

MINI-BUS GUIDANCE APP

Name: Danell Mathurin

PI: Z5162947

Course: TM470

EMA

Abstract

This report documents the development of a bus guidance application for the bus system in St. Lucia designed to improve usability for mainly passengers and some aspects for drivers. The project addressed the need for real-time route visualization, fare calculation, and intuitive navigation by leveraging React Native (Expo CLI) for cross-platform compatibility, Figma for UI prototyping, and MongoDB Atlas, Express and Node for backend data management.

The application was developed using an incremental approach, with each stage focusing on specific features. Increment 1 involved creating the user interface, using placeholder information for the map, routes, and fare details. Increment 2 focused on developing the database to store route and driver information. Increment 3 added features for simulating real-time bus locations, including polyline-based route visualization and GPS movement simulation. While the final system is not deployed for real-world use, the project demonstrates the feasibility of combining mapping technologies with location simulation to illustrate how a bus tracker application can function. The report reflects on the development process, challenges faced, and solutions applied, emphasizing technical implementation and project management practices. The outcomes provide insight into both the potential benefits and limitations of such applications and offer a foundation for future work on real-time public transport tracking systems.

Acknowledgement

I would like to express my heartfelt gratitude to my parents for their unwavering support and for funding my education, which made this journey possible. I am also deeply thankful to my girlfriend for her emotional support during challenging times, which helped me stay focused and motivated. I extend my appreciation to the tutors at the Open University for their guidance, encouragement, and valuable insights. Above all, I give thanks to God for guiding me and providing strength throughout this process.

Table of Contents

Project title	1
Problem description	3
Requirements and analysis	4
Impact of LSEPI/EPI on solution	9
Life cycle	10
Planning	12
Resources	13
Skills	14
Risks	15
Literature review	16
Work done	17
Review and reflection	40
References	43
Appendices	46
Appendix A - Ethics form	46
Appendix B - LESPI form	48
Appendix C – Consent form	49
Appendix D – Participation Sheet	50
Appendix E – Activity Log	50
Appendix F - Keeping in touch	50
Appendix G – Checklist for testing	50
Appendix H – Generative AI	50
Appendix I – Previous Plan	50

Table of tables

Table 1: Project Scope (Mathurin,2025)	9
Table 2:Evaluation Criteria (Mathurin,2025)	11
Table 3: LSEPI/EDI impact (Mathurin. 2025)	14
Table 4:Life cycle comparison (Mathurin, 2025)	15
Table 5: Resources (Mathurin, 2025).....	18
Table 6: Skills (Mathurin, 2025).....	19
Table 7: Risks(Mathurin,2025).....	20
Table 8: Literature review (Mathurin, 2025).....	Error! Bookmark not defined.

Tables of Figures

Figure 1: Incremental Lifecycle diagram (Mathurin,2025).....	16
Figure 2: Project plan (Mathurin,2025).....	17
Figure 3:Figma Design Part 1 (Mathurin, 2025)	30
Figure 4: Figma Design Part 2 (Mathurin. 2025)	30
Figure 5: Code Snippet from search bar (Mathurin,2025)	32
Figure 6:Code snippet for Google maps place holder (Mathurin, 2025)	33
Figure 7: Code snippet from bottom sheet(home) (Mathurin, 2025).....	34
Figure 8: Code snippet from custom button (Mathurin, 2025).....	35
Figure 9: Code snippet for adding button to bottom sheet (Mathurin, 2025).....	36
Figure 10: Code snippet from bottom sheet (details (Mathurin, 2025))	37
Figure 11: Code snippet from back button (Mathurin, 2025)	37
Figure 12: Code snippet for initializing initial stage (Mathurin, 2025)	38
Figure 13: Code snippet for async function (Mathurin, 2025)	38
Figure 14: Code snippet for calling async function (Mathurin, 2025).....	39
Figure 15: Completed user interface (Mathurin, 2025)	40
Figure 16: MongoDB database (Mathurin, 2025)	41
Figure 17: Route collection part 1(Mathurin, 2025)	43
Figure 18: Route collection part 2 (Mathurin, 2025)	44
Figure 19:Prices in database (Mathurin, 2025).....	45
Figure 20: Driver collection (Mathurin, 2025)	46
Figure 21: Environment Variables (Mathurin, 2025)	47
Figure 22:Using URI in server.js (Mathurin, 2025)	48
Figure 23: Using the port in server.js (Mathurin, 2025)	48
Figure 24:Code snippet for driverRoutes (Mathurin, 2025).....	49
Figure 25: Code snippet for routeRoutes (Mathurin, 2025)	50
Figure 26:Code snippet for driverService (Mathurin, 2025).....	51
Figure 27:Code snippet to display driver details (Mathurin, 2025).....	52
Figure 28: Driver details in UI (Mathurin, 2025)	53
Figure 29: App on physical android device. (Mathurin, 2025)	55

