<Carpool system>

Requirements Specification Document

Author1: Erik Wen Han Sun <es225ki >

Author2: Zejian Wang <zw222bb>

Contents

1. Introduction

This documentation presents the details of the Requirements Engineering process for a Carpool system. The purpose is to evaluate the requirements, clearly comprehend the needs of customers, assist engineers to process issues and set goals for the system by using methods that are well known and effective. The document identifies, analyzes and documents the requirements of this carpool system. It contains the system requirements, system interfaces, system constraints, business rule, and use case.

The carpool system is a GPS based mobile application for car-sharing that allows the users to publish their routes or search via inputted departure and destination to match the published routes to offer or get a ride for the purpose of convenience, environment-friendness and cost-saving. There is also a web version of the application used by the application administrator to administer the system, manage users' information. All information in the system is maintained in the database located in a web-server.

Section 2 identifies the stakeholders and functional and non-functional requirements, then performs a series of evaluations on the requirements, including requirement classifications, analysis, risk assessments, systematic validations, and test cases. Section 3 describes the interfaces that this system requires during the development process of the system, including User, hardware, and software interfaces. Section 4 states the business rules that apply to this carpool system, including driver risk assessment, customer validation, application agreement, discount, safety, application rules. Section 5 describes different aspects of constraints that the system encounters, including software implementation Languages, prescribed use of developmental tools, third-Party components or class libraries, platform support and resource limits. Section 6 presents a series of use cases that records the scene of the user and actor interacting with the system. The use cases are fully addressed and provide one or more scenes for the user to interact with in order to improve the requirements. Section 7 modeling quality attribute requirements by using quality attributes scenarios. They specify if the system fulfills the customer's needs on a more advanced level in terms of modifiability, availability, testability, performance, security, and usability. The system is developed based on a list of assumptions as Figure 1-1 shows.

Assumption No.	Description
1	This system doesn't support payment with cash
2	Only user who has Bankid can use this software system
3	The system only supports IOS and Android
4	This system only works in Sweden

Figure 1-1

2. System Requirements

2.1 Stakeholders

Users: The largest group of stakeholders. People who ended up using this system.

Development team: Engineers who develop the system.

UI and UX designers: Designers who design the graphic interfaces or engage closely with the users for feedback to improve the system.

Project manager: Manager who leads the direction of development of this system.

Maintenance team: Engineers who have to debug or fix errors for the system.

Business Owner: Whoever owns this system from a legal perspective.

Legal Department: For providing advice regarding laws and data protection regulations.

Authorities: Government department to authorize the business.

2.2 Functional Requirements

No.	Requirements Title	Description
FR1	**User Registration**	The user should be able to be registered and verified through email address, personal number, phone number and Bankid.
FR2	**User Login**	The user should be able to log in via username and password or Bankid.
FR3	**Book rides**	The user should be able to book rides if there's a match after inputting departure, destination and time.
FR4	**Role selection**	The user should be able to change his/her role to the other.
FR5	**Ride Information Display**	The system should be able to display the time, route and cost of the ride after the carpooling is done.
FR6	**Necessary Information Sharing**	The necessary personal information should be shared between the passengers and the driver after both accept the ride.
FR7	**Real-time Map Display**	The system should be able to show the route and current location of the vehicle during the whole trip.

FR8	**Search Rides**	The passenger should be able to search for rides by entering start and destination addresses in the system.
FR9	**Payment Method**	The system should be able to provide a way to let the passenger do the payment via Swish after both the passenger and the driver confirm the trip is finished.
FR10	**Driver Gets Payment**	The system should transfer the payment to the driver after the trip is finished.
FR11	**Rating System**	The passenger and the driver should be able to rate each other after the trip is finished. The rating is from 1 to 5 stars of each option associated with a comment.
FR12	**History Check**	The users should be able to check their trip history.
FR13	**Customer Service**	The system should provide online customer service for the users.
FR 14	**Users Communication**	The driver and passenger are allowed to call each other by phone number within a certain amount of time.
FR 15	**Display Matching Percentage**	The system displays matching percentages of routes after the carpooling.
FR 16	**Information Modification**	There should be a profile page where the users can manage and modify their information.
FR 17	**Unregistration**	the users are allowed to delete their account.
FR 18	**Ride cancellation**	The users are allowed to cancel the ride.
FR 19	**Estimated Ride Information Display**	The system should be able to display the estimated time, route and cost of the ride to the users after the ride is matched.
FR 20	**Block Service**	The system will suspend service to the user who violates the rules and regulations of the system.

Figure 2-1

2.3 Non-functional requirements

No.	Requirements Title	Description
NFR1	**Time Limit for User Login**	User login should be completed within 3 seconds
NFR2	**Time Limit for Ride Search**	The system should match a route for the users within 5 seconds
NFR3	**Time Limit for Payment Transfer for Driver**	The driver should get paid within one hour.
NFR4	**Time Period in Carpool History**	The users can select and view a specific period of time in their carpool history.
NFR5	**Number Mask**	The users can call each other, but only part of the number will be displayed.
NFR6	**Information Modification Limit**	The user is only allowed to modify their username and email every at least 6 months.
NFR7	**Revive Account**	After the user has unregistered their account, they can still revive it within 7 days.

