request from Murat **(passenger)**. I see the distance of travel and amount of fee. There is two option for me, accept the booking or reject. This booking is good offer for me and I accept it.

COMMUNICATION

I am Murat **(passenger)**. I made a booking request and Ahmet **(driver)** accept the booking request. I am waiting to much and from map application I am tracking the driver but there is no movement. I want to send a message to Ahmet **(driver)** to ask if there is something wrong. I switch to communicate page and send a message to driver.

3.3 USE CASE DIAGRAM



Use Case Diagram

3.4 USE CASES

USE CASE: LOGIN TO APPLICATION

Primary Actor/Actors: Passenger/Driver/Admin

Purpose: Passenger/Driver/Admin would log in to application

Preconditions:

- Application must be downloaded.
- Before login Passenger/Driver/Admin must be sign in before with required information like e-mail, telephone number etc.

Postconditions:

- Main page would open after login.

Basic flow:

1. Passenger/Driver/Admin enter their username or registered e-mail.
2. Passenger/Driver/Admin have option to select save my login information or not.
3. Passenger/Driver/Admin enter their password.
4. After login main page would appear.

Alternative flow event:

- If username does not exist in database warning would be displayed.
- If password is not correct error would be displayed

USE CASE: BOOKING A CAR (INSTANTLY)

Primary Actor/Actors: Passenger

Purpose: Passenger book a car with instantly option

Preconditions:

- Passenger should log in.
- There should be available driver.
- Payment should be succeeded.
- Driver should accept the booking offer from passenger.

Postconditions:

- Booking hour would marked as a

Basic flow:

1. Passenger login to application.
2. Passenger select booking option: instantly.
3. From map feature passenger can select a car to make a booking.

4. Passenger should select where he/she want to go to because payment calculator require this information.
5. Passenger can track location of driver after booking accepted.

Alternative flow event:

- If user try to make a booking with unavailable driver application would give an warning message.

USE CASE: BOOKING A CAR (SCHEDULED)

Primary Actor/Actors: Passenger

Purpose: Passenger book a car with scheduled option

Preconditions:

- Passenger should log in.
- Payment should be succeed, payment can be done at least 30 minutes ago of booking time.
- Driver who selected by application should accept the booking.

Postconditions:

- Passenger can track location of driver after booking accepted.

Basic flow:

1. Passenger login to application.
2. Passenger select booking option: scheduled.
3. Passenger should select where he/she want to go to because payment calculator require this information.
4. Before 30 minutes ago one available driver should assign for this passenger from application.
5. Passenger can track location of driver after booking accepted.

USE CASE: MAKE A PAYMENT

Primary Actor/Actors: Passenger

Purpose: Passenger make payment with their credit card.

Preconditions:

- Passenger should log in.
- Booking request should be accepted by driver.
- Last position of travel must be selected by passenger.
- Credit card information of passenger must be correct.

Basic flow:

1. Payment calculated by calculator according to initial and last position of travel.
2. Credit card information checked application. (Syntax check)
3. If credit card informations are correct payment be done with third party application.
4. Payment receipt would created by application, user can be download this or send by e-mail.

Alternative flow event:

- If credit card information is not correct error would be displayed.
- If there is not enough limit in passenger credit card error would be displayed.

Exceptional flow of events:

- If third party service for payment is not work correctly error should be displayed.

USE CASE: SEARCHING/TRACKING A CAR

Primary Actor/Actors: Passenger

Purpose: Passenger can search for an available driver or can track a car that they book.

Preconditions:

- Passenger should log in.
- Passenger should give access to use locations for that application from their mobile phone.

Basic flow:

1. Passenger switch to map section of application.
2. Passenger can select an area to find an available driver.
3. Passenger can arrange the size of area that they watch.
4. If user made a booking related driver would be marked.

Exceptional flow of events:

- If third party service for GPS is not work correctly map page would give an error to passenger.

USE CASE: RATING/COMMENT

Primary Actor/Actors: Passenger

Purpose: Passenger would make a rating or comment to driver.

Preconditions:

- Passenger should log in.
- Passenger must had a travel with driver who he/she want to make rating or comment.

Basic flow:

1. Passenger would open the driver profile.
2. Passenger can make a rating with stars.
3. There are 5 stars. Passenger would select what they want to give to driver.
4. There is also option to make a comment.
5. From related section passenger can make a comment for driver.

USE CASE: ACCEPT OR REFUSE BOOKING OFFER

Primary Actor/Actors: Driver

Purpose: Driver would accept or reject the request of passenger booking.

Preconditions:

- Driver should log in.
- There should be a booking offer.
- Driver should be arrive the place of passenger at least 20 minute.

Basic flow:

1. Passengers send an offer to driver.
2. Application sends a notification to driver.
3. After that there is two options for driver: Accept or Refuse.

USE CASE: COMMUNICATION

Primary Actor/Actors: Passenger/Driver

Purpose: Passenger and Driver want to communicate with each other.

Preconditions:

- Passenger should log in.
- Driver should log in.

Basic flow:

1. Passenger/Driver switch go to profile that they want to send a message.
2. There is a message feature in the application, when Passenger/Driver push the send a message button messaging page would open.
3. Passenger/Driver can send a message from message box.
4. If notifications on in related Passenger/Driver's phone they would see the message.

USE CASE: FIX THE APPLICATION ERRORS

Primary Actor/Actors: Admin

Purpose: Admin would fix the application error.

Preconditions:

- There should be error to fix.
- The error should be able to fix from application.

