Team 34 – Transport app – Uni bus

Created by Adam up2166905, Granville up2042436, Andrew up939395, Jude up2080988, Baraa up2041097 and Hasnain up961975

Specification

This app will be used to help users navigate around Portsmouth with the help of the Uni bus. This chapter will focus on the requirements that we will use to build our app. In this chapter we will explore how we collected to data to find out what our potential users want from a Uni bus app, the user requirements extracted from that data and the functional and non-functional requirements taken from the user requirements.

Interview Participants and Structure

The process for determining our interview participants is simple. Whilst any one will be able to use our app it will be most helpful for students and staff who can ride the Uni bus. So, we selected two students at random, one who already uses the Uni bus and one who does not.

We collected the following information from then before asking them more specific questions about our app; their age; gender; whether they use the Uni bus; and whether they use other transport apps. This information can be seen in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Gender | Age | Do they use the Uni bus? | Do they use other transport apps? |
| 1 | Male | 21 | Yes | Yes |
| 2 | Non-binary | 19 | No | Yes: Uber and Aqua Taxi |

The interviews took place in person with the interviewer first describing the basic idea being our app. That being that it is to be an app that will help students and staff of Portsmouth university navigate around Portsmouth and help them take the Uni bus. These interviews were created with the main idea to ask a few main questions to get the participant thinking and then to ask specific follow up questions depending on the participants answers in order to get a better idea of what the participant would really want from our app.

The main questions asked were:

1. What would you want in a Uni bus app to help you to travel around?
2. How would you visualise that app?
3. What non bus related aspects would you want to see in a bus app?
4. Would you want a login system and if so what would you want it to do?

These questions were designed to make the participant think more broadly and not influence their answers so that the data we collected would be of a high quality.

Ethics Review

The participants were also made to sign a consent form before taking this interview that allows us to use this information. The questions were also designed to not be personal whilst still gathering useful information about their opinions.

Results and Analysis

The information collected revealed that the participants want our application to have an interactive map with clear icons that will allow the user to see the bus route and their own location. The interviews also revealed that the participants what to see what times the bus arrives at each stop and an estimation of how long that is from the current time. The participants also highlighted that it would be useful for other buildings such as University of Portsmouth buildings and Portsmouth landmarks to appear on the map as this will mainly be used by University of Portsmouth students and staff. The participants also mentioned that a way finding tool would be useful to help navigate to bus stops and around Portsmouth. Finally, the participants do want a login system and for it to store the recently viewed bus stops, university buildings and landmarks. However, they showed concern for how the data would be stored.

User Requirements

From these interview answers we generated the following user requirements:

|  |  |
| --- | --- |
| ID | Description of Requirement |
| 1 | Interactive map that shows bus stops, university buildings, landmarks and your current location |
| 2 | The time table for the bus stops and an estimated time for busses that arrive soon |
| 3 | A login system that stores the recently viewed bus stops, university buildings and landmarks |
| 4 | The ability to way find around Portsmouth which includes getting the Uni bus |
| 5 | The ability to search for bus stops, university buildings and landmarks |
| 6 | The app should be available on the latest versions of both IOS and Android |

Functional Requirements

The following functional requirements were generated from the previous user requirements. We have sorted them by priority and include the ID from the user requirement that this functional requirement was generated from. We have also included a summary and a detailed description of the requirement.

Must Have:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | User requirement ID | Summary | Detailed Description |
| 1 | 5 | Search for bus stops | The user will input the name of the bus stop and then displayed on the screen will be the closest matching bus stops that the Uni bus stops at ranked in order of which bus stop names have the most adjacent matching letters in the right order to the name entered by the user. Only bus stops that have at least 3 matching adjacent letters will be displayed. All letters will first be turned to lower case for the purpose of this ranking. E.g. if the user entered “Cannon street” and there is a bus stop named “cannon street” then it will be ranked highest due to every letter matching, however if the user input “teerts nonnaC” then the bus stop “Cannon street” would not match every letter even though Cannon street is just teerts nonnaC backwards due to the order needing to be preserved so the best match between teerts nonnaC and Cannon street is the double “e” and double “n” making 2 matching letters. |
| 2 | 2 | Display bus timetable | The app will display a table containing every bus stop and then every time of the day that the Uni bus stops at that bus stop. |
| 3 | 1 | Display bus stops on a map | The app will display a map, similar to google maps or bing maps, and on that map it will display a marker at the location of every bus stop that the Uni bus stops at. |
| 4 | 1 | Displays Landmarks on map | The app will display a map, similar to google maps or Bing maps, and on that map it will display a marker at the location of every landmark that we have chosen to include. |
| 5 | 1 | Displays university buildings on map | The app will display a map, similar to google maps or Bing maps, and on that map it will display a marker at the location of every Portsmouth university building that we have chosen to include. |

