

# Deliverable #1 Software Requirement Specification (SRS)

SE 3A04: Software Design II – Large System Design

**Tutorial Number:** T01

**Group Number:** G6

**Group Members:**

- Jane Klavir
- Nathan Luong
- Areez Visram
- Jennifer Ye

## 1 Introduction

The following document is dedicated to showcasing the requirements of the taxi carpool app requested by a local taxi company. The app is dedicated to make rides cheaper and more convenient for their riders. This app is also meant to be a long-term project for the company, meaning this **SRS** is a living document which has the possibility of changing in the future. This document will explain the important stakeholders involved, requirements and viewpoints of each while also explaining the main functions of the app.

### 1.1 Purpose

This **SRS** is meant for developers, stakeholders and direct players in the planning and development of the app and should be understandable for all those involved. This **SRS** will further break down the functionalities, constraints, use cases and overall design of the project in further detail.

### 1.2 Scope

To develop this project the production of the main taxi request and offer app called TaxiCarPool, the Conversation Prompt Generator and a variety of databases will have to be produced. There will not be a need to produce a payment system into the app as this will be handled by the taxi company payment systems. The project will only project the cost of the ride but will not handle the payments as it is not part of the app's main functionality.

The TaxiCarPool app will perform the main functions of the app. This will include allowing users to request a carpool taxi, receive a list of possible rides to select and allow current rides to allow carpooling in their car. This product's objective is to perform the main functions of matchmaking users to a carpool to save money.

The Conversation Prompt Generator will randomly generate a conversation starter for the riders to use based on their profile. Their profile will specify if they want to use this feature and if so, what they are interested in. The generator will not be forced upon any of the users and can be switched off if desired. The goal of this product is to create a friendly environment when carpooling with others that one may not know. This function is mostly dedicated to benefit those who are carpooling to a further distance.

The creation of a database will hold all the necessary information for the app to function, making it easily

accessible and manageable. The information being stored can range from user information; name, email, phone number, to riding history, to preferences of taxi times, models, or other passengers.

### 1.3 Definitions, Acronyms, and Abbreviations

BE: Business Event is something that occurs between the client/stakeholder and the system.

SRS: Software Requirements Specification is a document that explains the requirements of the software being developed.

VP: Viewpoint, often in reference to a stakeholder/client interested in the system and how they view the business event.

CF: Creative Feature, an indication to one of the creative/innovative features

### 1.4 References

- Johns, K. (2022, July 7). Why design is the most important factor in a mobile app development?: ISHIR- mobile application development company india. ISHIR. Retrieved February 11, 2023, from <https://www.ishir.com/blog/9633/why-design-is-the-most-important-factor-in-a-mobile-app-development.htm>
- Office of the Privacy Commissioner of Canada. (2018, January 31). Summary of privacy laws in Canada. Office of the Privacy Commissioner of Canada. Retrieved February 25, 2023, from <https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/02-05-d-15/heading-0-0-2-1>
- Schoenfelder, N. (n.d.). What are good latency & ping speeds? Graphical Network Monitoring and Troubleshooting. Retrieved February 11, 2023, from <https://www.pingplotter.com/wisdom/article/is-my-connection-good>
- statcounter. (n.d.). Mobile Operating System Market Share Worldwide. StatCounter Global Stats. Retrieved February 11, 2023, from <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- UGEM. (2020, June 15). What are the core features of Minimalist app design? UGEM. Retrieved February 11, 2023, from <https://ugem.design/blog/minimal-app-design>

### 1.5 Overview

This **SRS** is structured to first give background information about the app. This will include how the app will interface with the taxi companies, along with the general functionalities and assumptions that are being made. The document will then follow with showcasing one of the main business events and its related use cases. This will explain how the main function will work across different scenarios and stakeholders in a visual manner. This will be further expanded upon in more detail throughout the document. It will include more business events with a variety of viewpoints and scenarios for both the function shown in the use case diagram and other main function of the project. This document will conclude by explaining the non-functional requirements involving presentation, performance, usability, security, and maintainability notes of the project.

