Group Assignment Submission Sheet

This sheet needs to be submitted to your tutor during workshop 7. Its purpose is to confirm that each member of your group has contributed to the document submitted and to verify the current membership of your group. Please identify the submitting member by putting a "yes" in his/her row.

Note that each student must also submit a **Group Evaluation Form** (available on FLO) which identifies your individual work on the assignment and enables you to evaluate the contribution of your other team members' efforts.

Assignment: Air-Carpooler

Group Name: Group 8

Date: 22/9/2019

Group Membership:

FAN	Name	Submitting Member?	Signature
true0030	Rhys Trueman		Rhys Trueman
roge0309	William Rogers		William Rogers
manu0083	Aishwarya Manumari	✓	Aishwarya Manumari

Submission Checklist:

Please confirm that the following have been completed before submitting your work.

[y] We have a copy of our submission.
[y] The submission document is single-space, 12-point Font.
[y] Pages are numbered.
[y] This sheet is completed and inserted as the first page in the assignment folder.

Software Requirements Specification

for

Air-Carpooler

Prepared by Aishwarya Manumari, Rhys Trueman and William Rogers

Flinders University

Software Engineering Principals and Practice

Semester 2 2019

Table of Contents

1. Introduction

1.1 Purpose – Rhys Trueman

The purpose of the Air-Carpooler app is to provide a ride sharing interface for people associated with Flinders University. The product will allow a user to offer a ride to or from a University Campus. User's will be able to search for available ride offers and will be required to pay the driver after a successful trip has been completed, either through AUD or in-app currency, both of which are handled by the app. User's will be able to rate one another after a completed trip, message one another to discuss details and negotiate fees, and to offer a special trip to somewhere other than a campus.

User's will be required to create an account for the app if they intend to use it, which will require University credentials in order to succeed.

The app will rely on external API's to handle location, navigation and payment.

1.2 Document Conventions – Aish Manumari

Main Section Titles

- Font : choice in Heading options (1) or default to Times
- Size : 18
- Style : Bold / capitalized / Underlined

Subsection Titles

- Font : Sub-Heading options (2) or default to Times
- Size : 16
- Style : Bold / capitalized/ colour/ Italics

Other Explanations

- Font : Subtitle / paragraph options or default to Arial or Times
- Size : 12
- Style : Normal text

Diagram & Analysis Models

- Written Language: UML (Unified Modeling Language)

1.3 Intended Audience and Reading Suggestions – Rhys Trueman

The intended audience of the following document is the developers of the application and the Flinders University person(s) responsible for monitoring development goals and progress. Section 1 features an introduction to the project, Section 2 features a broad overview of the app including functional and non-functional requirements. Sections 1 and 2 are recommended for

all readers, while the following sections detail lower level functions of the software and are recommended for the development team.

1.4 Project Scope – Aish Manumari

Air-Carpooler is an MVC (Model- View-Controller) web based application system where the structure of the design is divided into three logical components. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction. (Sommerville, 2011) As our project is a web based protocol the use of MVC will enable clear communication environment between users (providers and seekers). Registered users can access the system (account), view their profiles with FAN and password. While passengers add their routes from the application drivers can communicate with passengers via messaging system within the application to accept the ride request. After the end of the transportation seeker can rate the provider via rating/feedback system.

With a main purpose to serve Flinders staff and students who wish to carpool from various locations and travel between certain campuses this Air-Carpooler system can bring many advantages to its users. For instance, drivers and passengers spend less money on fuel and other commuting costs, thus is cost effective. Productivity between users is increased as less time spent in traffic. And due to the app only committed to flinders staff and students, both driver and passengers abide professionalism and security from the ride.

User details will be managed by a DBMS which will help to store and retrieve data. The HTTP protocol will enable communication between users over the network and the GUI will act as an interaction medium between users.

1.5 References

1. Lehr, S. 2018, *The Benefits Of Using Database Management Systems -RingLead. RingLead, viewed 28 August 2019,* <*https://www.ringlead.com/blog/the-benefits-of-using-database-management-systems/*> (Aish)
2. OAIC. 2019, *The Privacy Act*., viewed on 17 September 2019, <https://www.oaic.gov.au/privacy/the-privacy-act/> (Rhys)

3. TextBook: Sommerville, I. 2011, *Software Engineering*. 9th ed. Boston: Sommerville Pearson, p.155., viewed on 21 September 2019 <software-engineering-9th-ed-intro-txt-i-sommerville-pearson_-2011-bbs.pdf > (Aish)

2. Overall Description

2.1 Product Perspective – Aish Manumari

Air-Carpooler is an independent and self-contained web based application. Its purpose is to connect Flinders staff and students who wish to carpool from various locations such as Flinders University campus or other destinations.

2.2 Product Features – Aish Manumari

The purpose of provided functionalities is to develop a reliable and well-designed Air-Carpooler application which can benefit users (Flinders staff and students) who wish to carpool from various locations. Therefore the application must specifically oblige the following functional requirements.