Should Have:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | User requirement ID | Summary | Detailed Description |
| 6 | 2 | Estimated time of arrival on Timetable | On the bus timetable next to the times that the bus will arrive at each stop, the amount of time until it is that time should be displayed in minutes along with the word minutes. This should only be displayed for times that have not yet past that day, as in it should not give an estimated time of arrival for a bus that will arrive tomorrow. E.g. bus arrives at 5:50pm, current time is 5:40pm so it should display “10 minutes” |
| 7 | 1 and 2 | Click on bus stop to get the estimated time of arrival for that stop | When a user clicks on a bus stop, whether through the map or the search menu, the app should display the estimated time of arrival in minutes for the remaining times that day that a bus will stop at that bus stop. This is similar to how an estimated time of arrival is displayed on the timetable except it is only displaying the info for the chosen bus stop. |
| 8 | 1 and 2 | Click on bus stop to get the bus times for that stop | When a user clicks on a bus stop, whether through the map or the search menu, the app should display the timetable for that bus stop. As in it should display the times that a Uni bus stops at that bus stop. This is similar to the timetable except it is only displaying the info for the chosen bus stop. |
| 9 | 1 | Displays your current location on map | The app will display a map, similar to google maps or bing maps, and on that map it will display a marker at your current location if the app has been given permission to access your current location |
| 10 | 5 | Search for university buildings | The user will input the name of the university building and then displayed on the screen will be the closest matching Portsmouth university buildings ranked in order of which university building names have the most adjacent matching letters in the right order to the name entered by the user. Only university buildings that have at least 3 matching adjacent letters will be displayed. All letters will first be turned to lower case for the purpose of this ranking. This is the same algorithm that will be used to search for bus stops. |
| 11 | 5 | Search for Land marks | The user will input the name of the landmark and then displayed on the screen will be the closest matching landmarks ranked in order of which landmark names have the most adjacent matching letters in the right order to the name entered by the user. Only landmarks that have at least 3 matching adjacent letters will be displayed. All letters will first be turned to lower case for the purpose of this ranking.  This is the same algorithm that will be used to search for bus stops. |
| 12 | 2 | Timetable estimated time of arrival’s should update | When the user is viewing the timetable the estimated time of arrival’s displayed should update once every minute. Since the estimated time of arrival’s will be displayed in minutes this just means that they should all decrease by 1 when this update happens. For estimated time of arrival’s that would go below zero, meaning the time that that bus is supposed to stop at that bus stop in that day has now passed, the estimated time of arrival is removed. |
| 13 | 3 | User login | The user should be able to login using their email and chosen password. |
| 14 | 3 | User register | The user should be able to register for an account using their email and their chosen password. |
| 15 | 3 | User account saves recently viewed bus stops, landmarks and university buildings. | User that are using a registered account will save the 5 most recently viewed bus stops, landmarks and university buildings that the user accessed. This means the 5 most recently viewed bus stops, the 5 most recently viewed landmarks, and the 5 most recently viewed university buildings. |
| 16 | 3 | Recently viewed bus stops to be displayed back to the user. | When a user using a registered account clicks to open the search function for the bus stops, before the user types anything, their 5 most recently viewed bus stops will be displayed. Once the user types at least 3 characters into the search function these recently viewed bus stops will be replaced by the search results. Once the number of characters in the search function has dropped below 3, the recently viewed bus stops will reappear. |
| 17 | 3 | Recently viewed landmarks to be displayed back to the user. | When a user using a registered account clicks to open the search function for the landmarks, before the user types anything, their 5 most recently viewed landmarks will be displayed. Once the user types at least 3 characters into the search function these recently viewed landmarks will be replaced by the search results. Once the number of characters in the search function has dropped below 3, the recently viewed landmarks will reappear. This is similar to the recently viewed bus stops. |
| 18 | 3 | Recently viewed university buildings to be displayed back to the user. | When a user using a registered account clicks to open the search function for the university buildings, before the user types anything, their 5 most recently viewed university buildings will be displayed. Once the user types at least 3 characters into the search function these recently viewed university buildings will be replaced by the search results. Once the number of characters in the search function has dropped below 3, the recently viewed university buildings will reappear. This is similar to the recently viewed bus stops. |