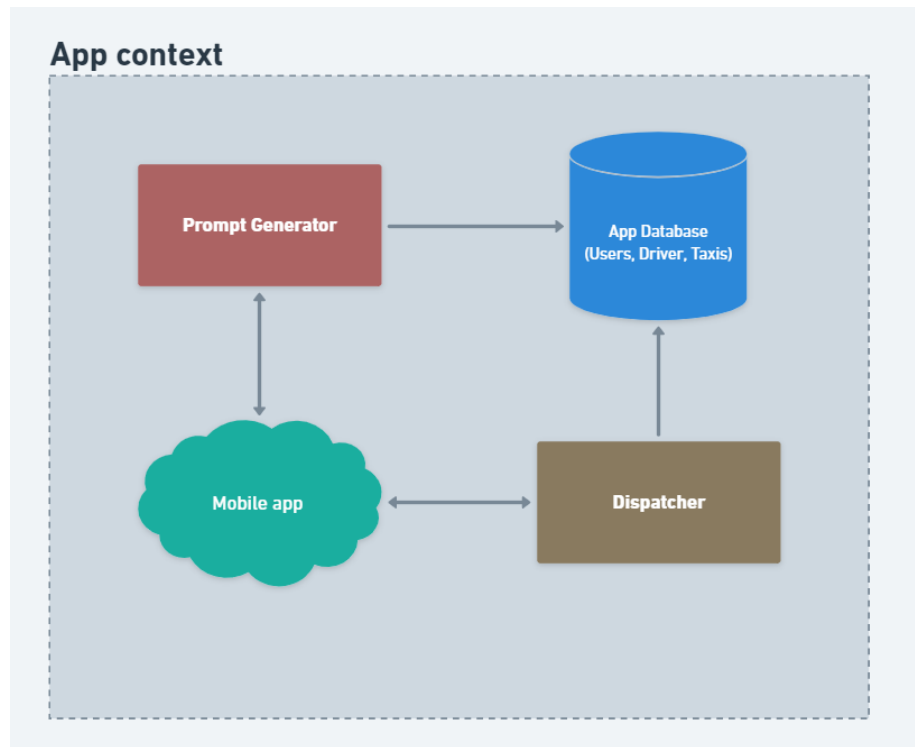
## 2 Overall Description

### 2.1 Product Perspective

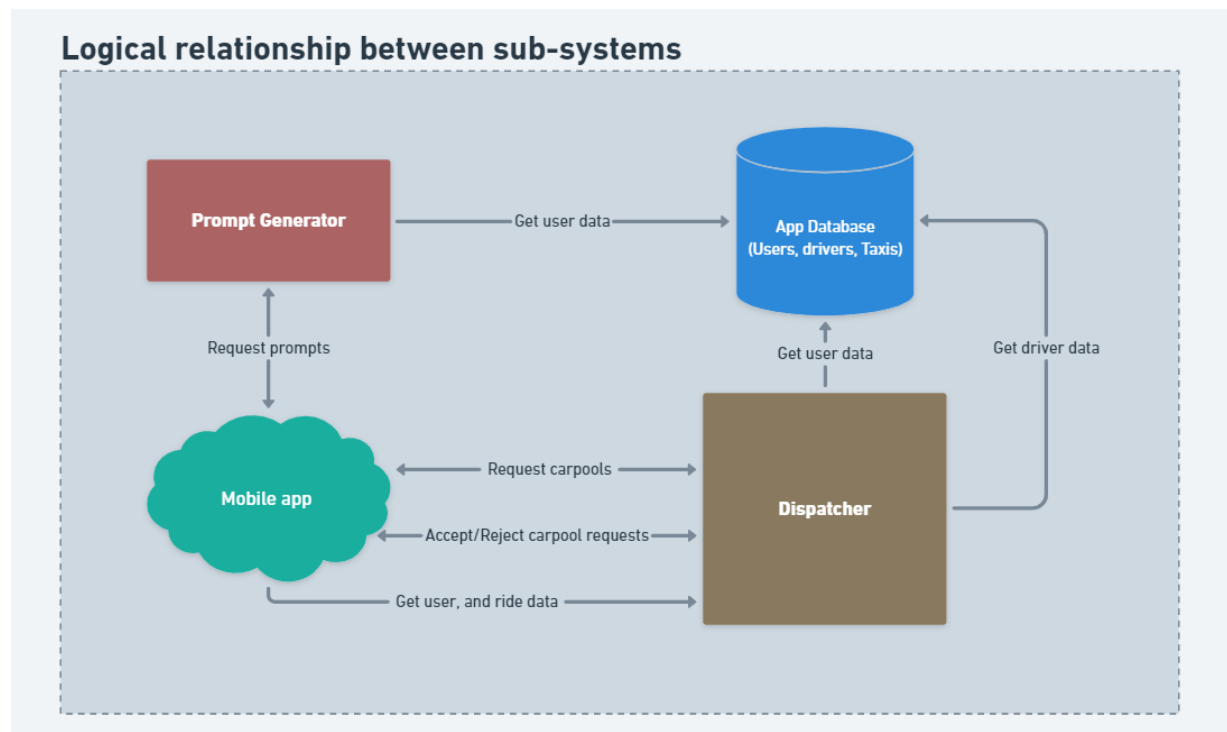
- a) The product being developed is an Android application which empowers the ability to book carpools via a user-friendly interface. The application will securely store customer personal information such as carpool request histories, and personal data inputted by the user. The product is not self-contained since its functionality depends on Google Maps for mapping integration. The product scope mainly covers a match making functionality to match potential carpools together. The feature will be implemented under a centralized dispatcher which communicate with the mobile app and user's database. Fare

determination will be covered under the scope of the app, however payment processing is external to the system.

