Deliverable #1 (Revised) : Software Requirement Specification (**SRS**)

SE 3A04: Software Design II – Large System Design

Tutorial Number: T03
Group Number: G8
Group Members:

- Adam Mak
- Eric Chen
- Justin Ho
- Ahmad Hamadi
- Kevin Ishak
- Jonathan Jiang

1 Introduction

1.1 Purpose

The purpose of this **SRS** is to define the functionality and characteristics that the carpool app should have. It should provide a detailed overview of all the processes and systems that needs to be present in the final product. The **SRS** will have a descriptive model, which will be represented visually in the form of a use-case diagram.

- Developers
- UI/UX Designers
- Product Managers
- Engineers
- System Architects
- Quality Assurance Testers

1.2 Scope

The software product is a taxi carpool service. It will provide a communication environment for passengers to carpool via taxis. This service is not meant to be replacement for taxi services (i.e. Uber-like services).

The objective of the software (local taxi system) being implemented is to transport users from one destination to another using a carpool method. This software allows user to enter a destination along with some optional search criteria that they want to arrive at. The software will then match the user with a taxi nearby which may or may not carry other individuals who are travelling in the same general direction. This benefits both the users, taxi driver as well as the environment by as it helps reduce costs, whether it be the fare for the ride or the reduction in environmental pollutants by the means of carpooling.

There are no higher-level specifications that exist, thus, any lower-level specifications must be consistent with this document, as it is the highest-level specification.

1.3 Definitions, Acronyms, and Abbreviations

Abbreviation/Acronym	**Definition**
API	Application Programming Interface
Carpool	To collectively share a ride (either to help reduce the cost of transit or as an act of being more environmentally "green")
DBMS	Database Management System
GoPlay Store	The default Android distribution platform
SMS	Software Requirements Specification
SRS	Simple Message Service
Users/Passengers	A reference to carpoolers and requesters

1.4 References

APA references sorted by occurence of when they appear in this document.

1. Simic, S. (2022, December 13). 7 ways to reduce server response time Improve your server speed. Knowledge Base by phoenixNAP. Retrieved March 5, 2023, from https://phoenixnap.com/kb/reduce-server-response-time#: :text=Google%20recommends%20you%20aim%20for,on%20the%20users'%20geographical%20positions.

2. Earls, A. R. (2019, July 19). Best practices for negotiating a SAAS SLA: TechTarget. Cloud Computing. Retrieved March 5, 2023, from https://www.techtarget.com/searchcloudcomputing/tip/Negotiate-a-SaaS-SLA-for-compliance-uptime-considerations

3. What is an audit? everything about the 3 types of audits. Ageras. (n.d.). Retrieved March 5, 2023, from https://www.ageras.com/blog/what-is-an-audit

4. Office of the Privacy Commissioner of Canada. (2021, December 8). The Personal Information Protection and Electronic Documents Act (PIPEDA). Office of the Privacy Commissioner of Canada. Retrieved March 5, 2023, from https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/

5. Home. TTA. (1970, February 7). Retrieved March 5, 2023, from https://www.thetransportationalliance.org/

6. Business insurance. NFP. (n.d.). Retrieved March 5, 2023, from https://www.nfp.ca/business-insurance/industry-expertise/taxis-limousines-and-busses

7. Taxi fare. Taxi Fare - TLC. (n.d.). Retrieved March 5, 2023, from https://www.nyc.gov/site/tlc/passengers/taxi-fare.page

1.5 Overview

In the order of occurrence, the remainder of the **SRS** contains an overview of the system, which includes product perspective, product functions, user characteristics, constraints, and dependencies. The **SRS** will have a descriptive model, which will be represented visually in the form of a use-case diagram. Furthermore, the document also includes various system-specific requirements which are consist of functional and non-functional requirements. Finally, at the end of the document, there is a record which represents the division of labour of all contributors to the project.

2 Overall Product Description

2.1 Product Perspective

The carpool app product is independent and self-contained, as it does not build upon any existing products/applications. However, the standalone system uses and owns interfaces to and from external systems, such as hardware, users, databases, and other **API**'s, to provide essential functionality to the app. Figure 1 shows how these external systems will interact in a block diagram.

The carpooling app is developed for Android devices, providing users owning these devices with easy access to this service. Also, the app's user interface interacts with an Android device to allow users to see information and interact with the application through a touchscreen.

The app sends and receives data from a database that stores information about users, rides, route information, and other related data. This is managed by a **DBMS**.

The carpooling app uses the Google Map **API** to provide maps and location-related services such as route planning and estimating travel times. The **API** provides information about the locations of users and destinations, as well as real-time traffic data, to help drivers make more informed decisions about their routes.