Could Have:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | User requirement ID | Summary | Detailed Description |
| 19 | 4 | Wayfinding from location to bus stops for walking | A route will be displayed on the map from the users current location to the chosen bus stop that the university bus stops at. The route displayed will use paths and roads that a person can walk along. |
| 20 | 4 | Wayfinding to university buildings | A route will be displayed on the map from the users current location to the chosen Portsmouth university building. The route displayed will use paths and roads that a person can walk along. This is similar to wayfinding to the university bus stops. |
| 21 | 4 | Wayfinding to landmarks | A route will be displayed on the map from the users current location to the chosen landmark. The route displayed will use paths and roads that a person can walk along. This is similar to wayfinding to the university bus stops. |
| 22 | 4 | Walking distance along route | The distance that the route will need the user to walk will be displayed in kilometres. |
| 23 | 4 | Estimated time of arrival for walking route | The estimated time that the route will take to walk will be displayed in minutes. This will be calculated using the UK average walking speed of 5 kilometres per hour. |
| 24 | 4 | Making part of the route to the destination be taking the university bus. | The route mentioned in the previous requirements will now include taking the university bus where needed. This means that the route will direct the user to the bus stop along the route, then it will instruct the user to take the university bus and will display the part of the route done by the bus on the map in a different colour. Then the app will instruct the user at which bus stop the need to get off the university bus. Then the walking route will continue. |
| 25 | 4 | Estimated time of arrival for entire route | The estimated time that the user will finish the route. The walking parts will be estimated using the UK average walking speed of 5 kilometres per hour. |
| 26 | 4 | Let the user choose between multiple routes | When the user chooses selects a destination that they want a route to, if there is a difference, let them choose between the shortest distance walked and the fastest route. There may be a difference for example if the route tells the user to take the bus for the least walked route however there might be a long wait time for the bus making it faster to walk. |

Non-Functional Requirements

The following non-functional requirements were also generated from the user requirements. These have also been sorted by priority.

Must have:

|  |  |  |
| --- | --- | --- |
| ID | User requirement ID | Summary |
| 27 | 1 | Our use of openlayers map will comply with their licence agreement |
| 28 | 2 | The timetable should be available 24 hours a day |

Should Have:

|  |  |  |
| --- | --- | --- |
| ID | User requirement ID | Summary |
| 29 | 6 | App will be available on Android 13 |
| 30 | 6 | App will be available on IOS 16 |
| 31 | 2 | Estimated time of arrival’s on timetable should update at least once per minute |

Could Have:

