

Uber - Software Requirement Specifications

SWE Project - Phase 1

Students:

BI03 - 2021170951 - دينا سعيد محمد محمود

BI03 - 2021170956 - ميار إيهاب محمد نجيب

BI03 - 2021170955 - ملك خالد حافظ خطاب

BI03 - 2021170954 - محمد شريف عبد الصادق صفا

DMM3 - 2021170936 - محمود هاني محمود أحمد

Table of Contents

Introduction	03
Glossary	04
User Requirements	05
Functional Requirements	06
Non-Functional Requirements	15
Use Case Diagram	16
Sequence Diagram	18

1. Introduction

This is the Software Requirements Specification (SRS) document for the Uber application's ride-booking feature. Our aim in this document is to examine requirements that should be available for the ride booking process to become an integral part of the Uber platform and its easy use. The ride-sharing system is geared towards completely transforming the means by which people access their transportation, so that all this could be done in a simple and efficient manner without the same tension. Furthermore, customers can define their own criteria about the time they prefer to travel, the comfort they want to get while journeying, availability at all times of day and night, and prices by choosing between different modes like bikes, taxi, or cars.

1.1. Purpose:

The main purpose of this SRS is to be a complete guide in ensuring that the ride booking feature is in line with the requirements. This is through strict adherence to these specifications such as ease and fluency for users to book for rides using the Uber application.

1.2. Scope:

The simple process of booking a ride on the Uber app is the main focus of this SRS document. It describes in detail each and every step, from the moment the user selects the destination to the confirmation he receives from the driver. Crucial features like choosing the type of ride, how drivers are assigned to users, tracking the driver's real-time location, and calculating the estimated time of arrival (ETA) for the ride are all thoroughly examined and explained so all stakeholders can fully understand how they work.

1.3. References:

Through this document key resources like Uber's design guidelines and Google Maps API docs are used.

2. Glossary

1. **Ride-booking feature:** A functionality within the Uber application that enables users to book rides from their current location to their desired destination.
2. **Ride-sharing system:** A transportation system facilitated by Uber that allows users to share rides with others heading in the same direction, providing cost-effective and efficient transportation options.
3. **Transportation modes:** Different modes of transportation available for booking through the Uber application, including bikes, taxis, and cars.
4. **Estimated Time of Arrival (ETA):** The approximate time it will take for a driver to arrive at the user's location after booking a ride.
5. **Software Requirements Specification (SRS):** A document outlining the detailed requirements for the development and implementation of a software system, in this case, the ride-booking feature of the Uber application.
6. **Stakeholders:** Individuals or entities with an interest in the development and operation of the ride-booking feature, including users, developers, designers, and management.
7. **Google Maps API:** Interface provided by Google Maps, used to integrate mapping and location-based services into the Uber application for features such as real-time tracking.
8. **Transparency:** The state of being transparent, ensuring openness and clarity in communication and transactions within the Uber application.
9. **Registration:** The process by which customers create accounts and provide necessary information to access and use the Uber application.
10. **Profiles:** User accounts within the Uber application where personal information, preferences, and payment methods are stored and managed.
11. **Real-time tracking:** The feature that allows users to track the location of their assigned driver and monitor the progress of their ride in real-time.
12. **Fares:** The cost associated with booking a ride through the Uber application, which is estimated and provided to users before confirming the booking.
13. **kepler.gl:** Uber's mapping tool tracks and maps journeys on its popular ride-hailing application. This tool enables web-based visualization of large-scale spatial data.
14. **Object-oriented:** Code is written in separate modules which will enhance the program's structure, making it more maintainable.

3. User Requirements

Uber's software is crafted around the user, putting convenience, transparency and safety first in the ride-booking process. In their registration, customers should be able to they maintain their profiles without problems and to allow them to adjust their revel in and garage of price inside ways to communicate with ease.

The software should offer a variety of modes of transport such as car, bike and scooter allowing users to opt for a different mode of transport of their choice either a quick solo or group time difference Users should have access to access to real-time tracking characteristics of They are privileged to ensure they are always informed and prepared for their journey. Clear estimates on fares before booking and flexible payment methods such as credit/debit cards, e-wallets and cash must be there, so that booking can be made easier and trust can be built Each ride requires users share their experience and provide feedback , which contributes to the continuous improvement of high quality service.

