Deliverable #1 Template: Software Requirement Specification (SRS)

SE 3A04: Software Design II – Large System Design

Tutorial Number: T02

Group Number: G3

Group Members:

- Mya Hussain
- Moamen Ahmed
- Jinal Kasturiarachchi
- Nivetha Kuruparan
- Aadil Rehan

IMPORTANT NOTES

- Be sure to include all sections of the template in your document regardless of 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.

1 Introduction

The SRS for a mobile taxi-sharing application aims to provide a convenient way for customers to arrange taxi carpools in order to minimize the cost of a trip. The application is being developed for the local taxi company, CarpoolClan, to attract more customers and increase revenue in the long term.

1.1 Purpose

The purpose of the SRS is to serve as a basis for communication between the developers and stakeholders to ensure that all parties have a clear understanding of the application's functionality, features, and technical requirements. The SRS will serve as a blueprint for the developers and acts as a reference throughout the process and testing phases, making sure that the final product meets the customer's needs and expectations.

The intended audience for SRS is the stakeholders involved in the development and implementation of the mobile taxi-sharing application, including:

1. Developers: The developers, software architects, and engineers who will be responsible for designing, building, and testing the app.
2. Customers: The customers who will be using the application to arrange taxi carpools, including information about destinations, taxi IDs, and estimated fares.
3. Project Manager: The person responsible for overseeing the process, making sure that the project stays on schedule.
4. Taxi Company: The local taxi company that has commissioned the development of the application and will be using it to attract more customers and increase revenue.
5. Investors: Anyone who has provided financial support for the project and wants to understand the requirements for the application.

1.2 Scope

SP1. DBMS

- DBMS are data storage, access, and query software systems. A database management system acts as a bridge between an end-user and a database, allowing users to create, read, update, and delete data in the database.

SP2. Carpool Service

- The carpool service facilitates communication between drivers and passengers using the service. This should enable users to create or join a carpool event. This service is not intended to be a replacement for taxi services.

SP3. Encryption Service

- The encryption service is a software component that secures sensitive information sent and stored by the mobile taxi-sharing application, such as customer account information. It encrypts and decrypts data using cryptographic algorithms, securing the information's confidentiality and integrity.

SP4. Creative feature

- Introducing our newest carpooling option, which will improve your riding experience! Users may now change the music and modify the temperature throughout the trip using MixMaster and WeatherWizard features, making their experience more customized and pleasant. Now you can travel in style, with your preferred music and temperature. Come along with us as we make carpooling more convenient and pleasant.

A mobile taxi carpool application for the local taxi firm, CarpoolClan, is being developed. The objective of the application is to provide customers with a simple and cost-effective way to arrange taxi carpools. To protect the security of private information, the application will contain a variety of software products such as an online mapping service, a carpool service, and encryption service. Some benefits include higher productivity, ease of use, reduce in cost, reduction in environmental pollutants and an improved customer experience. The application's main goals are to match the needs and expectations of the customer while also providing a safe platform for the communication and storage of private information.

There are no higher level specifications that exist, therefore, any lower level specifications must be consistent with this SRS, as it is the highest level specification

1.3 Definitions, Acronyms, and Abbreviations

Term	**Definition**
AC	Air Conditioning
AODA	Accessibility for Ontarians with Disabilities Act
API	Application Programming Interface
Carpool	The shared use of a car, often by people driving to the same location, to reduce the number of vehicles on the road and save money on expenses such as fuel and parking fees.
Customer	The individual who is requesting a carpool from the local taxi company
DBMS	Database Management System
Encryption	The process of transforming sensitive information into a secure, encoded format to protect its confidentiality and integrity, using cryptographic algorithms.
GPS	Global Positioning System
GUI	Graphical User Interface
PIPEDA	Personal Information Protection and Electronic Documents Act
QA	Quality Insurance
QR	Quick Response
SDLC	Software Development Life Cycle
SRS	Software Requirements Specification
UI	User Interface
UML	Unified Modelling Language
UX	User Experience

1.4 References

1. J. Baur, "Dynamic Route Optimization: What It Is and How It Can Benefit Your Business," DispatchTrack Blog, 10-Oct-2019. [Online]. Available: https://www.dispatchtrack.com/blog/dynamic-route-optimization. [Accessed: 06-Mar-2023].

2. Uber Technologies, Inc., "UberPOOL - Affordable Ridesharing," Uber, [Online]. Available: https://www.uber.com/au/en/ride/uberpool/. [Accessed: 06-Mar-2023].

3. Lyft, Inc., "Lyft Shared Rides for Riders," Lyft Help Center, [Online]. Available: https://help.lyft.com/hc/e/all/articles/115013078848-Lyft-Shared-rides-for-riders. [Accessed: 06-Mar-2023].

4. Provincial Heating. (2021, March 15). What is the Healthiest Room Temperature? [Blog post]. Retrieved from https://www.provincialheating.ca/blog/what-is-the-healthiest-room-temperature/

5. Office of the Privacy Commissioner of Canada, "The Personal Information Protection and Electronic Documents Act (PIPEDA)," Privacy Laws in Canada, [Online]. Available: https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/. [Accessed: 06-Mar-2023].

6. Accessibility for Ontarians with Disabilities Act Alliance, "The Act," Accessibility for Ontarians with Disabilities Act Alliance, [Online]. Available: https://aoda.ca/the-act/. [Accessed: 06-Mar-2023].