b) The general interaction can be visualized as follows:



## 2.2 Product Functions



- a) **User registration and profile editing:** Users can register to use the taxi service and later can edit or remove their information.
- b) **Requesting taxi:** Users can enter search criteria to request a taxi. Based on the user's profile, the app will return a list of best-fit matches. Users choose the desired match and other carpooler will be notified.
- c) **Reviewing offer:** The app displays offers information to the users to choose between.
- d) **Taxi check-in:** Users can scan or enter the taxi ID when it arrives to check in and use the service.
- e) **Fare calculation and display:** Trip prices are calculated, using a fixed rate per kilometer, and presented to the users before reaching their destination.
- f) **Google Maps Integration:** The app interface will use Google Maps to visualize pick-up, drop-off location.
- g) **Data encryption:** The software system will encrypt all messages when transmitting. Important information like password or user's postal code will also be encrypted when storing within the database.
- h) **Trips rating:** After each trip, users can enter their rating to help improving the service.
- i) **Prompts Generation:** Users of the taxi service can engage in a prompt-based game while carpooling. When the users activate prompts, the app will return a list of personalized prompts for carpoolers to engage with when they are on the same car.

## 2.3 User Characteristics

- a) Riders should be comfortable with using Android mobile devices.
- b) Riders might be sensitive with prices and therefore opt out for carpools instead of regular taxi services.
- c) Riders might be outgoing, socially engaged, and interested in meeting new people.
- d) Riders are most likely live in a big city since taxi services are highly available, and parking fees are expensive.
- e) Riders might have a busy schedule since taking public transport or maintaining their own car takes time and effort.

## 2.4 Constraints

- a) **Geographical:** The app can only operate on the land which the taxi service operate.
- b) **Internet:** The app functionality relies heavily on the internet to book, checkin, and view taxi's location.
- c) **Taxi Regulation:** Since the app is working in conjunction with the taxi provider, it must follow taxi law and regulation. For example, the fare structure used by the app must be approved by the Board of Transportation for some cities.
- d) **Data Privacy Regulation:** Personal data stored within the app needs to be encrypted and processed in a way that complies with the Canadian Personal Information Protection and Electronic Documents Act.
- e) **Technological:** Since the app is dependent on Google Maps, it can not extend to provide a better mapping experience than what Google Maps offers.

## 2.5 Assumptions and Dependencies

### a) Assumptions

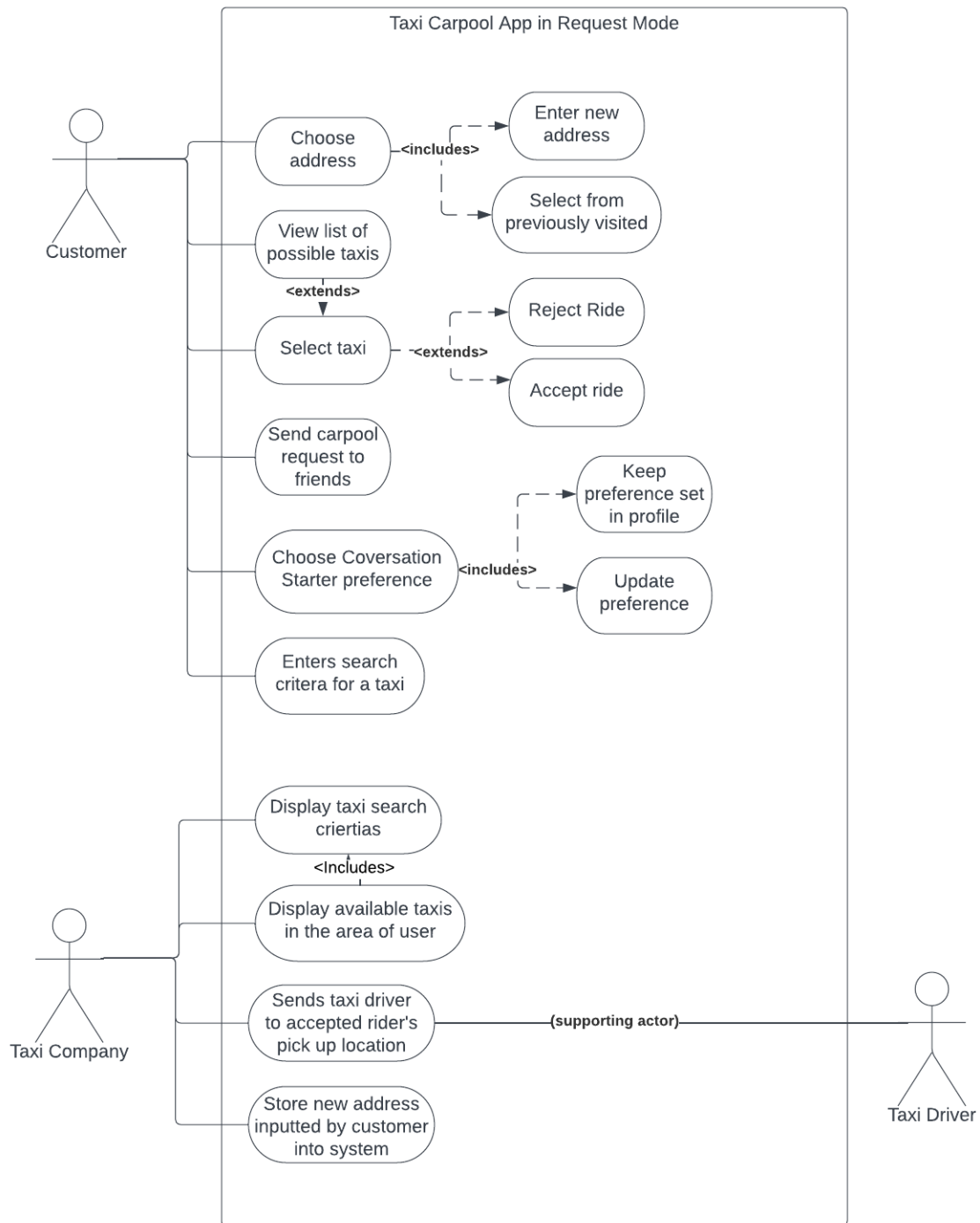
- Users have an android mobile device. If users uses IOS or other operating system, the app would not be available.
- Users have a highly available internet connection. The app will not function when internet connection is not present.
- Drivers information are provided and available via the app's database. If the assumption does not hold, driver's data will be pulled from an external data source.
- Prices are determined via a fixed rate per kilometer.
- Payment processing is external and not under the scope of the application. If the assumption does not hold, an integration with external Payment Service Provider need to be implemented.