1. **User Registration & Verification**

   Eligible users (i.e only Flinders staff/student) need to sign up to use the application. Users must provide their details such as first name, last name, Flinders FAN and preferred contact (phone or chat) to be verified and registered into the system and to oblige security concerns.

2. **User Login**

   Users sign in and sign out of their account using their Flinders registration. This helps to verify user's affiliation with Flinders University at registration before s/he commits to a ride.

3. **Add Ride**

   Users add transportation by specifying select once-off or longer term travel needs/offers and enter details such as date and time of travel; and, for offers, number of seats available and for seekers, number of travellers.

4. **Pick up/Drop off points**

   Allows users to pick their start and end destinations (i.e Bedford Park or Tonsley) with assumed standard carpool pick-up points.

5. **Online browsing**

System provides online browsing of providers/seekers within a user selected distance (e.g., 0.5km or 1 km) to the departure location.

6. Online Chat options

The users communicate and support any negotiation between carpoolers using online chat or phone call when a provider is selected (by default the seeker is responsible for initiating contact).

7. Special offers

Allow users to offer and search for special offers such as a ride to the Airport or interstate travel.

8. Sign up for Special offers

Allow users to sign up for special promotion email service where weekly emails are distributed with travel options and special offers. An unsubscribe option is also available.

9. Check Payment

Before the user makes a payment check for which incentive management system the user has indicated. If for gold coin payment or merit-point based payment when the ride is complete where merit points for carpool providers is awarded based on number of people they have helped & distance travelled.

10. Make Payment

After the payment check, user makes a contribution to cost of parking-permit for long-term carpool arrangement.

11. Negotiated Payment

Other negotiated arrangements for special trips user makes a negotiated payment.

12. Rate service & experience

Allow car-poolers to write feedback on aspects of the service and their experience. Thus, other users can review the user rating and ride comments to decide which transportation is better for them.

13. Rate Air-Carpooler users

Allow application administrators to generate various reports, such as weekly/monthly Air-Carpooler users, location distribution, peak usage, merit-point awarded, etc.

14. Delete Ride

A user may delete his/her ride. After the cancellation of the ride, the driver and other passengers in that ride will be informed by the system.

15. **Request Ride**

    The user may proceed with the transportation by sending a transportation request to the driver.

16. **Provider/Seeker Route Recommendation**

    The user may search for their suitable route transportation and specify the date and time of travel in which the driver of the transportation can comply.

2.3 User Classes and Characteristics – Rhys Trueman

Users shall be Flinders University Staff and Students, and possess a FAN and password to verify this. Users are expected to be able to have reasonable computer skills, to be able to navigate a GUI. It is expected, and necessary, that the majority of users will be passengers, with the minority of user's being driver's. It is expected that both classes will use the app with the same frequency, twice every week day.

2.4 Operating Environment – Will Rogers

The software will operate in the environment of either a phone or personal computer. Each of these types of devices can run two possible common operating systems.

For a phone device, iOS 12.4 (only available for iPhone 5s and later) or Android OS 10. For the personal computer, macOS 10.14.6 or Microsoft Windows 10.0.18362.267, also known as '1903'.

Devices would ideally have a GPS to find the user's location, with Google Maps application being able to coexist with the Carpooling application.

2.5 Design and Implementation Constraints – Rhys Trueman

The app shall comply with federal and state law, in addition to Flinders University Policies. The app will require driver's license verification before the option of offering a ride is made available to a user. Furthermore, the app will not require input from the driver while the vehicle is in motion.

The app will need to be responsive to user's, be reacting appropriately within 2 seconds to any input. The app will rely on google maps for navigation.

The app will require a FAN and password login to be used, in order to verify a user is associated with the University. The app shall be secure and have no possible way of accessing user's FAN numbers, passwords or any other information required of it's users.

2.6 User Documentation – Rhys Trueman

The app will be intuitive to use and navigate with a simple GUI, and should not require any manual. Upon first opening the app for the first time the app should walk the user through the basics on how to operate the app, including how to request a ride and how to offer a ride. The app will also have a short help section in the menu with the ability to refresh the tutorial mentioned above if the user wishes to be walked through the interface of the app once again.

2.7 Assumptions and Dependencies – Will Rogers

Assumptions are that the application will use a third-party application in Google Maps, the user interface will change throughout the development with different interfaces for phone and PC, and functions may be changed, added, or removed throughout development. It is dependent on data from the Flinders database on students and staff to verify accounts.

3. Specific System Requirements

3.1 User (Passenger and Driver) – All

3.1.1 Functional Requirement 1

Use case: Register

Actors: User

Type: primary, essential

Description: User registers into the system by providing their first name, Flinders FAN and contact details and signs up for an account to access the application.