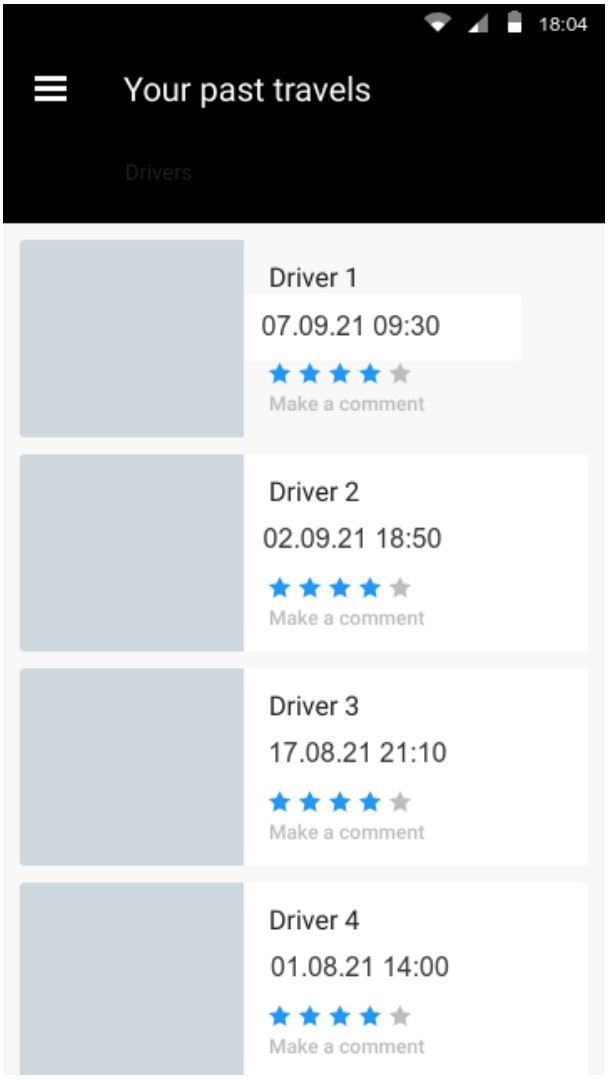
Basic flow:

1. Issue come from passenger or driver.
2. Admin try to understand reason for error.
3. If admin knows the how to fix error, he/she log in to application and fix the problem.
4. If admin don't understand the reason he/she create a report for developers to fix the problem.

4. USER INTERFACE MODEL

4.1 SCREEN MOCKUPS

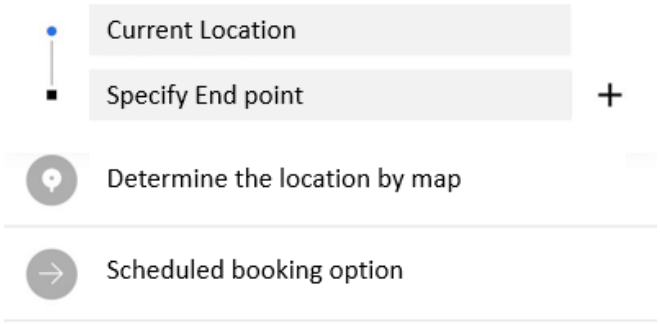
RATING & COMMENT

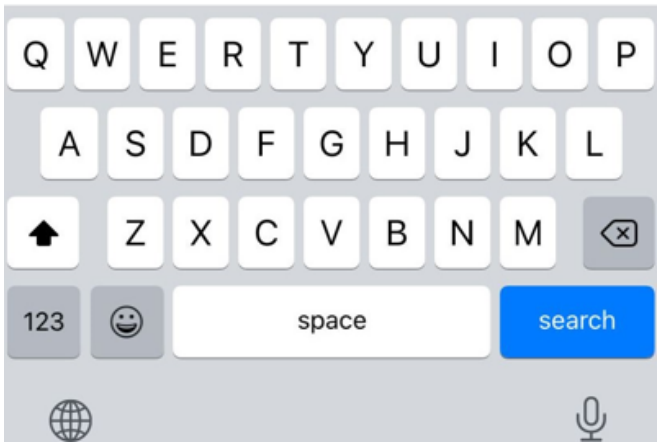


Rating&Comment Page

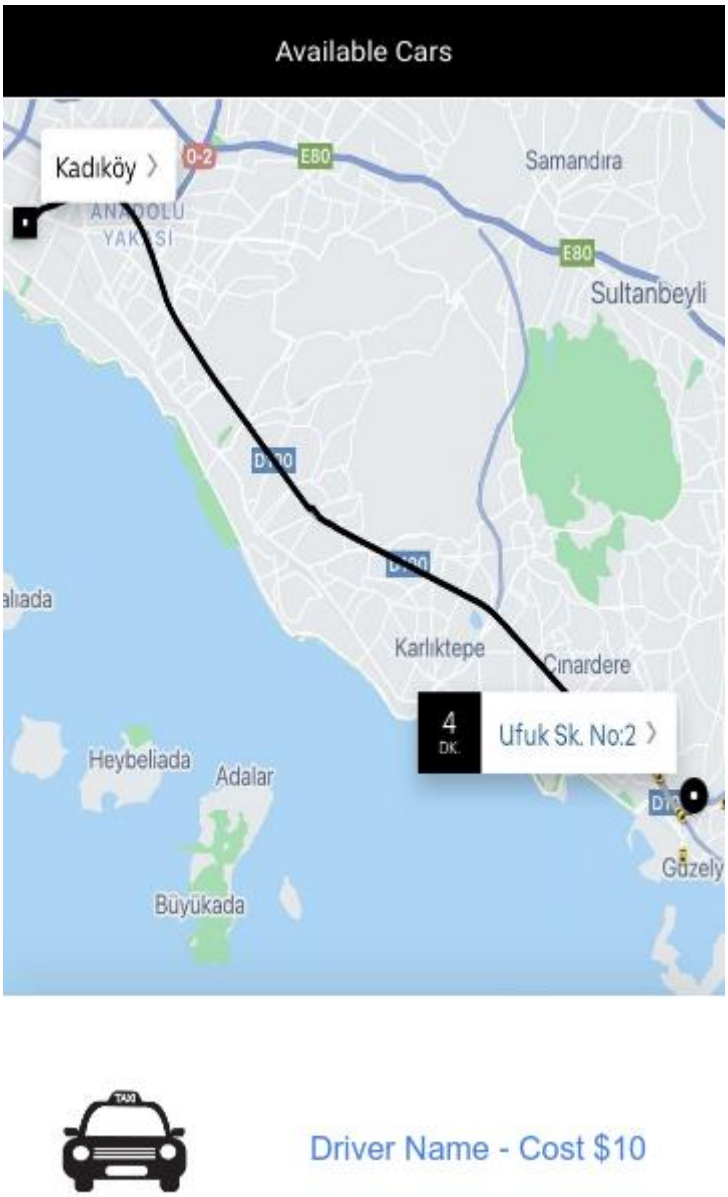
1. Passenger can rate drivers.
2. Drivers would have average rating in their profile.
3. Passenger can make a comment for drivers.
4. Passenger can make favorite drivers list.
5. Passenger can give some tag's for drivers. For example: friendly, funny.