To protect sensitive information such as passwords, payment information, transmitted messages, and other confidential data, the carpooling app uses a cryptosystem to encrypt this information before it is stored in the **DBMS**. When the data is needed, it is decrypted and used by the app. This ensures that information is protected from unauthorized access.

The carpooling app uses **SMS** functionality to provide communication between users to provide updates, confirm ride details, and more. For instance, when a ride is requested, the app may send a message to the driver to notify them of the request, and then another message to the passenger to confirm the ride details.

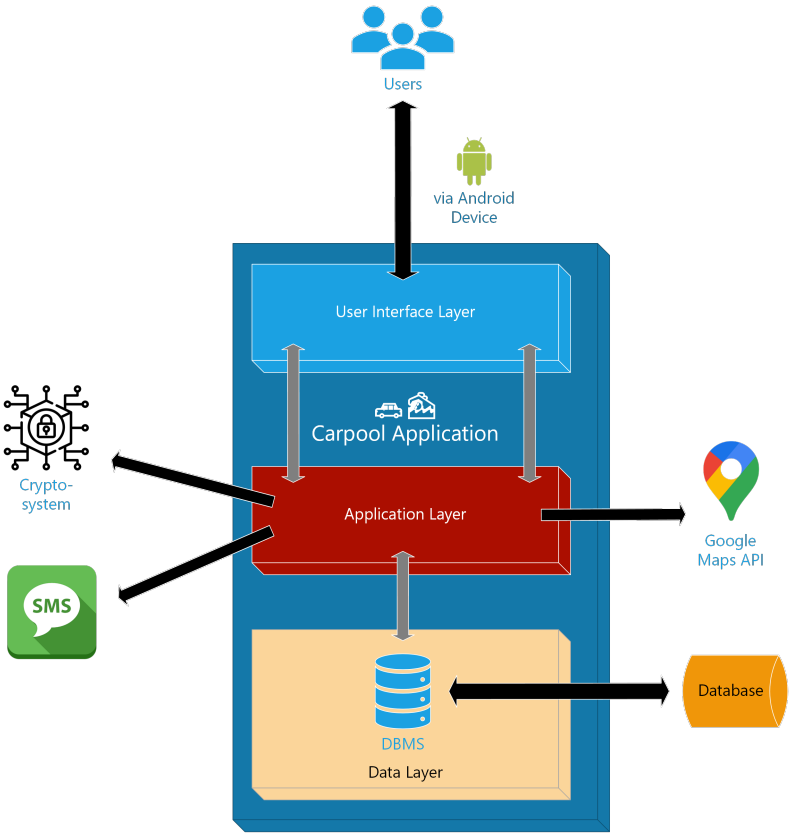

Figure 1: A block diagram showing the major layers of the carpooling application, interconnections, and external interfaces.

2.2 Product Functions

- App should compare carpool price with non carpool price and display the money saved as a point system.
- App should allow carpoolers to setup/join other carpoolers using the app.
- When requesting a carpool, carpoolers should be able to filter through different travel options (i.e. sort by closest cab/highest rated cab driver). The app should return a list of cab options according to the specification.
- Carpoolers looking to set up a carpool should be able to input the taxi ID , destination, maximum number of other carpoolers, as well as potential auxiliary information. They system should post this info and allow other carpoolers to join.
- Users should be able to create and delete accounts, which the system should store.
- The system should check if a user that wants to request/setup a carpool is a registered user. It should reject unregistered users.
- App should have a reward system that allows users to obtain free rides based on their accumulated cost savings.
- App should have a points system that keeps track of costs saved, and reward free rides once a carpooler exceeds a certain number of points.

2.3 User Characteristics

Users include both passengers and drivers, of which education level may vary. The users of the product are expected to have a basic knowledge of technology and are familiar with smartphones, which the app is developed for. Also, users must 16 years of age or older to use the app.

2.4 Constraints

- Legal and regulatory restrictions: The developer needs to ensure that the application complies with the local laws and regulations.
- Technical limitations: The developer may face limitations with regards to the technology used to build the application, such as compatibility with different platforms, scalability, and security.
- Budget and resources: The availability of financial and human resources can impact the scope and timeline of the project.
- Time limits: The project must be completed within the span of 4 months.

2.5 Assumptions and Dependencies

The carpool app will be available for installation/download through the Apple and Google Play store for users to use. It is assumed that Apple/Google Play will streamline our app upon the necessary guidelines being met (I.e., Safety, Privacy, Performance, Business, Design, Legal).

It is assumed that the Google Maps **API** will be available to use with an 100% uptime which our carpool app will utilize to plan its routes and navigate users around cities. If the Google Maps **API**, for some extenuating circumstances, is unable to provide the services for route navigation, then another GIS **API** system may have to be utilized (e.g., OpenLayers, ESRI, etc. . .).

