Software Requirements Specification

Version 1.0

December 16th, 2025

Software Requirements Specification (SRS) for SuperUber Ride-Sharing Application

Gary Kuepper

Table of Contents

1.0. Introduction

1.1. Purpose

The purpose of this document is to present the Software Requirements Specification (SRS) for a ride-sharing application. This document defines the functional and non-functional requirements of the system and serves as a reference for stakeholders, developers, and testers.

1.2. Scope of Project

The ride-sharing application enables customers to request transportation services and drivers to accept and complete ride requests using mobile devices. The system supports ride matching, pricing estimates, ride cancellation, notifications, ride completion, ratings, and monitoring of system behaviors.

1.3. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

2.0. Overall Description

2.1 Use case summary

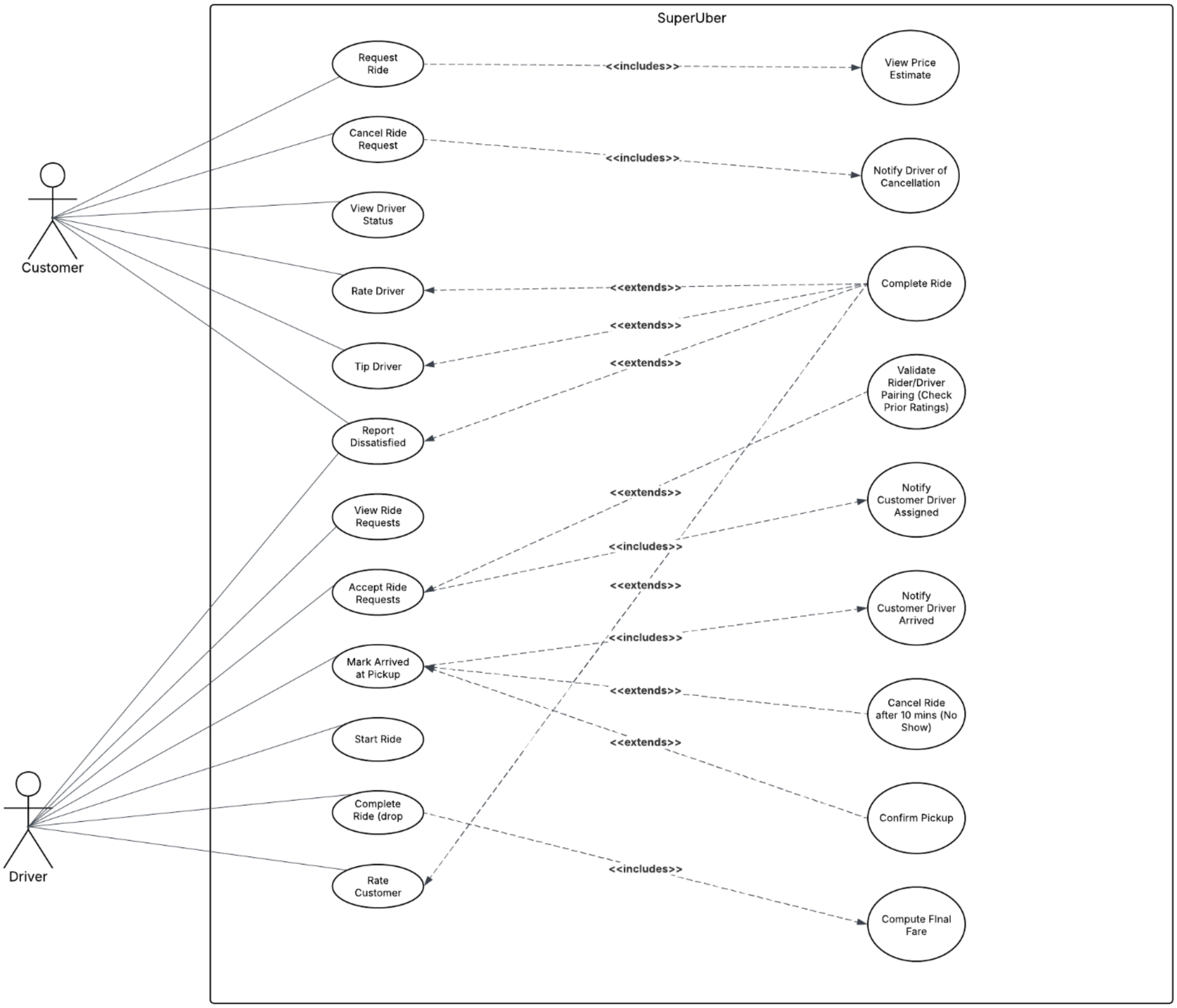

Figure 1 – Use case summary

The use case diagram illustrates the interactions between the two primary actors, the Customer and the Driver, and the ride-sharing system. Customers use the system to request and cancel rides, view price estimates and driver status, rate and tip drivers, and report dissatisfied rides, while drivers view and accept ride requests, mark arrival and pickup, complete rides, cancel pickups after a no-show waiting period, and rate customers. The diagram also represents key system behaviors such as notifications, fare calculation, and enforcement of business rules, including preventing driver–customer pairings based on prior unsatisfactory ratings, with required actions modeled using

<<include>> relationships and conditional behavior modeled using <<extend>> relationships.

2.2 Functional Requirements Specification

This section outlines the use cases for each of the active actors separately. The Customer and Driver are the primary actors in this system. Each actor interacts with the system through a mobile application to request, accept, and complete rides.
The Ride Request and Completion Process consists of the use cases listed below.

- A Customer uses the mobile application to request a ride by specifying a pickup location and destination.
- The system determines an estimated price for the ride based on distance and time of day and presents this estimate to the Customer.
- Available Drivers within a defined radius of the pickup location are notified of the ride request, including the destination and estimated price.
- A Driver selects and accepts a ride request. Once a request is accepted, it is no longer visible to other drivers.
- The Customer is notified that a Driver has accepted the request and is provided with an estimated time of arrival.