Figure 2-2

2.4 Classification of Requirements

Classification	Functional requirements No	Non-Functional requirements No
System	3, 5, 7, 8, 19	1, 2, 3
User interface	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20	1, 2, 3, 4, 5, 6, 7
Communications	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 20	1, 2, 3, 4, 7
Database	1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 17, 20	1, 6, 7
Security	1, 2, 6, 9, 10, 8, 16	5

Figure 2-3 [1]

2.5 Requirements Analysis

1. Does the requirement include premature design or implementation information?

2. Could the description of a requirement be broken down into several different requirements?
3. Is the requirement 'gold plating'?
4. Does the requirement mean that non-standard hardware or software must be used?
5. Is the requirement consistent?
6. Is the requirement ambiguous?
7. Is the requirement realistic?
8. Is the requirement testable? [3]

Question	FR 1	FR 2	FR 3	FR 4	FR 5	FR 6	FR 7	FR 8	FR 9	FR 10	FR 11	FR 12	FR 13	FR 14	FR 15	FR 16	FR 17	FR 18	FR 19	FR 20
1	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
2	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
3	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
4	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
5	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
6	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
7	Yes	Yes	Yes	yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
8	Yes	Yes	Yes	yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Figure 2-4

Question	NFR 1	NFR 2	NFR 3	NFR 4	NFR 5	NFR 6	NFR 7	NFR 8	NFR 9	NFR 10
1	No	No	No	No	No	No	No	No	No	No
2	No	No	No	No	No	No	No	No	No	No
3	No	No	No	No	No	No	No	No	No	No
4	No	No	No	No	No	No	No	No	No	No
5	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
6	No	No	No	No	No	No	No	No	No	No
7	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
8	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Figure 2-5

2.6 Systematic Risk Assessment

Systematic risk assessment of Functional requirements

H-High, M-Medium, L-Low

Risks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Performance	L	L	M	L	L	L	H	M	L	L	L	L	L	L	M	L	L	L	M	L
Safety and Security	M	L	L	L	L	M	L	L	H	M	L	L	L	M	L	M	L	L	L	L
Process	L	L	M	L	L	L	M	M	L	L	L	L	L	L	M	L	L	L	M	L
Implementation technology	L	L	L	L	L	L	H	M	M	L	L	L	L	L	M	L	L	L	M	L
Database	M	M	L	M	L	M	M	L	M	M	M	M	L	L	L	M	M	L	L	M
Schedule	L	L	L	L	L	L	M	M	L	L	L	L	L	L	M	L	L	L	L	L
External	M	M	L	L	L	L	H	L	M	M	L	L	L	L	L	L	L	L	M	L
Stability	L	L	M	L	L	L	M	M	M	L	L	L	L	L	L	L	L	L	M	L

Figure 2-6 [1]

Systematic risk assessment of Non-Functional requirements

H-High, M-Medium, L-Low

Risks	1	2	3	4	5	6	7
Performance	M	H	L	L	L	L	L
Safety and Security	L	L	M	L	H	L	M
Process	M	M	L	L	L	L	L
Implementation technology	M	H	L	L	L	L	L
Database	M	L	M	M	L	M	M
Schedule	L	L	L	L	L	L	L
External	L	L	M	L	L	L	L
Stability	M	L	L	L	L	L	L

Figure 2-7 [1]

2.7 Systematic Validation

Questions	Answer
Are the requirements complete?	Yes
Are the requirements consistent?	Yes
Are the requirements comprehensible?	Yes
Are the requirements ambiguous?	No
Is the requirements document structured?	Yes
Are the requirements traceable?	Yes
Does the requirements document as a whole conform to defined standards?	Yes

Figure 2-8 [3]

2.8 Test Cases

Test Case 01

Requirement Title	User Registration
Priority	High
Description	Test if the user can register in the system.
Precondition	The mobile application has been downloaded to the user's phone.
Test Steps	● Open the application ● Set username and password ● Input personal information required ● Input password on Bankid to confirm
Expected Result	Register the user successfully and the information is stored on the system's database.
Status(pass/fail)	pass

Test Case 02

Requirement Title	Payment Method
Related Requirement	Ride Information Display
Priority	High
Description	Test if the payment can be successfully done.
Precondition	Both passenger and driver confirm the trip is finished.
Test Steps	● The passenger pays with swish ● The passenger verifies with Bankid ● The passenger completes the payment ● The passenger checks the payment history record
Expected Result	The payment was successfully done and the payment transfer showed in Swish.
Status(pass/fail)	pass

Test Case 03

Requirement Title	Driver Gets Payment
Priority	High
Description	Test if the driver gets his/her payment within one hour from the system after the trip finished.
Precondition	The vehicle arrived at the destination.
Test Steps	● The driver confirms the trip finished ● The passenger confirms the trip finished ● The driver open Swish ● The driver check if the payment has been transferred
Expected Result	The payment has been transferred to the driver’s Swish.
Status	pass

Test Case 04

Requirement Title	Rating System
Priority	Medium
Description	Test if the rating system function works.
Precondition	The vehicle arrives at its destination.
Test Steps	● The driver confirms the trip finished ● The passenger confirms the trip finished ● The system pops up “Rate the carpool” window on passenger’s phone ● The passenger gives star to each evaluation criteria ● The passenger submit the evaluation
Expected Result	The system renews the rating score of the driver.
Problem	The result shown in the system may not be correct since the shown score is an average based on many reviews. So one single new review may not be able to change the average score.
Recommendation	A specific math tool can be used to improve the rating algorithm.

