

# Ride Sharing Website

COMPX374-20B (HAM) Software Engineering Industry Project

## Deliverable III

# Software Test Plan



THE UNIVERSITY OF  
**WAIKATO**  
*Te Whare Wānanga o Waikato*

*8<sup>th</sup> October 2020*

**Members of Group K:**

**Aaron Win**

**Lysa Phan**

**Victor Yao**

**Ryan Good**

# Table of Contents

- 1. Test Plan Identifier.....2**
- 2. Introduction.....2**
- 3. Test Items.....3**
- 4. Features to be Tested**
  - 4.1 Functionality Testing..... 3
  - 4.2 Attributes of Focus Testing ..... 12
    - 4.2.1 Performance Testing ..... 12
    - 4.2.2 Usability Testing ..... 13
- 5. Environmental Needs.....14**
- 6. Responsibilities.....14**
- Appendices ..... 16**

# 1. Test Plan Identifier

**Name of test plan:** RS010

Version No	Version date	Authors & Contact Information	Reference Document
1.0	8th Oct 2020	Aaron Win - aaronjw32@gmail.com Lysa Phan - lysa.grass@gmail.com Victor Yao - vowoov@hotmail.com Ryan Good - ryantgood1016@gmail.com	Software Design Specification document

## 2. Introduction

This test plan aims to document the important information required to effectively define the testing approach of the Ride Share website. This is a system testing as the whole web application is tested, along with the interaction of any external services or utilities. There are four main sections in this test plan, which are Test Items, Features to be Tested, Environmental Needs and Responsibilities. The *Test Items* section introduces all the items that will be tested within the scope of this document. In *Feature to be Tested* section, the tests for each feature are described in detail to identify and report the presence of any gaps or defects. The three testing attributes chosen to test the web application include functionality testing, performance testing, and usability testing. *Functionality Testing* section describes how the test performs and processes each item of the web application to ensure all the requirements stated in Software Requirement Specification report are met. The team has written some tests in the *Performance Testing* section to ensure the web application responds on time and nothing hold back its overall performance. The *Usability Testing* section provides several tests to reveal areas of confusion and difficulty that users may encounter when they attempt to complete a task. The purpose of this section is to improve the overall user experience when they are using the Ride Share website. Any special requirements needed for this web application are listed under the *Environmental Needs* section. The *Responsibilities* section identifies the responsibility of each team member for each aspect of the testing process. All the test listed have an assigned team member to be responsible for.

The intended audience for this document is the developer team, the testing team, the client and the convenors of Software Engineering Industry Project paper. Some parts of this test plan may be shared with the users whose input into the testing process is needed.

# 3. Test Items

## i) Functionality Testing:

- **Index Page version 1.0:** Integrated with the database and *Main Offers* page.
  - Display navigation bar and footer
  - Display image slider
  - Sign in with Facebook
- **User Profile Page version 1.0:** Integrated with the database and *Edit Profile* page.
  - Display user's information
  - Edit Profile button
- **Edit Profile Page version 1.0:** Integrated with the database and *User Profile* page.
  - Edit user's information
  - Update user's information
- **Main Offers Page version 1.0:** Integrated with the database, *Request/Offer Form* page and *Detailed Request/Offer* page.
  - Load Offers button loads Offers
  - Clicking an offer takes you to that offer's page
  - Clicking "Join" on the offer's page to join the ride gives the user the pertinent details of the offeree.
- **Request/Offer Form Page version 1.0:** Integrated with the database, *Main Offers* page and *Detailed Request/Offer* page.
  - Clicking and dragging the cursor across a map.
  - Entering into a search box.
  - Selecting one of the search results.
  - Clicking "Submit" when the form is incomplete.
  - Clicking "Submit" when the form is complete.
- **Detailed Request/Offer Page version 1.0:** Integrated with the database and *Request/Offer Form* page.
  - Load the user's detailed information correctly.
  - Display user's location using Google Map API
  - Changing user's 'party\_number' data in the database when clicking on the "Accept the ride" button.
  - Clicking "Join" on the offer's page to join the ride gives the user the pertinent details of the offeree.