- Prior to pickup, the Customer may cancel the ride request. If canceled, the Driver is notified of the cancellation.
- When the Driver arrives at the pickup location, the Driver indicates arrival through the application.
- If the Driver waits more than a defined time period and the Customer does not arrive, the Driver may cancel the pickup.
- When the Customer is picked up, the Driver indicates that the ride has started.
- The Driver transports the Customer to the destination and indicates ride completion upon drop-off.
- Both the Customer and Driver may rate the ride, and the Customer may optionally provide a tip through the application.
- The system records ratings and monitors ride activity to identify repeated cancellations, delays, or dissatisfied rides.
- The Driver becomes available to accept new ride requests after completing or canceling a ride.

3.0. Requirements Specification

3.1 External Interface Requirements

The ride-sharing system interacts with external services to support core functionality. These external interfaces include a GPS and mapping service used to determine customer and driver locations and calculate distances, a payment processing service used to handle fare payments and tips, and a notification service used to deliver ride status updates to customers and drivers. The internal design and implementation of these external systems are outside the scope of this document.

3.2 Functional Requirements

This section provides a detailed specification of the functional requirements of the ride-sharing system. The requirements are documented as use cases that describe interactions between the system and its primary actors. Each use case includes an Xref entry that maps the requirement to a corresponding use case in the UML diagram presented in Section 2. The logical structure of the data used by the system is described separately in Section 3.3.

3.2.1 Request Ride

Use Case Name	Request Ride
Xref	Section 2 Request Ride
Trigger	The Customer selects the option to request a ride.
Precondition	The Customer is authenticated and location services are enabled on the Customer's mobile device.
Basic Path	1. The Customer enters a destination on the app 2. The system obtains the Customer's current location using GPS 3. The system calculates an estimated ride price based on distance and time of day. 4. The system displays the estimated price to the Customer. 5. The Customer confirms the ride request 6. The system creates a ride request with status Requested and makes it available within a defined radius.
Alternative Paths	A1. If the Customer cancels prior to confirmation, the use caes ends with no ride request created.