Glossary

AI – Artificial Intelligence; technology that enables machines to perform tasks that usually require human intelligence.

AI-generated icons – Images created using AI tools (like ChatGPT) instead of being manually designed, allowing unique and copyright-safe visuals.

API – Application Programming Interface; a way for software to communicate and exchange data.

Async (Asynchronous) function – A programming function that runs tasks in the background without stopping the app.

Backend – The part of an application that handles data, calculations, and communication with the database. Users do not see it directly.

Bottom sheet – A sliding panel in a mobile app that shows extra content without leaving the current screen.

Chrome Developer Tools – Browser tools for testing, debugging, and improving web and mobile applications.

CLI (Command-Line Interface) – A tool where commands are typed in text form to control software.

CRUD – Create, Read, Update, Delete; basic operations for managing data in a database.

Digital prototyping – Creating an interactive digital version of a design to test layout and usability before actual development.

Expo CLI – A tool for building and testing React Native mobile apps quickly and easily.

Figma – A software tool used to design and prototype user interfaces for apps and websites.

GUI (Graphical User Interface) – A visual interface where users interact with software using icons, buttons, and images instead of text commands.

MongoDB – A database system that stores information in flexible, document-like formats instead of tables.

MongoDB Atlas – A cloud-based platform for managing MongoDB databases with a user-friendly interface.

MongoSH (MongoDB Shell) – A command-line tool for interacting with MongoDB databases.

NoSQL – A type of database that stores data in flexible formats rather than traditional tables.

Placeholder map – A temporary static image or component used instead of a real, interactive map during development.

Polyline – A connected series of lines on a map used to show a route.

React Native – A framework for building mobile apps that run on both Android and iOS using JavaScript.

React Native CLI – A setup tool for creating React Native apps from scratch, offering more control but requiring more experience.

REST (Representational State Transfer) – A set of rules for designing web APIs to handle data efficiently.

Search bar – A user interface component that allows users to type queries to find content or destinations.

Snap points – Predetermined positions where a bottom sheet can stop, showing more or less content.

Status indicator – A symbol (e.g., red or green dot) showing the current state of something, such as a driver's availability.

TouchableOpacity – A button component in React Native that visually responds when pressed.

User-centered design – A design approach focused on making an application intuitive, easy, and accessible for users.

Viewport – The visible area on a device screen where content is displayed.

GPS simulation – Generating artificial location data in software to test real-time tracking without using real devices.

Problem Description

The Problem:

The mini-bus system problem in St. Lucia is an issue seen by many, including those in authority, but it appears that no clear solution has been reached. The explanation below outlines how the current system works, using the route from the city of Castries to Dennery, which has a driving time of approximately 40 minutes including potential stops.

From Castries to Dennery

Crucial times are generally after work hours, from 4 PM onwards. This is also the time when most drivers decide to go off duty. By 5 PM, there are usually only 2–4 buses trying to meet the demand of passengers waiting, as there is about a 120-minute wait for each bus to return. Given the number of passengers waiting, it is easy to see how hectic the situation becomes.

On rare occasions where the number of buses outweighs the number of passengers, competitive driving is noticeable. Drivers also make comments such as, “I saw another bus going to Castries already so I don’t think I will go back there.” This is often said without any certainty that the number of passengers has increased.