Furthermore, the software can even integrate the accessibility feature into its design. This means that it will add features like screen reader compatibility and adjustable font size so that users in whatever format they may have can all use its functionality and also be able to use it by means of emergency help buttons, ride sharing options with reliable networks, appropriate ability to motivate their heritage checks Uber commitment to reinforce that with enthusiasm which is in stable traffic for all users is of utmost importance.

4. Functional Requirements

1. Display List Of Recent Destinations:

Description / Action	The system shall display a list of the recent destinations & user will be able to choose a destination from the list
Requirements / Inputs	Destinations picked by the user in the last 10 days
Input Source	The system's history of the destinations of the user
Pre-Condition	<ul style="list-style-type: none">- User has logged in- The User picked at least 1 ride since the installation of the application
Post-Condition	If the User chooses a destination, the "Select Ride Option" will be activated
Output	Display the List & Route from the Current location of the user to the desired Destination [in case the user chose from the list]

2. Set Pick-Up Point:

Description / Action	The user shall be able to set a Start location OR the Current Location of the user must be known based on GPS
Requirements / Inputs	Start/Current location
Input Source	User's key board or screen touch or GPS
Pre-Condition	<ul style="list-style-type: none">- User has logged in- In case of the user will not enter the Start location, GPS must be used
Post-Condition	The "Enter Destination" will be activated
Output	Pick-up point

3. Set Destination:

Description / Action	User must be able to set a destination for their ride manually
Requirements / Inputs	Desired destination
Input Source	User's keyboard or screen touch
Pre-Condition	"Set Pick-Up Point" was completed successfully
Post-Condition	The "Select Ride Option" will be activated
Output	Destination

4. Display Ride Options

Description / Action	The system must display all the available vehicle types (i.e. UberX, Comfort, Taxi, XL, Premier, Black) and calculate their estimated fares based on pick-up point, destination, ride option, and traffic data
Requirements / Inputs	<ul style="list-style-type: none">- Pick-up point- Available drivers within 1 km of the Pick-Up point of the user & their Ride options- Destination- Traffic data
Input Source	<ul style="list-style-type: none">- Pick-up point → Set Pick-Up Point function- Ride Options & Available drivers → System DB of the real-time drivers' locations- Destination → Set Destination function- Traffic Data → Google Maps real-time DB
Pre-Condition	"Set Destination" was completed successfully
Post-Condition	None
Output	Display for each available vehicle the Ride Information : <ul style="list-style-type: none">- The Fare Estimate- Type- Pick-up time- Price of the ride

5. Send Ride Request:

Description / Action	Users shall request a ride and receive details about their driver
Requirements / Inputs	List of the Ride Information for each available vehicle

Input Source	Output of the “Display Ride Options” function
Pre-Condition	“Display Ride Options” was completed successfully
Post-Condition	<ul style="list-style-type: none">- Add the Destination to the System History of the user’s recent destination- Send a ride request to the chosen driver
Output	Create a Ride Request with User details including: <ul style="list-style-type: none">- Pick-up point- Desired destination

5. Track Ride:

Description / Action	<ul style="list-style-type: none">- System must show the Ride Information and Driver Details in the User view- System must show the ride route in both the user and driver views using kepler.gl- System must notify the user when the driver reaches the pick-up point- System must notify user & driver when reaching the destination
Requirements / Inputs	<ul style="list-style-type: none">- Pick-up point- Destination- Ride Information- The driver’s details
Input Source	<ul style="list-style-type: none">- Output of the “Set Pick-Up Point” function- Output of the “Set Destination” function- Output of the “Display Ride Options” function- Output of the “Accept Ride” function
Pre-Condition	“Accept Ride” was completed successfully
Post-Condition	“Rate The Driver” will be activated

Output	<p>Display in the User view the Ride Information that includes:</p> <ul style="list-style-type: none">- The Fare Estimate- Type- Pick-up time- Price of the ride <p>Display in the User view the Driver Details that include:</p> <ul style="list-style-type: none">- Driver's Name- Driver's Photo- Driver's Rank- Car's type and color and Vehicle registration plate <p>Display in the User & Driver views</p> <ul style="list-style-type: none">- Journey map- Time to reach destination
---------------	---