A primary telecommunication company will be managing the **SMS** traffic of the carpool app as the app makes use of **SMS** functionality to inform users about the status of their ride. If for some reason, any of the telecommunication companies listed within this **SRS** document disbands, terminates the contract or breach some sort of confidentiality between both parties, and/or something similar in nature, then the **SRS** might have to change accordingly to fill up the vacant **SMS** management company.

It is assumed that the carpool app will be systematically be calculating the cost of the trip per person on a reduced per mileage basis.

2.6 Apportioning of Requirements

The following requirements are usually delayed until later versions to meet the initial launch deadline and budget, or to focus on delivering the most valuable features to users first.

1. Advanced features: These are features that are not essential to the core functionality of the app, but can be added later as the app evolves.
    - For example, an in-app tipping feature, or integration with a rewards program.
2. Non-critical performance optimizations: Improving the app's performance can be a continuous process, and some optimizations can be deferred until later versions.
    - For example, optimizations that improve the app's speed, reliability or efficiency, but are not essential to its basic functionality.
3. Complex user interfaces: Advanced user interfaces can be time-consuming to develop and may be deferred in favor of a simpler interface in the initial release.
    - For example, a more sophisticated design that includes augmented reality, or a chatbot interface.

4. Integrations with third-party systems: Integrating the app with other systems can be a significant effort, and this work may be deferred until later versions.
    - For example, integration with a payment system, or with a local transit system.
5. Support for older platforms: Supporting older platforms can be time-consuming, and this work may be deferred until later versions or only supported for critical security and bug fixes.
    - For example, supporting older versions of the iOS or Android operating systems.

It is important to prioritize the most critical requirements, such as the core functionality of booking a ride, and ensuring that the app is easy to use. Other requirements can be deferred until future versions to meet the initial launch deadline and budget.

3 Use Case Diagram

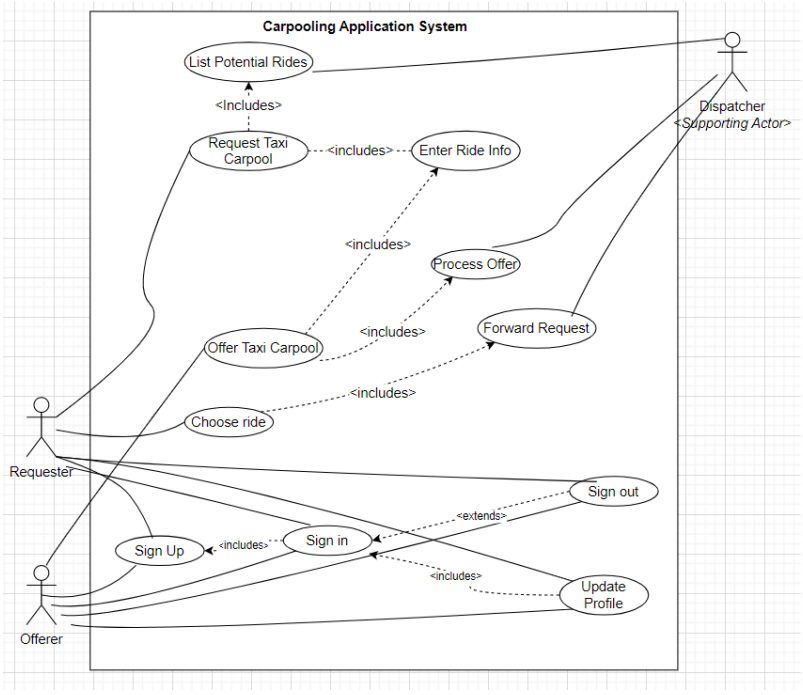

Figure 2: Use case diagram of the carpooling application.

4 Highlights of Functional Requirements

The business events we will present are:

BE1. Request a ride

BE2. Arrive at destination

BE3. Setup/Update user profile/account information

BE4. Offer a ride

BE5. Login/Logout of an account

BE6. Redeem a free ride

BE7. Delete user profile

The viewpoints we will consider are:

VP1. Requester

VP2. Offerer

Some business events only have 1 viewpoint, the passenger. This represents both the requester and offerer as their course of action will not differ from each other.

BE1. Request a ride (Precondition: Passenger must be logged on)

VP1. Requester

- S_1: System asks requester for a pickup location and a destination point and allows them to input a search preference (nearest cab, highest rating first, find specific offeror).
- E_1: Requester inputs a pickup location and a destination point, as well as any optional information.
- S_2: System checks if there are any available carpoolers.
  $S_{2.1}$: There are no available carpool events. The system displays that there are no available carpools ⟨**Branch EXIT**⟩.
  $S_{2.2}$: There are available carpool events. The system lists out the Carpool events that correspond to the users inputted preferences, as well as a choice to reject all carpool events.
- E_2: User decides whether to choose any of the carpool events.
  $E_{2.1}$: Requester rejects all carpool events. ⟨**Branch EXIT**⟩
  $E_{2.2}$: Requester chooses a carpool event.
- S_3: System notifies offerer of the request.