From Dennery to Castries

When heading to the city, passengers simply wait by the road until a bus arrives. It could be a 5-minute wait or even over an hour. The city is not the only stop, as passengers can board or leave a bus along the route, which is where pricing concerns become an issue.

Why is it a problem?

The above problem has a negative impact on all stakeholders, namely the driver, passenger and the public. This is a system which is utilised daily and has a significant impact on stakeholders.

Proposed Solution

A mobile application will be developed using React Native, tailored to passenger requirements, providing a user-friendly platform beneficial to improving trip planning and general satisfaction with the system. This solution would be based on web mobile technologies (TM352).

REQUIREMENTS ANALYSIS

Discussions with passengers

Two regular users of the current bus system on the route Dennery to Castries were consulted. One noted that there has been a recent attempt to alleviate some concerns, specifically improvements in availability during mornings, but the lack of availability persists at other crucial times, particularly after work hours. Both passengers outlined issues personally encountered with the underdeveloped system, such as the inability to effectively plan trips due to the lack of schedule knowledge.

One passenger also stated: "*Some time I'm waiting and waiting and no bus.... Nothing. Most times I must decide on an alternative route like catching a ride to another bus stop where the frequency of buses is higher or even in some situations decide to just walk.*" (Passenger, Discussion, 8 April 2025), elaborating on their perspective.

It was further highlighted that although in the moment passengers may be content with drivers speeding due to their extensive waiting time or tardiness, this has significant effects on road safety.

Scope

<u>Must have</u>	<u>Should have</u>
<ol style="list-style-type: none">1. Display driver's current work status<ul style="list-style-type: none">• Display on duty• Display off duty2. Estimated arrival times,<ul style="list-style-type: none">• The ETA is shown for the current location of a passenger3. Show if there is a delay<ul style="list-style-type: none">• If the expected time frame of arrival is exceeded, it is flagged as delayed• A delay is indicated4. Display Price of trip<ul style="list-style-type: none">• Prices of trip to the main stop point are displayed• The price of trip is adjusted based on destination change5. Display Routes	<ol style="list-style-type: none">1. Notify drivers to adjust their work status accordingly<ul style="list-style-type: none">• Notifications are sent to drivers prompting them to change their work status• Criteria to trigger notification• Drivers manually update status

<ul style="list-style-type: none">• The various routes are displayed• The main stopping point to be highlighted• Passenger can set their stop to get ETA <p>6. Real time location of minibus</p> <ul style="list-style-type: none">• A full display of the map• Real time location is shown on the map• An icon is used to display the bus movement	
<p><u>Could have</u></p> <ol style="list-style-type: none">1. Passenger bus request and driver response<ul style="list-style-type: none">• Driver receives incoming request• Driver can accept/ reject the request• Passenger is notified of driver's response2. Display number of available seats<ul style="list-style-type: none">• Driver can manually adjust seat count3. Automatically adjust driver's work status based on their location<ul style="list-style-type: none">• Location trigger set to adjust work status• Driver can override the decision• The change is reflected across the application	<p><u>Won't have</u></p> <ol style="list-style-type: none">1. Payment through application<ul style="list-style-type: none">• Transaction status shown• Digital receipt• Ensure that data is secure

Table 1: Project Scope (Mathurin, 2025)

Changes to the Scope

Initially, the work status was to be adjusted by the drivers themselves, but the application would not require their input, as the published version of the application would need to be developed in collaboration with the transport department, which would be responsible for overseeing all matters related to the minibuses.

Boundaries of solutions

Given the problem being addressed, making payments through the application would provide convenience to both drivers and passengers. However, due to the complexity of this feature, the focus on improving skills throughout development, and the time constraint, this feature will not be included in the application.

Ideally, the implementation of GPS location would be achieved through a device placed on the minibuses, which would involve additional cost. To ensure the application is viable before such expenses are made, the GPS location on the driver's phone will be utilised.