1.5 Overview

This document serves as the foundation for our project's specifications; the document's specifications will be detailed in subsequent parts.

The following section of the SRS will list the product perspective, product functions, user characteristics, constraints, assumptions, and requirement apportionment. The goal is to define the broad aspects that will affect the project as well as its needs. Following that, the document will include a use case diagram that describes how stakeholders will engage with each other through the various use cases.

In the functional requirement section, we will identify criteria that can be met within a system and properly tested. The non-functional requirement section will comprise needs such as look and feel, usability and humanity, performance, operational and environmental requirements, maintainability and support, cultural and political requirements, legal and security requirements.

To ensure that we cover the whole scope of the project, we must brainstorm and construct as many functional and non-functional requirements as possible before we finalize this use case diagram.

2 Overall Description

CarpoolClan is a taxi carpool mobile application, that allows customers to go on trips with other people heading to the same destination. This application allows customers to experience dynamic route optimization facilities[1] as well as have the ability to control the atmosphere of the trip (having full control over the AC/heater and music) to ensure an optimal and enjoyable ride for everyone.

2.1 Product Perspective

CarpoolClan is an entirely independent and self-contained system that uses several interfaces/tools to function. This application can be seen as similar to products such as UberPool[2] and Lyft Shared Ride[3], however, the main difference is that rather than connecting drivers to riders, CarpoolClan connects riders to other riders. The key aspect of this application is allowing customers who have a taxi to find other customers that also are heading to the same destination. The application will also determine whether it is optimal (cost and time efficiency) to pick up a certain requester, allowing the offerer to have full control of the trip.

Several interfaces and tools that will be used in the mobile taxi carpool application are the Google Maps integration to display potential routes and locations, a DBMS to store all user information, the Spotify integration to allow customers to create a playlist, and an encryption system to encrypt all messages.

2.2 Product Functions

- Register Profile: Allows users to create an account with CarpoolClan. Users would need to fill in their basic information before the account can be active.
- Sign into Profile: Allows users to sign into their profile to view their account information, request/offer a ride, and have access to any other features the application offers.
- Sign out of Profile: Allows users to sign out of their account.
- Edit Profile: Allows users to modify their account profile, such as revising their basic information.
- Delete Profile: Allows users the option to permanently delete their accounts.
- Centralized Dispatcher: Stores all relevant information about each taxi in the fleet. It also decided how to handle offers and requests for taxi carpools.
- Encryption System: All transmitted messages between customers and the Dispatcher are encrypted using a cryptosystem for confidentiality and integrity purposes.
- Scan Unique QR Code: Allows customers (offerers) to scan a QR code that represents the taxi ID. The QR code can be found in the taxi.
- Offer Taxi Carpool Service: Allows customers (offerers) to offer rides to other customers by inputting taxi details such as date, time, taxi ID, the maximum number of customers to share with, if MixMaster was selected, if WeatherWizard was selected, and destination. The offer will then be processed with the Dispatcher and displayed in any related searches.
- Request Taxi Carpool: Allows customers (requesters) to request a ride to the Dispatcher by inputting pick-up location, drop-off location and any additional preferences such as date, time, nearest cab, highest rating first, finding specific offerer, MixMaster, WeatherWizard, and female-only carpool. The Dispatcher will then return a list of all potential offers where the customer can then make a selection and allow the request to be processed.
- View New Request: The Dispatcher allows offerers to view a new request and presents the optimal option on whether to accept/reject the request. It also provides potential changes to their current trip, such as estimated fare, distance, time, etc.

- Accept Request: Allows offerers to accept the new request. The Dispatcher will modify the previous route the offerer was on and apply the necessary changes such as a new fare price. The requester will also receive a confirmation message with the trip details.
- Reject Request: Allows offerers to reject the new request. The Dispatcher will then send a notice to the requester and allow them to make a new selection.
- Display Fare: Displays the fare owed by each customer. Any changes made to the trip will be reflected in the fare.
- Ratings/Reviews upon Arrival: Allows customers an option to rate each other and provide reviews on their trip experience. All ratings will be anonymous, and the average score of all reviews will be displayed on the corresponding user profile.
- MixMaster: Allows customers the option to create a playlist of songs that the taxi can play during the trip. The offerer has the option to choose if they want to listen to songs on the trip or not. If the option is selected, then both the offerer and the requester can add and remove songs from the playlist. CarpoolClan will automatically play the first song on the playlist. Additionally, requesters can filter trips with or without the MixMaster feature.
- WeatherWizard: Allows the customer the option to have control of the AC and heater throughout the trip. The offerer has the option to choose if they want to listen to songs on the trip or not. If the option is not selected, then the default temperature is set. Additionally, requesters can filter trips with or without the WeatherWizard feature.

2.3 User Characteristics

- All Users: All users must be able to have constant access to the device as well as have general knowledge of how the app works as well as have valid identification. They must have good communication skills as well, in order to converse with other customers and taxi drivers.
- Customers:
  - Below 13: Unfortunately, these users would be unable to request and offer rides using CarpoolClan as they would need proper supervision. While making an account, the application will ask users to verify their age in order to make sure no customer is under the age limit.
  - 18 and above: These users are typical types of people using the carpool system such as university students or office workers. They are independent and do not need to rely on anyone when requesting and offering a ride. It is assumed that these people know how to navigate through a mobile app, as well as knowing the locations they want to be picked up/dropped off from.
  - Disabilities: These users are people with disabilities that need additional services.