Postcondition	A ride request exists and is available for Drivers to view and accept
Exception Paths	The Customer may abandon the use case at any time prior to confirmation
Other	

3.2.2 Cancel Ride Request

Use Case Name	Cancel Ride Request
Xref	Section 2 Cancel Ride
Trigger	The Customer selects “Cancel” for an active ride request prior to pickup
Precondition	A ride request exists with status Requested or Accepted, and the passenger has not been picked up.
Basic Path	1. The Customer selects “Cancel Ride” in the app. 2. The system prompts the Customer to confirm cancellation. 3. The customer confirms cancellation. 4. The system updates the ride request status to CancelledByCustomer. 5. If a Driver accepted the ride, the system notifies the Driver that the ride has been cancelled.
Alternative Paths	A1. If the Customer declines the confirmation prompt to cancel and the ride remains active.
Postcondition	A ride request is cancelled and is no longer available to Drivers
Exception Paths	If the ride has already transitioned to PickedUp, the system prevents cancellation and informs the Customer.

Other	

3.2.3 View Ride Requests (Driver)

Use Case Name	View Ride Requests
Xref	Section 2 View Ride Requests
Trigger	The Driver opens the “View Ride Requests” view
Precondition	A Driver is authenticated, marked as available, and has location services enabled.
Basic Path	1. The system obtains Driver’s current location 2. The system displays a list of customer ride requests within a defined radius 3. For each request system dispalys pickup location, destination, and estimated price.
Alternative Paths	A1. If no ride requests are available within the radius the system displays a “No rides found” message
Postcondition	The Driver has visibility into available ride requests.
Exception Paths	None
Other	

3.2.4 Accept Ride Request

Use Case Name	Accept Ride Request
Xref	Section 2 Accept Ride Request
Trigger	The Driver selects an available ride request and chooses “Accept”.
Precondition	A ride request exists with status Requested and the Driver is marked as available.
Basic Path	1. The Driver selects a ride request from the list.

	2. The system verifies the request is still available. 3. The system checks whether the Driver and Customer may be paired based on prior ride ratings. 4. If allowed, the system assigns the Driver to the ride request and updates status to Accepted. 5. The system removes the accepted request from other Drivers requests list 6. The system notifies the Customer that a Driver is on the way and provides an ETA
Alternative Paths	A1. If the request is no longer available, the system informs the Driver and returns to the ride request list
Postcondition	The ride request is assigned to a Driver and is no longer visible to other Drivers
Exception Paths	If the pairing is blocked due to prior unsatisfactory rating between Driver and Customer, the system prevents this match and suggests another request to the Driver.
Other	

3.2.5 Mark Arrived at Pickup Location

Use Case Name	Mark Arrived at Pickup Location
Xref	Section 2 Mark Arrived at Pickup Location
Trigger	The Driver marks on the app “Arrived at Pickup location”
Precondition	A ride request exists with status Accepted and the Driver is assigned to the ride.
Basic Path	1. The Driver arrives at the pickup location 2. The Driver marks “Arrived” on app 3. The system updates the rides status to Arrived. 4. The system notifies the Customer that the Driver has arrived.

Alternative Paths	None
Postcondition	The ride request status is arrived, and the Customer has been notified.
Exception Paths	If the ride is not in Accepted status, the system prevents the action and informs the Driver.
Other	

3.2.6 Cancel Pickup (No-Show)

Use Case Name	Cancel Pickup (No-Show)
Xref	Section 2 Cancel Pickup (No-Show)
Trigger	The Driver cancels ride after waiting at pickup location
Precondition	A ride status is Arrived and the Driver is assigned to the ride.
Basic Path	1. The system displays the elapsed waiting time since the Driver marked arrival 2. If the elapsed time is greater than 10 minutes, the Driver cancels the ride on the app. 3. The system updates the ride status to NoShow 4. The system marks the Driver as available to receive new requests
Alternative Paths	A1. If the elapsed time is 10 minutes or less, the system prevents cancellation and informs the Driver of the remaining required wait time.
Postcondition	The ride is canceled due to customer no-show, and the Driver becomes available.
Exception Paths	The Driver may abandon the cancellation action at any time before confirmation.
Other	