Global Scenario (GS1):

- S_1: System asks requester for a pickup location and a destination point and allows them to input a search preference (nearest cab, highest rating first, find specific offerer).
- E_1: Requester inputs a pickup location and a destination point, as well as any optional information.
- S_2: System checks if there are any available carpoolers.
  $S_{2.1}$: There are no available carpool events. The system displays that there are no available carpools ⟨**Branch EXIT** ⟩.
  $S_{2.2}$: There are available carpool events. The system lists out the Carpool events that correspond to the users inputted preferences, as well as a choice to reject all carpool events.
- E_2: User decides whether to choose any of the carpool events.
  $E_{2.1}$: Requester rejects all carpool events. ⟨**Branch EXIT**⟩
  $E_{2.2}$: Requester chooses a carpool event.
- S_3: System notifies offerer of the request.

BE2. Arrive at Destination (Precondition: Passenger is logged onto their account and are currently in a ride)

VP1. Passenger (Requester and Offerer)

- S_1: System displays the price of the ride for each passenger (price depends on distance travelled). System allocates points to the passenger depending on how much distance they travelled. The system prompts the passenger and asks if the passenger wants to redeem a free ride for a certain amount of their current amount of points.
- E_1: Passenger decides whether to get a free ride.
  $E_{1.1}$: Passenger trades their points for a free ride.
  $E_{1.2}$: Passenger does not trade their points for a free ride. The system prompts the passenger with rating options for the other drivers ⟨**Branch to VP1.E2**⟩
- S_2: System updates the displayed price of the ride to \$0. The system prompts the passenger with rating options for the other drivers.
- E_2: Passenger rates other passengers or chooses to skip.

Global Scenario (GS2):

- S_1: System displays the price of the ride for each passenger (price depends on distance travelled). System allocates points to the passenger depending on how much distance they travelled. The system prompts the passenger and asks if the passenger wants to redeem a free ride for a certain amount of their current amount of points.
- E_1: Passenger decides whether to get a free ride.
  $E_{1.1}$: Passenger trades their points for a free ride.
  $E_{1.2}$: Passenger does not trade their points for a free ride. The system prompts the passenger with rating options for the other drivers ⟨**Branch to GS2.E2**⟩
- S_2: System updates the displayed price of the ride to \$0. The system prompts the passenger with rating options for the other drivers.
- E_2: Passenger rates other passengers or chooses to skip.

BE3. Create account (Precondition: User is not logged into an account)

VP1. Passenger

- S_1: System prompts user for personal information (Full name, phone number, email).
- E_1: User responds to the system.
- S_2: System checks if the submitted username matches any pre-existing account usernames.
  $S_{2.1}$: Submitted username matches a pre-existing account username. ⟨**Branch to VP1.S1**⟩
  $S_{2.2}$: Submitted username does not match a pre-existing account username. User's personal information is added to the database.

Global Scenario (GS3):

- S_1: System prompts user for personal information (Full name, phone number, email).
- E_1: User responds to the system.
- S_2: System checks if the submitted username matches any pre-existing account usernames.
  $S_{2.1}$: Submitted username matches a pre-existing account username. ⟨**Branch to GS3.S1**⟩
  $S_{2.2}$: Submitted username does not match a pre-existing account username. User's personal information is added to the database.

BE4. Remove account (Precondition: Passenger is logged into their account)

VP1. Passenger

- S_1: System prompts passenger to confirm they want to remove their account.
- E_1: User decides whether they want to remove their account.
  $E_{2.1}$: User confirms they want to remove their account.
  $E_{2.2}$: User decides not to remove their account. ⟨**Branch EXIT**⟩
- S_2: System removes the passengers account information from the database.

Global Scenario (GS4):

- S_1: System prompts passenger to confirm they want to remove their account.
- E_1: User decides whether they want to remove their account.
  $E_{2.1}$: User confirms they want to remove their account.
  $E_{2.2}$: User decides not to remove their account. ⟨**Branch EXIT**⟩
- S_2: System removes the passengers account information from the database.