2.4 Constraints

- The app's budget is 0 dollars
- The app must be completed the week of April 10, 2023
- The app must be built using the Android Software Development Toolkit (SDK)
- The app must be able to adapt to different mobile screen sizes and orientations
- The processing power of the device must be able to withstand the requirements of the app.

2.5 Assumptions and Dependencies

- The user's device must have GPS capabilities
- The user's device must have an active internet connection
- The user's device must be able to run Android version 5.0 or later
- The system is a remote server
- All taxis and taxi owners are licensed
- All taxis are able to drive to all pick-off locations and destinations
- All taxi owners are able to drive to all of the pickup locations and destinations without having to stop for gas or for additional services

2.6 Apportioning of Requirements

Some requirements are frequently postponed until later versions of the project in order to focus on providing the most useful features to users first. These delays usually help the project meet its deadlines.

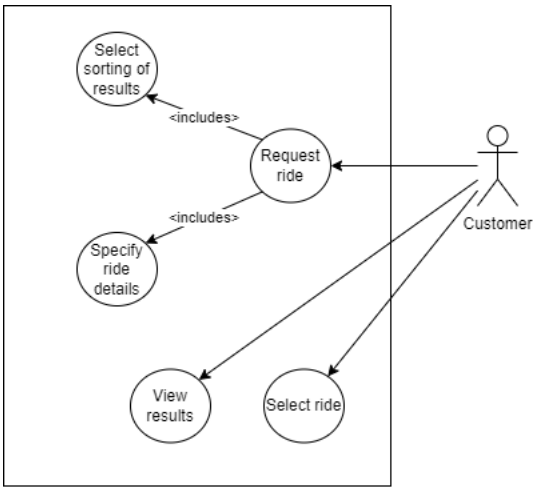
1) Complex UI: Advanced UI can be time-consuming to develop and may be delayed in favour of a simpler interface in the initial release

    1.1) Real-time Collaboration: Features such as in-app messaging and calling are not implemented in the first launch

2) Multi-Language Support: Support for multi-language is time-consuming and unnecessary for the app's first launch

3 Use Case Diagram

The following Use Case Diagram is for BE6 (Request Taxi Carpool):



The use case is triggered by a customer using the application indicating that they would like to request a ride, which prompts them to fill in details regarding the request and then provide them with a list of available rides. They are then able to select a ride from those options.

4 Functional Requirements

This S.R.S. document contains the following business events:

- $BE1$**:** Create user profile
- $BE2$**:** Reset password
- $BE3$**:** Login
- $BE4$**:** Delete profile
- $BE5$**:** Edit profile
- $BE6$**:** Request Taxi Carpool
- $BE7$**:** Communicate Offer To Offering Customer
- $BE8$**:** Ride Pickup
- $BE9$**:** Ride Drop-Off
- $BE10$**:** Make A Payment
- $BE11$**:** Offer Taxi Carpool
- $BE12$**:** Add a Song to Playlist
- $BE13$**:** Remove a Song To Playlist
- $BE14$**:** Turn Up AC
- $BE15$**:** Turn Down AC
- $BE16$**:** Open Window
- $BE17$**:** Close Window

BE1. Create User Profile

VP1 Customer

- E_1**:** Customer navigates to create user profile
- E_2**:** Customer enters all required personal information.
- S_1**:** System encrypts and saves personal data.
- E_3**:** Customer enters potential password
- S_2**:** System checks that password is a minimum of 8 characters, with a minimum of 1 number and a minimum of 1 uppercase and 1 lowercase letter

  $S_{3.1}$ System determines that the password meets the criteria, proceed to S3

  $S_{3.2}$ System determines that the password does not meet the criteria, loop to E3
- S_3**:** System encrypts and saves password
- S_4**:** System informs user that profile has been created
- S_5**:** System offers user the ability to stay signed in

  $S_{5.1}$ User accepts, and the system no longer requires this user to log in when accessing the software from this device.

  $S_{5.2}$ User rejects and the system requires login information every time the app is launched.

VP2 Dispatcher

- NA, Dispatcher exists as an algorithm built into the application.

Global Scenario User creates a profile in the app

- E_1**:** User navigates to create user profile
- E_2**:** User enters all required personal information.
- S_1**:** System encrypts and saves personal data.
- E_3**:** User enters potential password
- S_2**:** System checks that password is a minimum of 8 characters, with a minimum of 1 number and a minimum of 1 uppercase and 1 lowercase letter

  $S_{3.1}$ System determines that the password meets the criteria, proceed to S3

  $S_{3.2}$ System determines that the password does not meet the criteria, loop to E3
- S_3**:** System encrypts and saves password
- S_4**:** System informs user that profile has been created
- S_5**:** System offers user the ability to stay signed in

  $S_{5.1}$ User accepts, and the system no longer requires this user to log in when accessing the software from this device.

  $S_{5.2}$ User rejects and the system requires login information every time the app is launched.

BE2. Reset Password

Pre-Condition: BE3 Login

VP1 Customer

- E_1**:** Customer initiates a password reset
- S_1**:** System reads user data
- E_2**:** Customer enters potential password
- S_2**:** System checks that password is a minimum of 8 characters, with a minimum of 1 number and a minimum of 1 uppercase and 1 lowercase letter

  $S_{2.1}$ System determines that the password meets the criteria, proceed to S3

  $S_{2.2}$ System determines that the password does not meet the criteria, loop to E1
- S_3**:** System encrypts and saves password

VP2 Dispatcher

- NA, Dispatcher exists as an algorithm built into the application.