Test Case 05

Requirement Title	Rating System
Priority	Medium
Description	Test if the rating system function works
Precondition	The vehicle arrives in destination
Test Steps	● The driver confirms the trip finished ● The passenger confirms the trip finished ● The system pops up “Rate the passenger” window on passenger’s phone ● The driver gives star to each evaluation criteria ● The driver submit the evaluation
Expected Result	The system renews the rating score of the passenger.
Problem	The result shown in the system may not be correct since the shown score is an average based on many reviews. So one single new review may not be able to change the average score.
Solution	A Math tool can be used to check if the rating algorithm is appropriate.

Test Case 06

Requirement Title	Customer Service
Priority	Medium
Description	Test if the user can chat with the customer service online.
Precondition	The user logged in the system.
Test Steps	• A user opens the customer service section in the system • The user opens online chat box • The user input some text on the chat box • The user send the text • The administrator check the chat box on web version of this application
Expected Result	The user successfully reaches the customer service by chatbox.
Status	pass

Test Case 07

Requirement Title	User Communication
Related Requirement	History check
Priority	Medium
Description	Test if the passenger can call the driver.
Precondition	The passenger has logged in.
Test Steps	• The passenger opens the profile page • The passenger check his/her carpooling history • The passenger select the order in the history to see the phone number of the driver • The passenger checks the phone number • The passenger do the call operation
Expected Result	Call successed
Status	pass

3. System Interfaces

3.1. User Interfaces

There will be several user interfaces implemented, including a real-time map interface, a payment interface, a customer service interface, a rating interface, and a profile interface. Switching from one interface to another should be quick and fluid. All interfaces should be simple, straight forward, clean and elegant.

3.1.1. Look & Feel

The design is simple, straightforward, and elegant. The default background is primarily white and then faint yellow, the font should be appropriate to provide a pleasant feeling. The main function area should be appealing, and the real-time map can be zoom sensitive and provide an easy way to do location. The area for advertising should not be annoying. The news to remind the users should be eye-catching.

3.1.2. Layout and Navigation Requirements

After opening the application, the home page is presented, and it should only include two parts, one is the real-time map area, the other is the departure and destination. The departure place can be automatically located on the user's current location if the user opens his/her phone GPS function, if not, the application will remind the user to open it. Users can also manually input the departure location. For the destination, users can either swipe the map to locate the destination or manually input the address of the destination. The area that needs user input should be in eye-catching font. The figure-1 is the mobile application user interface after the user login, and figure-2 is the real-time map showing the route and current location of the vehicle. Figure-3 showing login interface of the web application.

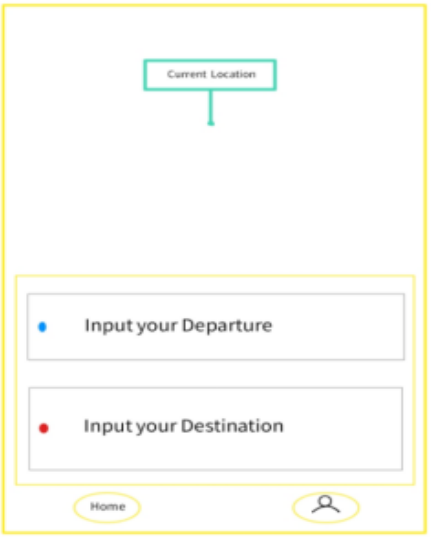

Figure 3-1

Figure 3-2

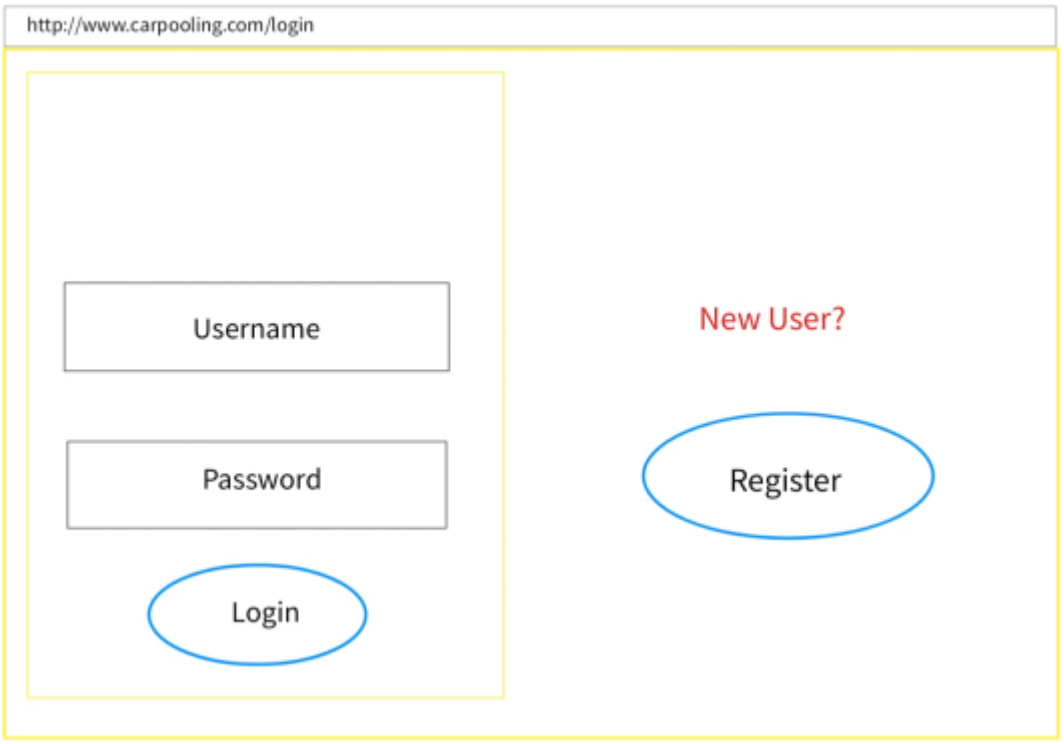

Figure 3-3

3.1.3. Consistency

After the first login, the user will not be required to enter the username and password unless the application has not been used for one month. When you launch the application, the home page is always displayed first, and the user's current location is marked on the real-time map if the GPS is enabled. If the user resets the user interface preference, the default one will no longer be displayed when open the application next time. If there are any development iterations after launch, the interface of each function should not be significantly changed, nor should the meaning of terms used in the application.