BE5. Offer a ride (Pre-condition: Passenger is logged into their account)

VP1. Offeror

- S_1: System asks offerors for information regarding the offer (destination, taxi ID, max number of customers to share with).
- E_1: Offeror inputs the carpool information.
- S_2 : System waits for a carpool request from a requester. The system provides user with the options to cancel the carpool offer (stop the carpool event entirely) or to begin the ride (begin the ride with the current accepted requesters).
  $S_{2.1}$: System receives request from the requester. System forwards the request to the offeror of the carpool event. In addition, they are presented with changes to their trip fare, distance, total estimated travel time, a distance – trip fare ratio, and the updated route that they will be taking
  $S_{2.2}$: System receives a "cancel the carpool" response from the user. ⟨**Branch EXIT**⟩
  $S_{2.3}$: System receives a "begin the ride" response from user. ⟨**Branch EXIT**⟩
- E_2: Offeror chooses whether or not they want to accept the carpool request.
- S_3:
  System checks for offerors response.
  $S_{3.1}$: System adds the requester to the carpool event if the offeror accepts the request. The requester is notified that their request has been accepted. System asks offeror if they would like to close the carpool event/ begin the ride.
  $S_{3.2}$: The requester is notified that their request has been denied. System asks offeror if they would like to close the carpool event/ begin the ride
- E_3: Offeror chooses whether or not to end the carpool event, begin the ride, or keep looking for more requests.
  $E_{3.1}$: Offeror chooses begin the ride. ⟨**Branch EXIT**⟩
  $S_{3.2}$: Offeror chooses cancel carpool event. ⟨**Branch EXIT**⟩
  $S_{3.3}$: Offeror chooses to look for more requests. ⟨**Branch VP1.S2**⟩

Global Scenario (GS5):

- S_1: System asks offerors for information regarding the offer (destination, taxi ID, max number of customers to share with).
- E_1: Offeror inputs the carpool information.
- S_2 : System waits for a carpool request from a requester. The system provides user with the options to cancel the carpool offer (stop the carpool event entirely) or to begin the ride (begin the ride with the current accepted requesters).
  $S_{2.1}$: System receives request from the requester. System forwards the request to the offeror of the carpool event. In addition, they are presented with changes to their trip fare, distance, total estimated travel time, a distance – trip fare ratio, and the updated route that they will be taking
  $S_{2.2}$: System receives a "cancel the carpool" response from the user. ⟨**Branch EXIT**⟩
  $S_{2.3}$: System receives a "begin the ride" response from user. ⟨**Branch EXIT**⟩
- E_2: Offeror chooses whether or not they want to accept the carpool request.
- S_3:
  System checks for offerors response.
  $S_{3.1}$: System adds the requester to the carpool event if the offeror accepts the request. The requester is notified that their request has been accepted. System asks offeror if they would like to close the carpool event/ begin the ride.
  $S_{3.2}$: The requester is notified that their request has been denied. System asks offeror if they would like to close the carpool event/ begin the ride
- E_3: Offeror chooses whether or not to end the carpool event, begin the ride, or keep looking for more requests.
  $E_{3.1}$: Offeror chooses begin the ride. ⟨**Branch EXIT**⟩
  $S_{3.2}$: Offeror chooses cancel carpool event. ⟨**Branch EXIT**⟩
  $S_{3.3}$: Offeror chooses to look for more requests. ⟨**Branch GS5.S2**⟩

BE6. Login to an account (Precondition: Passenger is currently not logged into an account)

VP1. Passenger

- S_1: System prompts passenger for login information (username and password).
- E_1: Passenger responds to the system.
- S_2: System authenticates login information.
  $S_{2.1}$: Customers login info matches an account and they are granted access.
  $E_{1.2}$: Customers login info matches an account and they are granted access ⟨**Branch VP1.S1**⟩

Global Scenario (GS6):

- S_1: System prompts passenger for login information (username and password).
- E_1: Passenger responds to the system.
- S_2: System authenticates login information.
  $S_{2.1}$: Customers login info matches an account and they are granted access.
  $E_{1.2}$: Customers login info matches an account and they are granted access ⟨**Branch S1**⟩

BE7. Update account information (Precondition: User is logged onto their account)

VP1. Passenger

- S_1: System displays their previous personal information form and prompts the passenger to make changes(Full name, phone number, email).
- E_1: Passenger fills in the information.
- S_2: System updates the account information in the database.

Global Scenario (GS7):

- S_1: System displays their previous personal information form and prompts the passenger to make changes(Full name, phone number, email).
- E_1: Passenger fills in the information.
- S_2: System updates the account information in the database.

5 Non-Functional Requirements

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF-A1. The app should conform to a standardized colour scheme which is to be used throughout the app. The choice of colour should be consistent all throughout the app and the must provide enough vibrance amongst each other so that in no shape of form gives off a "dull" image.
Rationale: Having a modern aesthetic and consistent color scheme help create a more engaging and enjoyable user experience while also reinforcing the brand and establishing a sense of professionalism. The vibrance also makes the app more accessible for those with visual impairments.

LF-A2. The app should be organized by related content with little to no cluttering of text/images to provide for visually appealing and easy to read navigation. The cluttering of text/images should follow a basis of having at minimum 5 vw/vh between unrelated images/text bodies and 2.5 vw/vh for related ones.
Rationale: Organizing content and following a reasonable minimum spacing guideline help create a visually appealing and user-friendly experience that is easy to navigate and accessible to all users.