Global Scenario User requests a password reset

- E_1**:** User initiates a password reset
- S_1**:** System reads user data
- E_2**:** User enters potential password
- S_2**:** System checks that password is a minimum of 8 characters, with a minimum of 1 number and a minimum of 1 uppercase and 1 lowercase letter

  $S_{2.1}$ System determines that the password meets the criteria, proceed to S3

  $S_{2.2}$ System determines that the password does not meet the criteria, loop to E1
- S_3**:** System encrypts and saves password

BE3. Login

Pre-Condition: BE1 Create User Profile

VP1 Customer

- S_1**:** System prompts customer for a username and password.
- E_1**:** User enters a username and password.
- S_2**:** System authenticates username and password.
  - $S_{2.1}$ Credentials are valid the customer is logged into their account.
  - $S_{2.2}$ Credentials are invalid loop to S1

VP2 Dispatcher

- NA, Dispatcher exists as an algorithm built into the application.

Global Scenario System prompts user to login

- S_1**:** System prompts user for a username and password.
- E_1**:** User enters a username and password.
- S_2**:** System authenticates username and password.
  - $S_{2.1}$ Credentials are valid the user is logged into their account.
  - $S_{2.2}$ Credentials are invalid loop to S1

BE4. Delete Profile

Pre-Condition: BE1 Create User Profile, BE3 Login

VP1 Customer

- E_1**:** Customer navigates to their settings page and initiates a profile removal
- S_1**:** System asks the Customer if they are sure they want to delete their profile
  - $S_{1.1}$ User responds yes, system deletes the profile and all associated data and logs the user out.
  - $S_{1.2}$ User responds no, system does not delete the profile

VP2 Dispatcher

- NA, Dispatcher exists as an algorithm built into the application.

Global Scenario User requests to delete profile

- E_1**:** User navigates to their settings page and initiates a profile removal
- S_1**:** System asks the user if they are sure they want to delete their profile
  - $S_{1.1}$ User responds yes, system deletes the profile and all associated data and logs the user out.
  - $S_{1.2}$ User responds no, system does not delete the profile

BE5. Edit Profile

Pre-Condition: BE1 Create User Profile, BE3 Login

VP1 Customer

- E_1**:** Customer selects edit on their user profile page entering editing mode
- S_1**:** System makes profile fields editable by the user
- E_2**:** Customer makes changes to their profile data in the respective fields
- E_3**:** Customer submits their changes to the system
- S_2**:** System validates that the changes are legal

  $S_{2.1}$ System determines changes are not legal and informs the user of violations, no changes are made to the profile and original data is restored in all fields from the saved data.

  $S_{2.2}$ System determines changes are legal and saves changes accordingly. All changes are now reflected in the user profile page.
- S_3**:** System exits edit mode, all fields on the profile page become static

VP2 Dispatcher

- NA, Dispatcher exists as an algorithm built into the application.

Global Scenario User requests to edit profile

- E_1**:** User selects edit on their user profile page entering editing mode
- S_1**:** System makes profile fields editable by the user
- E_2**:** Customer makes changes to their profile data in the respective fields
- E_3**:** User submits their changes to the system
- S_2**:** System validates that the changes are legal

  $S_{2.1}$ System determines changes are not legal and informs the user of violations, no changes are made to the profile and original data is restored in all fields from the saved data.

  $S_{2.2}$ System determines changes are legal and saves changes accordingly. All changes are now reflected in the user profile page.
- S_3**:** System exits edit mode, all fields on the profile page become static

BE6. Request Taxi Carpool

Pre-Condition: BE1 Create User Profile, BE3 Login

VP1 Customer

- E_1: Customer indicates that they want to request a Taxi carpool.
- E_2: Customer must enter in data pertaining to their ride including but not limited to their starting location, ending location, number of passengers, intended pickup time and if they want music for the duration of the ride.
- E_3: Customer must choose at most one of the following sorting algorithms for the results: Sort by earliest arrival time, Sort by number of passengers small-big, Sort by number of passengers big-small, Sort by highest driver ratings, Sort by lowest distance.
- E_4: Customer submits the data to the system, branch to VP2 then proceed to S2
- S_2: System displays results generated by dispatcher to user
- E_5: User selects a desired route or aborts
  - $E_{5.1}$ if selects desired route proceed to S3
  - $E_{5.2}$ if aborts End Event
- S_3: Selected route is logged by the system, branch to BE7.VP2 then proceed to S4
- S_4: System displays the logged response from the offerer to the customer
- E_6: User chooses to modify their search or choose another result
  - $E_{6.1}$ if the offerer denied the request proceed to S5
  - $E_{6.2}$ if the offerer accepted the request proceed to S6
- S_5: Provide the user with the option of modifying their search or select another result.
- E_7: User chooses to modify their search or choose another result
  - $E_{7.1}$ if the user chooses to modify their search loop to E2
  - $E_{7.2}$ if the user chooses to request another result loop to S2
- S_6: The accepted ride is saved to the users profile

VP2 Dispatcher

- S_1: Requires BE6.VP1.E1-E4
- S_2: Dispatcher reads all data provided in BE6.VP1.E1-E3
- S_3: Dispatcher fetches all currently available offers that it saved
- S_4: Dispatcher calculates the feasibility of all options given the data provided by the customer
- S_5: Dispatcher generates a response to send back to the user
  - $S_{3.1}$ if no options were feasible the response states that there were no available carpool options under the given constraints, proceed to S7
  - $S_{3.2}$ if there were 1 or more feasible options include all in the response, proceed to S6
- S_6: Response is sorted by the users given sort criteria
- S_7: System stores the response