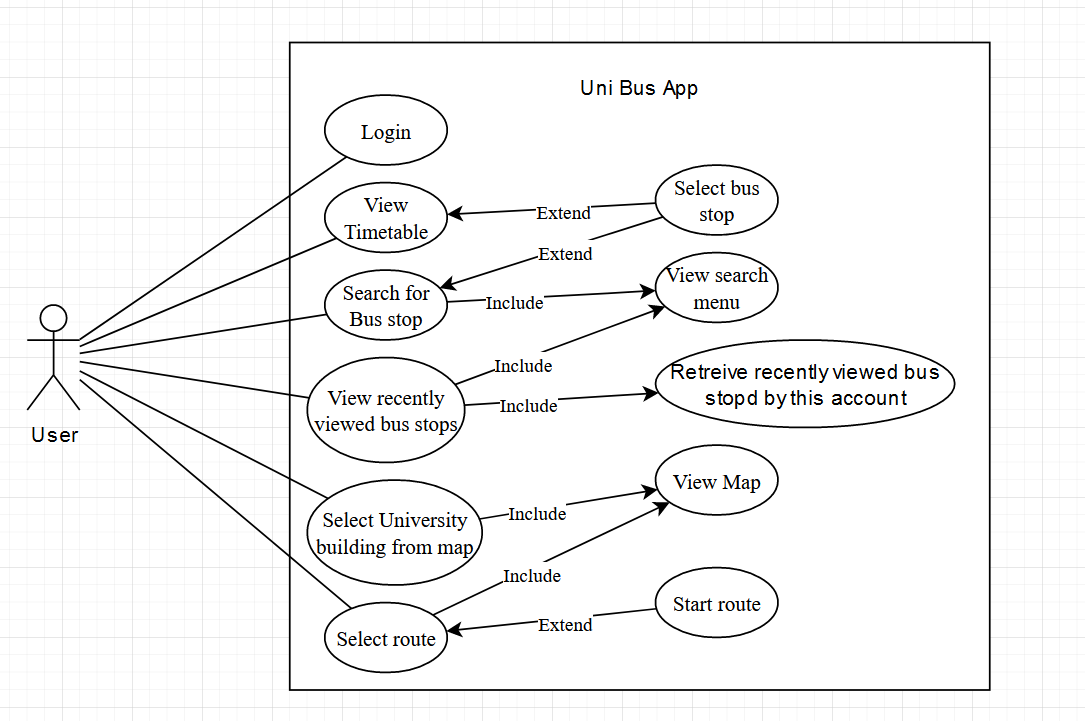
|  |  |  |
| --- | --- | --- |
| ID | User requirement ID | Summary |
| 32 | 4 | Wayfinding time estimated time of arrival’s should update at least once per minute |
| 33 | 4 | Wayfinding distance estimate’s should update at least once per minute |

Design

This chapter will focus on the use cases for our app and the model we will use to create our app.

Use Cases

Below is our use case diagram in which we have chosen 6 use cases to display, we will go into further detail for the bottom 5 just below the diagram.



Use Case 1:

|  |  |
| --- | --- |
| Category | Info |
| Summary | View Timetable |
| Description | The user is to view a screen that shows a table including every bus stops and then all the times in a day that a bus is scheduled to stop at that bus stop |
| Pre-condition | The app is connected to our database via the internet |
| Post-condition | The timetable is displayed |
| Input data | Click to view the time table screen |
| Output data | The bus stops in a table along with all the times that a bus will stop at that bus stops. For busses that have not yet arrived that day an estimated time of arrival will be given. (The amount of time till the bus is scheduled to arrive) |
| Valid case | The user selects the button to display the timetable. A request is sent to the server, the server retrieves the timetable from the database and sends the information back to the client where it is displayed on screen |
| Error case | The user selects the button to display the timetable. A request is sent to the server. The server is unable to fulfil the request and so an error message is displayed to the user asking them to try again later |

Use Case 2:

|  |  |
| --- | --- |
| Category | Info |
| Summary | Search for bus stops |
| Description | The user is to view a search bar in which they can type to search for bus stops |
| Pre-condition | The app is connected to our database via the internet |
| Post-condition | The bus stops matching the input text is displayed |
| Input data | The name of a bus stop |
| Output data | Bus stops with similar names to the input text as a click-able option, which when clicked will display that bus stop on the map as well as its schedule. |
| Valid case | The user clicks to search through the bus stops. The client requests a list of all bus stops from the server, the server retrieves the bus stop names from the database and fulfils the request. Then when a user enters a bus stop name the bus stop is found in the list and displayed to the user |
| Error case | The user clicks to search through the bus stops. The client requests a list of all bus stops from the server, the server retrieves the bus stop names from the database and fulfils the request. Then when a user enters a bus stop name the bus stop is not found in the list and an error is displayed to the user saying no bus stops have been found |