LF-A3. The font family will remain consistent throughout the interface of the app. The size of the font will be increased amongst reasonable intervals and there shall be no difference in text size which varies ever so slightly. There shall not be font size of something of the sort: 13.5, 13.7 and 13.9. Instead, a font size with more appropriate intervals will be an interval of 10, 12, 16 as it provides an easy way to distinguish the size of the text and remain consistent all throughout the app.

Rationale: Consistent font sizes make it easier for users to read and understand the text. If there are variations in font size, it can cause confusion and make it difficult for users to read and comprehend the content.

5.1.2 Style Requirements

LF-S1. The app shall have a monochromatic appearance, with appropriate font size and font type.
Rationale: See rationale for LF-A1.

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH-EOU1. The app should be relatively easy to learn for beginners and should require no more than 3 minutes of tutorial time to learn. A tutorial will be provided upon creating of a new account that helps navigate the user through the app if they so choose to.

UH-EOU2. The app should be accessible, and convenient for the general public to use. The app shall allow users to create an account through various forms of media (e.g. Gmail OAuth, **SMS** Login, Apple Login, etc...) so that users with only one source of media are not limited.
Rationale: The requirement for all accessibility and navigation buttons to be intuitive and easy to find ensures that users can easily navigate the app's interface and access the content they need. Organizing related content in an easy and intuitive way improves user experience and minimizes frustration when searching for information.

5.2.2 Personalization and Internationalization Requirements

UH-PI1. The app shall have the basic US English language implemented in its initial run and can take on various languages afterwards.
Rationale: Initial target audience is based in US and Canada, whose primary languages are both English. Internationalization will be considered in later iterations of the application, in which other languages mu

5.2.3 Learning Requirements

UH-L1. The app should not require users to have any pre-existing knowledge beside knowing where the user wants to get to in order to use the apps features.
Rationale: An application that does not require pre-existing knowledge is more accessible to users with varying levels of technical expertise, which leads to greater accessibility and market audiences.

5.2.4 Understandability and Politeness Requirements

UH-UP1. The carpool app should not make use of any vulgar speech in any dialogue/informational sections throughout the app.
Rationale: The carpool app should be appropriate to use for all ages and should maintain a degree of professionalism. Vulgar speech can limit the target audience and impair reputation.

UH-UP2. Has buttons labelled with different actions the user can take.
Rationale: It should be clear to users what they are doing when interacting with the app.

5.2.5 Accessibility Requirements

UH-A1. App must have a text-to-speech functionality for visually impaired users.
Rationale: Visually impaired users are unable to see the user interface, and thus they are not able to use the app's functionality even though they are just as capable of using the carpool's services. Text-to-speech increases accessibility for those that are visually impaired.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR-SL1. The app should run reliably fast with a responsive user interface with minimal lag. The latency shall not exceed 200ms when transitioning between different pages on the app and that new content should be displayed within the 200ms.
Rationale: A slow user interface risks users losing patience, becoming frustrated, and tending towards competitors. This is vice-versa with consistently high-responsive user interface. Google recommends aiming for a latency of no more than 200ms, and that it is consistent for all users.[1]

PR-SL2. The app must also calculate a suitable travel route for carpool users within 5 seconds of matching.
Rationale: Similar rationale to PR-SL1. However, since calculating routes take up much more computational power than displaying context on a screen, a time of 5 seconds accounts for that while maintaining a degree of efficiency.

5.3.2 Safety-Critical Requirements

PR-SC1. The system should provide information about the correct vehicle for their trip. This includes the make/model of the car that the driver is driving, the license plate attributed with the car, and the location of the car/driver when they are ready to receive passengers.
Rationale: Ensuring correct information helps to ensure that passengers are getting into the correct vehicle with a verified driver, and not unauthorized or unsafe vehicles/drivers. This prevents instances of fraud, theft, or other safety concerns.

PR-SC2. Route should not be recalculated when driver is in motion. Travel route should only be able to be changed/updated accordingly if there's a faster route (e.g., due to traffic or something) and that the car has been stationary for at least 15 seconds.
Rationale: This helps to ensure that the driver remains focused on the road and does not become distracted by changes to the travel route.

PR-SC3. System should not disclose sensitive personal information of any user to other user. This includes the profile of the user along with any information that may identify the passengers who are carpooling in.
Rationale: All information pertaining to users of the app besides their name and "rating" can be considered private information and will not be shared with any other passengers and carpoolers.

5.3.3 Precision or Accuracy Requirements

PR-PA1. The locations of pick-up/destinations should be accurate within 50m.
Rationale: Inaccurate destinations cause inconveniences for all parties participating in the carpool, such as having to walk further than expected or missing the carpool altogether because the taxi's location is inaccessible from the user's location.