Global Scenario Customer requests taxi carpool

- E_1: Customer indicates that they want to request a Taxi carpool.
- E_2: Customer must enter in data pertaining to their ride including but not limited to their starting location, ending location, number of passengers, intended pickup time and if they want music for the duration of the ride.
- E_3: Customer must choose at most one of the following sorting algorithms for the results: Sort by earliest arrival time, Sort by number of passengers small-big, Sort by number of passengers big-small, Sort by highest driver ratings, Sort by lowest distance.
- E_4: Customer submits the data to the system,
- S_1: Dispatcher reads all data provided
- S_2: Dispatcher fetches all currently available offers that it saved
- S_3: Dispatcher calculates the feasibility of all options given the data provided by the customer
- S_4: Dispatcher generates a response to send back to the user
  - $S_{4.1}$ if no options were feasible the response states that there were no available carpool options under the given constraints, proceed to S7
  - $S_{4.2}$ if there were 1 or more feasible options include all in the response, proceed to S6
- S_5: Response is sorted by the users given sort criteria
- S_6: System stores the response
- S_7: System displays results generated by dispatcher to user
- E_5: User selects a desired route or aborts
  - $E_{5.1}$ if selects desired route proceed to S3
  - $E_{5.2}$ if aborts End Event
- S_8: Selected route is logged by the system, branch to BE7.VP2 then proceed to S4
- S_9: System displays the logged response from the offerer to the customer
- E_6: offerer accepts or denies request
  - $E_{6.1}$ if the offerer denied the request proceed to S5
  - $E_{6.2}$ if the offerer accepted the request proceed to S6
- S_{10}: Provide the user with the option of modifying their search or selecting another result.
- E_7: User chooses to modify their search or choose another result
  - $E_{7.1}$ if the user chooses to modify their search loop to E2
  - $E_{7.2}$ if the user chooses to request another result loop to S2
- S_{11}: The accepted ride is saved to the users profile

BE7. Communicate offer to offering customer

Pre-Condition: BE1 Create User Profile, BE3 Login, BE6.VP1.S1-S3

VP1 Customer

- S_1**:** Requires VP2
- S_2**:** System receives the offer and calculates the new cost of the trip for the offerer alongside the expected delay in minutes and the new distance to be travelled.
- S_3**:** System performs a calculation to determine if the new route is worth it by determining the similarity of the new and previous route, this similarity is displayed as a percentage to the user to help them decide whether it is optimal to accept or refuse the request.
- S_4**:** The system displays the offer alongside the previous calculations to the offerer
- E_1**:** Customer chooses a response, to accept or deny the request
  $E_{1.1}$ if the customer chooses to accept, the response is 'accepted'
  $E_{1.2}$ if the customer chooses to deny, the response is 'denied'
- S_5**:** The response of the customer is logged in the system

VP2 Dispatcher

- E_1**:** Dispatcher reads selected route
- S_2**:** Dispatcher sends request to the offerer, branch VP1

Global Scenario Communicate offer to offering customer

- E_1**:** Dispatcher reads selected route
- S_2**:** Dispatcher sends request to the offerer
- S_3**:** System receives the offer and calculates the new cost of the trip for the offerer alongside the expected delay in minutes and the new distance to be travelled.
- S_4**:** System performs a calculation to determine if the new route is worth it by determining the similarity of the new and previous route, this similarity is displayed as a percentage to the user to help them decide whether it is optimal to accept or refuse the request.
- S_5**:** The system displays the offer alongside the previous calculations to the offerer
- E_2**:** Customer chooses a response, to accept or deny the request
  $E_{2.1}$ if the customer chooses to accept, the response is 'accepted'
  $E_{2.2}$ if the customer chooses to deny, the response is 'denied'
- S_4**:** The response of the customer is logged in the system

BE8. Ride Pickup

Pre-Condition: BE6 Request Taxi Carpool

VP1 Customer

- S_2**:** System keeps track of upcoming saved rides and notifies the user both 10 minutes before the ride arrives, if applicable, and when the ride arrives.
- E_1**:** The customer gets in the vehicle and the ride starts
- S_5**:** If playing songs is enabled for the ride. Users can add and remove songs from a playlist. See BE12, BE13
- S_6**:** The System allows the user to request that the driver turn up or down the AC, turn up or down the central heating, or open or close the windows from the app. See BE14, BE15, BE16, BE17
- S_7**:** The system has a real-time display of the current cost of the ride for the customer at any point in time for the duration of the ride.
- E_2**:** The driver arrives at the passenger's destination.
- S_8**:** The system displays the final cost of the ride to the user
- E_3**:** The user pays their final cost branch BE9, proceed S9
- S_9**:** The system provides the user with a survey to rate each of the passengers in their carpool alongside the driver.

VP2 Dispatcher

- N/A, Dispatcher exists as an algorithm built into the application.