SEARCHING/TRACKING





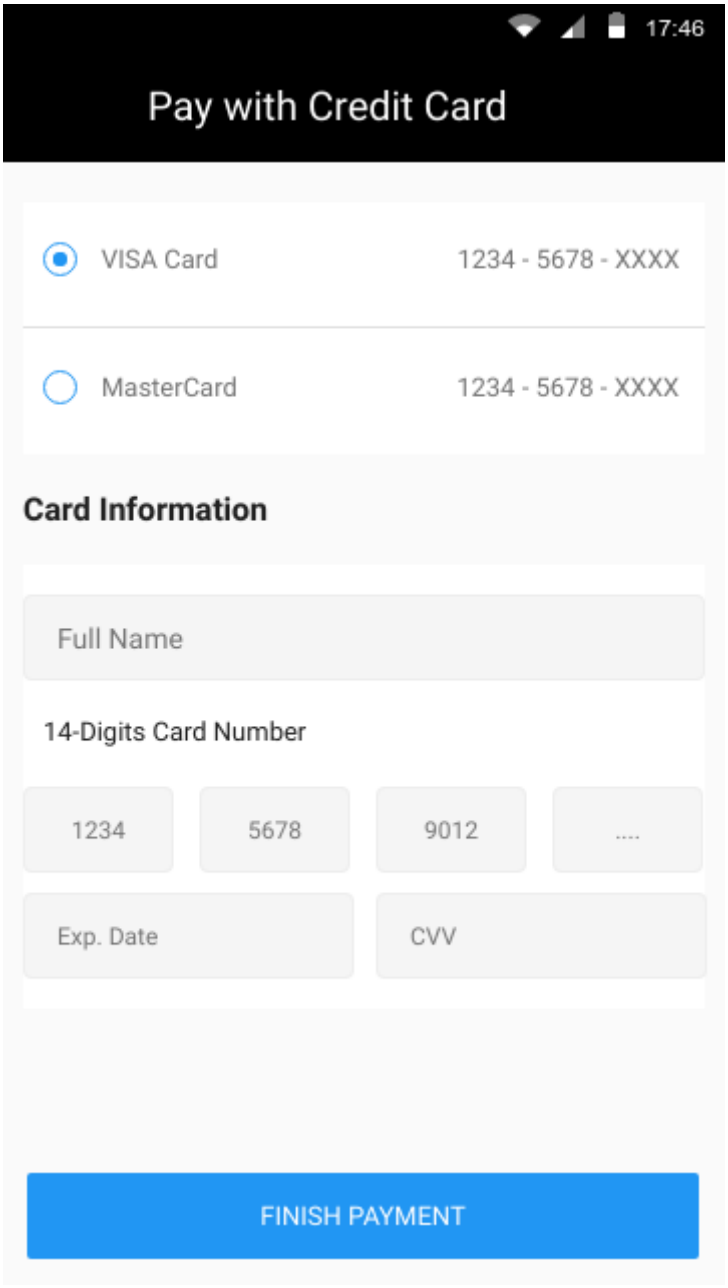
Specify the End Point of Travel with Scheduled Option



Available Drivers and Costs Page

1. Passenger should track the car and check out the estimated time of arrival.
2. Position of car can be determined by third party application that uses GPS.
3. There should be a map page and, in this page, passenger could be able to see their own position and also all driver positions.
4. Passenger can search available driver from a certain area that he/she specified.