## ii) Performance Testing:

- Logging in with Facebook occurs within a reasonable time.
- Loading the current Ride Offers occurs within a reasonable time.
- 50 users on separate devices all attempt to load all Current Offers at once.

## iii) Usability Testing:

- How long it takes the user to find out how to, then proceed and Login with Facebook.
- How long it takes a user to join a current Ride offer that suits them.
- How long it takes a user to create a new Ride offer.

# 4. Features to be Tested

## 4.1 Functionality Testing

Index Page (Lysa Phan)				
DFD Process 1. Verify Account <b>Inputs:</b> Student Info, Verified Student Info, Verification Result <b>Outputs:</b> Student Info, Verified Student Info				
Item to Test	Test Description	Pass/Fail Criteria	Level of risk	Reason
Display navigation bar and footer	The tester must be able to see the information in the navigation bar and footer at all times. The tester can ensure that each button takes them to the respective web page. <b>Action:</b> 1) View the navigation bar and footer 2) Click on buttons within the navigation bar (Home, About, Login with Facebook) or within the footer section (Privacy, Security, Term).	<b>Pass:</b> The logo and company information in the footer section is displayed. Each navigation button redirects the user to a respective destination.  <b>Fail 1:</b> User cannot see navigation bar or footer or both.  <b>Fail 2:</b> The buttons in the navigation bar and/or footer do not redirect people to the right destination.	Low	The likelihood of this risk is low as the navigation bar and footer are coded to always appear in the web page. The impact is medium because the user will have difficulty to navigate around the web pages if the navigation bar is not appeared. The overall level of risk is low as the risk is extremely hard to occur.

Display image slider	<p>The tester should be able to see the image slider with text description and button in each image as well as and navigate between images.</p> <p><b>Action:</b></p> <ol style="list-style-type: none"> <li>1) View image slider</li> <li>2) Navigate left/right between images</li> <li>3) View text description of each image</li> <li>4) Click on the button displayed on the image.</li> </ol>	<p><b>Pass:</b> The web page should display all images in the image slider consecutively. Each image displays its own text description and button.</p> <p><b>Fail 1:</b> The images in the image slider do not appear.</p> <p><b>Fail 2:</b> The images do not slide automatically after a certain period of time.</p> <p><b>Fail 3:</b> User cannot navigate between the images.</p> <p><b>Fail 4:</b> The text description in each image is not displayed.</p>	Low	<p>The likelihood is low as the images are stored in the hosting space folder, not inserted from another web link. The risk such as an expired link will not occur. The impact is low as this feature does not affect the main function of the website.</p>
Sign in with Facebook	<p>The tester should be requested to log in from Facebook platform.</p> <p><b>Action:</b></p> <ol style="list-style-type: none"> <li>1) Click on “Log in/Sign in with Facebook” button</li> <li>2) Redirected to Facebook login page</li> <li>3) Enter valid Facebook account details and submit.</li> </ol>	<p><b>Pass:</b> User is redirected to Facebook platform. Facebook successfully verifies a user’s account. The user is taken back to the website and their Facebook information such as name, email, and profile picture are stored in the website’s database.</p> <p><b>Fail 1:</b> User is not redirected to Facebook platform.</p> <p><b>Fail 2:</b> User’s Facebook account is not verified.</p> <p><b>Fail 3:</b> User’s information from Facebook is not stored in the database.</p>	High	<p>The likelihood is medium as there could be some problems with the access permission from Facebook. The web page may not be granted the permission to access the user's Facebook account. The impact is high because this website entirely relies on the login function of Facebook platform. The user cannot have further access on the website if the Facebook verification process is failed.</p>

User Profile Page ( <i>Lysa Phan</i> )				
Item to Test	Test Description	Pass/Fail Criteria	Level of risk	Reason
Display user's information	The tester should see all the personal information displayed.	<p><b>Pass:</b> All information of the user from the database is displayed.</p> <p><b>Fail:</b> Cannot load the user's information from the database</p>	High	The likelihood is medium. If there is an issue with the database or the connection to the database, the risk will occur. The impact is high as the user details will be displayed for other users to see and contact when the match happens. If the user information is not displayed, the users cannot contact each other to have further discussions about the ride-sharing.
Edit Profile button	When the "Edit Profile" button is clicked, it should take the tester to another web page where they can edit their profile details.	<p><b>Pass:</b> User is redirected to Edit Profile page.</p> <p><b>Fail:</b> User remains in the profile page or is redirected to the wrong page.</p>	Medium	The impact is medium because the user will not be able to edit their profile details if this button does not work. The overall level of risk is low because the likelihood is extremely low (As this button's function is only redirecting page.)