Use Case 3:

|  |  |
| --- | --- |
| Category | Info |
| Summary | View recently viewed bus stops |
| Description | When the user selects the search bar but before they start typing. If they are logged-in their recently viewed bus stops should display |
| Pre-condition | The app is connected to our database via the internet and the user is logged in |
| Post-condition | The recently viewed bus stops are displayed |
| Input data | Click to view the search bar but no text input to the search bar |
| Output data | The recently viewed bus stops for that account |
| Valid case | The user clicks to search for a bus stop. The client requests the recently viewed bus stops from the server. The server retrieves the bus stops from the database then fulfils the request from the client. The client then displays the recently viewed bus stops |
| Error case | The user clicks to search for a bus stop. The client requests the recently viewed bus stops from the server. The server cannot fulfil the request so an error message saying the recently viewed bus stops cannot be viewed at this time |

Use Case 4:

|  |  |
| --- | --- |
| Category | Info |
| Summary | Select University building from map |
| Description | When the user is using the interactable map, they can click on an icon for a university building and then they should be displayed with the name of that university building |
| Pre-condition | The app is connected to our database via the internet |
| Post-condition | The selected university building’s name is displayed |
| Input data | The click on that university buildings icon on the map |
| Output data | The name of the selected university building |
| Valid case | When the app is launched the client requests the names and locations of all university buildings from the server. The server retrieves the university buildings from the database then fulfils the request from the client. These are then displayed with a marker at their location. The user clicks on the icon of a university building on the map. The client then displays the name of this building |
| Error case | When the app is launched the client requests the names and locations of all university buildings from the server. The server does not respond so the university buildings cannot be displayed and an error message is given saying that the app could not connect to our server. Since no university buildings are received, they cannot be displayed on the map and so the user cannot click on them |

Use Case 5:

|  |  |
| --- | --- |
| Category | Info |
| Summary | Select route |
| Description | When the user has selected a destination and starting location for their route they should be able to select 1 of a few optional routes. |
| Pre-condition | The app is connected to our database via the internet and the user has selected a destination and starting location for their route |
| Post-condition | The route selected is displayed and the other possible routes disappearing from the screen |
| Input data | A click on the route that they wish to follow |
| Output data | The route being displayed on the map |
| Valid case | The user selects the way finding tool and inputs a destination and start location. Then the client calculates the best possible routes for the user and if a possible route might require taking the Uni bus a request to the server is made for the timetable. The server retrieves the timetable from the database then fulfils the request from the client. These routes are then displayed to the user and the user then selects a route. That route then becomes the only route displayed on the map and the other possible routes disappear |
| Error case | The user selects the way finding tool and inputs a destination and start location. Then the client calculates the best possible routes for the user and if a possible route might require taking the Uni bus a request to the server is made for the timetable. The server does not respond and so the client only displays routes not using the Uni bus and displays an error message stating it cannot connect to the server. user then selects a route. That route then becomes the only route displayed on the map and the other possible routes disappear |

Architectural model

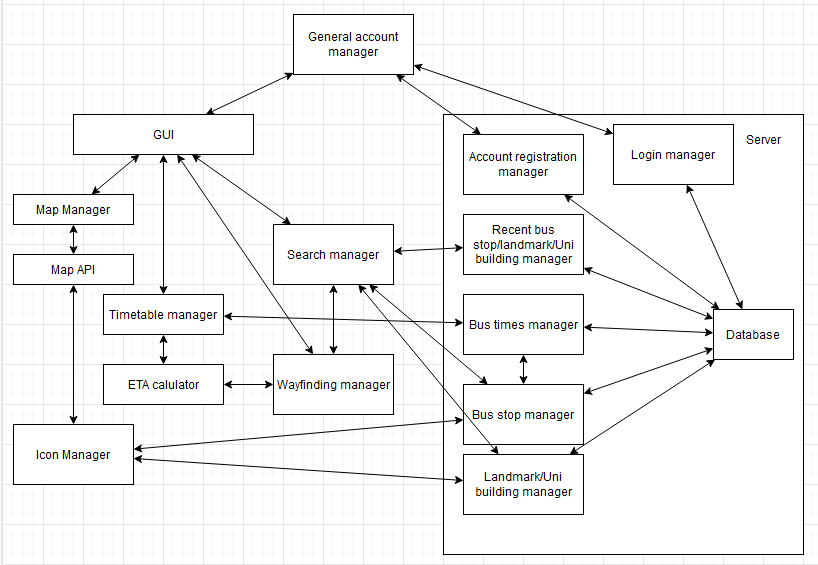
Our model uses the client-server architecture. This was chosen due to the fact that each client has no need to communicate with each other but every client does need the same common information about the bus time table and user accounts. This is a perfect case for the client-server model.