## 2.6 Apportioning of Requirements

### a) Real-time trip tracking via Google Maps Interface.

### b) Accurate personalized prompts based on user's trips history, behaviour, and more.

### 3 Use Case Diagram



## 4 Functional Requirements

Note: Requirements in blue are “extra” and can be ignored in initial design stages

### BE1. Customer creates new user profile

#### VP1.1 Customers

$S_1$  System asks for customer’s information

$E_1$  Customer inputs their information

$S_{1.1}$  Certain information must be unique

$S_2$  System asks customer to verify email or phone number

$E_2$  Customer verifies email or phone number

$S_3$  System stores customer profile in database

#### VP1.2 Taxi company

N/A

#### VP1.3 Taxi drivers

N/A

#### VP1.4 Investors

$S_1$  Every new 1000 sign-ups, system sends notification to investors

#### VP1.5 Legal

$S_1$  System securely stores user data to comply with GDPR privacy

#### Global Scenario for BE1

$S_1$  System asks for customer’s information

$E_1$  Customer inputs their information in all mandatory fields

$E_{1.1}$  Certain information must be unique

$S_2$  System asks customer to verify email or phone number

$E_2$  Customer verifies email or phone number

$S_3$  System registers customer profile as user in system

$S_{3.1}$  System sends new user data to taxi company

$E_{3.1}$  Taxi company logs data in their database

$S_{3.2}$  System securely stores user data to comply with GDPR privacy

$S_{3.3}$  Every new 1000 sign-ups, system sends notification to investors

### BE2. User edits their profile

#### VP2.1 Customers

$S_1$  System asks customer to authenticate themselves

$E_1$  Customer completes authentication process

$S_2$  System displays user profile

$E_{2.1}$  Customer edits information

$E_{2.2}$  Customer saves edits, given that all mandatory information is entered

$S_3$  System saves user profile

#### VP2.2 Taxi company

N/A

#### VP2.3 Taxi drivers

N/A

VP2.4 Investors

N/A

VP2.5 Legal

$S_1$  System securely stores user data to comply with GDPR privacy

Global Scenario for BE2

$S_1$  System asks customer to authenticate themselves

$E_1$  Customer completes authentication process

$S_2$  System displays user profile

$E_{2.1}$  Customer edits desired information

$E_{2.2}$  Customer saves edits, given that all mandatory information is entered

$S_3$  System saves user profile

$S_{3.1}$  System securely stores user data to comply with GDPR privacy

**BE3.** Taxi carpool is requested

VP3.1 Customers

$S_1$  System asks customer to input their destination

$E_1$  Customer inputs destination

$S_2$  System asks customers to enter search criteria

$E_2$  Customer inputs search criteria

$S_3$  System returns best matches

$E_3$  Customer selects taxi to send their carpool request to

$S_{4.1}$  System sends customer's request to carpool offerer

$S_{4.2}$  Customers are returned to selection phase (i.e. after inputting destination) if request is denied, branch to  $S_2$

