**Software Requirements Specification**

*Version 1.0*

June 17, 2020

**Management of a transport company**

*Hany Salloum*

*Lian Aljundi*

*Basma Dekar*

*Zolfakkar Aldaher*

*Submitted in partial fulfillment of the requirements of Software Engineering 2*

**1.Introduction**

1.1. Purpose

The purpose of this document is to present a detailed description of YourRoad applications and transportation system.

It will explain the purpose and features of the applications, the interfaces

of the applications and what the system will do. This document is intended for the passengers, the drivers and the developers of the system.

1.2. Scope of Project

This software system will be an online booking for local journeys. This system will be designed to improve booking quality and customer comfort by easily choosing preferred time and destination and many others features.

1.3. Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Booking | the interface where a passenger can book his journey online and choose whatever characteristics he needs for this journey |
| Tickets | the interface where a passenger can check his tickets |
| Last Trips | the interface of which the user can check out his previous journeys and trips |
| My Trip | the interface of which the user can check the live status of his current journey |

1.4. References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

1.5. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the applications

**2.0. Overall Description**

2.1 System Environment

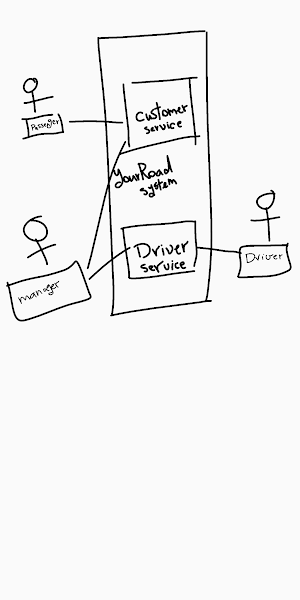
The YourRoad System has three active actors and one cooperating system.

The Passenger, the driver accesses the interfaces via internet. Through apps. The manager

accesses the entire system directly.

2.2 Functional Requirements Specification

This section outlines the use cases for each of the active actors separately.

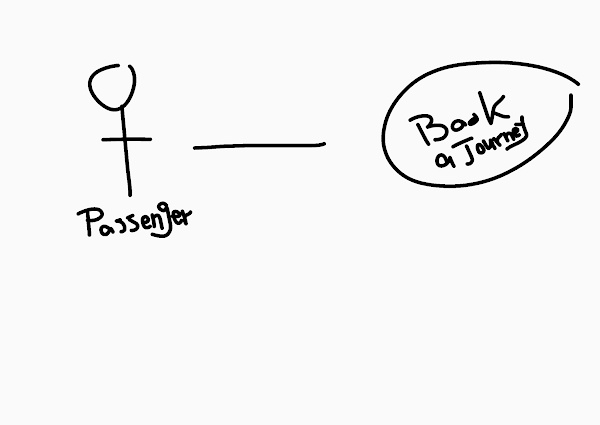


2.2.1 Passenger Use Case

Use case: Book your journey

Brief Description:

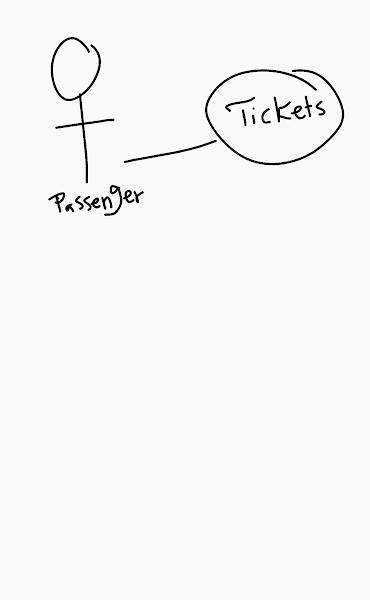
The passenger accesses the booking tab, chooses which time and destination he'd like to travel to.



Use case: Tickets

Brief Description:

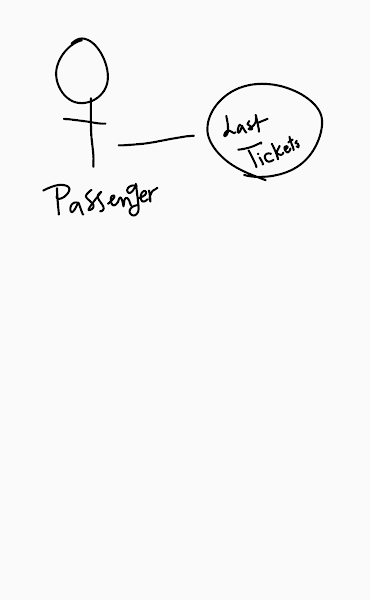
The passenger accesses the tickets tab, checking the tickets he has.