Edit Profile Page ( <i>Lysa Phan</i> )				
Item to Test	Test Description	Pass/Fail Criteria	Level of risk	Reason
Edit user's information	There are certain fields that the tester can edit and there are some fields where the tester should not be allowed	<p><b>Pass:</b></p> <p>- The user can edit certain fields such as phone number, suburb, city, language, and gender.</p>	High	The impact is high because if the user cannot edit their details, other users may not be able to

	to edit. The name and email fields are imported from Facebook so they cannot be changed here.	<ul style="list-style-type: none"> <li>- The user can change their profile picture.</li> <li>- The user is not allowed to edit the name and email fields.</li> <li>- The user is permitted from changing their rating score and the number of trips they have done.</li> </ul> <p><b>Fail 1:</b> The user cannot edit phone number, suburb, city, language, and gender fields.</p> <p><b>Fail 2:</b> The user cannot change their profile picture.</p> <p><b>Fail 3:</b> The user is allowed to edit the name and email fields.</p> <p><b>Fail 4:</b> The user can change their rating score and the number of trips they have done.</p>		contact them. The likelihood is low because the user can easily edit their personal information in the text fields.
Update Profile button	The changes in personal details should be updated in the database when the tester clicks this button. It should also redirect the tester back to their Profile page.	<p><b>Pass:</b> All changes are saved and updated in the database. The user is redirected to the Profile page.</p> <p><b>Fail 1:</b> User remains in the same page or is redirected to the wrong page.</p> <p><b>Fail 2:</b> The changes are not updated in the database.</p>	High	The likelihood is medium as there can be an issue with the database connection. The impact is high as the user needs to update their contact details to keep them up-to-date so that other users are able to contact them.

### Main Offer Page (*Ryan Good*)

#### DFD Process 2. Displaying Potential Matches

**Inputs:** Filtered Student Info, ID of selected student (who offered the ride)

**Outputs:** Filtered Student Info, ID of selected Student



Item to Test	Test Description	Pass/Fail Criteria	Level of risk	Reason
Load Offers	The correct obtaining and displaying of current active offers is to be tested. Test data will be added to the database to facilitate this test.	<p><b>Pass:</b> Once the “Load Offers” button is pressed, offers made up of the test data (The Filtered Student Info) will be displayed on the page.</p> <p><b>Fail 1:</b> Once the button is clicked, nothing visually happens.</p> <p><b>Fail 2:</b> The active offers are displayed but in an incorrect format</p>	Medium	Once the button is clicked, in almost all cases, something is displayed on the screen, even if the database is empty, but if for some reason, the offers are not displayed, this is detrimental as the offers are the core functionality that facilitates the ride sharing.
Click an offer takes you to that offer’s page	This is to test if you click an active offer, it takes you to a page about that offer with more details. At Least 2 Test Offers will be added to the database to facilitate this test.	<p><b>Pass:</b> Clicking of the active offer takes the user to a page about that offer with more details and a join button.</p> <p><b>Fail 1:</b> Clicking the active offer takes the user nowhere, or to a blank page.</p> <p><b>Fail 2:</b> Clicking the active offer takes the user to a page with more details, but details for the wrong active offer.</p>	Medium	The more detailed offer page should always display, with the correct information, but if it does not, a user might join the wrong offer of a ride. Meaning that the user cannot accomplish their goal.
Click “Join” on the offer’s page to join the ride gives the user the pertinent details of the offeree.	This test will see what happens when a user attempts to Join a ride. A Test Offer will be added to the database to facilitate this.	<p><b>Pass:</b> Clicking the Join button, the user is given the details (ID is given as well to facilitate the match, but as data, not as information for the user) of the offeree so that they can continue and finalise the details of the Ride. The ID of the Selected Student is then sent back to the “Student Database”.</p> <p><b>Fail:</b> The user clicks join and is not given the details they need to proceed with the Ride.</p>	Medium	Medium Risk as the page should not have any trouble giving the user the offeree’s details if the offer page already works correctly, but if this is not the case, Rides cannot be correctly facilitated and as a result the purpose of the web app cannot be achieved.