3.1.2 Functional Requirement 2

Use case: Verify

Actors: User

Type: primary, essential

Description: If the user is eligible (i.e Flinders staff member or student) their details are verified with Flinders University at the registration before confirmation of registration.

3.1.3 Functional Requirement 3

Use case: Sign In

Actors: User

Type: primary, essential

Description: Once the registered user has an account, they can sign into the application which allows for the next set of functional requirements

3.1.4 Functional Requirement 4

Use case: Sign Out

Actors: User

Type: primary, essential

Description: Allows the registered user to sign out of their account when they are finished with the application.

3.1.5 Functional Requirement 5

Use case: Single Fare

Actors: User (provider/seeker)

Type: primary, essential

Description: The user has an option to choose their travel for a single ride (once-off) and therefore continues adding their route.

3.1.6 Functional Requirement 6

Use case: Long-term Fare

Actors: User (provider/seeker)

Type: primary, essential

Description: If the user has chosen longer term travel needs/offer, they must enter details of date and time of travel and for offers, number of seats available and for seekers, number of travellers.

3.1.7 Functional Requirement 7

Use case: Add Route

Actors: User (driver)

Type: primary, essential

Description: The user picks the location of where they want to go with departure or arrival time, and then the application generates a route for getting there. Other users can see this route.

3.1.8 Functional Requirement 8

Use Case: Delete Route

Actors: User (driver)

Type: primary, essential

Description: Deletes any added routes

3.1.9 Functional Requirement 9

Use Case: Request Ride

Actors: User (passenger)

Type: Primary, essential

Description: This use case begins when a user requests a ride through the app. The app prompts the user for a pickup time, destination and location. Once entered the request is sent to the server.

3.1.10 Functional Requirement 10

Use Case: Online browsing

Actors: User (passenger)

Type: primary, essential

Description: Passenger makes a search request to find a desired route which a driver has added to the departure location.

3.1.11 Functional Requirement 11

Use Case: Send Message

Actors: User

Type: primary, essential

Description: A passenger or driver sends a message to the driver or passenger through the application.

3.1.12 Functional Requirement 12

Use Case: Reply Message

Actors: User

Type: primary, essential

Description: Reply to the message sent by the driver or passenger

3.1.13 Functional Requirement 13

Use Case: Block User

Actors: User

Type: optional

Description: Block user's profile from any form of communication through the application.

3.1.14 Functional Requirement 14

Use Case: Rate User

Actors: User

Type: Primary, Essential

Description: Use case begins with a user electing to rate a fellow user they've recently interacted with. They select a score from 1 to 5 that then contributes to the other users average score.

3.1.15 Functional Requirement 15

Use Case: Call User

Actors: Users

Type: Primary, Essential

Description: Use case begins with a User A pressing the call button in a chat with another User B. User B then has a notification tell them such and has the option to confirm or deny the call.

3.1.16 Functional Requirement 16

Use Case: Offer Special

Actors: User (passenger)

Type: Primary, Essential

Description: Use case begins when a user decides to make a special offer (a ride that doesn't begin/end at a University Campus) such as a trip to the airport or a trip interstate. App prompts user to give the destination and time of departure.

3.1.17 Functional Requirement 17

Use Case: Search Special Offer

Actors: Users (passenger)

Type: Primary, Essential

Description: Use case begins when a User A decides to browse special offers posted by users. User A can request to be a passenger on the trip, User B (who made the special offer) is notified and prompted for a confirmation or denial.

3.1.18 Functional Requirement 18

Use Case: Sign up for promotional email

Actors: User (passenger)

Type: Primary, Essential

Description: Use case begins with a User selecting the sign up button which adds their email to a list of email addresses to send the weekly email to.

3.1.19 Functional Requirement 19

Use Case: Unsubscribe from promotional email

Actors: User (passenger)

Type: Primary, Essential

Description: Use case begins with a user electing to opt out of promotional email, removing them from the list of email addresses that the email is sent to.

3.1.20 Functional Requirement 20

Use Case: Change Language

Actors: User

Type: secondary

Description: Allows user to choose a language for them to fully understand the application.

3.1.21 Functional Requirement 21

Use Case: Check Payment

Actors: User (passenger)

Type: primary, essential

Description: Before the user makes a payment check for Gold Coin payment or merit-point based payment for once-off ride. Merit points for carpool providers is awarded based on number of people they have helped and distance travelled.

3.1.22 Functional Requirement 22

Use Case: Make Payment

Actors: User (passenger)

Type: primary, essential

Description: Once the payment check is approved user makes payment which will contribute to the cost of long-term parking for the driver.

3.1.23 Functional Requirement 23

Use Case: Negotiated Payment

Actors: User (passenger)

Type: primary, essential

Description: Users with other negotiated arrangements i.e. special trips make a negotiated payment.

3.1.24 Functional Requirement 24

Use Case: Verify Driver's License

Actors: User (driver)

Type: primary, essential

Description: Driver submits their driver's license number to be verified. This use case, if and only if it is successful, enables the user to offer rides.