$S_{4.3}$  Customers notified if request is accepted

$E_4$  Customers can track location of designated cab

VP3.2 Taxi company

$S_1$  System computes estimated payment split based on taxi company's fares

$S_2$  System notifies taxi company of new carpool

VP3.3 Taxi drivers

N/A

VP3.4 Investors

N/A

VP3.5 Legal

N/A

Global Scenario for BE3

$S_1$  System asks customers to input their destination

$E_1$  Customer inputs destination

$S_{2.1}$  System computes estimated payment split based on taxi company's fares

$S_{2.2}$  System asks customers to input search criteria

$E_3$  Customer inputs search criteria

$S_4$  System returns best matches

$E_4$  Customer selects taxi to send their carpool request to

$S_{5.1}$  System sends customer's request to carpool offerer



*S*<sub>5.2</sub> Customers are returned to selection phase (i.e. after inputting destination) if request is denied, branch to *S*<sub>2</sub>

*S*<sub>5.3</sub> Customers notified if request is accepted

*E*<sub>5</sub> Customers can track location of designated cab

**BE4.** Taxi carpool is offered

VP4.1 Customers

*S*<sub>1</sub> System asks customer to scan QR code

*E*<sub>1</sub> Customer scans QR code

*S*<sub>2</sub> System asks customer to input offer information

*E*<sub>2</sub> Customer inputs offer information

*S*<sub>3</sub> System shows offer to other customers looking to join a taxi

*E*<sub>3</sub> Customer looking to join a taxi requests to join carpool

*S*<sub>4</sub> Dispatcher returns potential match, displaying updated estimated fare, distance, time, and optimality measure

*E*<sub>4.1</sub> Customer accepts match

*E*<sub>4.2</sub> Customer rejects match, return to *S*<sub>3</sub>

*E*<sub>4.3</sub> Customer aborts offer mode

*S*<sub>4</sub> Offer is removed from list of active offers

VP4.2 Taxi company

*S*<sub>1</sub> New offer sent to dispatcher

*E*<sub>1</sub> Dispatcher stores offer in pool of live offers

VP4.3 Taxi drivers

*S*<sub>1</sub> Driver is given pick-up location and directions (i.e. detour) for new customer in carpool

*E*<sub>1</sub> Driver follows directions of detour to pick-up carpooler

*S*<sub>2</sub> Once detour is complete and carpooler is in the taxi, system displays original destination with updated directions

VP4.4 Investors

N/A

VP4.5 Legal

N/A

Global Scenario for BE4

*S*<sub>1</sub> New offer sent to dispatcher

*E*<sub>1</sub> Dispatcher stores offer in pool of live offers

*S*<sub>2</sub> System asks customer to scan QR code

*E*<sub>2</sub> Customer scans QR code

*S*<sub>3</sub> System asks customer to input offer information

*E*<sub>3</sub> Customer inputs offer information

*S*<sub>4</sub> System shows offer to other customers looking to join a taxi

*E*<sub>4</sub> Customer looking to join a taxi requests to join carpool

*S*<sub>5</sub> Dispatcher returns potential match, displaying updated estimated fare, distance, time, and optimality measure

*E*<sub>5.1</sub> Customer accepts match

*S*<sub>5.2</sub> Offer is removed from list of active offers

*S*<sub>5.2.1</sub> Driver is given pick-up location and directions (i.e. detour) for new customer in carpool

*E*<sub>5.2.1</sub> Driver follows directions of detour to pick-up carpooler

*S*<sub>5.2.2</sub> Once detour is complete and carpooler is in the taxi, system displays original destination with updated directions

*E*<sub>5.2</sub> Customer rejects match, return to *S*<sub>3</sub>

*E*<sub>5.3</sub> Customer aborts offer mode

**BE5.** Taxi carpool arrives at destination

VP5.1 Customers

*S*<sub>1</sub> System displays fare to each customer

*E*<sub>1</sub> Customer pays taxi driver [out of scope for system]

*S*<sub>2</sub> Driver inputs that payment is complete

*S*<sub>3</sub> System prompts customer to rate the other customers they shared a ride with, with an option to leave a review

*E*<sub>3</sub> Customers input rating and optional review

*S*<sub>4</sub> System stores rating and review in customers' profile

VP5.2 Taxi company

*E*<sub>1</sub> Drivers obtain payment from customers

*E*<sub>2</sub> Driver inputs that payment is complete

VP5.3 Taxi drivers

*E*<sub>1</sub> Driver obtains payment from customers

*E*<sub>2</sub> Driver inputs into system that payment is complete

*S*<sub>2</sub> System logs end of carpool

VP5.4 Investors

*S*<sub>1</sub> Every new 10 000 completed carpools, system sends notification to investors

VP5.5 Legal

N/A

Global Scenario for BE5

*S*<sub>1</sub> System displays fare to each customer

*E*<sub>1</sub> Customer pays taxi driver

*E*<sub>1.1</sub> Drivers obtain payment from customers [out of scope for system]