## Create Request/Offer (*Aaron Win*)

### DFD Process 3. Filling out Form (Request/Offer Form Page)

**Inputs:** ID of selected student

**Outputs:** Request details added to the Database

Item to Test	Test Description	Pass/Fail Criteria	Level of risk	Reason
Click and drag the cursor across a map.	Testing whether the map API drag effect works.	<b>Pass:</b> The map moves smoothly with the cursor.  <b>Fail:</b> Any other outcome.	Low	This functionality is derived entirely from Google Maps API so there should be no real risk.
Enter into a search box.	Both maps have a respective search box, one for the start location and one for the destination. For this test, the first few letters of a place are entered to test the first half of the autocomplete mechanic.	<b>Pass:</b> A list of possible results appear below the search box as per the autocomplete.  <b>Fail:</b> Any other outcome.	Medium	There is a moderate amount of script code, the correctness of which this test's success depends on. Effect would be medium/high as the full location name needs to be sent to the database.
Select one of the search results.	(Assuming the previous test passed.) After typing into one of the search boxes, a list of predictions appears below. For this test one of these is selected to test the second half of the autocomplete mechanic.	<b>Pass:</b> The selected result is filled into the search box, and the respective map goes to the selected location.  <b>Fail:</b> Any other outcome.	Medium	Same as the above test.
Click "Submit" when the form is incomplete.	All fields in the form must be filled in before being submitted. Here, the ability of the page to handle the attempted submission of a form with at least one field left empty is tested.	<b>Pass:</b> The form fails to submit. Instead, the first empty field displays a popup which indicates the user must fill it in.  <b>Fail:</b> The form is submitted successfully.	Medium	All input elements contain the 'required' label which should prevent an incomplete form from being submitted. However, the effect is high as incomplete data in the database would likely cause other parts of the system to fail.

Clicking “Submit” when the form is complete.	The test involves submitting a completed form, seeing the new request/offer in the main page with limited details and checking the database to ensure full details are there.	<p><b>Pass:</b> Request/offer details are added to the database. The user is taken to the main page.</p> <p><b>Fail 1:</b> The form fails to submit for any reason.</p> <p><b>Fail 2:</b> The user is taken to a non-existent page.</p>	High	There is a lot of code and integration thus a good possibility of failure. The consequences are also significant, as the user would be stuck after submitting the form.
--	---	---	------	---

### Detailed Request/Offer Page (*Victor Yao*)

DFD Process: 4. Finding a Match

**Inputs:** Detailed user information retrieved from the database (name, role, party\_number, time\_to\_leave, home\_address, city, zip\_code, email, contact\_number, comments, locationLatitude, locationLongitude)

**Outputs:** Make change in ‘party\_number’ when user hits “Accept” button.

Item to Test	Test Description	Pass/Fail Criteria	Level of risk	Reason
Load the user's detailed information	The system should be able to retrieve user's details from the database and display them onto this Detail Page.	<p><b>Pass:</b></p> <ul style="list-style-type: none"> <li>- Sending requests and successfully making connections to the database.</li> <li>- Retrieving data from the database to the page.</li> <li>- User's information is displayed in the right place on the page.</li> </ul> <p><b>Fail 1:</b> Fail to make connection to the database.</p> <p><b>Fail 2:</b> Fail to retrieve database data and get them displayed on the page.</p> <p><b>Fail 3:</b> The retrieved data got shown in the wrong place on the page.</p>	Medium	Not much could go wrong when retrieving data from the database in this case where there are going to be a huge number of users accessing the database at the same time. But since the data contains personal information, there are certain levels of risks to be considered.