3.2 Application Admin – Will Rogers

3.2.1 Functional Requirement 1

Use Case: Generate Report

Actors: Admin

Type: primary, essential

Description: Allows application admin to generate various reports on weekly or monthly Air-Carpooler users, given a list which an admin picks.

4. External Interface Requirements

4.1 User Interfaces – Rhys Trueman

Upon first opening the app the user will be required to login in, underneath the field boxes will be a create an account option. If the user logs in successfully they will be taken to the main menu, a mockup of which is provided in Appendix B, Figure 3.

From this menu a user can navigate a selection of other menus including to “request a ride”, “offer a ride”, “message a user”, “rate a user”, with all menus resembling the above style. The green help button in the top right corner would be available on every screen, and would show a short animation displaying the purpose and functions of all buttons on the screen.

4.2 Hardware Interfaces – Aish Manumari

Since Air-Carpooler is a web-based system, it is compatible with all browsers and can be run on any operating system and processor. (i.e android, apple) Hence no special hardware interface is required. This web application will work using the standard GUI and the communication between users via app will be facilitated through network protocols.

4.3 Software Interfaces – Aish Manumari

Database Management System will be required for the Air-Carpooler application for the following reasons:

Improved Data Sharing and Security: Having a proper DBMS helps to organise and assess data in a more efficient way. (Lehr, 2018) For example user and route information stored in the database allows carpoolers to share or book a ride quickly and effectively across various locations. While promotes effective data integration of the organization’s operations.

Increase in productivity of the end user: Having a DBMS increases end-user productivity. (Lehr, 2018) Therefore empowers them to make quick and informed decisions that can contribute to the effective system functionality.

The use of MySQL Database management system is also required to manage the data stored in a database for quick processing, proven reliability and ease of use. It is an open source interface for the end-users which helps to organise user details in table and rows in a well formatted form for developers to create and manage the database successfully.

HTTP Protocol for communication over the internet.

Google Maps Server is required for the Carpooler software to provide details about the geographical area while showing convenient routes from Flinders to Tonsley.

4.4 Communication Interfaces – Will Rogers

When signing up, the application will send an automatic verification email to the user, which would go to the student's Flinders email. The chatting between passengers and drivers, a messaging function within the application is supported using HTTP protocol. This chat can be accessed with both PC and mobile phone devices.

5. Other Non-functional Requirements

5.1 Performance Requirements – Will Rogers

Capacity

The application has to be able to store data of multiple users, when considering the number of students and staff, it should be able to accommodate 20,000 users with non-continuing student and staff accounts being deleted.

Response Time

It would be expected that a maximum of 3,000 users would use the application at the one time and it should be able to respond to all of them.

Device Sensitivity

A user can be expected to typically have download speeds of around 44Mbps which will therefore grant a fast connection to the app unless the connection to the Internet is unreliable on the user's end.

5.2 Safety Requirements – Will Rogers

Passengers will be able to give a safety rating to drivers by using the feedback systems, to help prevent users from potential harm through dangerous driving by holding drivers accountable. Sexual harassment or any verbal abuse can contribute to this rating, as well that drivers can also give a safety rating to passenger who might be 'ill-behaved' or distracting. Low ratings can give the passengers to refuse a specific driver or the driver to refuse a specific passenger.

5.3 Security Requirements – Rhys Trueman

User's will enter University Credentials in order to create an account for use with the app, FAN credentials will not saved by the app server but sent to a Flinders University server for validation. Any saved information on the app database will be encrypted. When the user enters their login credentials to the app the username/email will be displayed as it is typed but the password box will display a black dot for every character typed, for example "●●●●" if you were to type a 4 letter password.

5.4 Software Quality Attributes – Rhys Trueman

Availability

The app will be available from 7am to 10pm on weekdays during the semester, and 8am-6pm during study break periods, with no more than 10 seconds of downtime during any week day.

Accuracy

The app should accurately (to within 5 minutes) calculate an estimated travel time, barring road works and emergencies.

Maintainability

The app should be easily maintainable for long term use and such that it can be changed when/if the Universities needs change. This will be ensured by a thorough documentation of the source code and the software components will be developed to be as loosely coupled as possible.

Reliability

The app should be considered better or as usable as competing apps, trials with focus groups will be done to ensure this.

6. Other Requirements – Rhys Trueman

The app will comply with The Privacy Act 1988 and will disclose to user what type of data and why the data is being collected.

The sub modules will be made generic where possible such that they could be adapted for any future software the University produces.

The database used by the app will be encrypted, to protect user's privacy, and is required to be reliable.