In our model most of the processing is done client side, such as which bus stops match the search. However these client side managers will then make a server call to a server side manager. The point of the server side managers is to request information from the database on behalf of the client side managers. The only expectations on the server side are the Account registration manager and the Login manager which along with requesting information from the database on behalf of the client side managers, also process the information as they deal with sensitive information about user accounts which should stay on the server. The client side managers interact with the GUI and either process the information themselves, such as the wayfinding manager, call upon the server such as the search manager, or ask the ETA calculator to perform a calculation for it.

The brokers architecture was considered, however, the client needs to be able to receive the same information no matter which server it is talking to and the broker model complicates this process unnecessarily for our needs. Especially since we do not expect a large number of users due to our app being focused of specifically university of Portsmouth staff and students.

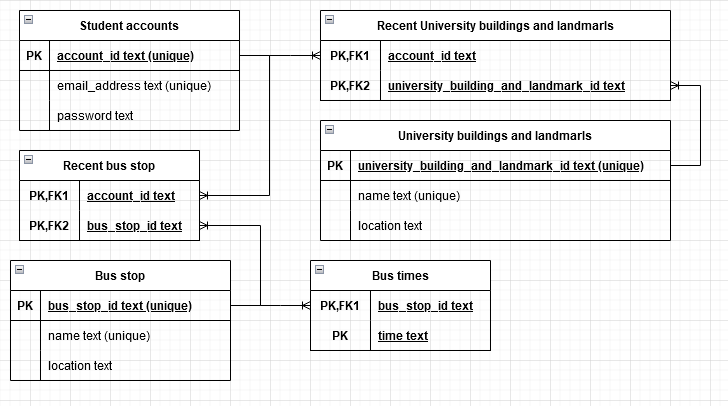
For our client side managers we believe an event based model is best as most of our requirements are triggered based upon user action and require very little processing.

The pipe and filter model was also looked at however our app would result in lots of very short parallel pipes and so we believe that using such a model would over complicate our app when all we need are a few managers to talk to managers on the server to talk to our database.



Database Design

Our database will store both information about our user accounts but also information about the bus timetable. This will allow us to relate the recently selected bus stops, landmarks and university buildings to the user accounts whilst also creating an easy way to store the timetable without wasting space.



Implementation

Software Reuse

We will be using open layers maps to allow us to easily create an interactable map. Open layers is a free tool used to manipulate maps and make them interactable with icons and so is perfect for our application.

Technologies and languages

We be programming most of our project in dart with the help of flutter. In order to make our map provided through open layers maps interactable a little bit of JavaScript will also have to be used.

Target Devices and Operating Systems

Our target devices are smart phones with the aim to bring our app to IOS 16 and Android 13 allowing most modern smart phones to run our app.

Version control

We have set up a GitHub project at the following link: <https://github.com/AdamMurrayPortsmouth/Uni-bus-app-SETAP>

This will allow us to control all the versions of our project along with the help of GitHub desktop which we have installed on each of our devices.

Issues and Bugs Management

We will use GitHub to manage and assign Issues and bug to the team. These issues will be brought forward in 2 week sprints and the team will each select an issue/bug to implement/fix before moving onto the next issue/bug.

Feasibility Check

We are a team of 6 and have reviewed our requirements and still believe that the second semester will provide us with enough time to complete all 33 requirements due to most of these requirements being relatively simple tasks. We do believe this will not have use finishing early however do to the large number of tasks. Thus we are happy that this is a feasible project.