7. Cancel Ride:

Description / Action	The system shall allow users to cancel a ride and notify the driver
Requirements / Inputs	Ride Information
Input Source	Output of the "Confirm Ride" function
Pre-Condition	"Display Ride Options" was completed successfully
Alternative Flow	<p>Flow 1: Cancellation Fees = 0 [The cancellation is within the allowed time frame]</p> <p>Flow 2: Cancellation Fees = extra time * 5 EGP [The cancellation is not within the allowed time frame]</p>
Post-Condition	<ul style="list-style-type: none">- "Track The Ride's Route" will be deactivated- Update the Driver State in the DB- Add cancellation fees to the user account

Output	<ul style="list-style-type: none">- Display the Cancellation Fees in the user view- Send cancel notification to driver account
---------------	---

8. Rate The Driver:

Description / Action	The user shall rate the driver after reaching the final destination
Requirements / Inputs	Rate from 1 to 5 starts
Input Source	The value of the slide bar which is set by the user in the user interface
Pre-Condition	The "Track Ride Route" indicates the end of the ride
Post-Condition	Update the Driver's Rank at the driver DB
Output	<ul style="list-style-type: none">- Display "Thank you for your feedback" at User view- Display The rank for the driver at the Driver view- Initiate the Home page for both views

9. Sign Up:

Description / Action	The system will create a new account for the user
Requirements / Inputs	Email, password, phone number, name, user type(User/Driver)
Input Source	The user's keyboard or screen touch

Pre-Condition	The user doesn't have an account
Post-Condition	Update User/Driver DB with a new account
Output	Initiate Home Page

10. Login:

Description / Action	The system must check on the login info to allow each type of user to enter its specific view (User view/Driver view)
Requirements / Inputs	Email, password
Input Source	User's keyboard or screen touch
Pre-Condition	Users have an account
Post-Condition	In the case of the driver account, the driver status will be set to Ready for Accepting Rides in the Driver DB
Output	Initiate Home Page

11. Accept Ride:

Description / Action	The driver shall accept the requested ride from the user
Requirements / Inputs	- Ride request - Drive Details

Input Source	<ul style="list-style-type: none">- Output of the “Send Ride Request” function- Driver DB
Pre-Condition	<ul style="list-style-type: none">- The driver has logged in.- Driver status must be Ready
Post-Condition	<ul style="list-style-type: none">- Driver status must be updated to Taken in the driver DB- Send driver information to the “Track Ride” function
Output	<p>The Ride Information includes:</p> <ul style="list-style-type: none">- Driver’s Name- Driver’s Photo- Driver’s Rank- Car’s type and color and Vehicle registration plate- Price of the ride

12. Decline Ride:

Description / Action	The system shall allow the driver to cancel a ride and notify the user.
Requirements / Inputs	Ride Information
Input Source	Output of the “Send Ride Request” function
Pre-Condition	“Display Ride Options” was completed successfully
Alternative Flow	<p>Flow 1: Cancellation Fees = 0 [The cancellation is within the allowed time frame]</p> <p>Flow 2: Cancellation Fees = extra time * 5 EGP [The cancellation is not within the allowed time frame]</p>

Post-Condition	<ul style="list-style-type: none">- “Track The Ride’s Route” will be deactivated- Update the Driver State in the DB- Add cancellation fees to the driver account- Notify the user
Output	<ul style="list-style-type: none">- Display the Cancellation Fees in the driver's view- Send cancel notification to the user account

13. Track Driver’s Location:

Description / Action	The system must track the driver's location and status.
Requirements / Inputs	<ul style="list-style-type: none">- Current Location of driver- Driver Status
Input Source	<ul style="list-style-type: none">- GPS- Driver DB
Pre-Condition	The driver has logged in.
Post-Condition	None
Output	Display a map of all the drivers’ locations and status in the driver view

5. Non-Functional Requirements

1. Dependability and Security

- a. Implement authorization technique for using email and password.
- b. The passwords of the user shall be encrypted in DBMS for security purposes.
- c. Implement robust authentication mechanisms to prevent unauthorized access.
- d. The application shall implement the User privacy services set by the international laws.

2. Scalability

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

3. Portability

- a. The application works with different environments (i.e. IOS, Android, Hawaii)
- b. The application must adapt the UI effectively considering the flexible layout, stretchable images, and pixel densities.

4. Usability

- a. According to the feedback form associated with the application, 90% of the users and drivers will rate the application's usability with a score of 4 to 5 out of 5.
- b. The driver/user can answer a quick survey about the application's sections after using the Uber application for 1h with an 85% score.