3.1.4. User Personalization & Customization Requirements

After the route match, the price of the upcoming trip should be displayed on the screen automatically. The discount price is determined by the user's accumulated travel miles and their identities–students or elderly adults. The real-time map should display the current location as well as the remaining travel miles. The user can also customize the style of that interface.

3.2. Interfaces to External Systems or Devices

3.2.1. Software Interfaces

This mobile application interacts with the GPS application already installed in the phone to give geographical information, it interacts with the Google Map to have real-time map function. MySql is used for database management of all the users' information, the mobile application

interacts with Swish for the user to do payment, it also interacts with Bankid for the user registration and payment, it also interacts with text input software to do the input function.

3.2.2. Hardware Interfaces

The physical GPS chip is managed by its application already installed in the phone. The web server is the web application hosted on, listening on the web standard port, port 80. There is also a hardware connection to the database server, which is managed by the operating systems of both. The monitor, mouse and keyboard are also the hardware interfaces used.

3.2.3. Communications Interfaces

The communication between the server and the client is made by HTTP or HTTPS protocol. The communication between the subparts of the system is handled by its underlying operating system for the application.

4. Business Rules

4.1 Registration rules

- The system requires the driver to provide their driver's license when registering an account.
- The system requires the driver to upload an eyesight examination report annually when registering.
- The type of vehicle should be a car, the system would not allow drivers with other kinds of vehicles to register.
- The system does not allow users who are suffering from a disease that has the potential to affect the driving to register as a driver.

4.2 Safety rules

- The driver is responsible for the safety of this carpool instead of the system.
- Passengers are not allowed to carry flammable, toxic, radioactive and other dangerous items, otherwise they are not allowed to get into the car.
- The driver's vehicle should be checked and maintained on a regular basis according to local laws and regulations, and then the driver must report the statuses of the car and upload related documents in the system.
- The system requires the driver to upload insurance documents of the car when registering an account
- If the passenger feels dangerous due to unsafe driving or violation against safety obligations, fines and warnings will be issued, and if the circumstances are serious, the service will be stopped for the driver.
- Passengers who interfere with driving during the ride are warned in minor cases. If the case is more serious, the passenger will be suspended from all services for a period of time

4.3 Application Rules

- If the user decides to cancel the carpool, then the system will charge them a cancellation fee.
- If the users engage in a conflict that couldn't be resolved privately, the system will contact authorities to be involved.
- If a user decides to use the system, then they automatically agree to all the terms of use of this system.
- If the user decides to use this system, then it means that all the payments should go through the system. There are no direct payments between users.

4.4 Carpool Fee Rules

- Carpool fee consists of basic, mileage, duration fee. Basic fee refers to the minimum amount charged. Mileage fee refers to the fee charged according to the travel distance, Duration fee refers to the fee charged according to the travel time.
- The system charges 10 % of the payment as commission.
- Fees can be represented in 3 ways: estimated, payable, and actual fee.
- Estimated fee is calculated with an algorithm after the passenger has entered the start and destination addresses. The system is not responsible for it. Payable fee is based on the travel mileage and time. Actual fee is what the passenger is actually paying for after deducting the discount from the payable fee.

5. System Constraints

5.1 Software Implementation Languages

- The communication between the system and database is MySQL
- The web version application is developed using HTML, Javascript and css
- The mobile application is written in Java
- The third-party libraries or components used for this system must be adaptable by the chosen programming languages.

5.2 Prescribed Use of Developmental Tools

- Git is used to manage the source code of the carpooling system.
- Lucid chart is used in UML design.
- BootStrap is used in the creation of the web application.

5.3 Third-Party Components or Class Libraries

- The carpooling system uses third-party components or class libraries, mainly libraries for handling payment processing or querying data.
- Any third-party components or libraries must be properly licensed and meet the security and reliability requirements of the carpooling system.

5.4 Platform Support

- The mobile application is only able to run on Android and IOS operating systems .
- The system has a web version, but it is only accessible by engineers and administrators for testing and development.

5.5 Resource Limits

- The carpooling system is implemented on mobiles phones, as a result provided resources are not as powerful as the ones on pc, mainly RAM and network (user would most likely use data instead of wifi, and is limited by phone signal).
- As the system grows with more users, features and data, requiring more storage and RAM. Performance could be affected.