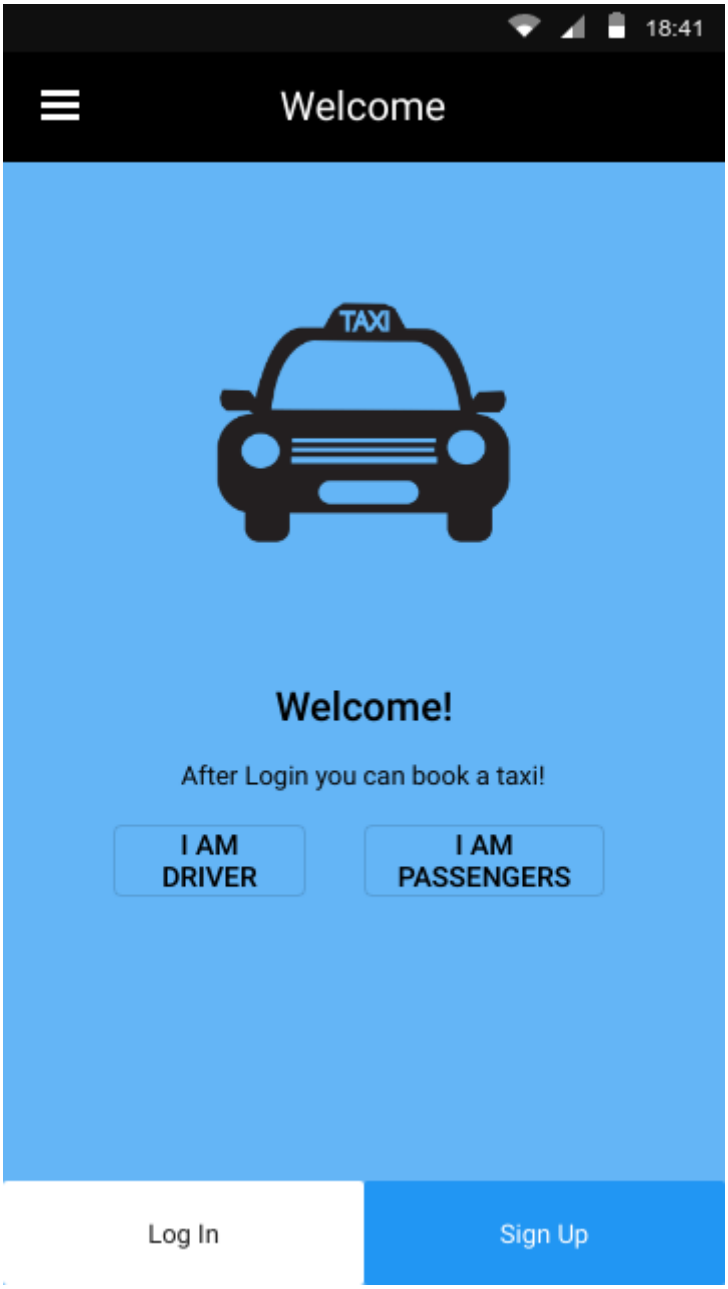
PAYMENT



Payment Page

1. Payment can be done with credit card.
2. Passenger can save credit card information in the application.
3. Payment would be done before travel.
4. Cost should be evaluated according to km that traveled.
5. For payment calculator, last position of travel must be selected by passenger.

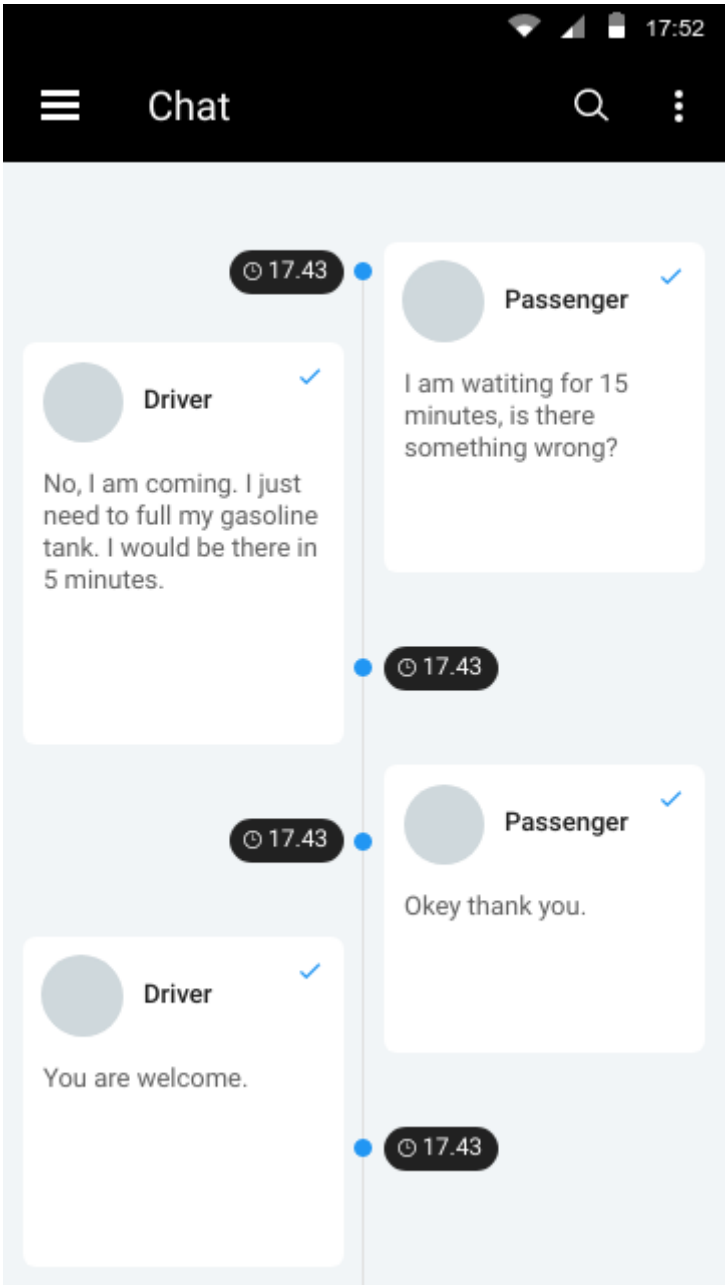
REGISTRATION



Login Page

1. There should be a Passenger and Driver section in the application.
2. There should be an authentication page.
3. Each passenger and driver should have unique id in database.
4. Personal nationality number can be unique id.
5. Driver and passenger login information should be saved in this application.

