Group 11

Software Requirements Specifications

–Ride-hailing and Dispatching platform

Table of Contents

1. Introduction

1.1 Purpose

"Ride-hailing and Dispatching platform" is designed to provide convenient transportation services to our valued customers. The target audience includes both passengers and drivers who seek a user-friendly and reliable platform for their transportation needs. The goal of this system is to provide a reliable and user-friendly platform that simplifies the process of booking rides and offers various transportation options to meet the customers' diverse needs.

1.2 Scope

"Ride-hailing and Dispatching platform" is the name of the product which is going to be produced. The product will mainly focus on the following things:

- Provide a user-friendly interface for passengers to book rides, including selecting vehicle types, specifying pickup and drop-off locations, choosing payment methods and rating the drivers,
- Enable passengers to view detailed driver information, including vehicle types, license plate numbers, ratings, and track their booked orders.
- Enable drivers to receive and accept ride requests, report their arrival at the destination, and settle the fare.

The ride-hailing and dispatching platform will be utilized by passengers and drivers who seek convenient, reliable, and efficient transportation services. The platform aims to provide the following benefits, objectives, and goals:

- Convenience: Passengers can easily book rides from their preferred locations, select suitable vehicle types, and make seamless payments. Additional features like saving frequent destinations and carpooling options cater to various needs.
- Reliability: Passengers can rely on the platform to connect them with verified drivers and track the progress of their booked rides.able vehicle types, and make seamless payments.
- Efficiency: Drivers can efficiently manage ride requests, navigate to destinations, and settle fares, enhancing their overall productivity.

1.3 Definition, acronyms, and abbreviations

1.3.1 User: An individual who accesses and interacts with the ride-hailing platform as a passenger or driver.

1.3.2 Valet Service: A service provided by drivers to transport a customer's vehicle from one location to another.

1.3.3 Carpooling: The practice of multiple passengers sharing a ride in a single vehicle, usually on similar routes.

1.3.4 Third-party Payment Gateway: An external service that handles the secure processing of payment transactions.

1.3.5 GPS: Global Positioning System, a satellite-based navigation system that determines precise location coordinates.

1.4 References

1.4.1 « IEEE Recommended Practice for Software Requirements Specifications ».(1998) The Institute of Electrical and Electronics Engineers, Inc. To access the document, you can visit the IEEE's official website or reach out to them for further information on obtaining the recommended practice.

1.4.2 〈軟體需求規格書〉(2014)。中央研究院。Please refer to their official website or contact them directly for access to the document.

1.5 Overview

The remaining content of the SRS will be organized as follows:

1.5.1 Overall Description:

This section will provide an overview of the system, including Product Perspective, Product Functions, User Characteristics, Constraints, and Assumptions and Dependencies.

1.5.2 Specific Requirements:

This section will outline the detailed requirements of the ride-hailing platform, covering both functional and non-functional aspects. It will include Functional Requirements and Non-functional Requirements.

2. Overall Description

2.1 Product perspective

The system operates as a standalone platform, accessible by users on their mobile devices. It interfaces with internal and external services such as (a).GPS (b).Payment gateways (c).Database (d).Feedback mechanisms (e).Carpooling

2.1.1 GPS

Develop a GPS system for tracking the users and cars location precisely.

2.1.2 Payment gateways

Develop 3 kinds of payment gateway for transaction processing including (a).Cash (b).Digital Pay (c).Credit card.

2.1.3 Database

Develop 2 separate databases to store the passengers and drivers personal history and information respectively.

2.1.4 Feedback mechanism

Passengers can rate the drivers, and drivers could check the feedback stored in the system.

2.1.5 Carpooling

Develop 2 main algorithms for carpooling. (a).Path algorithm (b).Payment algorithm

2.1.5.1 Path algorithm

Search the passengers with the same or similar route, and get the best pick-up and drop-off position.

2.1.5.2 Payment algorithm

Compute the payment for each passenger taking the same car according to pick-up and drop-off position.