7. Reflection – Aish

This Software Requirement Specification appropriately addresses all the functional and non-functional requirements for the proposed software system. It is written in language that can be easily followed and understood by the users, stakeholders and developers to refer back when developing the system. This document is intended to provide a detailed overview of the web-based application 'Air-Carpooler' to its users, developers and stakeholders. It also includes use-case diagrams and data flow models to provide a pictorial view and effectively convey the mandatory requirements of the system architecture.

The Specifications provided in this document thoroughly acknowledge the Software Requirement Specification principles and describes each component of the Air- Carpooler application to a high level of accuracy. The system is designed to be performable, scalable and available to all its users, developers and stakeholders. It also ensures the privacy of user data is secured while allowing the application to be maintainable is an easy manner. While this application is not ready to be deployed for public use however should enable developers to implement the design specification into a working prototype. However, a constraint with this specification is that information such as number of users, cost figures in construction and development of the application is not given to allow flexibility for developers and stakeholders in how they want to proceed when designing the system on a software level. Hence additional elicitation and analysis reports when developing the application can be useful to allow tangibility between users and developers.

Appendix A: Glossary – Will & Aish

API	Application Programming Interface, functions that allow creation of applications to access features or data of another service/application
DBMS	Database Management System
FAN	Flinders Authentication Name, used to log onto a range of Flinders University services
GPS	Global Positioning System is capable of receiving signals from satellites to calculate a device's location
GUI	Graphical User Interface
HTTP	Hyper-Text Transfer Protocol
MySQL	Is a system software, open-source relational database system
OS	Operating System
PC	Personal computer
USER	can be driver or the passenger / provider or seeker
MVC	Model-View-Controller, is a software architectural pattern commonly used for developing user interfaces where the related program is divided into three logical components, model, view, and control.