COMMUNICATION



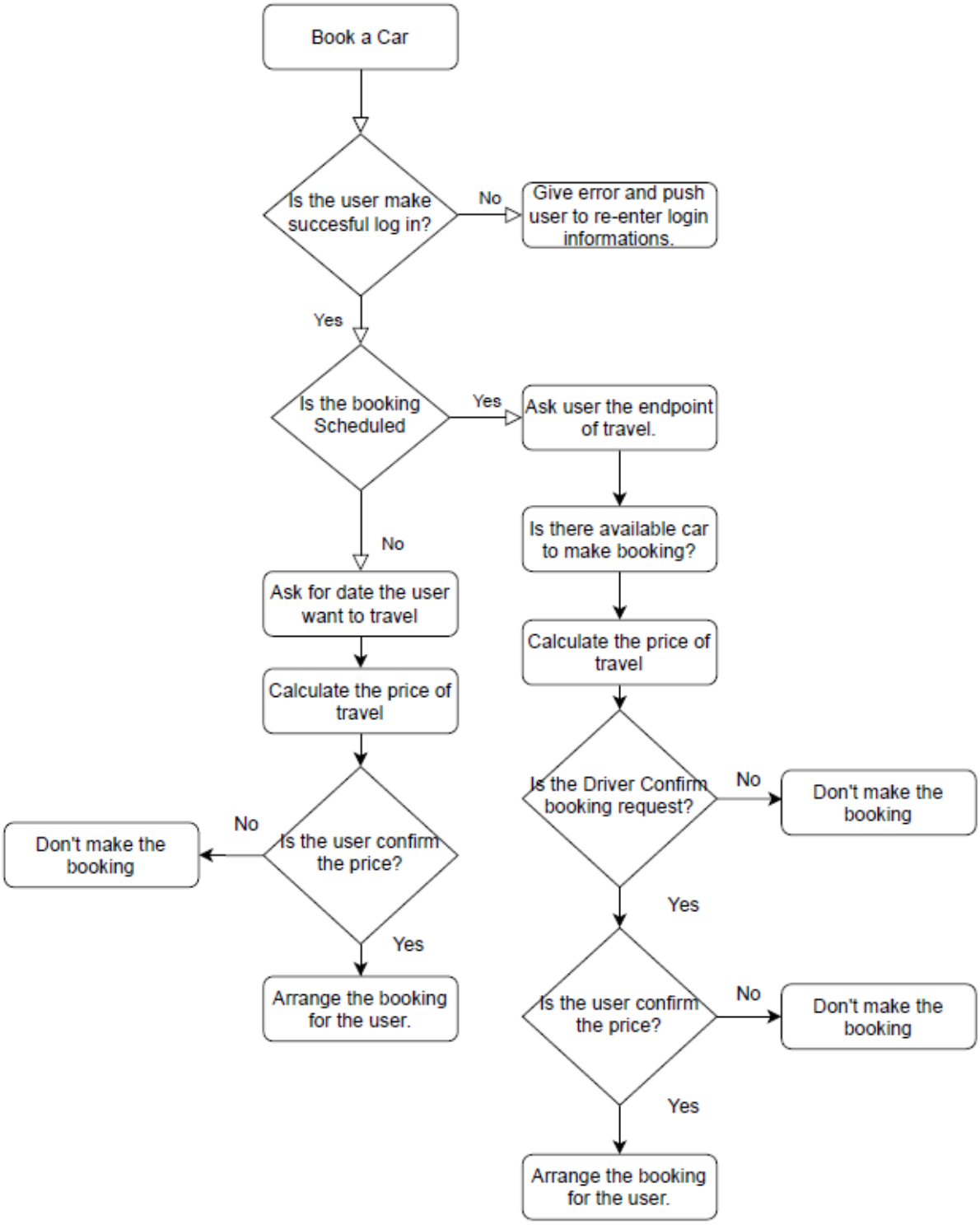
Communication Page

1. There should be options for communication between passenger and driver.
2. Messaging through application is one option.
3. Driver and passenger can also communicate by their personal phone.
4. In sign in page phone and e-mail information should be asked.
5. Without phone information sign in cannot be done.
6. All messages between driver and passenger should save in database.