2.2 Product function

The Product function includes the functions:(a).User Registration and Login. (b).Ride Booking. (c).Driver Information Display. (d).Rating and Reviews. (e).Ride Tracking. (f).Driver Ride Management. (g).Integration with Payment Gateways. (h).Carpooling

2.2.1 User Registration and Login

The system allows both passengers, and drivers, to register and create their accounts.

2.2.2 Ride Booking

Passengers can easily book rides by selecting vehicle types, specifying locations, preferred payment methods, booking the ride in the future, and requiring valet service.

2.2.3 Driver Information Display

Passengers can view detailed driver information, including vehicle types, car plate numbers, and ratings.

2.2.4 Rating and Reviews

Passengers can rate their ride experience and provide feedback on drivers. The system stores and displays the driver ratings and reviews to help maintain a reliable and safe environment.

2.2.5 Ride Tracking:

Passengers can track the progress of their booked rides, including real-time updates on the driver's location, estimated arrival time, and other ride details.

2.2.6 Driver Ride Management

Drivers receive ride requests from passengers and can accept or decline them based on availability.

2.2.7 Integration with Payment Gateways

The system integrates with third-party payment gateways. Payment methods include cash, digital payment options, and credit cards.

2.2.8 Carpooling

Carpooling options are available, where the system matches passengers with similar routes to share rides.

2.3 User characteristics

There are 3 kinds of users, including passengers, drivers and administrators. Passengers only need to be familiar with mobile use, and drivers need to have a license and also familiar with the platform. The administrators should be proficient with information management. They should understand the platform's administrative tools, such as monitoring system performance, and addressing user inquiries or issues.

2.4 Constraints

2.4.1 Regulatory Policies

The ride-hailing platform must comply with applicable laws, regulations, and policies related to transportation services, data privacy, and financial transactions.

2.4.2 Hardware Limitations

The platform should consider the hardware capabilities of these devices.

2.4.3 Interfaces to Other Applications

The ride-hailing platform may need to integrate with third-party applications or services.

2.4.4 Audit Functions

The system should incorporate auditing capabilities to track and log important activities, such as user registration, ride bookings, payment transactions, and feedback submissions.

2.4.5 Control Functions

The platform should provide administrative controls to manage users, verify driver credentials, and handle exceptional scenarios such as ride cancellations, disputes, or emergency situations.

2.4.6 Safety and Security Considerations

This includes implementing measures to protect user information, secure payment transactions…etc.

2.5 Assumption and dependencies

2.5.1 Availability of Mobile Devices

It is assumed that users will have access to compatible mobile devices (smartphones or tablets) with internet connectivity to use the ride-hailing platform.

2.5.2 Availability of Location Services

The ride-hailing platform assumes that users' mobile devices have enabled location services (e.g., GPS) to provide accurate information about their current location.

2.5.3 Integration with Third-Party Services

The ride-hailing platform will integrate with external services such as mapping services, payment gateways, and driver verification systems.

3. Specific Requirement

F1. Log in

Description: The system should check if the username and the password is in invalid format.

F1.1. Enter the username and password

Input: type the username and password

Output: prompt to show if the username and password is in valid format or not

F1.2. Press the Login button

Input: press/ click the button

Output: prompt to show if the username and password exist or not.

F2. Book a ride

Description: The system shall find a driver to take the order of the passenger

F2.1. Choose the place to Pick-up location and Drop-off location

Input: click the map or type the address to indicate the location

Output: prompt to show the time it takes for the driver to arrive

F2.2. Press the confirm button

Input: press/ click the button

Output: prompt to show if the order is successful or not.

F3. Driver's information

Description: The system should be able to show the passenger the information of the driver.