3.2.7 Confirm Passenger Pick Up

Use Case Name	Confirm Passenger Pick Up
Xref	Section 2 Confirm Passenger Pick Up
Trigger	The Driver confirms passenger picked up on app
Precondition	A ride status is Arrived and the Customer is present for pickup
Basic Path	1. The Driver confirms the Customer is present 2. The Driver confirms pickup on app 3. The system updates the ride status to PickedUp and records pickup time.
Alternative Paths	None.
Postcondition	The ride begins
Exception Paths	If the ride status I not Arrived the system prevents action
Other	

3.2.8 Complete Ride (Drop-Off)

Use Case Name	Complete Ride
Xref	Section 2 Complete Ride
Trigger	The Driver selects “Complete Ride”
Precondition	The ride status is PickedUp and the Driver is assigned to the ride
Basic Path	1. The Driver arrives at the destination. 2. The Driver selects “Complete Ride” 3. The system updates the ride status to Completed and records the drop-off time. 4. The system computes the final fare based on distance and time of day.

	5. The system records the completed ride and marks the Driver as available for new requests.
Alternative Paths	None.
Postcondition	The ride is completed and the final fare has been computed and recorded.
Exception Paths	None.
Other	

3.2.9 Rate Ride

Use Case Name	Rate Ride
Xref	Section 2 Rate Ride
Trigger	The Customer or Driver selects option to rate completed ride.
Precondition	A ride exists with status Completed
Basic Path	1. The system prompts the rater to submit a rating for the completed ride. 2. The rater selects a rating value 3. The system records the rating and includes the ride, rater, and ratee, 4. The system updates the Customer and/or Drivers rating
Alternative Paths	A1. If the rater skips rating the use case ends with no rating recorded.
Postcondition	A rating is recorded, or the rating step is skipped.
Exception Paths	None.
Other	

3.2.10 Tip Driver

Use Case Name	Tip Driver
Xref	Section 2 Rate Ride
Trigger	The Customer selects the option to tip the Driver for a completed ride.
Precondition	A ride exists with status Completed
Basic Path	1. The system displays tip amount options 2. The Customer selects tip or enters tip amount 3. The Customer confirms tip 4. The system records the top and associates it with the ride and Driver
Alternative Paths	A1. The Customer cancels tip entry; no tip is recorded.
Postcondition	A tip is record or no tip is recorded if cancelled.
Exception Paths	If payment processing fails (external), the system informs the Customer that the tip could not be completed.
Other	

3.2.11 Report Dissatisfied Ride

Use Case Name	Report Dissatisfied Ride
Xref	Section 2 Report Dissatisfied Ride
Trigger	The Customer or Driver selects the option to report dissatisfaction with a ride.
Precondition	A ride exists with status Completed
Basic Path	1. The Customer or Driver selects “Report an Issue” for a ride 2. The system prompts the user for a reason and description

	3. The user submits the report 4. The system records dissatisfaction report and associates it with the ride and reporting user. 5. The system flags the ride for monitoring and review.
Alternative Paths	A1. The user abandons the report; no report is recorded
Postcondition	A dissatisfaction report is recorded
Exception Paths	None.
Other	

3.2.11 Monitor Ride and User Behavior

Use Case Name	Monitor Ride and User Behavior
Xref	N/A System-level requirement
Trigger	Ride lifecycle events and user actions
Precondition	The system is operational and ride activity data is being recorded.
Basic Path	1. The system records ride state transitions and associated timestamps for each ride. 2. The system monitors customers who repeatedly request and then cancel rides or are not present at the pickup location and time. 3. The system monitors drivers who do not arrive at the pickup location within the estimated arrival time. 4. The system monitors drivers who do not reach the destination within the estimated trip duration. 5. The system monitors dissatisfaction reports and unsatisfactory ratings submitted by customers or drivers.