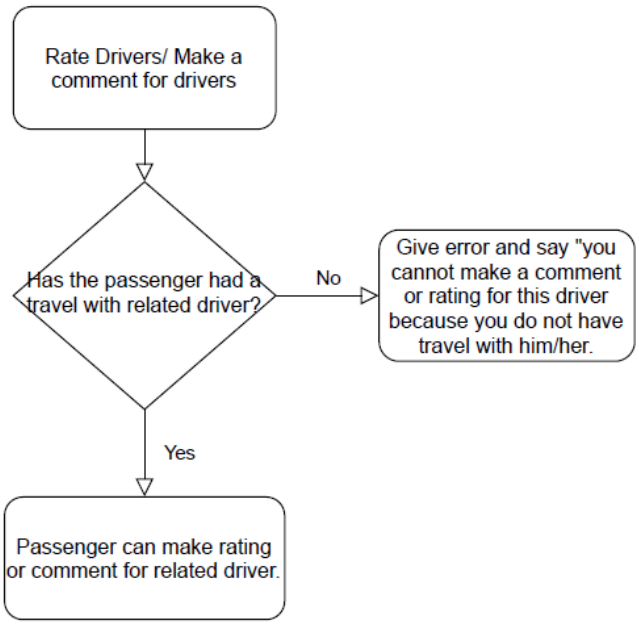
5 FLOW DIAGRAMS

5.1 FLOW DIAGRAMS

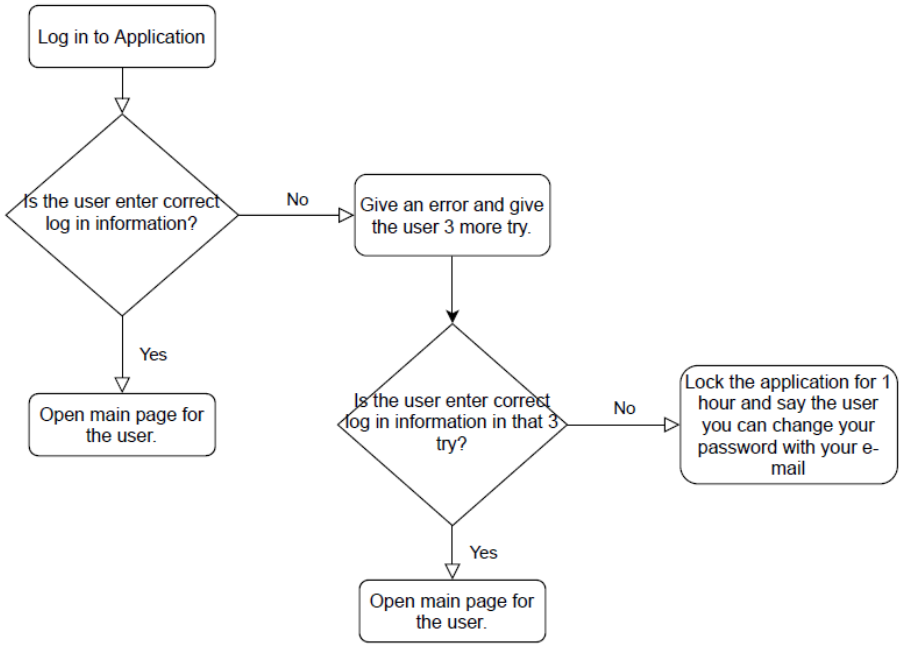
5.1.1 Book a Car



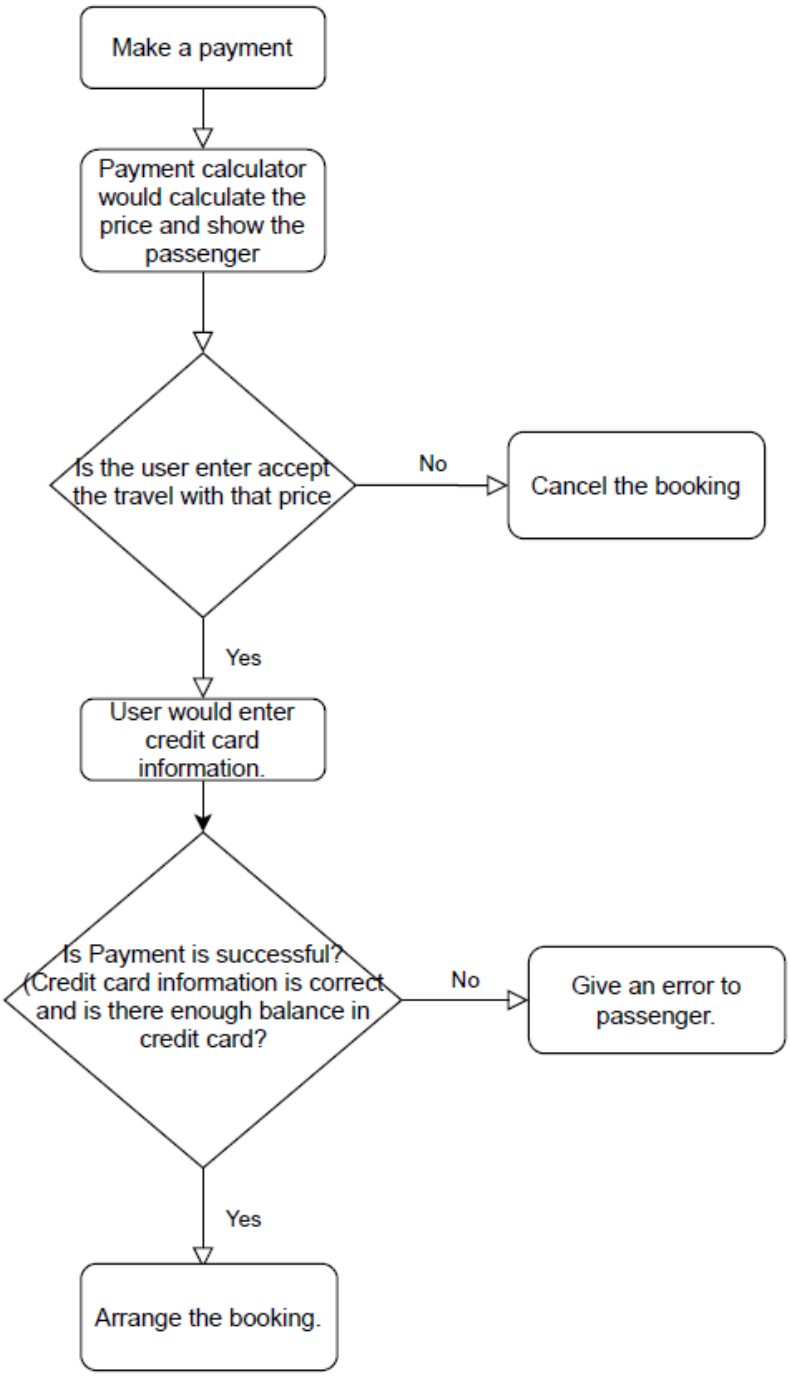
5.1.2 Rate Drivers/ Make a comment for drivers