Testing

|  |  |  |
| --- | --- | --- |
| Requirement ID | Description of Test | Inputs for Test |
| 1 | To search for bus stops we will enter the name of a bus stop and see if it appears and if any other bus stops appear that shouldn’t | Any name of a bus stop, e.g. Elm Grove |
| 2 | To test the bus timetable we will click to look at the screen that shows the timetable and check that all the times are correct | Click to bring up the timetable screen |
| 3 | To test if the bus stops are displayed on the map we will open the screen displaying the map and check that there is a marker at the correct location for each bus stop and that there are no extra bus stop markers | Click to open the map |
| 4 | To test if the landmarks are displayed on the map we will open the screen displaying the map and check that there is a marker at the correct location for each landmark and that there are no extra landmarks markers | Click to open the map |
| 5 | To test if the university buildings are displayed on the map we will open the screen displaying the map and check that there is a marker at the correct location for each university building and that there are no extra university buildings markers | Click to open the map |
| 6 | To test the estimated time of arrival’s on the bus timetable we will click to look at the screen that shows the timetable and check that all the estimated time of arrival’s are correct according to the bus times displayed | Click to bring up the timetable screen |
| 7 | To test the estimated time of arrival’s for a particular bus stop we will select that bus stop then check that the estimated time of arrival’s displayed are correct according to the bus times displayed | Click on bus stop either through the map for search menu |
| 8 | To test the bus times for a particular bus stop we will select that bus stop then check that the bus times displayed are correct | Click on bus stop either through the map for search menu |
| 9 | To test if our current location is displayed on the map in the correct location we will turn on our devices location, ensure that the app had permission to access our location, then compare the location viewed on our map to where other mapping apps such as google maps and apple maps display our location | Click to open the map |
| 10 | To search for university buildings we will enter the name of a university building and see if it appears and if any other university buildings appear that shouldn’t | Any name of a university building, e.g. Anglesea |
| 11 | To search for landmarks we will enter the name of a landmark and see if it appears and if any other landmarks appear that shouldn’t | Any name of a landmark, e.g. Guildhall |
| 12 | To check that the timetable estimated time of arrival’s update we will open the timetable, record an estimated time of arrival, then wait a period of time such that it should have updated, then record the estimated time of arrival again. If it has updated correctly the estimated time of arrival should have changed | Click to bring up the timetable screen |
| 13 | The user should be able to enter their correct username and password and then be logged into their account | Enter email and password |
| 14 | The user should be able to register for an account by entering an email and password. The email must not be registered to an account already | Enter an email and password |
| 15 | We can select bus stops, landmarks and university buildings and then query the database to check that the 5 most recently viewed have been saved | Select bus stops, landmarks and university buildings |
| 16 | To check that the recently viewed bus stops are being displayed back to the user, go to the search function for bus stops but don’t type anything, then check the bus stops that are displayed, these should be the 5 most recently viewed bus stops | Click to search for bus stops, then don’t input any text |
| 17 | To check that the recently viewed landmarks are being displayed back to the user, go to the search function for landmarks but don’t type anything, then check the landmarks that are displayed, these should be the 5 most recently viewed landmarks | Click to search for landmarks, then don’t input any text |
| 18 | To check that the recently viewed university buildings are being displayed back to the user, go to the search function for university buildings but don’t type anything, then check the university buildings that are displayed, these should be the 5 most recently viewed university buildings | Click to search for university buildings, then don’t input any text |
| 19 | When the user selects a route to a bus stop from the current location, a route that takes the user from the current location to the bus stop should be displayed on the map | Click to select way finding, select a bus stop, click to select starting location as current location, select route |
| 20 | When the user selects a route to a university building from the current location, a route that takes the user from the current location to the university building should be displayed on the map | Click to select way finding, select a university building, click to select starting location as current location, select route |
| 21 | When the user selects a route to a landmark from the current location, a route that takes the user from the current location to the landmark should be displayed on the map | Click to select way finding, select a landmark, click to select starting location as current location, select route |
| 22 | When a route is selected the distance along that route should be displayed, we can test this by creating a walking route along a distance we already know and checking if they are the same | Click to select way finding, select a destination, click to select starting location as current location, select route |
| 23 | When a route is selected the estimated time of arrival along that route should be displayed, we can test this by creating a walking route we already know how long it takes to walk and checking if the estimated time of arrival given is the same as our time | Click to select way finding, select a destination, click to select starting location as current location, select route |
| 24 | We select a destination so the app should offer the user a route that includes taking the Uni bus | Click to select way finding, select a destination, click to select starting location as current location, select route |
| 25 | When a route is selected the estimated time of arrival along that route should be displayed, we can test this by creating a route that requires the Uni bus we already know how long it takes to travel and checking if the estimated time of arrival given is the same as our time | Click to select way finding, select a destination, click to select starting location as current location, select route |
| 26 | After the user has selected the destination and starting point, multiple routes should be displayed if applicable, so we can select a destination and starting location such that multiple routes should be displayed, such as if the fastest route requires more walking but a slower route allows you to take the Uni bus and walk less | Click to select way finding, select a destination, click to select starting location as current location, select route |
| 27 | We check that our software complies with the open layers licence agreement by ensuring that our app includes the BSD copyright notice | N/A |
| 28 | We check the timetable at multiple hours of the day and ensure that it gets displayed during all of them | N/A |
| 29 | The app should be compatible with Android 13 | N/A |
| 30 | The app should be compatible with IOS 16 | N/A |
| 31 | We test the timetable estimated time of arrival’s update test with a period of 1 minute | N/A |
| 32 | We test the wayfinding estimated time of arrival’s with a period of 1 minute | N/A |
| 33 | We test the wayfinding distance estimates with a period of 1 minute | N/A |