*E*<sub>1.2</sub> Driver input that payment is complete

*S*<sub>2</sub> Once payment approved, system prompts customer to rate the other customers they shared a ride with, with an option to leave a review

*E*<sub>2</sub> Customers input rating and optional review

*S*<sub>3</sub> System stores rating and review in customers' profile

*S*<sub>4</sub> System logs end of carpool

*E*<sub>4.1</sub> Every new 10 000 completed carpools, system sends notification to investors

**BE6.** Carpoolers activate conversation prompts

VP6.1 Customers

*S*<sub>1</sub> Precondition: Carpool has started

*E*<sub>1</sub> Customers indicate that they want to enter prompts mode

*S*<sub>2</sub> Random prompt is generated

*E*<sub>2.1</sub> Customers interact between each other based on prompt

*E*<sub>2.2</sub> Customers select new prompt, branch to *S*<sub>2</sub>

*E*<sub>2.3</sub> Customers exit prompt mode

VP6.2 Taxi company

- N/A
- VP6.3 Taxi drivers
  - $E_1$  Driver begins commute once new customer is seated in taxi
  - $S_1$  System registers that carpool has started
- VP6.4 Investors
  - N/A
- VP6.5 Legal
  - N/A
- Global Scenario for BE6
  - $S_1$  Precondition: Carpool has started
  - $E_1$  Driver begins commute once new customer is seated in taxi
  - $S_2$  System registers that carpool has started
  - $E_2$  Customers indicate that they want to enter prompts mode
  - $S_3$  Random prompt is generated
    - $E_{3.1}$  Customers interact between each other based on prompt
    - $E_{3.2}$  Customers select new prompt, branch to S2
    - $E_{3.3}$  Customers exit prompt mode

## 5 Non-Functional Requirements

### 5.1 Look and Feel Requirements

#### 5.1.1 Appearance Requirements

- LF-A1. The colour scheme of the application must match the branding colours of the company  
**Rationale:** For marketing purposes, the colour scheme within the app need to match the branding colours of the company so the brand becomes well-recognized and there is no confusion among users about the brand.

#### 5.1.2 Style Requirements

- LF-S1. The app must have a UI that is clear (makes important information obvious), consistent, aesthetically pleasing  
**Rationale:** Users are more likely to use and come back to mobile apps which have UIs that are clear, consistent and simple. This helps increase the user base and branding power of the app (Johns, 2022).

### 5.2 Usability and Humanity Requirements

#### 5.2.1 Ease of Use Requirements

- UH-EOU1. The user must not encounter any unclear prompts from which they do not know how to proceed  
**Rationale:** If the app is prompting the user to do something, that prompt must be clear and easy to understand so it is easy for the user to use the app.
- UH-EOU2. The user must not encounter any errors while using the app, and if they do, what to do next must be clear  
**Rationale:** The app must not show the user any errors while they are using it to prevent confusion and to make the app easy to use. On the chance an error does appear to the user, it must be clear on what the user has to do next to remove the error or continue. If the error is generic and confusing to understand, the app is no longer easy to use and leaves the user in confusion.

### 5.2.2 Personalization and Internationalization Requirements

- UH-PI1. The product will operate in both English and French and the user will be able to set their preferred language  
**Rationale:** The app will be available only in Canada, so it needs to contain support for both the national languages of Canada and allow the user to choose their preferred language
- UH-PI2. The product will allow the user to set their preference for dark mode and light mode  
**Rationale:** To make the app more personal, users must be able to customize the appearance to their liking. Its important for users to enjoy the appearance of the app so allowing them to personalize dark mode or light mode will help contribute to that.

### 5.2.3 Learning Requirements

- UH-L1. The product must be able to be used immediately without any training, with no learning curve required to learn how to effectively use the app.  
**Rationale:** Mobile apps designed for general users should have no learning curve. Upon using the app for the first time, the app should be designed in such a way that the user immediately knows how to use it. If users take a while to learn how to effectively use the app, they will not come back to it.

### 5.2.4 Understandability and Politeness Requirements

- UH-UP1. The product must use standard English and French words that can be found in a standard dictionary, with no slang or acronyms.  
**Rationale:** The app is aimed to be used by anyone in Canada, so the symbols and language used must be understandable by the general Canadian population. Using symbols and language or slang that is specific to a certain group or culture would make the app hard to understand for some of the target users.

### 5.2.5 Accessibility Requirements

- UH-A1. The product must provide captions for audio content to allow users with hearing impairments to access that content.  
**Rationale:** A successful mobile app should be accessible to all users, no matter what physical impairments they may face. The app should allow users with hearing impairments to still access any audio content within the app through captions, symbols or some other visual means so they are still able to access the content.

## 5.3 Performance Requirements

### 5.3.1 Speed and Latency Requirements

- PR-SL1. All responses from the product to the user must be within 50 milliseconds  
**Rationale:** The perfect latency is between 20 to 40 milliseconds (Schoenfelder). If all product-to-user responses for our app are 50 milliseconds are better, this is an excellent latency and will bring users back to the app.