6. Use-Cases

6.1 Use Case 1

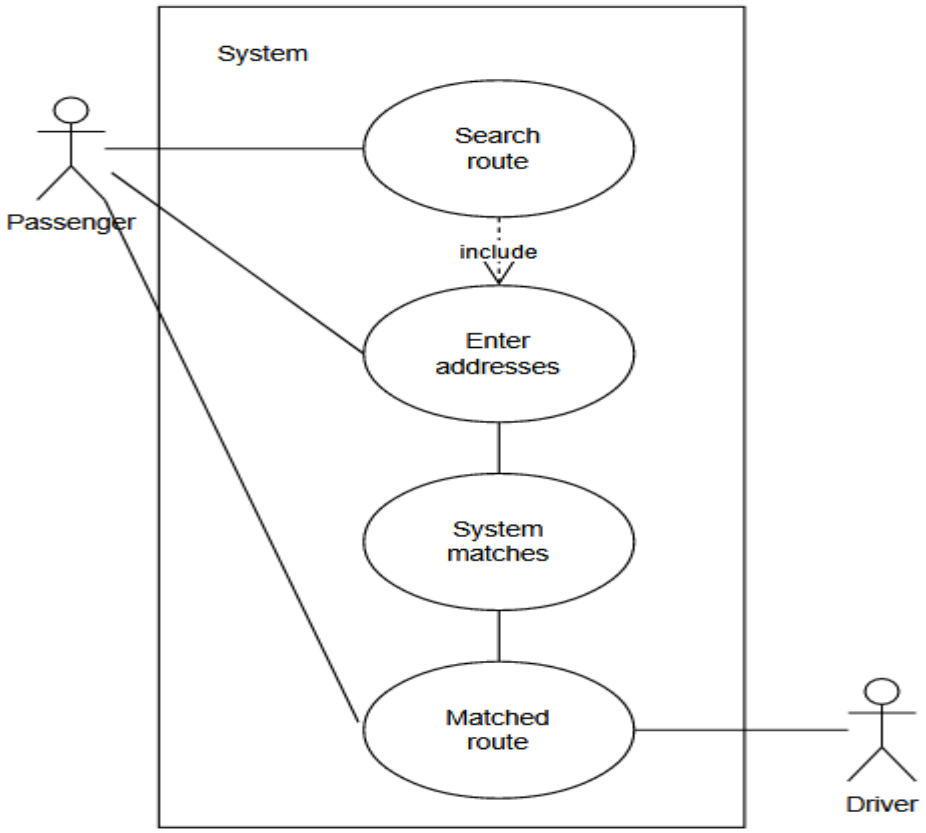

Figure 6-1 The passenger books a ride

Use Case (UC) Section	**Comment/Description**
Brief Description	This use case describes the details of a passenger booking a ride.
Actor brief description	Involves a passenger.
Precondition	1. Both the passengers must be logged in. 2. There must be at least one route already in the system.
Basic flow of events	1. The passenger enters the start and destination address and searches for a ride. 2. The system computes the information. 3. The system presented matched routes and percentages. 4. The passenger requests the most desirable route. 5. The driver accepts the request.

Alternative flows	If there's no match after the system's computation: 1. Then the system shows a message to clarify that. 2. The passenger retry the process again. If the driver refuses the passenger's request: 1. The system shows the passenger the message 2. The passenger tries match another ride
Key Scenarios	The drivers: 1. post their routes The passenger : 1. enter the start and destination address 2. Search the routes
Post-conditions	1. The passenger is picked up by the driver. 2. The pickup is confirmed by both. 3. The destination has arrived. 4. The payment is swished by the user. 5. The payment is transferred to the driver.
Special requirements	1. The passenger should not enter start and destination addresses that are across the entire country.
Open issues	1. How long does it take for the passenger to get a match? 2. What if the passenger never gets a match? Does the system have any backup solutions?

Figure 6-2 [2]

6.2 Use Case 2

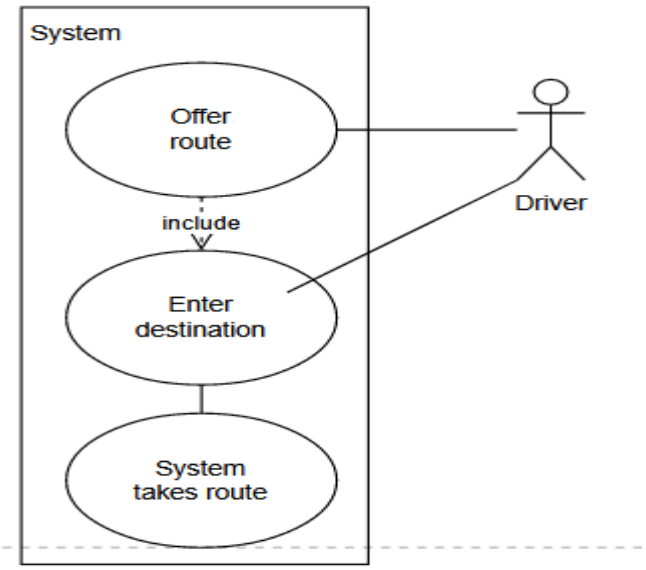

Figure 6-3 The driver publishes a ride

Use Case (UC) Section	**Comment/Description**
Brief Description	This use case describes the detail of the process that a driver offers a carpool.
Actor brief description	Main actor is the driver. The passenger will be involved as well.
Precondition	The user must have chosen the driver mode when they log into the application.
Basic flow of events	1. A driver opens the main page of the application. 2. The driver enters the start and destination. 3. The driver offers this route to the system. 4. The system computes and tries to match a passenger. 5. The driver waits for passenger requests
Alternative flows	Canceling the carpool 1. After the driver offers the route to the system 2. The driver changes his mind about carpooling 3. The driver press “cancel matching” button in the system 4. The system remove the offer Change the route or destination 1. After the driver offers the route to the system. 2. The driver changes her mind about the destination. 3. The driver presses the “cancel matching” button in the system. 4. The system removes the offer. 5. The driver enters new start and destination addresses. 6. The driver offers this route to the system. 7. The system computes and tries to match a passenger. 8. The driver waits for passenger requests.
Key Scenarios	1. The driver enters the start and destination addresses. 2. The driver offers the route to the system. 3. The system computes and tries to match a passenger.
Post-conditions	1. The system has one more route in the system. 2. The driver gets a match. 3. A passenger gets a ride.
Special requirements	1. The driver should enter the start and destination addresses to offer the ride to the system some time before his plan to leave.
Open issues	1. What if the driver never gets a match?