Critical Analysis

Leadership

Adam Murray (up2166905) acted as the group leader through the whole semester as he was willing to do so and before he took charge progress was very slow. Progress sped up once Adam was leading the group as Adam assigned tasks and made sure work was done. Adam was accepted as the leader due to having by far the most experience and skills in both programming as an individual and as part of a team in both a university setting and industry. However, Adam’s leadership skills were highly lacking could have used help from other members of the group who were better skilled in leadership. Adams leadership was unorganised at times and lead to the group still working too slowly and too heavily relied on individual members. The group might have operated more effectively if everyone took a turn being the leader as this style of group heavily relied on Adam to function. However, given the way the group performed before Adam took charge, had he not taken charge and made himself the sole leader the group probably would have suffered even more.

Progress monitoring

We had weekly meetings to discuss the work done that week, the content of the lecture from that week, and any new work that needed to be assigned. These assignments were sometimes put on GitHub as issues and Adam encouraged the group to utilise GitHub as much as possible to collaborate on tasks issued to multiple people. These meetings were sometimes useful, however quite often not all of the group would turn up and heavily relied on Adam and Jude to get the meeting on track. In retrospect, for many weeks progress was too slow and this lead up to having a disproportionate amount of work to have to do in the last 3 weeks of the semester. A good way to track progress and make sure everyone was always doing what they were assigned was never found. Instead we ended up in a cycle of incomplete work being asked for feedback, feedback being given, the work not fully being edited according to the feedback, then the next week the still incomplete work would once again be asked for feedback, similar feedback would be given and the cycle would continue for many weeks. The most successful periods were right at the start of the project and right at the end, this was because work stopped being assigned and instead the group came together for in person meetings and did the work together. This way everyone worked on the issue at hand together and so no one could put of doing the work for a week and instead of feedback being given that was mostly ignored, everyone’s ideas would come together and the work would be made right there and then.

Conflict resolution

Conflict in the group was resolved in 1 of two ways. First, if it was a decision in an area where Adam was particularly confident then the group would follow what Adam would suggest. The biggest example of this is the plan to use opensource maps to create the interactive for the app, this was Adam’s idea and went unchallenged. The second way was by a majority vote that would take place among all members present at that meeting. An example of this occurred when creating an early draft for the use case diagram. There was a suggestion to add the ability to favourite bus stops, the group was split and so after a bit of debate a vote was held, 3-2 in favour of adding the feature and adding it to the use case diagram. This feature was later removed however. Overall conflict was quite rare in this group as most members looked to Adam for the solution and often was happy to go along with his ideas. That being said, conflict did arise outside of the work itself and tension grew in the group as time went on due to the varied levels of effort put into the project. This conflict never created major issues however it was also never dealt with properly.

Contributions Table

|  |  |  |
| --- | --- | --- |
| Name | Up number | Percentage |
| Adam | up2166905 | 19 |
| Jude | up2080988 | 19 |
| Granville | up2042436 | 17 |
| Andrew | up939395 | 16 |
| Hasnain | up961975 | 16 |
| Baraa | up2041097 | 13 |