Evaluation criteria for solution (or predictions)

Scope (M or S or C, not W) for development project OR hypothesis for research project	Evaluation criterion
Display driver's current work status (On duty/Off duty)	Display on duty- verify visibility in response to changes Display off duty- verify visibility in response to changes
Estimated arrival times	The ETA is shown for the current location of a passenger - Verify the accuracy in real time at multiple locations
Show if there is a delay	If the expected time frame of arrival is exceeded, it is flagged a delayed - The estimate should not be offset by more than 5 minutes. Delay indicated – There is a visual indicator to show the delay
Price of trip	Prices of trip to the main stop point is displayed - Price is available for starting a trip The price of trip is adjusted based on destination change – The price is adjusted accurately for multiple locations.
Routes	The various routes are display – The route specified for the current application is visible. The main stopping point to be highlighted – An indicator is visible to show the main stopping point Passenger can set their stop to get ETA – Use different destinations to ensure that the ETA is accurate.
Real time location of minibus through the GPS of driver's smartphones	A full display of the map – A map with the visible indicators and is interactable. Real time location is shown on the map – The location of the passenger is shown with a viable indicator

	An icon is used to display the bus movement – a bus icon is visible and moves in real time.
Notify drivers to adjust their current work status accordingly	Notifications are sent to drivers prompting them to change their work status – When a criterion is met a driver get the notification Criteria to trigger notification – The expected time of shift end and if they go off route. Drivers manually update status – The changes made by the driver is visible through the work status indicator utilized

Table 2:Evaluation Criteria (Mathurin,2025)

Approach to solution of criteria

Display driver's current work status (On duty/Off duty)

As previously mentioned, the published version of the application would be done in collaboration with the transport department, and they would have all the information about the drivers, including schedules and shift start and end times. This information would be stored in a MongoDB database, and the status would be updated based on the criteria in the database, according to the current time.

Estimated arrival times, show if there is a delay.

This would be implemented through the google maps API. Displaying the distance away in real time and the estimated arrival time to the passenger's current location.

Price of trip

The prices of the trips would be stored in a database created using mongo DB. This would hold all prices at the various pick-up and drop-off points along the route. The database would then be connected to the application using express.js & node.js API.

Routes

The routes would be stored in the MongoDB database displaying the main routes to and from the main locations as different routes. (e.g. Dennery to Castries and Castries to Dennery would be different.)

As with Castries to Dennery there are different drop-off points which cause the prices vary.

Real time location of minibus

Considering the implementation of GPS equipment is a boundary.

This feature would be implemented using Google maps API.

A full version of the application would be done in association with the Transport Department so the details of each driver would be available and able to be placed in the database.

Each driver would have a unique ID which would differentiate them from the passengers given that both would be using the same application. The app would access the GPS location of the phones utilized by the bus drivers.

The drivers and their ID would be placed in the database also showing their normal working hours which would be used to automatically turn on and off their GPS.

If the driver goes off route for more than 2 minutes, then their location would automatically be turned off.

Notify drivers to adjust their current work status accordingly.

The work status of the driver would automatically be adjusted based on their predefined work times which would have been added to the database and linked to the application.