	6. The system flags rides, customers, or drivers that exceed predefined monitoring thresholds for review.
Alternative Paths	A1. Monitoring thresholds and evaluation criteria may be updated by system configuration without modifying this use case.
Postcondition	Ride activity and user behavior are evaluated, and any detected issues are recorded and flagged for further review.
Exception Paths	None.
Other	This use case represents automated system behavior and does not involve direct interaction with system actors.

3.3. Logical Structure of the Data

This section describes the logical structure of the data used by the ride-sharing system. The data model is represented using an Entity-Relationship (ER) diagram and accompanying descriptions of each entity, including primary keys, attributes, and relationships. The logical data model supports the functional requirements described in Section 3.2, including ride requests, cancellations, pickups, drop-offs, ratings, tipping, dissatisfaction reporting, and system monitoring.

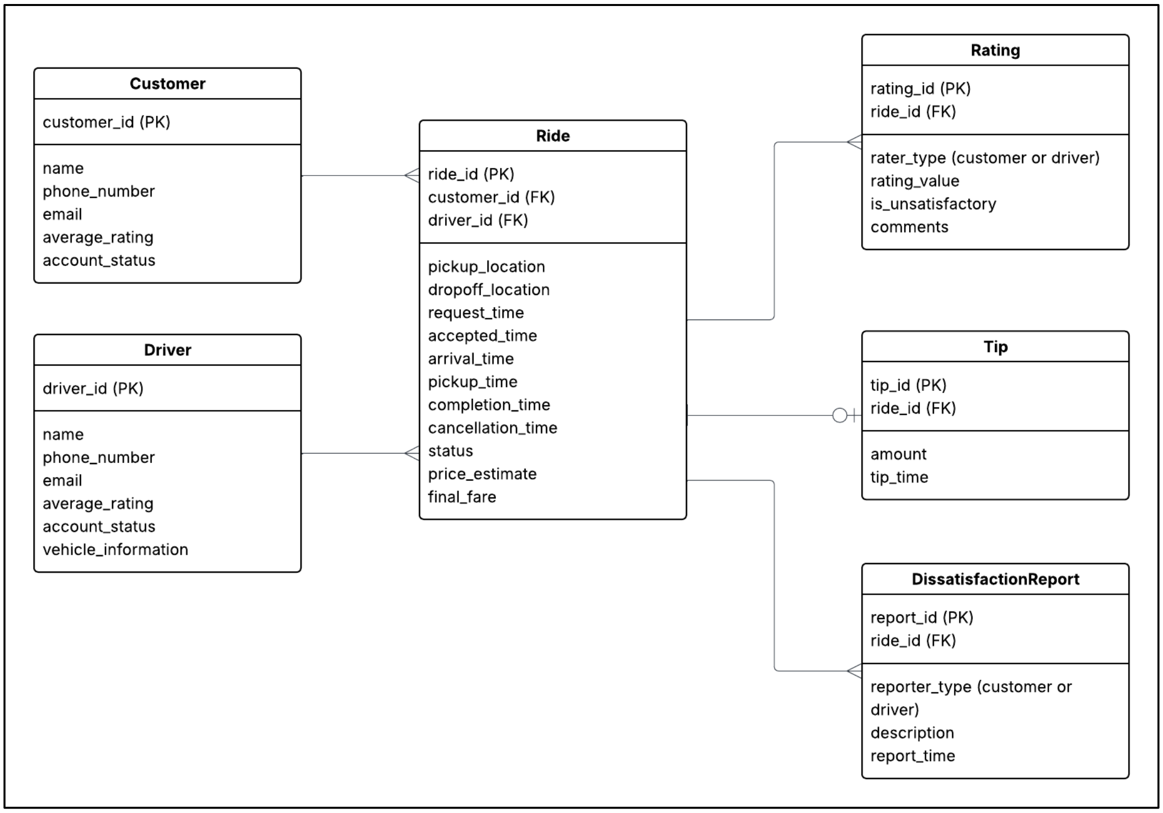

Figure 3 – Entity Relationship Diagram

The data element descriptions for each of these data entities are as follows:

Customer Data Entity

Data Item	Type	Description	Comment
customer_id	Integer	Unique identifier for each customer	Primary Key
name	Text	Customer’s full name	
phone_number	Text	Customer’s phone number	
email	Text	Customer’s email address	
average_rating	Decimal	Average rating based on completed rides	
account_status	Text	Current status of the customer account	

Driver Data Entity

Data Item	**Type**	**Description**	**Comment**
driver_id	Integer	Unique identifier for each driver	Primary Key
name	Text	Driver’s full name	
phone_number	Text	Driver’s phone number	
email	Text	Driver’s email address	
average_rating	Decimal	Average rating based on completed rides	
account_status	Text	Current status of the driver account	
vehicle_information	Text	Vehicle details provided by the driver	

Ride Data Entity

Data Item	**Type**	**Description**	**Comment**
ride_id	Integer	Unique identifier for each ride	Primary Key
customer_id	Integer	Identifier of the customer requesting the ride	Foreign Key
driver_id	Integer	Identifier of the driver assigned to the ride	Foreign Key, nullable until accepted
pickup_location	Text	Pickup location for the ride	
dropoff_location	Text	Drop-off location for the ride	
request_time	DateTime	Time the ride was requested	
accepted_time	DateTime	Time the driver accepted the ride	
arrived_time	DateTime	Time the driver arrived at pickup	
pickup_time	DateTime	Time the passenger was picked up	
completion_time	DateTime	Time the ride was completed	
cancellation_time	DateTime	Time the ride was canceled	
status	Text	Current state of the ride	
price_estimate	Decimal	Estimated price shown before acceptance	

final_fare	Decimal	Final calculated fare	

Rating Data Entity

Data Item	Type	Description	Comment
rating_id	Integer	Unique identifier for each rating	Primary Key
ride_id	Integer	Identifier of the related ride	Foreign Key
rater_type	Text	Indicates whether the rater is a customer or driver	
rating_value	Integer	Numeric rating value	
is_unsatisfactory	Boolean	Indicates if the rating is unsatisfactory	
comments	Text	Optional comments about the ride	

Tip Data Entity

Data Item	Type	Description	Comment
tip_id	Integer	Unique identifier for each tip	Primary Key
ride_id	Integer	Identifier of the related ride	Foreign Key
amount	Decimal	Tip amount provided by the customer	
tip_time	DateTime	Time the tip was recorded	Optional

DissatisfactionReport Data Entity

Data Item	Type	Description	Comment
report_id	Integer	Unique identifier for each report	Primary Key
ride_id	Integer	Identifier of the related ride	Foreign Key
reporter_type	Text	Indicates whether the reporter is a customer or driver	
description	Text	Description of the issue reported	
report_time	DateTime	Time the report was submitted	

3.4. Nonfunctional requirements

The ride-sharing system will operate on cloud-based servers with high-speed Internet connectivity. The specific hardware used to host the system will be determined by the service provider. The system assumes the use of external services for GPS location, mapping, and payment processing to support ride requests, navigation, and fare transactions. The performance of the Customer's and Driver's mobile applications will depend in part on the hardware and network connectivity of the user's device rather than characteristics of the system itself.

The Customer and Driver applications will run on mobile devices using standard smartphone operating systems. These devices are expected to provide location services, network connectivity, and notification capabilities. The reliability of these features is dependent on the user's device and network and is considered external to the system.

The servers hosting the ride-sharing system will provide security controls to prevent unauthorized access to system data. Write and update access will be restricted to authenticated users and system components. Communication between mobile applications and backend services will be encrypted. There is no restriction on read access to publicly available information.

Payment processing and notification delivery are handled by external services and are therefore outside the direct control of this system. The security and availability of these services are assumed to be provided by their respective vendors.