Display user's location using Google Map API	Through Google Map API, the page should be able to display the user's location on the map using the latitude and longitude data retrieved from the database.	<p><b>Pass:</b></p> <ul style="list-style-type: none"> <li>- Successfully making connections to the database and retrieving the latitude and longitude data.</li> <li>- Instantiating a Google Map object and pin-pointing user's location using latitude, longitude data. The map is also manipulatable (click &amp; drag).</li> </ul> <p><b>Fail 1:</b> Fail to successfully retrieve the latitude and longitude from the database.</p> <p><b>Fail 2:</b> Fail to instantiate a Google Maps onto a page.</p> <p><b>Fail 3:</b> Fail to pin-point the user's location on the map.</p> <p><b>Fail 4:</b> Fail to be able to click &amp; drag, zoom-in, zoom-out the map.</p>	Medium	Although this step involves dealing with user's sensitive location data and home coordinates, Google Map API is a trust-worthy party when it comes to protecting user's information.
<p>Change user's 'party_number' data in the database when clicking on the "Accept the ride" button.</p> <p>**'party_number' is the number of people from each party, which could be just one person to four people because normally there are only five seats in a car.</p>	<p>When clicking on the "Accept the ride" button, the 'party_number' for both users should be changed to the total of 'party_number's from both parties.</p> <p>E.g. Tom (party_number is 1) accepts the ride on Bob's detailed page (party_number is 2). Their party_number data in the database should be changed to 3.</p>	<p><b>Pass:</b></p> <ul style="list-style-type: none"> <li>- Successfully making connections to the database.</li> <li>- Successfully retrieving the 'party_number' data of both users and then making an addition of that.</li> <li>- Successfully updating the 'party_number' for both users in the database.</li> </ul> <p><b>Fail:</b> Fail to do those processes shown above.</p>	Low	No sensitive information involved during this process.

## 4.2 Attributes of Focused Testing

Performance was selected as an attribute to test as the web application relies heavily on communicating back and forth with a database. Thus, the performance and the time it takes to facilitate this transfer of data is important and crucial to the positive experience that the user has with the Ride Share website. It needs to be the case that the results of button clicks, page updates and other functions happen quickly so as to not waste the user's time, prevent them from doing anything else whilst the transfer of data occurs, or give them the impression that something has gone wrong.

Usability is also selected as an attribute to test as this product is a web application which people will use to sort out transport quickly and easily to and from some destination. If it is difficult, confusing or time consuming to browse current ride offers, or to create a new offer, users will immediately be put off and likely decide to not use this web application again. The usability of this web application should allow the user to complete their goals without hindrance or untoward hesitation. Thus, it is a good idea to test how easy this web application is, so that the developers can provide the best and most convenient experience to a user possible.

### 4.2.1 Performance Testing

Item to Test	Test Description	Pass/Fail Criteria
Logging in with Facebook occurs within a reasonable time.	The tester will login to the web application via Facebook and the time it takes for that to be processed and the user presented with the next screen will be measured.	<b>Pass:</b> The time it takes to login with Facebook is under 3 seconds.  <b>Fail:</b> The time it takes to login with Facebook is greater than 3 seconds.
Loading the current Ride Offers occurs within a reasonable time.	The tester will try to load the current Ride Offers. 3 Test Offers will be added to the database for the purposes of this test. The time it takes for the Current Offers to be retrieved and displayed to the user will be measured.	<b>Pass:</b> The time it takes to load the Current Ride Offers is under 2 seconds.  <b>Fail:</b> The time it takes to load the Current Ride Offers is greater than 2 seconds.
50 users on separate devices all attempt to load all Current Offers at once.	50 testers will attempt to load all Current Offers at the same time, to see if the web app and database can handle the capacity of users simultaneously. 3 Test Offers will be added to the database for the purposes of this test. This	<b>Pass:</b> All users get served the Current Ride offers in 10 seconds or less.

	could be performed manually with 50 Testers, or a program could be made to simulate 50 different users, all attempting to do the same thing.	<p><b>Fail 1:</b> The users get nothing or an error is returned to their page, likely because the number of simultaneous requests crashed the web app/database.</p> <p><b>Fail 2:</b> The users get served the Current Ride offers in greater than 10 seconds.</p>
--	--	--