Use case: last trips

Brief Description:

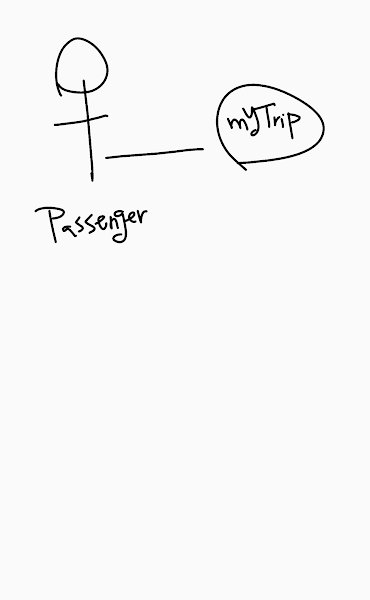
The passenger accesses the last Trips tab, views last Trips he had.



Use case: my trip

Brief Description:

The passenger accesses the my trip tab, checking for the details of the current trip he's on.

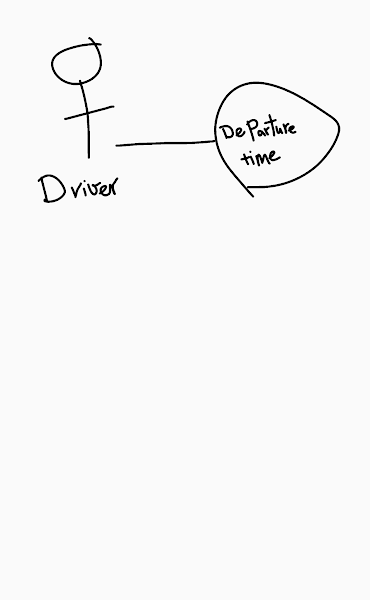


2.2.2 Driver Use Case

Use case: supplying departing time

Brief Description:

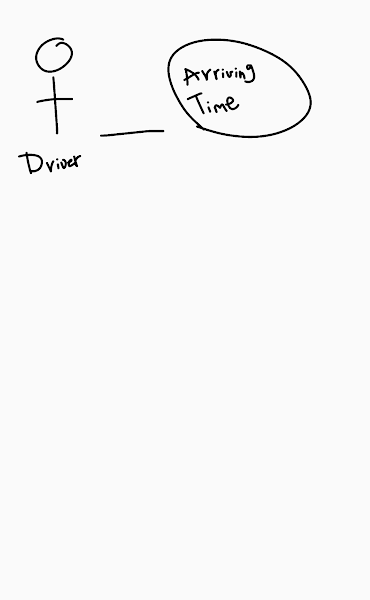
The driver presses a button and sends current time of departure.



Use case: supplying arriving time

Brief Description:

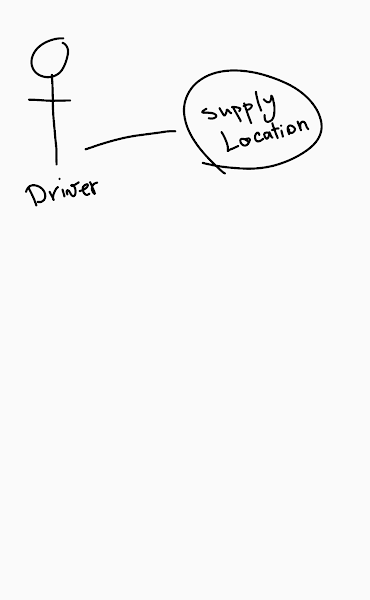
The driver presses a button and sends current time of arriving.



Use case: supplying live location

Brief Description:

The driver presses a button and sends his live location.



***2.3 User Characteristics***

The user is expected to be owned an android phone and be able to accsess the internet. The main screen of the application will have fileds to fill it with the trip requirnment of each user that include the depart date, depart time, return date, return time, from ,to ,passengers number, trip type.

The user is expected to be android literate and to be able to use button, pull-down menus, and similar

***2.4 Non-Functional Requirements***

Here you will be the customer, anyone who has an Android phone version 5.0 or more can use the application

The application development was based on the Android Studio environment in addition to using databases MySQL

exchanging the data between the application and database via the Internet

**3.0. Requirements Specification**

***3.1 External Interface Requirements***

***Requesting a link with the database to install the reservation***

***The rule fields define the time, hour, and day fields and give the manager available trips in time and place and his role to check on company reservation***

***3.2 Functional Requirements***

The Logical Structure of the Data is contained in Section 3.3.1.