5. Performance

- a. The system must quickly match drivers to user ride requests to ensure timely service within 5 sec.
- b. The user who has an 8 Mbits internet connection speed, shall be able to enter a page of the system in less than 1 second.

6. Maintainability

The Code follows the object-oriented methodologies.

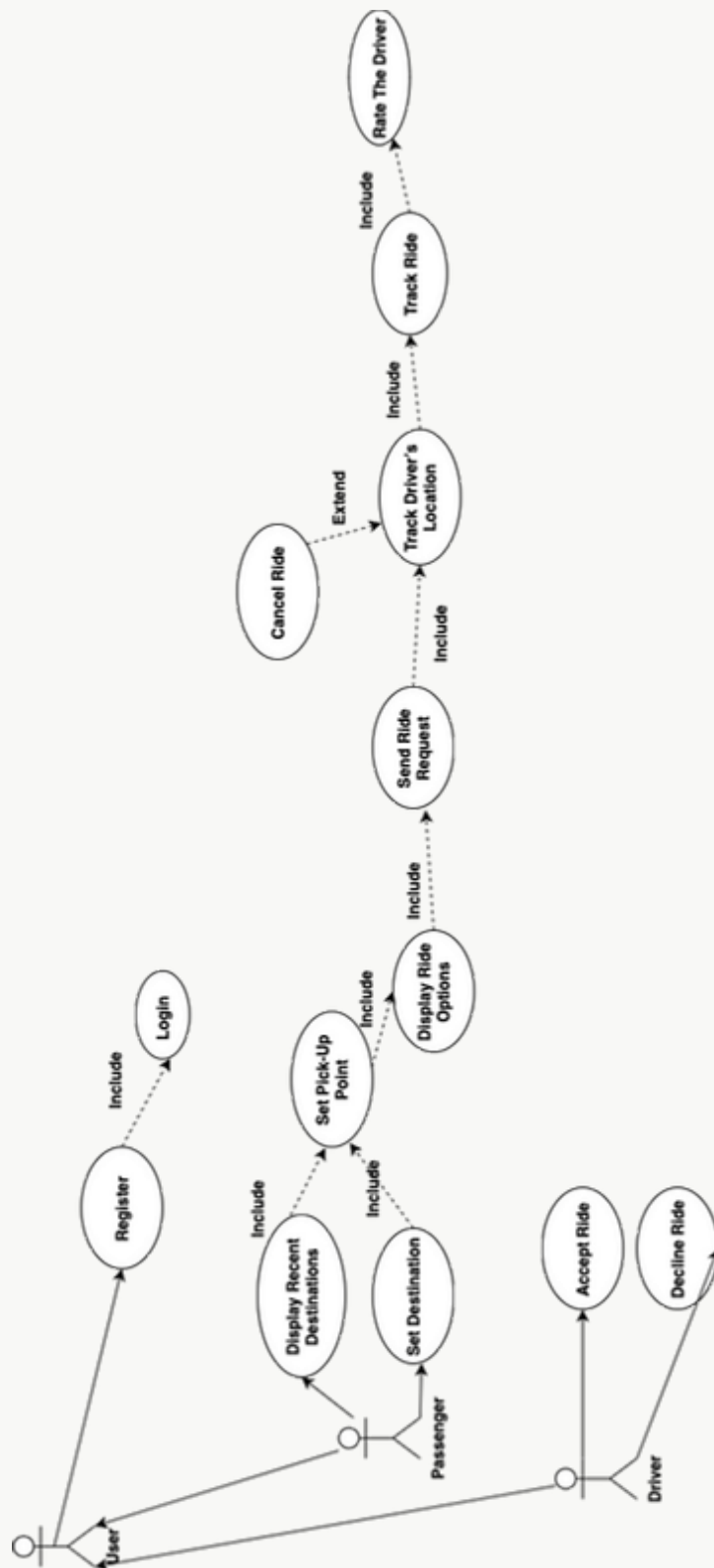
7. Reliability

When the system crashes it will return at most 1 hour.

6. Use Case Diagram

For Better View:

https://drive.google.com/file/d/1g_VchJAYxNXt7oCbJplkz07BaekHwO8M/view?usp=sharing



7. Sequence Diagram

For Better View:

https://drive.google.com/file/d/1KZ2QnliHKe65yOij3nXwzS1f8YZb-fly/view?usp=sharing_