## 4.2.2 Usability Testing

Item to Test	Test Description	Pass/Fail Criteria
How long it takes the user to find out how to, then proceed and Login with Facebook.	The tester who has never used the web app before will attempt to log in via Facebook in a reasonable amount of time.	<p><b>Pass:</b> The user figures out how to log in with Facebook and does so in less than 30 seconds.</p> <p><b>Fail 1:</b> The user is not able to find out how to login with Facebook.</p> <p><b>Fail 2:</b> The user logs in with Facebook, but it takes them longer than 30 seconds to do so.</p>
How long it takes a user to join a current Ride offer that suits them.	The tester who has never used the web application before will attempt to join an active Ride offer that suits them in a reasonable time. 3 Test Offers will be added to the database for the purposes of this test. The test assumes the user has already logged in and is at the main offers screen.	<p><b>Pass:</b> The user figures out how to join the active offer they want and does so in less than 30 seconds.</p> <p><b>Fail 1:</b> The user is not able to find out how to join the offer they want.</p> <p><b>Fail 2:</b> The user finds and joins the offer they want but it takes them longer than 30 seconds to do so.</p>
How long it takes a user to create a new Ride Share offer.	The tester who has never used the web application before will attempt to create a new Rise Share offer in a reasonable time. The Test user will act as the “driver” for the offer during the test. The test assumes the user has already logged in and is at the main offers screen.	<p><b>Pass:</b> The user figures out how to create a Ride Share offer in less than 2 minutes.</p> <p><b>Fail 1:</b> The user is not able to find out how to create a Ride Share offer.</p>

		<b>Fail 2:</b> The user figures out how to create a Ride Share offer, but it takes them longer than 2 minutes.
--	--	--

## 5. Environmental Needs

There is no special hardware or power requirement required to perform these test plans. There are also no specific versions of other supporting software needed for the tests to run successfully. However, for continuity sake, an up-to-date and recent web browser should be used when testing this web application. Any older or out of date browsers are not expected to be used for this test plan. A computer or laptop is the only required tool for this test plan. A smart phone should not be used to test this website. In order to perform any of the tests mentioned above, the tester must make use of communication tools such as web, client/server, network and router.

Test data will be provided by creating 3 ride-sharing offers, which will be stored in the database. Each of these will be used to test different parts of the web pages and its functionality. Multi-part features will be tested individually during the normal running of the feature, in the order they are encountered.

The tests will be performed with a database empty besides the test data. Therefore, no other users should use the web application while testing is occurring, as this will add more data to the database.

To test the sign in function, a Facebook account is needed. For security purpose, the private information of Facebook account (except email address) must not be revealed during the testing process.

## 6. Responsibilities

As shown in the table in Section III, the four team members are responsible for the testing of the features that each person does the coding:

- Lysa is responsible for the features on the Index Page, User Profile Page and Edit User Profile Page:
  - Display navigation bar and footer
  - Display image slider

- Sign in with Facebook
  - Display user's information
  - Edit Profile button
  - Edit user's information
  - Update Profile button
- Ryan is responsible for the features on the Main Page:
  - Load Offers
  - Clicking an offer takes you to that offer's page
  - Clicking "Join" on the offer's page to join the ride gives the user the pertinent details of the offeree.
- Aaron is responsible for the features on the Request Form Page:
  - Clicking and dragging the cursor across a map.
  - Entering into a search box.
  - Selecting one of the search results.
  - Clicking "Submit" when the form is incomplete.
  - Clicking "Submit" when the form is complete.
- Victor is responsible for the features on the Detailed Request/Offer Page:
  - Load the user's detailed information correctly.
  - Display user's location using Google Map API
  - Changing user's 'party\_number' data in the database when clicking on the "Accept the ride" button.
  - Clicking "Join" on the offer's page to join the ride gives the user the pertinent details of the offeree.

The team make use of online resources to solve problems during the developing and testing process. For the resources such as the server, remote database that is provided by the client, the team can go to the client for information. In addition, the team can go to the lecturers if the team encounters something difficult to solve.

For all the testing items not covered in the plan, the team would design the test plan on the development level and also review the impact of one functionality on other features of the project. The team would make close contact with the client should any major were to be done to certain features.