Figure 6-4 [2]

6.3 Use Case 3

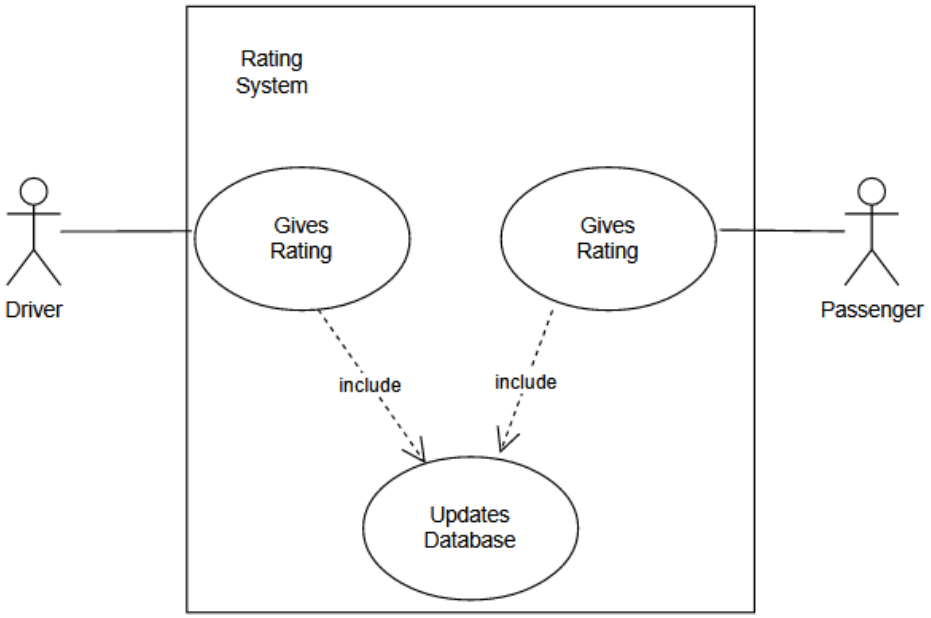

Figure 6-5 Users rates each other

Use Case (UC) Section	**Comment/Description**
Brief Description	This use case describes the details of users rating each other after they carpooled.
Actor brief description	This use case involves driver and passenger that had carpooled together.
Precondition	One driver and at least one passenger must have carpooled together.
Basic flow of events	1. The car arrives at the destination. 2. The passenger opens the application and pays for the ride. 3. The passenger then switches to the rating page in the application. 4. The passenger rates the driver and the car ride. 5. The driver opens the application and gets to the rating area. 6. The driver rates the passenger(s). 7. The rating is updated in the users' profile
Alternative flows	If there's more passengers 1. Passenger A opens the application and pays for the ride. 2. The passenger then switches to the rating page in the application 3. Passenger B opens the application and pays for the ride. 4. The passenger then switches to the rating page in the application ……. 5. The driver opens the application and gets to the rating area. 6. The driver rates the passengers.

	Does Not rate 1. Passenger A opens the application and pays for the ride. 2. The passenger closes the application directly without rating.
Key Scenarios	The driver 1. Rates the passenger(s) The passenger 1. Rates the driver The profile is updated
Post-conditions	1. The destination has arrived. 2. The payment is paid. 3. The rating is completed. 4. The profiles are updated.
Special requirements	1. The user should give each other honest ratings.
Open issues	1. What if a user does something against the law during the carpool? What should the users do? Is there a section in the rating to address this kind of issue? 2. How much tax are the users paying for the carpool?

Figure 6-6 [2]

6.4 Use Case 4

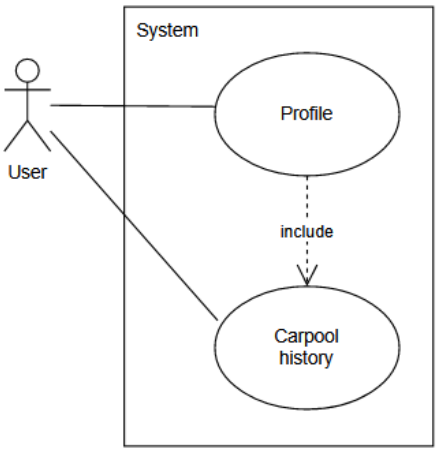

Figure 6-7 View carpool history

Use Case (UC) Section	**Comment/Description**
Brief Description	This use case describes the details of users checking their carpool history.

Actor brief description	This use case involves all kinds of users.
Precondition	The user must have at least one entry in the carpooling history.
Basic flow of events	1. The user opens the application and enters as any role. 2. The user opens the profile page and selects the carpool history area. 3. The system displays at least one entry of carpool history and total mileage (and possible discount based on mileage if the user qualifies). 4. The user presses one of the entries for detailed information. 5. The system displays the details of that carpool, with start and destination addresses, time, cost, and other involved users.
Alternative flows	None
Key Scenarios	1. The user presses the carpool history area in the profile page.
Post-conditions	1. The carpool history is shown. 2. The user is aware of their total mileage.
Special requirements	1. The user must have used the carpool system before.
Open issues	1. Is there any specification in this section for deep users other than the total mileage?