PR-PA2. Estimated times of pickup/arrival should be accurate within 5 minutes.
Rationale: Inaccurate pickup/arrival times cause unnecessary delays (early arrival delays the taxi and late arrival delays the user). This may turn people away from using the app as this is an inconvenience to all parties participating in the carpool.

5.3.4 Reliability and Availability Requirements

PR-RA1. The app should functionally run with an uptime of 99.5%.
Rationale: While it is not safety-critical to have the app crash or go on maintenance, any downtime directly affects any users that are looking for carpools, in a carpool. This can also cause confusion and complications between users and the taxi company in terms of payment. Furthermore, it may turn customers away from using the app is it is not reliable. Many of today's leading SaaS vendors offer at least 99.5% availability.[2]

5.3.5 Robustness or Fault-Tolerance Requirements

PR-RFT1. In the event of an internet outage, the system should prevent users from using the app, and should Continue displaying any committed trips.
Rationale: Upon any sort of outage, the calculated route should still be displayed, ensuring partial functionality of the app.

PR-RFT2. In the event the system goes down, any ongoing trips should continue.
Rationale: See rationale for PR-RFT1.

5.3.6 Capacity Requirements

PR-C1. The functional capacity of the app should scale with the number of users. No user should be restricted due to lack of functional capacity
Rationale: To avoid any downtime which might turn away users, the app should scale with the number of users to maintain constant uptime.

PR-C2. The application installation size shall not exceed 40 MB.
Rationale: Similar rationale to PR-C1. The size of the app should also reflect efficiency and be boilerplate free, suggested by 40 MB.

5.3.7 Scalability or Extensibility Requirements

PR-SE1. The system shall adapt to any increase in user influx while maintaining performance.
Rationale: See rationale for PR-C2.

5.3.8 Longevity Requirements

PR-L1. N/A

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE-EPE1. Expected physical environments include urbanized areas which have paved roads and cellular/wireless connectivity.
Rationale: The standards set for this app assume usage in these conditions, as the Google Maps API and route calculation require known roads, which are mostly paved, and constant connectivity to wireless/cellular networks is also assumed.

5.4.2 Requirements for Interfacing with Adjacent Systems

OE-IA1. All the tech stack used in the development of the carpool app is expected to interface correctly with all other components of our app. In extreme cases where any of the interfaces (e.g. Google Maps **API**/MongoDB) exhibits severely poor performance, it may recommended to swap out the interface with another one similar (e.g. OpenLayers/MySQL).
Rationale: The different technologies used in the development of the app and its services should be functional. Backup and replacement technologies are important in case of outages.

5.4.3 Productization Requirements

OE-P1. System shall must be tested on multiple devices before deployment.
Rationale: We want to make sure the app works for all users with a wide variety of Android devices.

OE-P2. System shall be accessible to customers in the **GoPlay** store.
Rationale: The current version of the app is designed for Android platforms, as such will be

delivered through the GoPlay store or APKs. To deploy apps on iOS requires a lengthier process and may be considered in the future.

5.4.4 Release Requirements

OE-R1. The application shall be released to the **GoPlay** store.
Rationale: See rationale for OE-P2.

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS-M1. In the event that maintenance/shutdown is required, users must be informed via the app's homepage a week in advance.
Rationale: This ensures users are not kept in the dark and will expect an outage, lowering the chance of situations which may leave users frustrated.

MS-M2. New feature updates should not conflict with previous versions of the app and can be pulled by updating through the App/Google Play store when users decide to update the app - This can also be set to automatic for automatic updates.
Rationale: Updates to the app should improve its performance and functionality, users should have agency to update or not, however in the case of mandatory updates where cross-version functionality is affected, the user will be informed updates have occurred and prevent operation of the app until version is up to date.

5.5.2 Supportability Requirements

MS-S1. Forward Compatibility with Android devices in newer versions.
Rationale: Naturally, the general population will update their devices when newer versions of Android releases. This should not break or crash the application, as people seldomly rollback versions just to use the app.

MS-S2. The repo to the project's main repository shall be made public to allow for improvements and suggestions from other developers.
Rationale: Along with users, other developers can provide valuable feedback, particularly things that are too technical for a user to notice. Having more feedback and improvement ideas help support the app with bug fixes, missing features, and so forth.

MS-S3. The source code shall be fully documented through Javadoc comments and shall adhere to correct coding styles.
Rationale: Having coding standards make it easier for the project to be maintained and extended. Consist coding styles and thorough comments allow anyone involved in the project to look at any part of the code, comprehend it, and change it regardless of when it was written during the project. It will also reduce code complexity, which in turn is more likely be better safeguarded against bugs.