3.2.1 yourroad

|  |  |
| --- | --- |
| **Use Case Name** | yourroad |
| **XRef** | Section 2.2.1, yourroad  SDD, Section 7.1 |
| **the user** | Make reservations with an app |
| **Basic Path** | The user opens the app, the reservation interface appears, chooses the place, time, and flight they want, and presses OK where you go to the database and install the reservation |
| **Alternative Paths** | * In step 2, Go to the company and book with it |
|  |  |
| **Exception Paths** | cancellation of reservation. |
| **Other** | The reservation list is created |

3.2.2 Add reservation

|  |  |
| --- | --- |
| **Use Case Name** | Add reservation |
|  |  |
| **user** | The user selects to add a new reservation to the database. |
| **Basic Path** | * The user opens the app, the reservation interface appears, chooses the place, time, and flight they want, and presses OK where you go to the database and install the reservation. |
| **Alternative Paths** | Go to the company and book with it |
| **Exception Paths** | cancellation of reservation. |
|  |  |

3.2.3 Update reservation Status

|  |  |
| --- | --- |
| **Use Case Name** | Update reservation Status |
| **user** | The user selects to update the status of an reservation in the database. |
|  |  |
| **Basic Path** | * The user chooses the reservation they want to modify * The rule displays and updates the reservation. |
| **Alternative Paths** | Go to the company and amend the reservation |
| **Post condition** | The database has been updated. |
| **Exception Paths** | If the reservation is not in the database, it will be canceled and re-booked |
| **Other** | This use case can be used to amend a reservation or to re-reserve |

**3.2.4 Remove reservation**

|  |  |
| --- | --- |
| **Use Case Name** | Remove reservation |
| **XRef** | Section 2.2.4, Remove reservation  SDD, Section 7.12 |
| **User** | The user selects to remove reservation from the database. |
|  |  |
| **Basic Path** | * The system provides a list of all reservations * The user chooses to reserve * The system displays the reservation information and asks the manager to confirm the deletion * The manager confirms the deletion. |
| **Alternative Paths** | Go to the company and confirm the deletion |
| **Post condition** | The reservation is removed from the database. |
| **Exception Paths** | The user may abandon the reservation |

**3.2.5 Check Status**

|  |  |
| --- | --- |
| **Use Case Name** | Check Status |
| **Trigger** | The user has selected to check the trips available in a time he want |
| **Precondition** | The user has an internet connection and put all information’s about the trip he want |
| **Basic Path**  **Basic Path** | The system presents a list of all availble trips organized by their infromation |
| **Alternative Paths** | None. |
| **Post condition** | The requested information has been displayed. |
| **Exception Paths** | The user may abandon the ticket at any time. |

**3.2.6 Send Communication**

|  |  |
| --- | --- |
| **Use Case Name** | Send Communication |
| **Trigger** | The user selects to send a communication to the database |
| **Precondition** | The user has accessed the available tickets from database |
| **Basic Path** | The user select an option to a trip he want  He send it to data base  Data base received it an send him the available trips |
| **Alternative Paths** | None. |
| **Post condition** | The information was in the selected option only |
| **Exception Paths** | The user may abandon the ticket at any time. |

**3.2.7 Confirmation of reservation appointment**

|  |  |
| --- | --- |
| **Use Case Name** | Confirmation of reservation appointment |
| **Trigger** | Booked the ticket from the company |
| **Precondition** | The user have to verify his bank account |
| **Basic Path** | The user select the ticket he want  He pay it from company  The ticket will be not available any more |
| **Alternative Paths** | None |
| **Post condition** | The ticket cant edit after the user bought it |
| **Exception Paths** | The user may abandon the ticket at any time. |

**3.2.8 Removing ticket**

|  |  |
| --- | --- |
| **Use Case Name** | Removing ticket |
| **Trigger** | Select ticket from the list of user to remove it |
| **Precondition** | The user have to lose 10% from the ticket cost |
| **Basic Path** | The user select a ticket he want to remove it  He sent a remove request to the database  The database remove the ticket and send the user an notification |
| **Alternative Paths** | none |
| **Post condition** | You cant return the ticket you have to buy a new one |

***3.3 Detailed Non-Functional Requirements***

**3.3.1 Logical Structure of the Data**

Below is the sequence of data that will be stored in the internal manager database :

User 🡪 select his trip options 🡪 review the available option 🡪 choose one 🡪 send it to manager database

The data descriptions of each of these data entities is as follows:

**User data entity :**

|  |  |  |  |
| --- | --- | --- | --- |
| Data item | Type | Description | Comment |
| source city | text | name of city | ---- |
| destination city | text | name of city | ---- |
| depart date | Date | trip depart day | ---- |
| return date | Date | trip return day | ---- |
| depart time | Time | trip depart time | ---- |
| return time | Time | trip return time | ---- |
| trip type | Radio button | trip is direct or indirect | ---- |
| passenger | spinner | number of tickets want to | can buy more the one |

**3.3.2 Security**

The server on which the application resides will have its own security to prevent unauthorized write/delete access. There is no restriction on read access.

The PC on which the database Manager resides will have its own security. Only the manager will have physical access to the machine and the program on it. There is no special protection built into this system other than to provide the user access to the Online information to choose a trip