# Appendices

#16 Meeting with Convenors: The Software Development team and the Convenors discussed the groups previous Design Document and talked about where they were at with the project. They pitched the overall concept of their web app to the convenors, so they would have a better understanding of what was done in the project and where the team needed to go next. The team realised that they needed to get on with actually coding and developing the project, lest they run out of time.

#19 Meeting with Convenors: The Software Development team and the Convenors discussed where they were at with the project. Particularly, how the project was going to be delivered to the client, with the result being that once the key features of the product were complete and coded, it could be physically used and tested to make sure it actually works with users, then can be developed further until the final due date.

#24 Meeting with Convenors: The Software Development team and the Convenors discussed where they were at with the project. They narrowed down what was and was not needed for the Test Plan, in relation to their project's implementation and scope. They were also briefed on the last assignment for the course, the presentation and what that entailed.

No	Date	Duration	Attendees	Note	Task Allocation
16	17th Sept 2020  1:20pm	12 minutes	Victor Yao Aaron Win Lysa Phan Ryan Good <i>Lecturers</i>	<ul style="list-style-type: none"><li>• Pitched an overview of our app</li><li>• Discussed our design document</li><li>• Team was encouraged to start the implementation as soon as possible</li></ul>	None
17	17th Sept 2020  1:32 pm	2 ½ hours	Victor Yao Aaron Win Lysa Phan Ryan Good	<ul style="list-style-type: none"><li>• Began work on implementation</li><li>• Created basic skeleton of site (html, css, js)</li><li>• Allocated tasks to begin functionality implementation</li></ul>	- Lysa: Facebook Verification - Ryan: Health Claim - Aaron: New requests

					- Victor: Database tables
18	23rd Sept 2020  1.00pm	38 minutes	Victor Yao Aaron Win Lysa Phan Ryan Good <i>Rohesia</i>	<ul style="list-style-type: none"> <li>A focus should be placed on future adaptability and scalability in the design of our pages</li> </ul>	None
19	24th Sept 2020  1:20pm	19 minutes	Victor Yao Aaron Win <i>Lecturers</i>	<ul style="list-style-type: none"> <li>Start thinking about how the project is going to be delivered to the client</li> <li>Test plan</li> </ul>	None
20	30th Sept 2020  1:00pm	5 minutes	Victor Yao Aaron Win Lysa Phan Ryan Good <i>Rohesia</i>	<ul style="list-style-type: none"> <li>A user profile was shown and discussed</li> <li>The focus of the discussion was what information personal information should be shown to the user and when</li> </ul>	None
21	30th Sept 2020  1:10pm	42 minutes	Victor Yao Aaron Win Lysa Phan Ryan Good	<ul style="list-style-type: none"> <li>What task each person was doing was discussed</li> <li>Organized and allocated tasks for the next sprint</li> </ul>	- Lysa: User Profiles - Victor: Request Page - Aaron: Offer page - Ryan: Main page, showing all active offers
22	3rd Oct 2020  12:00pm	72 minutes	Victor Yao Aaron Win Lysa Phan Ryan Good	<ul style="list-style-type: none"> <li>Try to access database</li> <li>Discuss test plan</li> </ul>	None
23	7th Oct 2020  1.00pm	25 minutes	Victor Yao Aaron Win Lysa Phan Ryan Good <i>Rohesia</i>	<ul style="list-style-type: none"> <li>Earliest Departure Time/ Latest Arrival Time are better names for start/end time of rides</li> <li>Could add calendars to the request form</li> </ul>	Continuing to work on the tasks already assigned
24	8th Oct 2020  1:20pm	14 minutes	Victor Yao Aaron Win Lysa Phan Ryan Good	<ul style="list-style-type: none"> <li>Discuss the test plan and the presentation.</li> <li>Presentation due October 30th</li> </ul>	None

			<i>Lecturers</i>		
25	11th Oct 2020  11:00am	11 minutes	Victor Yao Aaron Win Ryan Good	<ul style="list-style-type: none"> <li>Discuss test plan</li> </ul>	None
26	11th Oct 2020  4:45pm	90 minutes	Victor Yao Aaron Win Lysa Phan Ryan Good	<ul style="list-style-type: none"> <li>Work on test plan</li> </ul>	None