Input: click the "driver's information" button

Output: Show the page which contains driver's informationF2. Book a ride

F4. Rating

Description: The system shall let the passenger to rate the driver based on the ride experience

F4.1. select 1 to 5 stars to rate the driver

Input: click the stars

Output: prompt to show how many stars the passengers select

F4.2. Press the confirm button

Input: press/ click the button

Output: prompt to show if the rating is successful or not

F5. Look up the booked order

Description: The system shall let the passenger to see the order which has been booked

F5.1. switch to the history order page

Input: swipe the page

Output: page switch to the history order page

F5.2. choose the booked order the passenger would like to see

Input: press one of the booked order

Output: page switch to the chosen order page to show the details

F6. Save the location

Description: Select a location on the map or enter the text of the address to save it in the system.

F6.1 Select address

Input: The position on map

Output: The address full name

F6.2 Enter address

Input: The address text

Output: A mark show on the map indicating the location

F6.3 Save

Input: Press the “save” button or “cancel” button.

Output: Save the position into the system if press “save” and the address is valid, otherwise, go back to the function selected menu.

F7. Valet Service

Description: If the passengers have his or her own car, however not suitable for driving, passengers can book the service to look for the driver to drive passengers’ car to transport.

F7.1 Select valet service

Input: Press “valet” button.

Output: The system shall go into valet booking mode.

F7.2 Choose the place to Pick-up location and Drop-off location

Input: Click the map or type the address to indicate the location

Output: Prompt to show the time it takes for the driver to arrive

F7.3 Choose the payment method

Input: Select the payment method by toggle circle.

Output: Show the payment page.

F7.4 Press the confirm button

Input: Press “confirm” button

Output: Prompt to show if the order is successful or not.

F8. Carpooling

Description: Passengers should choose if their journey will be shared with other passengers or not.

F8.1 Choose to share the journey or not

Input: Click the checkbox to select or remove the selection.

Output: fill the checkbox if the checkbox is blank, otherwise, blank it.

F8.2 Choose the place to Pick-up location and Drop-off location

Input: Click the map or type the address to indicate the location.

Output: Prompt to show the time it takes for the driver to arrive. If you choose to share the journey, show the place to take the ride.

B)

1. Validity:

- To ensure validity, it is important to thoroughly understand the customer's needs and expectations. This can be achieved through market research, customer surveys, interviews, and user feedback.
- Conduct a comprehensive analysis of the customer requirements and map them to the system functions. Identify any gaps or missing functionalities.
- Validate the system functions by comparing them to the identified customer needs. Ensure that the functions provided by the system align with and best support those needs.

2. Consistency:

- To identify requirements conflicts, perform a thorough review of all the requirements and specifications.
- Look for contradictions or inconsistencies between different requirements. For example, if one requirement states that drivers should be able to accept ride requests, but another requirement states that drivers cannot decline requests, there is a conflict.
- Resolve conflicts by discussing and prioritizing requirements with stakeholders. It may be necessary to modify or redefine requirements to ensure consistency throughout the system.

3. Completeness:

- Review the customer requirements and ensure that all necessary functions are included in the system.
- Use techniques such as requirement traceability matrix or cross-referencing to verify that each customer requirement has been addressed by at least one system function.
- Conduct regular reviews and discussions with stakeholders to identify any missed or overlooked functions. Gather feedback from potential users to validate the completeness of the system requirements.

4. Realism:

- Assess the available budget and technology to determine if the requirements can be implemented within these constraints.
- Collaborate with technical experts to evaluate the feasibility of implementing each requirement. Consider factors such as development resources, technology limitations, and time constraints.

- Perform cost and technology analyses to determine if any adjustments or compromises are necessary. This may involve prioritizing certain requirements or exploring alternative solutions.

5. Verifiability:

- Establish measurable criteria and metrics to verify the achievement of each requirement.
- Define test cases and procedures that can be used to validate and verify the system's compliance with each requirement.
- Conduct testing activities, such as functional testing, integration testing, and user acceptance testing, to ensure that the requirements are met.
- Implement a verification process to regularly review and validate the system requirements against the implemented functionalities.