Appendix B: Analysis Models

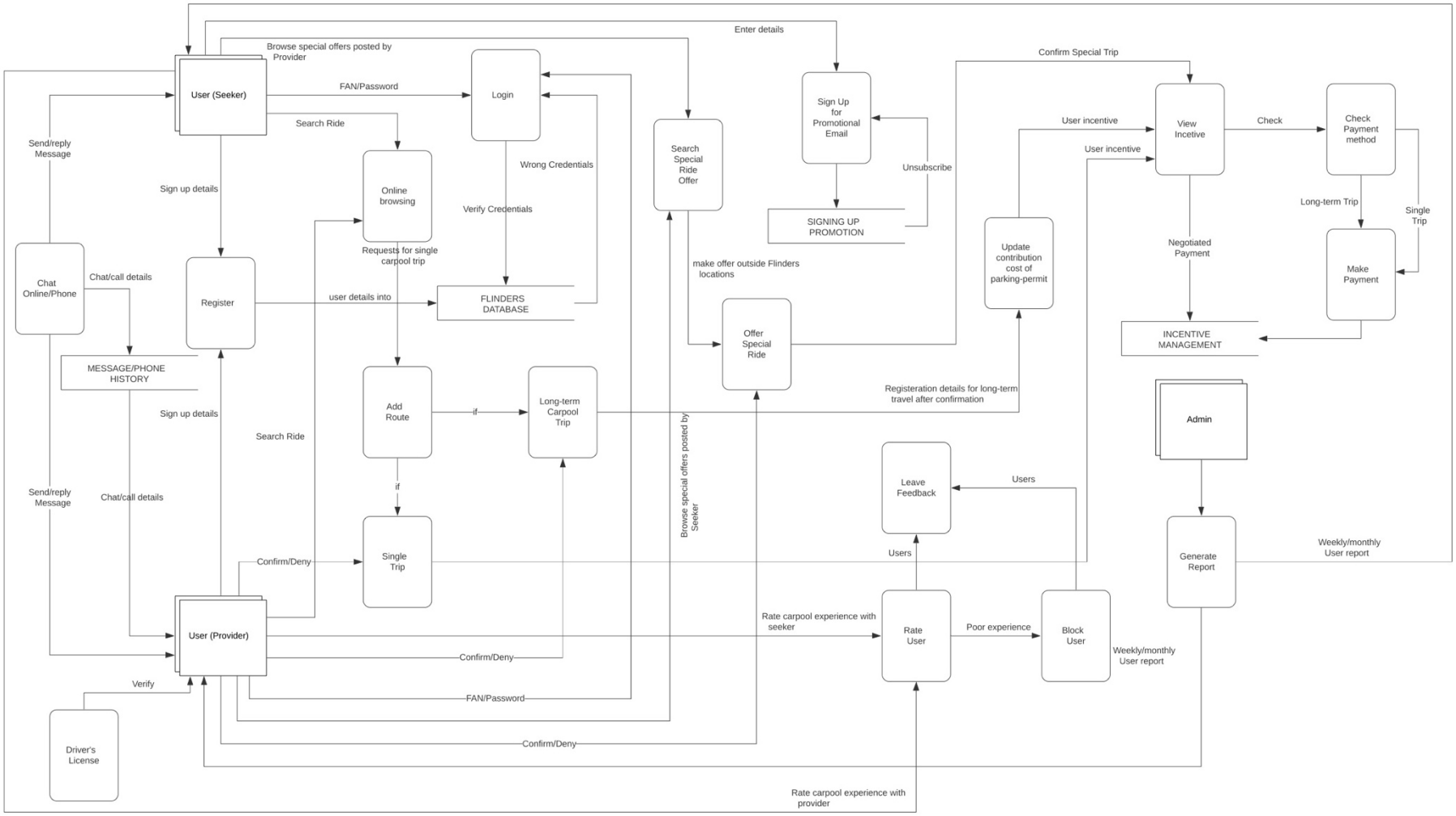

Figure 1: Level 1 Data Flow Diagram for Air-Carpooler, Aishwarya Manumari

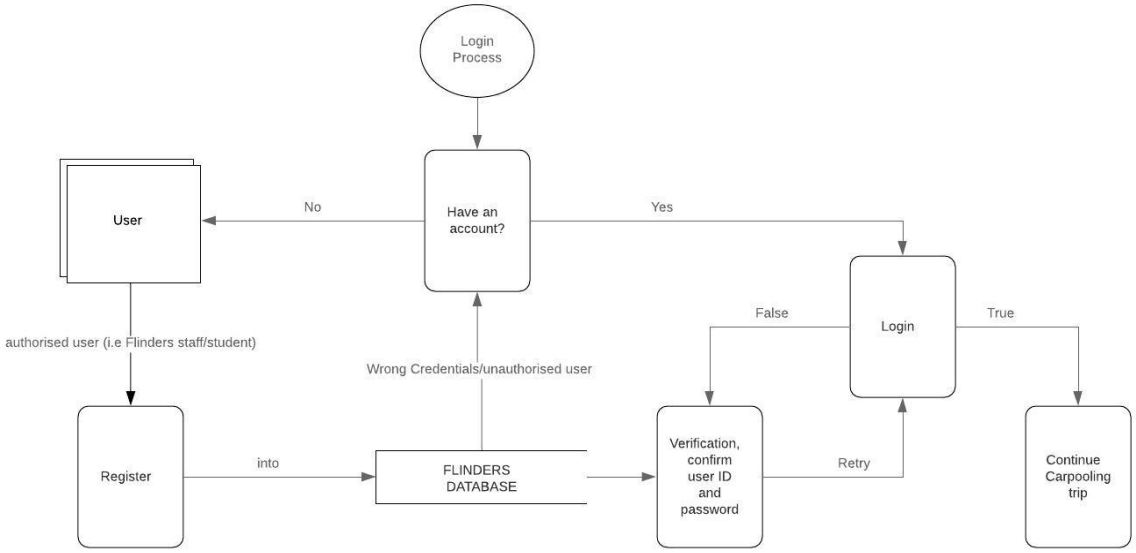

Figure 2: Level 2 Data Flow Diagram for User Login Process, Aishwarya Manumari

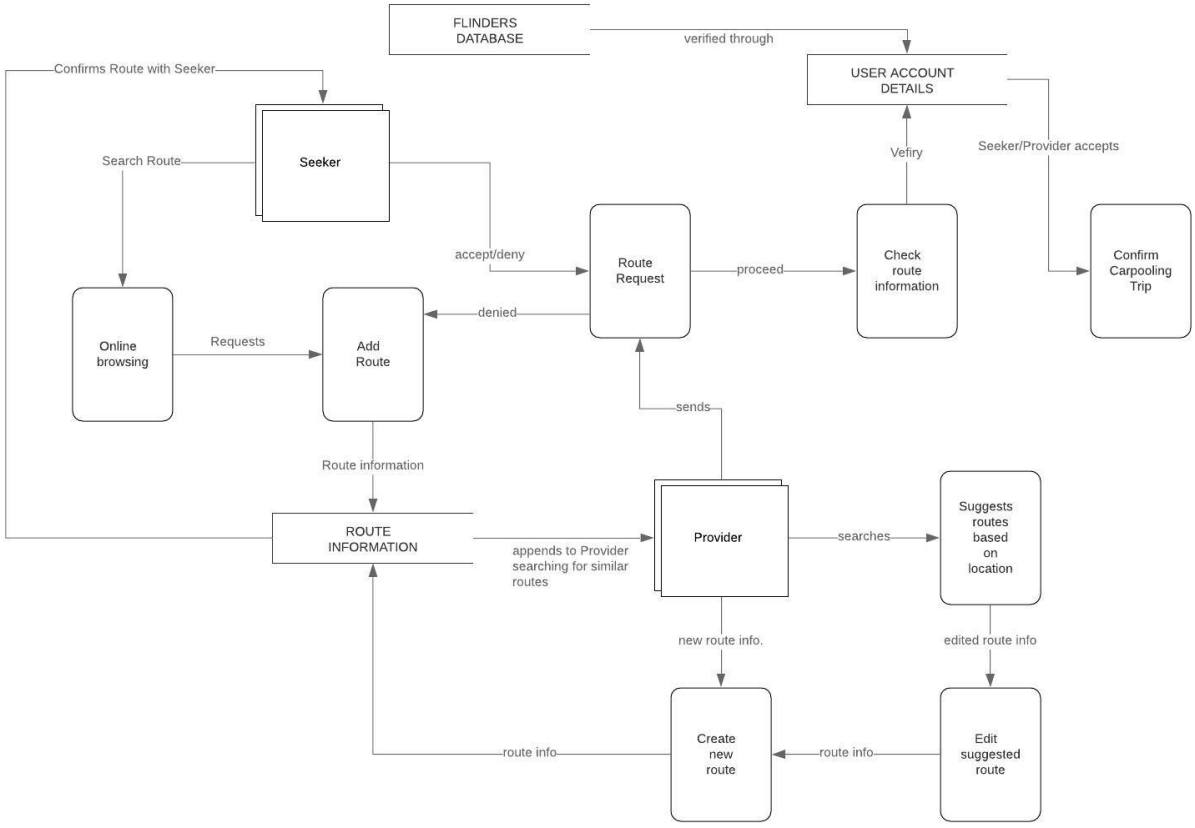

Figure 3: Level 2 Data Flow Diagram for Single Route Trip, Aishwarya Manumari