Figure 6-8 [2]

6.5 Use Case 5

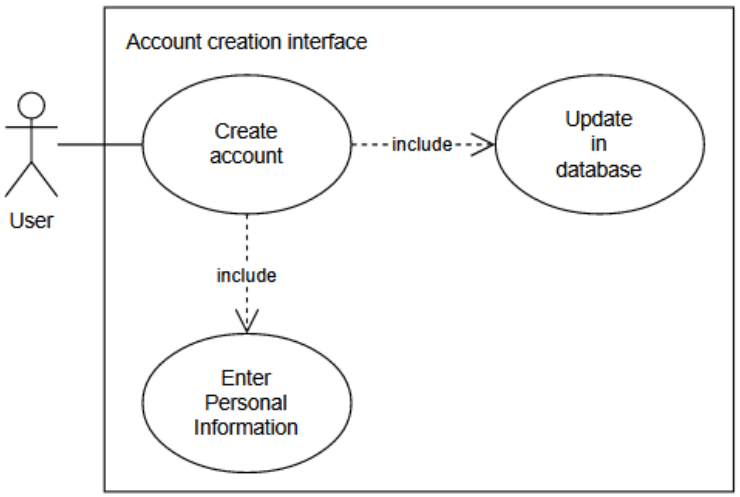

Figure 6-9 Create account and browse through the app

Use Case (UC)	**Comment/Description**

Section	
Brief Description	This use case describes the details of a user using the application for the first time, creating an account, and later browsing through the functionalities.
Actor brief description	Any user.
Precondition	1. The user should have already downloaded the application. 2. The user has all the required information for account creation.
Basic flow of events	1. The user opens the account page in the application. 2. The user creates an account by entering relative information including Username, password, email, phone number, and Bankid. 3. The user verifies with Bankid. 4. The account is create and the user logins with it 5. The user gets to the home page and starts exploring the application.
Alternative flows	When the user enters invalid information: 1. The system shows an error message. 2. The user enters valid information. 3. The account is successfully created.
Key Scenarios	1. The user enters registration information. 2. The user verifies themselves with Bankid.
Post-conditions	1. The user has been verified. 2. The account has been created.
Special requirements	1. The user must enter real information about themselves 2. The user must enter information in a relatively correct format.
Open issues	1. What happens if the user has changed their phone number and they enter their old phone number? 2. What if the user verifies themselves with another person's Bankid?

Figure 6- 10 [2]

6.6 Use Case 6

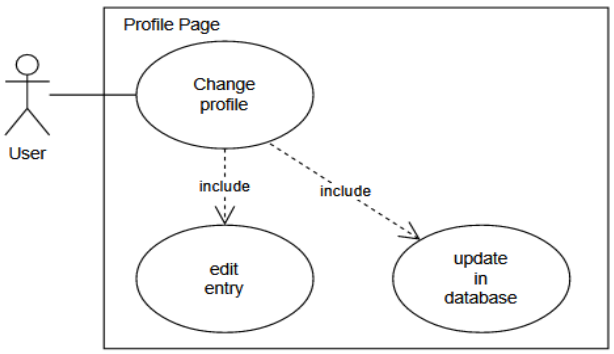

Figure 6-11 Update profile

Use Case (UC) Section	Comment/Description
Brief Description	This use case describes the details of a user changing their profile.
Actor brief description	Any user.
Precondition	The user should be registered and logged in.
Basic flow of events	1. The user selects the profile page in the application. 2. The user sees the user information entry and selects it. 3. The user changes the information that they want to change, options are: Username, password, email, phone number and saves it. 4. The system updates the profile page.
Alternative flows	If the user changes their phone number: 1. In other user's carpooling history that involves this user will also change.
Key Scenarios	The user: 1. Edits information in the profile page and saves it. The system: 1. Updates the information.
Post-conditions	1. The information in the profile is edited. 2. The system is updated.
Special requirements	1. The user must replace their profile information with real ones.
Open issues	1. What if the user tries to change the Bankid to someone else's?

Figure 6-12 [2]

6.7 Use Case 7

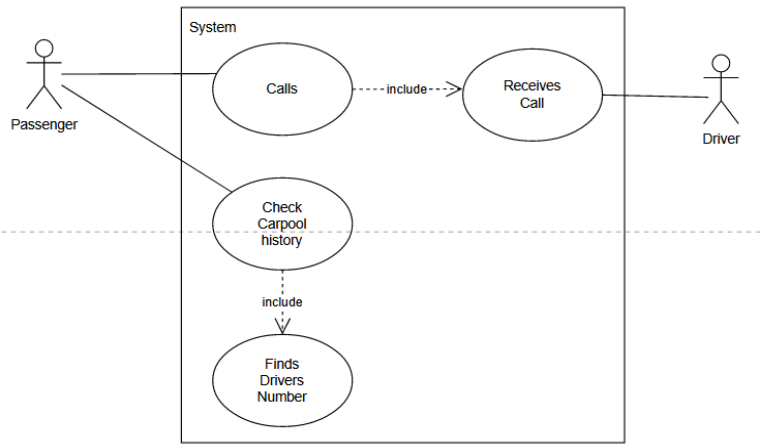

Figure 6-13 A passenger calls the driver(or other passenger)