5.5.3 Adaptability Requirements

MS-A1. N/A

5.6 Security Requirements

5.6.1 Access Requirements

SR-AC1. Users can only access their accounts after entering their own set password and email.
Rationale: Sensitive information are stored in accounts and should remain confidential to the user. Unwanted access can lead to a violation of LR1. and unwanted use of the app.

5.6.2 Integrity Requirements

SR-INT1. N/A

5.6.3 Privacy Requirements

SR-P1. User's private information will always be kept private and only accessible by the user and certain components of the application.
Rationale: By assuring users that their personal information will be kept confidential and secure, it builds trust between the users and the software application, leading to increased user adoption and retention.

SR-P2. All personal data stored in the database shall be encrypted, and data shall only be decrypted once the user has authorized access to view or modify the information.
Rationale: This adds another layer of security to ensure any unauthorised access of the database will not be able to retrieve any sensitive information about users.

SR-P3. All data transmitted between users shall be encrypted from when the sender sends the message to when the system confirms that the receiver is authorized to view the message. **Rationale:** This adds another layer of security to prevent cybercriminals from intercepting and misusing these messages which may be sensitive and go against PIPEDA regulations[4].

5.6.4 Audit Requirements

SR-AU1. Internal audits will be conducted once every month per fiscal period and external audits will be conducted at random intervals throughout the fiscal year to check for discrepancies.
Rationale: By conducting internal and external audits, the software application can demonstrate its commitment to compliance with regulatory requirements and best practices, build trust with users and stakeholders, and reduce the risk of financial and reputational losses by detecting and acting on discrepancies.[3]

5.6.5 Immunity Requirements

SR-IM1. N/A

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

CP-C1. Documentation shall include an inclusion and diversity statement to ensure all users have the right to access and utilize the website.
Rationale: As the target audience is made up of a range of different social and ethnic backgrounds, genders, ages, etc., the app should demonstrate a commitment to building an inclusive and welcoming product for all people.

5.7.2 Political Requirements

CP-P1. N/A

5.8 Legal Requirements

5.8.1 Compliance Requirements

LR-COMP1. The application shall not violate any Canadian or American laws. For instance, all data will be encrypted so that personal information privacy is preserved.
Rationale: By complying with laws such as the Personal Information Protection and Electronic Documents Act (PIPEDA) in Canada, the application can avoid any legal liabilities and potential lawsuits.[4]

LR-COMP2. Inform users about any personal information collected by the application.
Rationale: This will enable them to make informed decisions about its usage and ensure compliance with PIPEDA regulations.[4]

LR-COMP3. Prior to obtaining user consent to collect personal information, provide a clear explanation of the nature, purposes, and consequences of data collection by the application.
Rationale: This will allow users to make diligent and informed decisions about how they use the application whilst complying to PIPEDA regulations.[4]

LR-COMP4. Allow users to withdraw their consent to have their personal information collected at any time.
Rationale: Preventing further data collection by the application is the user's choice according to PIPEDA regulations.[4]

5.8.2 Standards Requirements

LR-STD1. Taxi companies partnered with the application must ensure licensing requirements for their taxi drivers.
Rationale: Taxi drivers must hold a valid driver's license, taxi operator's license, and background check. This is to ensure that taxi drivers are qualified, trustworthy, and safe to transport passengers.[5]

LR-STD2. Taxi companies partnered with the application must have adequate insurance coverage to protect all passengers in any taxi.
Rationale: This is to ensure that passengers are protected in case of accidents, and taxi companies are held accountable for any damages.[6]

LR-STD3. Taxi companies partnered with the application must have their taxi fares exceed the city or province's taxi fare regulations.
Rationale: This is to ensure that passengers are charged reasonable and fair prices for taxi services.[7]

A Division of Labour

This sheet must be signed by all team members.

Team Member	Contribution
Adam Mak	Introduction, product perspective (2.1), use case diagram (changed after feedback), NFR's + rationales, references, convert to LaTeX. Signed by: Adam Mak
Eric Chen	Assumptions and dependencies, BE1, BE2 (Has been changed after receiving feedback), BE3, BE6, Reviewing/editing all BE's, BE7 Global Scenario, NFRs 5.1, 5.2, 5.3 (first half) and collectively some of the other NFRs with the team. Signed by: Eric Chen
Justin Ho	Introduction, product functions (2.2), BE1, BE2, all global scenarios except for BE7, part of 5.3.5 and 5.3.6. Signed by: Justin Ho
Ahmad Hamadi	Use case diagram (changed after feedback), overall description 2.4 (constraints), NFR's 5.6.3, 5.6.4 and collaborated with other NFRs, scope. Signed by: Ahmad Hamadi
Kevin Ishak	Apportioning of Requirements, Legal Requirements and citations. Signed by: Kevin Ishak
Jonathan Jiang	Overview, user characteristics, contributions to BEs (most revised), contributions to NFRs, NFR rationales. Signed by: Jonathan Jiang