### 5.3.2 Safety-Critical Requirements

- PR-SC1. All taxi services, drivers and other carpoolers will go through an external (out of app) safety check before being allowed to offer rides or join carpools on the app.  
**Rationale:** Our app connects users with random members of the public, whether that be taxi drivers or fellow carpoolers. To ensure the safety of all users of the app, all drivers and carpoolers will have to go through a safety verification process to verify their intentions and ensure that the safety of the riders is guaranteed.

### 5.3.3 Precision or Accuracy Requirements

- PR-PA1. All monetary amounts displayed within the application must be accurate to two decimal places  
**Rationale:** Given that this app will involve payments for the taxi rides, all payment amounts will be displayed to two decimal places, which is accurate to the number of dollars and cents for the transaction. For example, “the total cost of the ride is \$3.40”
- PR-PA2. All times must be displayed in the form HH:MM PM/AM where HH is between 01 and 12.  
**Rationale:** The app may show estimated time of arrival, which will be shown using the HH:MM format, showing only the hours and minutes and not anything more. If the app showed seconds or below it would be inaccurate and confusing for users. For example, “the estimated time of arrival is 10:04 PM”
- PR-PA3. All distance measurements must be accurate to one decimal place.  
**Rationale:** The app may show distance to a destination or how far the driver is from picking you up. These should be shown with one decimal place as more decimal places would make it harder to understand and would make it lose meaning. For example, “your driver is 3.2 km away”

### 5.3.4 Reliability and Availability Requirements

- PR-RA1. The product must be available for use at all times when Internet is available, but will be unavailable for 5 hours a month for maintenance. The downtime will be in the middle of the night when the user activity is the lowest.  
**Rationale:** An app that allows users to book carpool rides should be available 24/7 because users should be able to book a carpool at any time of the day on any day of the year. However, it is unrealistic that the app will be available 24/7, so there will be chosen periods of downtime that affect as little of the user base as possible.
- PR-RA2. The product must perform without failure in 98% of use cases  
**Rationale:** Failures are expected to occur in any software application, but if the app performs without failure for the majority of uses then it can be deemed to be a reliable application that only fails on the rare occasion. 98% is chosen because given the expected size of user base the app projects to have, a 98% failure rate would mean very few failures for a large set of users.
- PR-RA3. The mean time to restore the product following a failure must not be greater than 10 minutes unless a malicious attack occurs, in which case the app will stay unavailable until the threat has been completely removed.  
**Rationale:** When the product does fail, it should recover and become usable again within 10 minutes max, otherwise users will begin to question the reliability and usage of the app. If an attack causes the app to go down, 10 minute recovery time is unattainable, so the app will only be restored once the threat has been completely removed.

### 5.3.5 Robustness or Fault-Tolerance Requirements

- PR-RFT1. The product must continue to be usable if external services it relies on go down, barring the Internet/cell tower failures.  
**Rationale:** The app will rely on some external third party services, and it needs to be tolerant to faults in those systems. For example, if the app interfaces with an API from the taxi service and that API goes down, the app needs to continue to be usable. If an AWS region goes down and the app uses AWS cloud services, it must change to another region to ensure it continues to be usable. However if the Internet or cell towers go down, this is out of our control and the app will not be able to be used.

### 5.3.6 Capacity Requirements

PR-C1. The product must be able to handle up to 1000 users simultaneously at any given time

**Rationale:** Although the app will not start with 1000 concurrent users right away, 1000 is a good estimate as to how many concurrent users the app may have within a few years of business. Given that Uber, a similar ride sharing application has millions of concurrent users, 10000 is a realistic capacity limit for this app.

### 5.3.7 Scalability or Extensibility Requirements

Not applicable.

### 5.3.8 Longevity Requirements

Not applicable.

## 5.4 Operational and Environmental Requirements

### 5.4.1 Expected Physical Environment

Not applicable.

### 5.4.2 Requirements for Interfacing with Adjacent Systems

Not applicable.

### 5.4.3 Productization Requirements

OE-P1. The product must be distributed as an Android mobile application

**Rationale:** The product being developed is a mobile app, therefore it should only be distributed as a mobile app and nothing else.

### 5.4.4 Release Requirements

Not applicable.

## 5.5 Maintainability and Support Requirements

### 5.5.1 Maintenance Requirements

MS-M1. The product must be designed in a maintainable way by following the open-closed principle and strong object-oriented techniques. It will be designed in such that future changes are able to be easily added or removed (small changes take less than a week to implement)

**Rationale:** Writing maintainable software allows for quicker implementation and debugging. Iterations of the product will be able to be released faster if maintainable code and design principles are followed.

### 5.5.2 Supportability Requirements

Not applicable.