Global Scenario Customer is Picked up

- S_2**:** System keeps track of upcoming saved rides and notifies the user both 10 minutes before the ride arrives, if applicable, and when the ride arrives.
- E_1**:** The customer gets in the vehicle and the ride starts
- S_5**:** If playing songs is enabled for the ride. Users can add and remove songs from a playlist. See BE12, BE13
- S_6**:** The System allows the user to request that the driver turn up or down the AC, turn up or down the central heating, or open or close the windows from the app.
- S_7**:** The system has a real-time display of the current cost of the ride for the customer at any point in time for the duration of the ride.
- E_2**:** The driver arrives at the passenger's destination.
- S_8**:** The system displays the final cost of the ride to the user
- E_3**:** The user pays their final cost branch BE9, proceed S9
- S_9**:** The system provides the user with a survey to rate each of the passengers in their carpool alongside the driver.

BE9. Ride Drop-off

Pre-Condition: BE8 Ride Pickup

VP1 Customer

- E_1**:** The driver arrives at the passenger's destination.
- S_1**:** The system displays the final cost of the ride to the user
- E_2**:** The user pays their final cost branch BE10, proceed S2
- S_2**:** The system provides the user with a survey to rate each of the passengers in their carpool alongside the driver.

VP2 Dispatcher

- N/A, Dispatcher exists as an algorithm built into the application.

Global Scenario Customer is dropped-off

- E_1**:** The driver arrives at the passenger's destination.
- S_1**:** The system displays the final cost of the ride to the user
- E_2**:** The user pays their final cost branch BE10, proceed S2
- S_2**:** The system provides the user with a survey to rate each of the passengers in their carpool alongside the driver.

BE10. Make A Payment

VP1 Customer

- [specifics are outside of the scope]

VP2 Dispatcher

- N/A

Global Scenario Make a payment

- [specifics are outside of the scope]

BE11. Offer Taxi Carpool

Pre-Condition: BE8 Ride Pickup

VP1 Customer

- E_1**:** Customer scans QR code inside the vehicle that represents a taxi ID
- E_2**:** Customer selects ride-sharing and enters required information some of which include the maximum number of people willing to carpool with and the maximum delay the customer is willing to endure in minutes and if the offerer will allow music in the carpool.
- S_1**:** System saves ride offering

VP2 Dispatcher

- S_1**:** Dispatcher receives a request to offer a ride-share
- S_2**:** System saves the request to the list of ongoing open requests

Global Scenario Offer Taxi Carpool

- E_1**:** Customer scans QR code inside the vehicle that represents a taxi ID
- E_2**:** Customer selects ride-sharing and enters required information some of which include the maximum number of people willing to carpool with and the maximum delay the customer is willing to endure in minutes and if the offerer will allow music in the carpool.
- S_1**:** System receives a request to offer a ride-share
- S_2**:** System saves the request to the list of ongoing open requests

BE12. Add A Song to Playlist

Pre-Condition: BE8 Ride Pickup

VP1 Customer

- E_1**:** User enters the name and artist of a song
- S_1**:** System validates that the song exists using the Spotify API

  $E_{2.1}$ if Spotify returns valid instances, choose the first song occurrence when sorted by number of streams, this is the selected song, and add it to the playlist

  $E_{2.2}$ else do not edit the playlist.

VP2 Dispatcher

- N/A

Global Scenario Add a song to Playlist

- E_1**:** User enters the name and artist of a song
- S_1**:** System validates that the song exists using the Spotify API

  $E_{2.1}$ if Spotify returns valid instances, choose the first song occurrence when sorted by number of streams, this is the selected song, and add it to the playlist

  $E_{2.2}$ else do not edit the playlist.

BE13. Remove A Song From Playlist

Pre-Condition: BE8 Ride Pickup

VP1 Customer

- E_1**:** User selects a song from the playlist from a drop-down.
- S_1**:** System removes the selected song.

VP2 Dispatcher

- N/A

Global Scenario Remove a song form playlist

- E_1**:** User selects a song from the playlist from a drop-down.
- S_1**:** System removes the selected song.

BE14. Turn Up AC

Pre-Condition: BE8 Ride Pickup

VP1 Customer

- E_1**:** User indicates they want to raise the temperature by x degrees
- S_1**:** System reads the requested change.
  - $S_{2.1}$**:** if the change does not cause the car to exceed the maximum allowed temperature perform the change
  - $S_{2.2}$**:** Else do not raise the temperature and inform the user that the maximum allowed temperature would be exceed if the change was applied.

VP2 Dispatcher

- N/A

Global Scenario Turn AC Up

- E_1**:** User indicates they want to raise the temperature by x degrees
- S_1**:** System reads the requested change.
  - $S_{2.1}$**:** if the change does not cause the car to exceed the maximum allowed temperature perform the change
  - $S_{2.2}$**:** Else do not raise the temperature and inform the user that the maximum allowed temperature would be exceed if the change was applied.

BE15. Turn Down AC

Pre-Condition: BE8 Ride Pickup

VP1 Customer

- E_1**:** User indicates they want to lower the temperature by x degrees
- S_1**:** System reads the requested change.
  - $S_{2.1}$**:** if the change does not cause the car to exceed the minimum allowed temperature perform the change
  - $S_{2.2}$**:** Else do not lower the temperature and inform the user that the minimum allowed temperature would be exceed if the change was applied.

VP2 Dispatcher

- N/A

Global Scenario Turn AC Down

- E_1**:** User indicates they want to lower the temperature by x degrees
- S_1**:** System reads the requested change.
  - $S_{2.1}$**:**, if the change does not cause the car to exceed the minimum, allowed temperature perform the change
  - $S_{2.2}$**:** Else do not lower the temperature and inform the user that the minimum allowed temperature would be exceed if the change was applied.