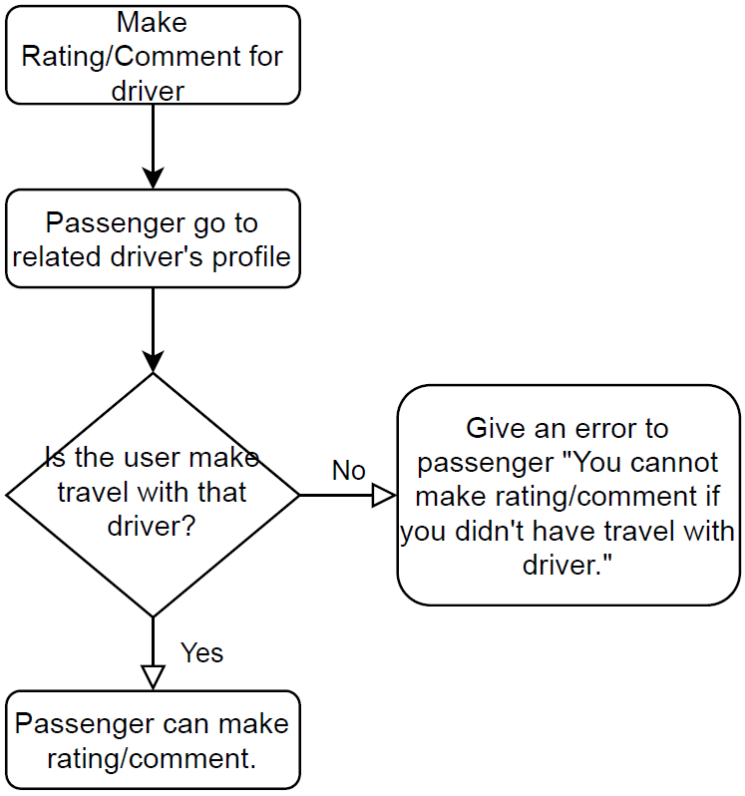
5.1.3 Log in to Application



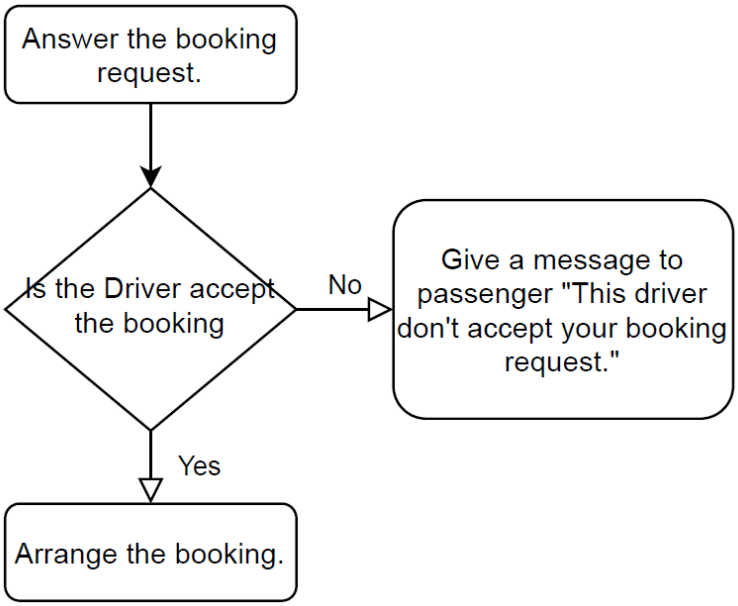
5.1.4 Make a Payment



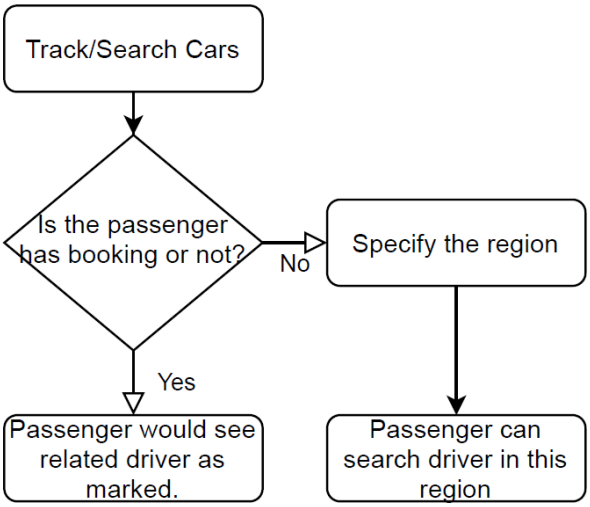
5.1.5 Make a Comment for Driver



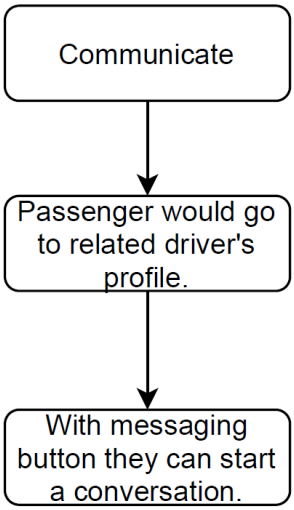
5.1.6 Answer the Booking Request



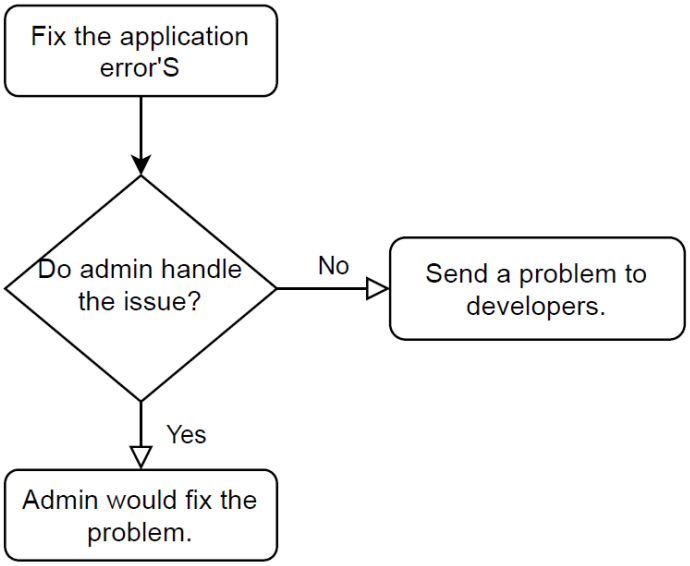
5.1.7 Track/Search a Car



5.1.8 Communicate

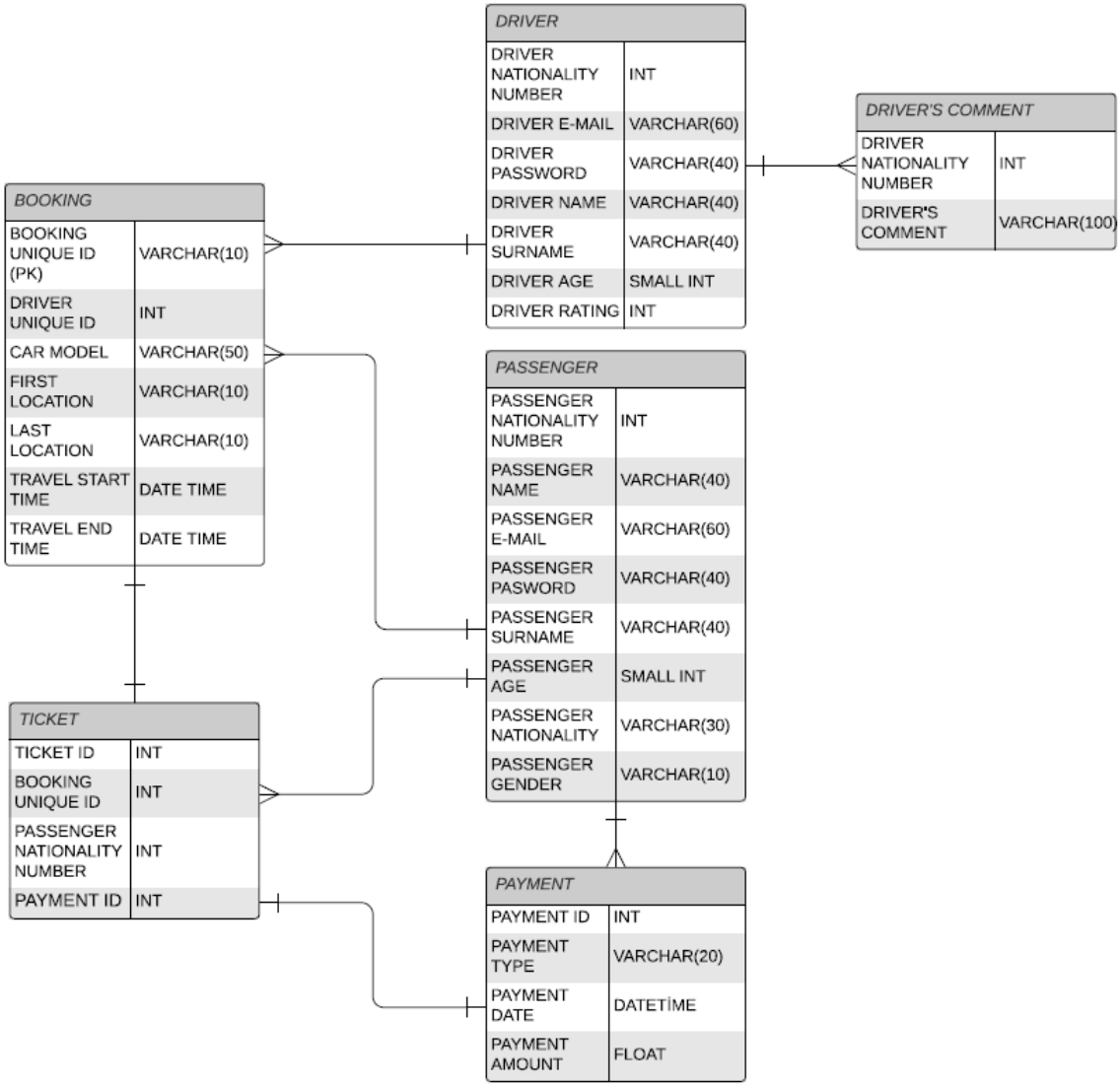


5.1.9 Fix Application Error's



5.2 GENERAL DATA MODEL

This chapter includes the Entity relationship diagram and so data that is stored in database.



E-R diagram.

Primary key selected as booking **unique ID.**

5.3 IMPORTANT DATA CONSIDERATIONS

Data would be sent to database as an API with **JSON** format.