Impact of LSEPI / EDI on solution

Act / law / guidance	Purpose	Relevance to project (if none, say why none)	Effect on project	Implementation approach
Computer Misuse Act 1990 (legislation.gov.uk, 1990)	<ul style="list-style-type: none"> Unauthorized access Malicious use of systems and data 	<ul style="list-style-type: none"> It is a software development project The GPS locations will be available to the public The routes and prices will be stored data 	<ul style="list-style-type: none"> Data available through the application can be utilized or alter for malicious purposes 	<ul style="list-style-type: none"> Ensure what is stored is secured Utilize a common API (Google maps) for real time tracking to improve security Ultimately given the time constraint all security risks may not be address so the final product may not be ready for real world use.
Copyright (GOV.UK, 2023)	<ul style="list-style-type: none"> Protect manage and enforce copyright 	<ul style="list-style-type: none"> Pondering on concepts from existing application to aid in designing the user interface Any content utilized in the application which may stimulate copyright 	<ul style="list-style-type: none"> The application could be futile until the copyright issues are solved. The originality of the project will be up for questioning 	<ul style="list-style-type: none"> Constantly ensure that my implementations are within the copyright guidance
Data Protection Act 2018 (legislation.gov.uk, 2018)	<ul style="list-style-type: none"> Protect personal information 	<ul style="list-style-type: none"> The location of the buses will derive from the GPS location of the driver's phones. 	<ul style="list-style-type: none"> Potential surveillance risks 	<ul style="list-style-type: none"> Request consent Ensure the information of is secure There will not be no way to distinct which specific bus identified to prevent surveillance risks
Design Principle FTW (Nielsen, 2013)	<ul style="list-style-type: none"> Provide guidance to create an ideal user interface 	<ul style="list-style-type: none"> The aim is to create a user interface which is simple and easy to use 	<ul style="list-style-type: none"> Some user may not be tech savvy so the lower the learning curves the better. New users can make immediate use 	<ul style="list-style-type: none"> Have someone who is not tech savvy test the user interface Consider simplicity through the design
Freedom of information request (GOV.UK, n.d)	<ul style="list-style-type: none"> Request to view public information held by public authorities 	<ul style="list-style-type: none"> None – All the information required will be obtained through my own means. 	Not applicable	Not applicable
British computer Society (BSC) (bsc The Chartered Institute for IT, 2025)	<ul style="list-style-type: none"> Ensuring ethical, responsible and professional behaviour 	<ul style="list-style-type: none"> The personal information of driver(location) Who can access the prices, and the routes information stored Considering how the public would be affected from the solution 	<ul style="list-style-type: none"> Promotes improved security, ethical consideration and high professionalism through development and the final product. 	<ul style="list-style-type: none"> Ensure that all steps take into consideration the purpose of BSC.
Equality Act 2010(legislation.gov.uk, 2010)	<ul style="list-style-type: none"> Ensuring equality, respect and fairness 	<ul style="list-style-type: none"> The benefits of the solution to one stakeholder should not affect another negatively 	<ul style="list-style-type: none"> Stakeholders may consider the solution to be biased towards the passenger. 	<ul style="list-style-type: none"> Ensure a balance is still considered to reduce bias towards passenger and considering the drivers and the public
Health and safety at work act 1974(legislation.gov.uk, 1974)	<ul style="list-style-type: none"> Promotes safety and reduces risks for workers and the public. 	<ul style="list-style-type: none"> Bus drivers rushing due to passenger pressure → road safety concern. App may encourage safer behaviors if delays and bus locations are transparent. Bus drivers rushing due to passenger pressure → road safety concern. App may encourage safer behaviors if delays and bus locations are transparent. 	<ul style="list-style-type: none"> The system should avoid incentivizing dangerous driving (e.g., competing to be first). Must not add distraction to drivers (no complex input during driving). 	<ul style="list-style-type: none"> Only allow passenger-facing features; driver inputs should be minimal or automated. Highlight safety over competition in system design.

Meet the requirements of equality and accessibility regulation (GOV.UK, 2024)	<ul style="list-style-type: none">Ensures services are accessible to all, including those with disabilities	<ul style="list-style-type: none">Some users may have low digital literacy or disabilities (visual impairment, colour blindness, mobility issue)	<ul style="list-style-type: none">If not designed inclusively, app could exclude part of the population	<ul style="list-style-type: none">Follow WCAG accessibility guidelines.Use high contrast colour schemes and simple navigation.Consider screen reader compatibility.
Consumer Rights Act 2015(legislation.gov.uk, 2015)	<ul style="list-style-type: none">Protect consumers when using goods or services	<ul style="list-style-type: none">Passengers relying on pricing info must not be misled.Trip cost data must be accurate and transparent.	<ul style="list-style-type: none">Misrepresentation of prices could undermine trust.	<ul style="list-style-type: none">Ensure that trip price calculation is correct.Provide disclaimers if prices may vary due to external factors.
Environmental Principles Policy Statement (legislation.gov.uk, 2021)	<ul style="list-style-type: none">Encourages sustainable and responsible technology development.	<ul style="list-style-type: none">Your app may reduce waiting times and unnecessary bus usage, lowering fuel consumption and emissions.	<ul style="list-style-type: none">Positive impact on community transport efficiency and environment.	<ul style="list-style-type: none">Highlight potential benefits in reducing traffic congestion and supporting sustainable transport planning.