BE16. Open Window

Pre-Condition: BE8 Ride Pickup

VP1 Customer

- E_1**:** User indicates they want to open the window
- S_1**:** System reads the requested change.
  - $S_{2.1}$**:** if the window is closed open it
  - $S_{2.2}$**:** if the window is open tell the user that the window is already open

VP2 Dispatcher

- N/A

Global Scenario Open Window

- E_1**:** User indicates they want to open the window
- S_1**:** System reads the requested change.
  - $S_{2.1}$**:** if the window is closed open it
  - $S_{2.2}$**:** if the window is open tell the user that the window is already open

BE17. Close Window

Pre-Condition: BE8 Ride Pickup

VP1 Customer

- E_1**:** User indicates they want to close the window
- S_1**:** System reads the requested change.
  - $S_{2.1}$**:** if the window is open close it
  - $S_{2.2}$**:** if the window is closed tell the user that the window is already closed

VP2 Dispatcher

- N/A

Global Scenario Close Window

- E_1**:** User indicates they want to close the window
- S_1**:** System reads the requested change.
  - $S_{2.1}$**:** if the window is open close it
  - $S_{2.2}$**:** if the window is closed tell the user that the window is already closed

5 Non-Functional Requirements

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF-A1. *The app should utilise the main colour combo of yellow and white.*
Rationale: The inclusion of CarpoolClan's primary colours in the app's design can enhance the company's branding and help customers identify the app with recognizable taxi colours. This can potentially increase the app's user base and encourage users to continue using the app as their primary carpooling service.

LF-A2. *The app should provide a dark mode with the main colour combo of yellow, white, and black.*
Rationale: As a significant number of mobile users prefer using dark mode, the app should provide an easy-on-the-eyes colour scheme that remains consistent with CarpoolClan's primary colours.

LF-A3. *All characters in the app should be ASCII characters.*
Rationale: To ensure simplicity in the app design as well as increase support for older mobile devices, the app should only use ASCII characters.

5.1.2 Style Requirements

LF-S1. *When questioned about their experience, 80% of typical users should characterize the application as fast and responsive.*
Rationale: It is important to ensure that the app provides a positive user experience by being described as fast and responsive. If 80% of the users describe the app as such, it reflects that the app is meeting its goal and users are more likely to continue using it.

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH-EOU1. *The app should provide a highlighted or hinted solution to errors made by the user.*
Rationale: If the user makes a mistake while using the app (ex. filling in wrong information error, selecting restricted actions), the app shall provide a short or detailed solution that does not clutter the user experience and responsiveness of the app.

UH-EOU2. *A user under the age of 55 should be able to use the app to request or offer a carpool within 5 minutes.*
Rationale: It's important to ensure that users under the age of 55 can use the app to request or offer a carpool without error. A significant portion of the population falls within this age range, and targeting this demographic can increase the app's potential user base and improve their overall satisfaction with the app, leading to increased usage.

5.2.2 Personalization and Internationalization Requirements

UH-PI1. *The app should allow the user to edit their account profile details, such as Name, address, or gender.*
Rationale: The app should allow users to edit their profile details, such as name, address, and gender. This gives users greater control over their personal information, improves the accuracy of user data, promotes transparency and accountability, and ultimately contributes to a better user experience.

5.2.3 Learning Requirements

UH-L1. *The app should provide a link to a walk-through to inform the user how to request or offer carpools.*
Rationale: To provide clear instructions on how to request or offer a carpool, the app should offer users a video walk-through accessible via a link in the support section. This will serve as a direct resource for users to learn about the process in an easy-to-understand format.

5.2.4 Understandability and Politeness Requirements

UH-UP1. *The app should use follow proper English grammar and spelling.*
Rationale: Using proper English grammar and spelling is important for effective communication, establishing credibility and professionalism, maintaining consistency, and making the app accessible to a user base.

UH-UP2. *The app should not use vulgar language or offensive language of any kind.*
Rationale: Using vulgar or offensive language in an app can be disrespectful, unprofessional, and can have legal implications. Avoiding such language is important for maintaining a positive user experience, retaining users, and upholding the app's professionalism and brand values.

5.2.5 Accessibility Requirements

UH-A1. *The app should be compatible with screen readers on devices running Android version 5.0 or later.*
Rationale: The app should accommodate users who depend on screen readers to navigate their mobile devices. The design of the app should allow users using screen readers to have a quick and seamless experience with the app.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR-SL1. *The app should respond to selections within an average of 5 seconds or less.*
Rationale: To provide a seamless and responsive experience to users, the app should minimize lag when prompted with selections to seamlessly transition to the next page or selection.

PR-SL2. *The app should recalculate altered routes within 1 minute when both users accept the carpool offer.*
Rationale: To minimize idling time and increase ride length, the app should recalculate the new route within 1 minute of accepting the carpool offer.

5.3.2 Safety-Critical Requirements

PR-SC1. *The app should provide filters for gender that will narrow the available carpool options.*
Rationale: To maximize user comfort, it's essential to recognize that some individuals may not feel at ease sharing an enclosed space with someone of another gender. Therefore, users should be given the option to select their preferred riding partner. The application is available to all and although users should be able to choose who they feel comfortable riding with, the app is all-inclusive and any legal user may use the app without discrimination.

PR-SC2. *When creating their account, the app should prompt the user for verification that they are above the age of 18.*
Rationale: To ensure the safety of the user, it is crucial to confirm that they are of legal age, thereby preventing minors from being exposed to potential harm from strangers.