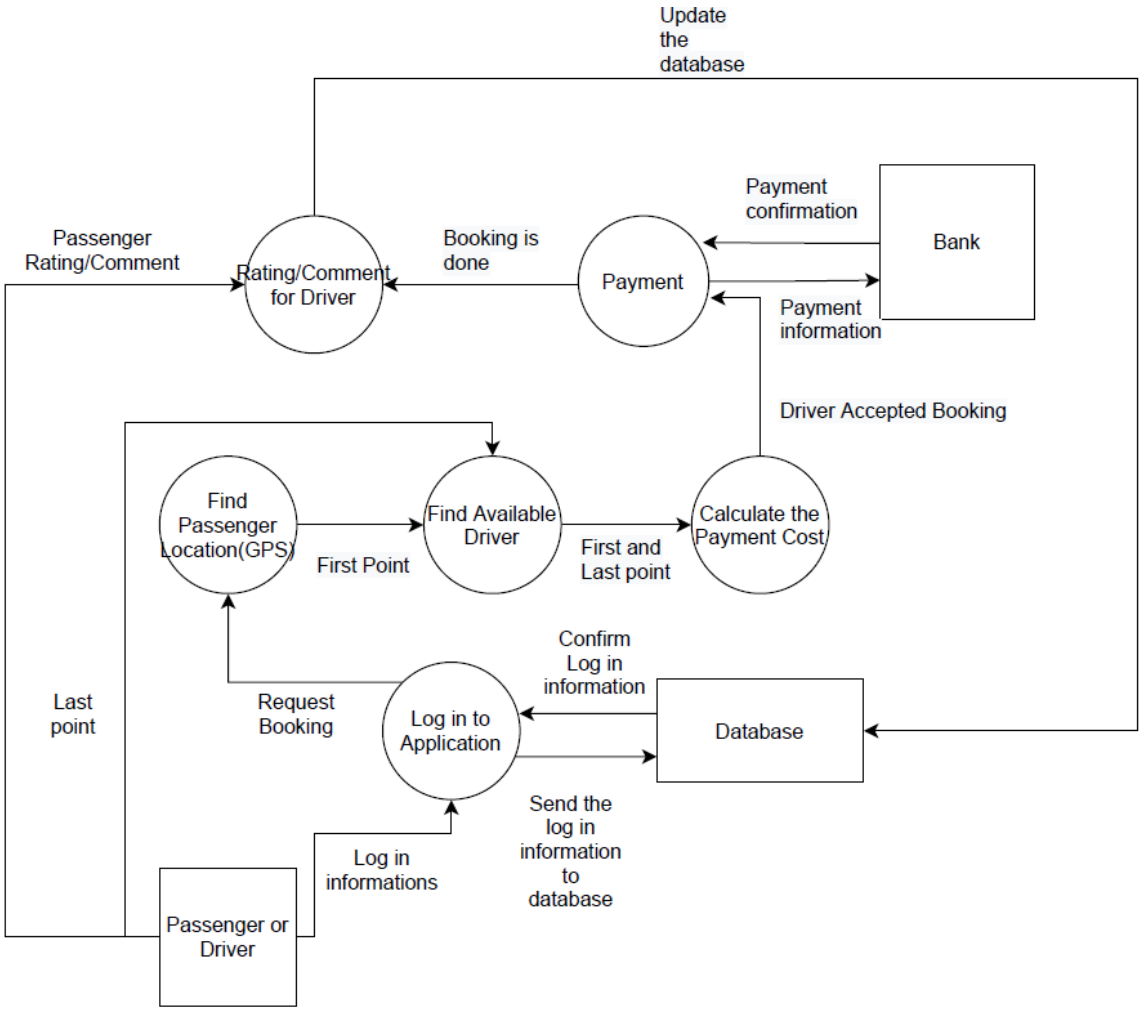
- JSON stands for JavaScript Object Notation
- JSON is a text format for storing and transporting data.
- JSON is "self-describing" and easy to understand.
- JSON is language independent.

One JSON file example for driver:

```
'{"Driver Nationality Number":19250706426, "Driver E-
mail":"bulutharunmurat@gmail.com", "Driver password":"123456", "Driver
name":"Harun Murat", "Driver surname":"Bulut", "Driver Age":26, "Driver
Rating":4.8}'
```

5.4 DATA FLOW

In this chapter there is a data flow diagram, with this diagram it can be seen that what data come from and where it is arriving. And processes can be seen as well.



Data Flow Diagram