Table 3: LSEPI/EDI impact (Mathurin. 2025)

[The completed ethics form is in the appendices](#)

[Consent Form](#)

[Participation Sheet](#)

LIFE CYCLE

Life cycle	Pros in project	Cons in project	Useful in project?	Accept / reject / why
Iterative	<ul style="list-style-type: none">• Flexibility• Detection of errors• Promotes prototyping	<ul style="list-style-type: none">• Difficult to predict a time frame.• Could lead to time wasting	Yes – it's flexibility would be beneficial and prototyping.	The development is under strict time frame where skills must be developed throughout. Having to iterate through stages would lead to wasted time.
Waterfall	<ul style="list-style-type: none">• Promotes meeting of deadlines• Organized planning	<ul style="list-style-type: none">• No flexibility• Difficult to revert to previous stages	No- with the lack of experience more flexibility is required	Skills are continuously being developed so being flexible to change current focus would be a good asset.
Incremental	<ul style="list-style-type: none">• Promotes prototyping• Detection of errors• Flexibility	<ul style="list-style-type: none">• Potential issues when integrating components• Requires more testing	Yes – refining the project continuously while still making progress.	This would ensure that components are being developed adequately. If the deadline creeps up, then a lot of working functionalities would have already been developed
Agile	<ul style="list-style-type: none">• Continuous testing• Flexibility• Consideration of user reviews• Faster	<ul style="list-style-type: none">• Has a very strict time frame• Require team collaboration	No – require a team effort and more strict setting.	This project will be conducted by one individual so such a strict time frame would not be possible.

Table 4:Life cycle comparison (Mathurin, 2025)

Chosen life cycle

From the comparison of the project life cycles regarding the impact each would have on this project, the incremental life cycle was chosen. It permits continuous progress throughout the project in an organized manner, also ensuring each component implemented works adequately. The possibility of developing a prototype would be a great initiative toward the success of the project.