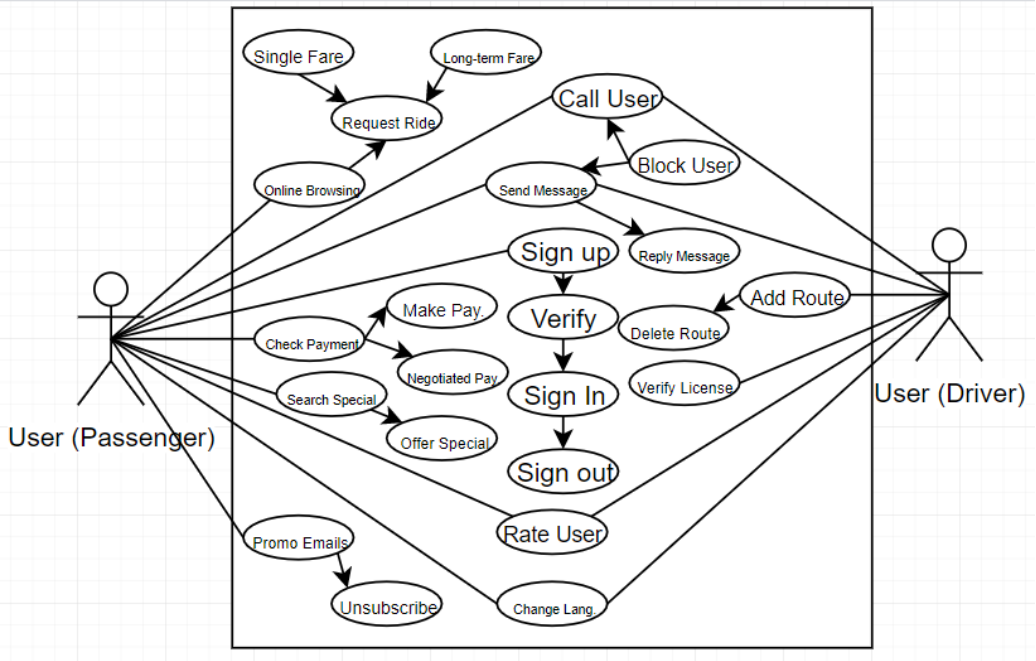

Figure 4: User, Use Case Diagram - Will Rogers

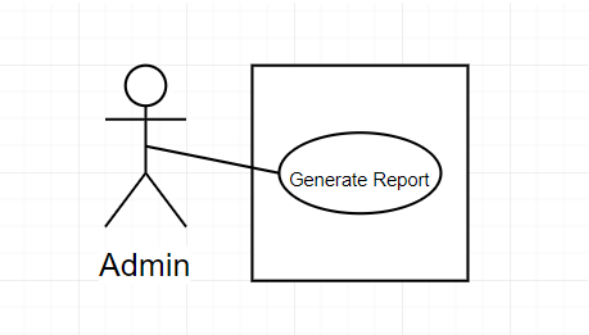

Figure 5: Admin, Use Case Diagram - Will Rogers

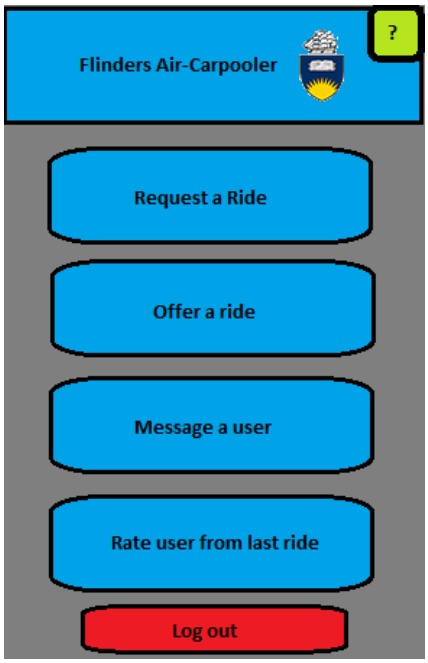

Figure 6 : A mockup for the application's main menu, Rhys Trueman

Appendix C: Elicitation Report

Survey Questions (Closed)