### 5.5.3 Adaptability Requirements

MS-A1. The product must be able to run on all versions of all Android mobile devices from Version 10.0 and onwards.

**Rationale:** Android makes up 69.74% of the mobile device market (statcounter). The product being built is a mobile application, so in order for the large majority of mobile device users to use the app, it must run on and Android. Furthermore, we are developing in Java which will only run

on Android. Version 10.0 and onwards are the versions still maintained so the app should function on all versions from 10.0 onwards.

## 5.6 Security Requirements

### 5.6.1 Access Requirements

SR-AC1. General users of the application must not have access to identifying and private information of users that would allow them to be identified or hacked.

**Rationale:** The app is designed to be used by any member of the general public, members of the general public should not have access to private or sensitive information about other users or about the taxi services

### 5.6.2 Integrity Requirements

SR-INT1. The system must not be able to be infiltrated by outside threats that look to access/change the data by having security measures such as encryption and proper authentication.

**Rationale:** All software systems come with the threat of external hackers. We must ensure that the system is secure and cannot be hacked, otherwise data can be changed and it will no longer be consistent, violating data integrity

### 5.6.3 Privacy Requirements

SR-P1. The product must make the user aware that they are entering private information whenever they must so the user can make an informed decision on whether they want to include that information.

**Rationale:** The user should be aware whenever they are entering personal or private information so they can make an informed decision on whether they want to enter that information.

SR-P2. The product must guarantee that all private information is kept secure

**Rationale:** Any private information from the user such as their password, address and other sensitive information should be secured through verified security measures, hashing, locking, encryption and authentication.

### 5.6.4 Audit Requirements

SR-AU1. All releases of the product must conform to the guidelines of the Android mobile app store (found here: <https://play.google.com/about/developer-content-policy/>)

**Rationale:** Whenever an app is released to the app store, whether brand new or an updated release, it must go through the app store's review process. Audits must be done within the company before the app is released to ensure the app passes all guidelines.

### 5.6.5 Immunity Requirements

SR-IM1. The product must implement security measures such as encryption and authentication to make it immune to infection from unauthorized malicious attackers

**Rationale:** There are many types of malware that can attack software, SQL injections, trojan horses, viruses. The system must be immune to infection from all of these, to ensure data privacy and security is maintained at all times.

## 5.7 Cultural and Political Requirements

### 5.7.1 Cultural Requirements

CP-C1. All numbering systems, road systems and any other systems must use the Canadian standard

**Rationale:** The app will initially be launched in Canada only, so all systems will use the Canadian standard. Metric system for numbering and measurements, Canadian road systems, everything that requires a national standard will use the Canadian standard.

### 5.7.2 Political Requirements

CP-P1. The product must not use any emojis, language or symbols that could be considered offensive, such as any emojis involving skin colour, as well as common symbols that are globally known to be offensive.

**Rationale:** Given that our app is designed to be used by anyone in the general Canadian population, we must ensure that no controversial language or emojis are used. Many cultural, ethnic and religious groups will exist within the userbase and some words or symbols or emojis might be offensive to a certain group, which must be prevented.

## 5.8 Legal Requirements

### 5.8.1 Compliance Requirements

LR-COMP1. The product must comply with Canadian Privacy Laws, specifically PIPEDA, which states that all identifiable information of a person should be kept private. (Office of the Privacy Commissioner of Canada, 2018)

**Rationale:** This app deals with lots of personal user information, so it must be implemented in a way that complies Canadian Privacy Laws, specifically PIPEDA as this is the act it falls under.

### 5.8.2 Standards Requirements

Not applicable.



## A Division of Labour

GROUP 06

### DELIVERABLE ONE - DIVISION OF LABOUR

01 Introduction + 03 Use case Diagram - Jennifer Ye

02 Overall Description - Nathan Luong

04 Functional Requirements - Jane Klavir

05 Non-Functional Requirements - Areez Visram

Nathan

Jy

J. Klavir

any

## B Creative Features

Below is a list of creative/innovative features that were brainstormed by the team. The features that were ultimately chosen for implementation are bolded.

- Friends (preferred to carpool with friends)
- Pre-book (pre book carpools with others for big events)
- **Prompts + social games in the car (add preferences on what kind of level of socialness you want)**
- **Emergency button**
- In app payments (integrate with PSP's)

## IMPORTANT NOTES

- Be sure to include all sections of the template in your document regardless whether you have something to write for each or not
  - If you do not have anything to write in a section, indicate this by the *N/A*, *void*, *none*, etc.
- Uniquely number each of your requirements for easy identification and cross-referencing
- Highlight terms that are defined in Section 1.3 (**Definitions, Acronyms, and Abbreviations**) with **bold**, *italic* or underline
- For Deliverable 1, please highlight, in some fashion, all (you may have more than one) creative and innovative features. Your creative and innovative features will generally be described in Section 2.2 (**Product Functions**), but it will depend on the type of creative or innovative features you are including.