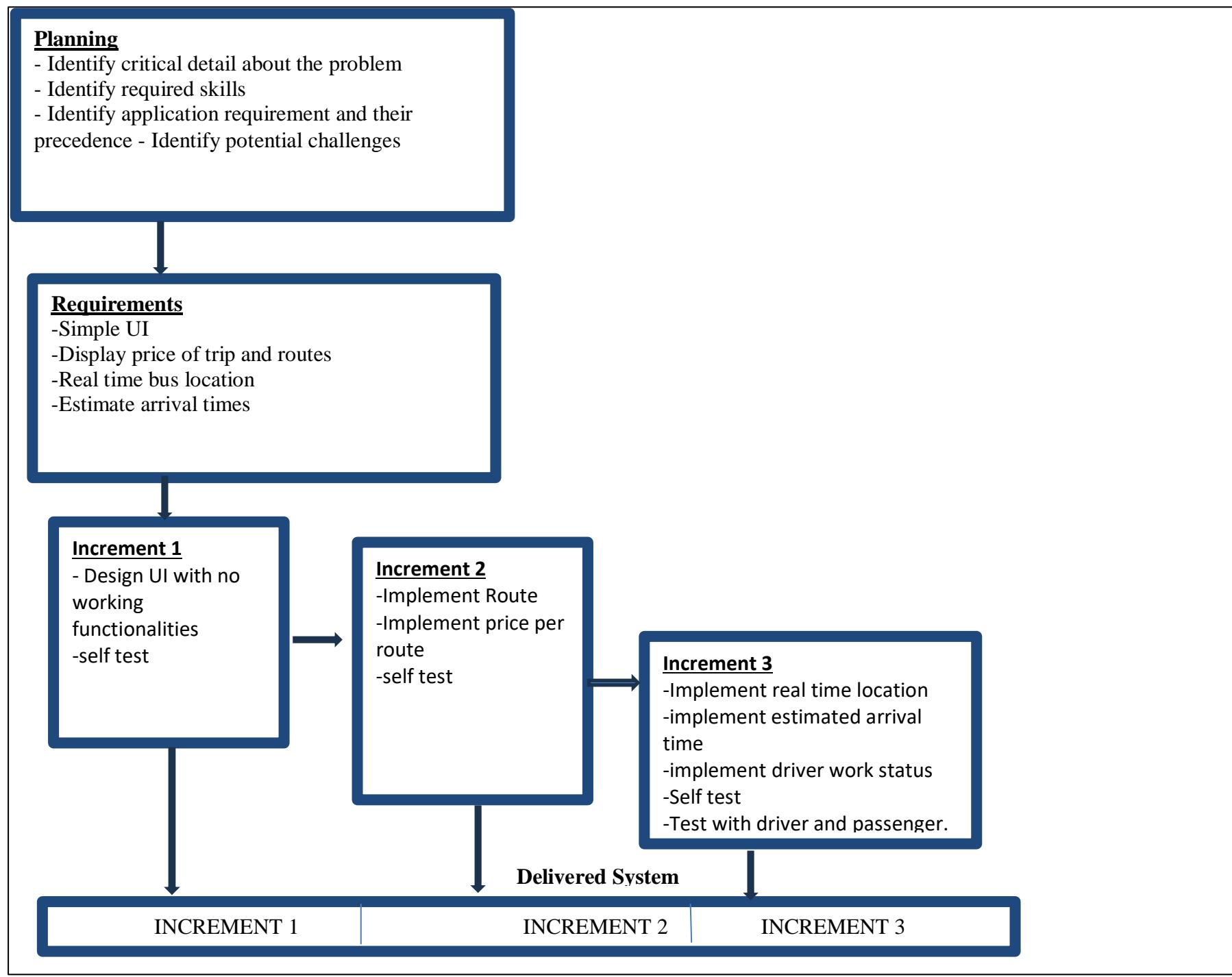


Figure 1: Incremental Lifecycle diagram (Mathurin,2025)