Use Case (UC) Section	**Comment/Description**
Brief Description	This use case describes the details of a passenger calls the driver(or other passenger)
Actor brief description	A passenger. A driver. Possibly other passengers.
Precondition	1. The passenger and driver must have been assigned to the same carpool. 2. The passenger wants to call the driver for a particular reason.
Basic flow of events	1. The car arrives at the passenger's destination and the passenger pays for the ride. 2. The passenger gets off. 3. The car continues the ride to the driver's or other passenger(if any) destination. 4. The passenger realizes that he has forgotten his bag in the car. 5. The passenger calls the driver after finding the driver's phone number in the carpool history. . 6. The driver picks up the call and discusses where to meet up with the passenger.
Alternative flows	If the driver answer the call: 1. The passenger finds other passenger's phone number in the carpool history 2. The passenger calls that other passenger. 3. They discuss possible solutions.
Key Scenarios	The passenger: 1. Find the phone number in the carpool system and call it.
Post-conditions	1. The destination has arrived. 2. The forgotten item is returned.
Special requirements	1. The driver can only pick up the phone under safe circumstances.
Open issues	1. What if the driver and the other passengers knew about the forgotten item and purposely refused to pick up the phone?

Figure 6-14 [2]

7. References

[1] N.Abbas (2023). 2dv608 lecture 1: Requirements Engineering [PowerPoint slides].

[2] N.Abbas (2023). 2dv608 lecture 2: Requirements Modeling [PowerPoint slides].

[3] N.Abbas (2023). 2dv608 lecture 3: Requirements Validation and management [PowerPoint slides].

Quality Attributes Scenarios (Task 2)

Modifiability Scenario

A user wants to change their information in the profile page. After the user edits the information in the profile. The system is updated with new information. The user is able to see the profile page in the system has been updated with the new updated information.

QAS element	Description
Stimulus	Wishes to change the profile information
Source of Stimulus	User(doesn't matter what the role is)
Environment	When the system is released to public
Artifact	system
Response	The information is updated in the profile page
Response Measure	Within a few seconds

Figure 1 [2]

Availability Scenario

A potential user wants to create an account in the system. The user creates an account with personal information including username, password, phone number, email address and personal address.

QAS element	Description
Stimulus	Creates account for the first time without entering Bankid
Source of Stimulus	New User
Environment	After the system application is released
Artifact	Login and create account interface
Response	The system returns a message saying that the process can not continue.
Response Measure	The response should be given almost instantly.

Figure 2 [2]

This scenario indicates the availability of the application is limited to users who don't have Bankid.

Testability Scenario

A developer or the maintenance team wants to test route matching functionality within the system. They use the web version of the application to test the processes and results. The tester will go through the process as a normal user would.

QAS element	Description
Stimulus	Tries to enter the start and destination address to match a route.
Source of Stimulus	Developer or Maintenance team
Environment	Throughout the development period. Or if the system has gotten negative feedback from the users that relates to this functionality.
Artifact	Route matching interface in the system.
Response	The system will match routes even if the matching percentage is low.
Response Measure	The matching process take no longer than 5 seconds

Figure 3 [2]

Performance Scenario

This scenario focuses on the performance of the system application when the user has grown to a large scale. The scenario tests the system's performance when a passenger searches for a ride and there's a large number of users offering rides, searching rides, paying, rating, changing profile at the same time.

QAS element	Description
Stimulus	Searches for a ride
Source of Stimulus	Passenger
Environment	After the system has been released for a while to the point there's a large user base
Artifact	Route matching interface in the system.
Response	The system presents the matched route
Response Measure	The system takes up to 10 seconds

Figure 4 [2]

This scenario indicates that the system's performance will be affected by the amount of users logged in the system.

Security Scenario

Any user tries to login into the system with another user's username that they remember from a carpooling history. The user enters the correct username and a false password.

QAS element	Description
Stimulus	Tries to login with someone else's username and incorrect password
Source of Stimulus	User
Environment	When the system is at runtime
Artifact	Login interface of the system
Response	The system responds with an error message saying that the password is incorrect
Response Measure	The system gives the response within one second

Figure 5 [2]

The scenario proves that the system is able to protect the user accounts if there's a lowing level hacking.

Usability scenario

A user logs into the system successfully, the user adapts and learns about the system as they switch pages.

QAS element	Description
Stimulus	The user wants to explore the functionalities in the system
Source of Stimulus	User
Environment	When the system is at runtime
Artifact	All accessible interfaces of the system
Response	The system response with expected results
Response Measure	The systems response within half a second

Figure 6 [2]

Appendix – Time Report

Date	Member	Activity	Time(hours)
Jan 31	Erik, Zejian	Read instructions and Lecture slides	3
Feb 1	Erik, Zejian	Form ideas and questions	4
Feb 2	Erik, Zejian	Discuss known and unknown	2.5
Feb 3	Erik	Starts with requirements QA with Nadeem	6
Feb 3	Zejian	Discuss requirements and make most of them certain	6
Feb 4	Erik	Starts with the rest of section 2	8
Feb 4	Erik, Zejian	Modified, add ,and delete some Requirements	7
Feb 5	Erik	Finishes section 2 and starts business rules	4
Feb 5	Zejian	Finished most of the section3 of system interface	4

Feb 6	Erik	Finish business rules and start system constraints. Ask Nadeem more questions	9
Feb 6	Zejian	Modified section 3 of system interface and starts section 2 of Test cases	8
Feb 7	Erik	Finishes system constraints and start use cases	7
Feb 7	Zejian	Finished section 2 of Test cases	8
Feb 8	Erik	Finishes use case and starts task 2 QAS and finished task 2	7
Feb 8	Zejian	Modified business rules and introduction	8
Feb 9	Erik, Zejian	Format the document and fix small details	5
Feb 10	Erik, Zejian	Modification	3