1. What is the preferred payment method?
    - Paypal
    - Credit
    - Cash
    - Other (please specify)
2. Which method of customer feedback is preferable?
    - Rating (1-10)
    - Leave a Comment
    - Other (please specify)
3. How do you support negotiation between carpoolers?
    - Online chat
    - Phone call
    - Other (please specify)
4. Will there be options for picking a specific car/driver?
    - Yes
    - No
5. How do you prove that carpooling is cost-effective compared to a transit route? Check all that apply.
    - Comparison in cost of transit vs the cost of a ridesharing program (ex: cost per hour)
    - Measure of kilometrage cost -actual or projected, for a transit route (riders per hour)
    - We don't attempt to prove its cost - effective, because it is considered part of our mix of mobility services.
    - Other (please specify)
6. What operating system will the server be using?
    - Windows
    - macOS
    - Linux
7. How will you be handling cash methods and payment?

8. What times will the app need to be available?

   __:__ to __:__ Every Day

   __:__ to __:__ Business Days

9. What is the maximum allowed downtime during a given business day?

   __ minutes a day

10. How does the app verify it's users are either students or faculty of Flinders?
    - Staff/Student FAN login
    - Other (please specify)

11. What data is the app recording of its users? previous routes, times, timetables

12. How is the app distributed to users?
    - App store
    - Website
    - Other

13. Do you have any existing software that the app will need to be compatible with?

Interview Questions (Open)

1. What is the aim of the product (a mobile application or a website)?
2. How might the chat feature be properly utilised if car-poolers are driving?
3. How should the application go about if servers fail?
4. How should a passenger repay the car-pooler?
5. What's the format of the report you would like?
6. What is your budget?
7. How is the application marketed for the staff and students of Flinders?
8. What privacy and security are you offering to your customers?
9. What information will you need about the car owner from a security and comfort perspective?
10. How does this application benefit the future and help better your system?
11. How are you willing to charge for the transportation (mobility costs, service fee, fuel cost, commuting charges)?
12. What is your payment method(paypal, cash, credit)?
13. Do participants receive vouchers on their fares/ carpooling services?
14. How have you incorporated feedback from people who use the application?

15. How do you maintain customer and driver's information confidential from the third party (database system)?
16. How are you willing to make your product stand out from competitors?

Appendix D: Team Meetings

Date Time	**Apologies and Absentees**	**1. Agenda Items** **2. Action Items**
14/08/19 12:00PM	n/a	1. Agenda Items a. Allocation of parts for Section 2 2. Actions Items . Section 2 / All / 20/8/2019
21/08/19 12:00PM	n/a	1. Agenda Items a. Think about elicitation, survey and interview questions for Appendix E b. Allocate parts for Sections 4 and 5 c. Section 2 is complete 2. Action Items . Sections 4 and 5 / All / 27/8/2019
28/08/19 12:00PM	n/a	1. Agenda Items a. Get at least 12 survey and interview questions, begin working of Use Cases b. Section 4 still not completed, to be completed 2. Action Items . Use Cases / Rhys + Will a. Certain parts of Section 4 / Aish b. Survey + Interview Questions / All
03/09/19 10:00AM	Will Rogers - sick	1. Agenda Items a. Continue working on Use Cases b. Begin thinking about use case diagram c. Check survey/interview questions 2. Action Items . Introduction to SRS / Rhys + Aish a. Begin use case diagram / Will
11/09/19 12:00PM	n/a	1. Agenda Items a. Finish use case diagram b. Create user interface idea c. Work on data flow diagrams 2. Action Items . Use Case Diagram / Will a. User Interface / Rhys

		b. Data Flow Diagram / Aish
18/09/19 12:00PM	n/a	1. Agenda Items a. Finish the SRS before the holidays b. Consider reflection c. Organise extra meeting tomorrow to further progress 2. Action Items . Data Flow Diagram / Aish a. Glossary and other little bits / Will + Aish
19/09/19 3:00PM	n/a	1. Agenda Items a. Finish reflection/ Aish b. Finish data flow diagrams c. Format the document correctly d. Any little changes that need to be done to be completed during break 2. Action Items . Data Flow Diagram / Aish a. Formatting and Finishing touches / Aish