PR-SC3. *When creating their account, the app should prompt the user for verification that they have parental permission if they are age 13 or above and not yet legal age.*
Rationale: To ensure the safety of under-aged users, the app has the responsibility to verify the intentions of parents who permit their child(s) to ride with other users. The app is not responsible for users lying about their age.

5.3.3 Precision or Accuracy Requirements

PR-PA1. *The temperature control settings for WeatherWizard should be limited to between 20 and 25 degrees Celsius.*
Rationale: Limiting WeatherWizard's temperature control settings to between 20 and 25 degrees Celsius is important because it helps to ensure that users do not create temperature extremes that can be hazardous to their health, promoting user safety [4].

5.3.4 Reliability and Availability Requirements

PR-RA1. *The search request should be cancelled if the system is unable to find a suitable carpool for the user within 1 minute of match-making.*
Rationale: Cancelling a search request if a suitable carpool cannot be found within a minute is a practical approach to prioritize the user's time, prevent system overload, and ensure transparency in the service provided.

5.3.5 Robustness or Fault-Tolerance Requirements

N/A

5.3.6 Capacity Requirements

PR-C1. *The app should be able to manage a minimum of eight simultaneous users: four requesting a carpool and the other four offering a carpool.*
Rationale: To ensure that the app can facilitate carpooling effectively, it must be capable of managing eight users simultaneously. This means that users must be able to both request and accept a carpool offer from another user, requiring the app to support all four pairs of users' actions at the same time.

5.3.7 Scalability or Extensibility Requirements

N/A

5.3.8 Longevity Requirements

PR-L1. *The app should have an expected lifetime of until the end of April 10, 2023.*
Rationale: It is crucial for the app to be fully operational when submitted for review, which also means that it should remain operational until the end of the semester, April 10, 2023.

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

N/A

5.4.2 Requirements for Interfacing with Adjacent Systems

OE-IA1. *The app should properly interface with the Spotify API.*
Rationale: In order to allow users to select and listen to music during their taxi ride, MixMaster will utilize the Spotify API to add songs to a queue.

OE-IA2. *The app should properly interface with the Google Maps Platform.*
Rationale: To provide users with a recognizable and user-friendly visual representation of their carpool route, the app will integrate the Google Maps Platform for displaying route information for both users requesting and offering carpools.

5.4.3 Productization Requirements

N/A

5.4.4 Release Requirements

N/A

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS-M1. *Software updates will be released when a new feature has been added to the app.*
Rationale: Releasing software updates when a new feature has been added to the app is a practical approach to ensure that the app remains reliable, engaging, and competitive in the marketplace, providing the user with the best possible experience.

5.5.2 Supportability Requirements

N/A

5.5.3 Adaptability Requirements

N/A

5.6 Security Requirements

5.6.1 Access Requirements

SR-AC1. *The carpool selections that have been matched should only be visible to the user who has requested said carpools.*
Rationale: This measure is in place to guarantee that no user can view another user's carpool selections or information, either deliberately or accidentally, at any point in time.

5.6.2 Integrity Requirements

SR-INT1. *Only the user requesting carpools may modify the inputted carpooling information used to match-make carpool requests.*
Rationale: No other user or the system itself should be able to modify or make changes to the information inputted by the user.

SR-INT2. *Only the user offering carpools may modify the inputted carpooling information used to match-make carpool offers.*
Rationale: No other user or the system itself should be able to modify or make changes to the information inputted by the user.

5.6.3 Privacy Requirements

SR-P1. *All transmitted data and user login credentials should be encrypted using AES encryption.*
Rationale: Encrypting all transmitted data and user login credentials using AES encryption is a practical and effective approach to ensure the security and confidentiality of sensitive user information, protect against cyber threats, and comply with data protection regulations.

5.6.4 Audit Requirements

N/A

5.6.5 Immunity Requirements

N/A

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

N/A

5.7.2 Political Requirements

CP-P1. *The app should only use official graphics from CarpoolClan, MixMaster, and WeatherWizard as icons.*
Rationale: To prevent the use of offensive or copyrighted material in the app, only materials created and/or distributed by CarpoolClan, MixMaster, and WeatherWizard shall be utilized in the design.

5.8 Legal Requirements

5.8.1 Compliance Requirements

LR-COMP1. *The app must comply with the Personal Information Protection and Electronic Documents Act (PIPEDA).*
Rationale: PIPEDA requires organizations to obtain consent for the collection, use, and disclosure of personal information, limit the collection of personal information to what is necessary, use appropriate safeguards to protect personal information, and provide individuals with access to their personal information and the ability to request corrections [5]. Non-compliance can result in fines and reputational damage.

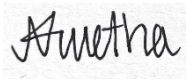
LR-COMP2. *The app must comply with the Accessibility for Ontarians with Disabilities Act (AODA).*
Rationale: The AODA is a law in Ontario that requires organizations to make their products and services accessible to people with disabilities [6]. Compliance means ensuring accessibility, which is a legal and moral responsibility. It can promote inclusivity, improve reputation, and failure to comply can result in penalties.

5.8.2 Standards Requirements

N/A

6 Division of Labour

Include a Division of Labour sheet that indicates the contributions of each team member. This sheet must be signed by all team members.

- $Q1, 5$**:** Moamen 
- $Q2, 4$**:** Nivetha 
- $Q3, 4$**:** Jinal 
- $Q3, 4$**:** Mya 
- $Q1, 5$**:** Aadil