Planning

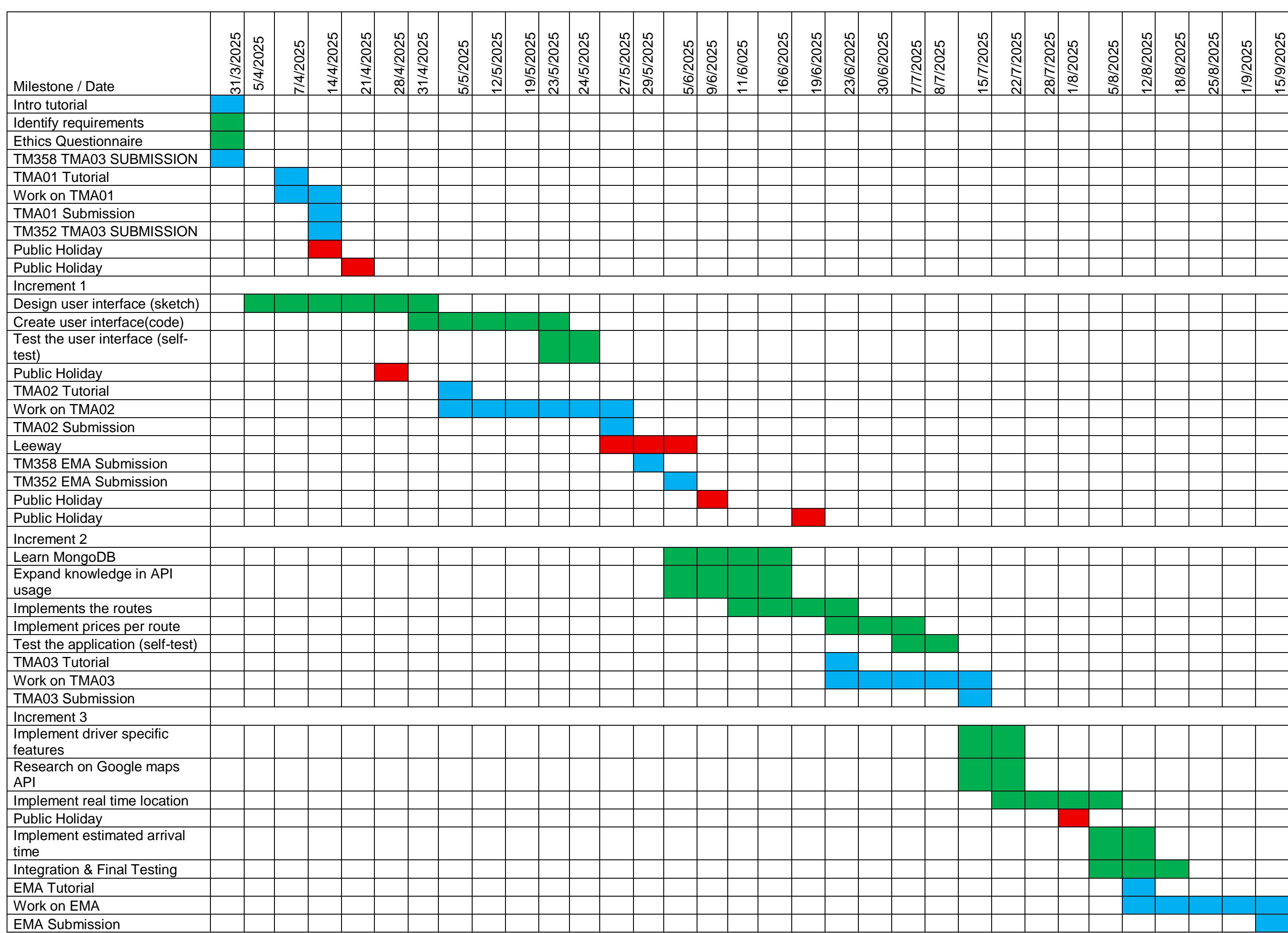


Figure 2: Project plan (Mathurin, 2025)

<u>KEY</u>	
Project work	Green
Assignments	Blue
Holidays & Leeway	Red

Previous plan

Discussions

Before Increment 1

A one-week extension was required (April 15th – 22nd) due to hardware failure, which caused the loss of all progress made on the report while adapting to new software. All other activities prior to this were completed in a timely fashion. This extension caused a slight delay, but the project was realigned with the original plan.

Increment 1

For Increment 1, designing the user interface using Sketch and then through the development software was completed successfully and on schedule allowing progress to move on to implementation in the application. Similarly, coding the user interface was completed on time. However, testing was delayed by a day, being done on the 25th rather than the 24th of April due to pressure from other module commitments. This pressure also contributed to requesting an extension during that period (May 27th – June 3rd)

Increment 2

Increment 2 was expected to start on June 5th but commenced on June 7th, as a brief rest was needed after submitting work for another module the day before. Confidence with MongoDB was achieved a day earlier than expected, based on the understanding that skills would be further reinforced through practical work with the software. Expanding API knowledge was originally planned to start simultaneously with MongoDB, but this did not occur due to full focus being placed on MongoDB. API work began afterward on June 8th and continued until the planned stop date of June 16th.

Instrument 3

This increment was originally scheduled to start on July 15th but was delayed until July 24th due to delays in previous increments. During this stage, it became clear that completing all planned features was unlikely, so priority was given to implementing the most essential functionality, particularly real-time location simulation and route visualization.

Resources, Skills and Risks

Resource	When needed	Why needed	How to obtain / already got it	Problems if unavailable
Passengers	<ul style="list-style-type: none"> Planning Final testing 	<ul style="list-style-type: none"> To identify requirement and to further understand the problem Testing 	<ul style="list-style-type: none"> I know regular passengers 	<ul style="list-style-type: none"> Produce may not satisfy the need of the users Development plan may have poor consideration
Minibus Driver	<ul style="list-style-type: none"> Planning Final testing 	<ul style="list-style-type: none"> To identify requirement and to further understand the problem Testing 	<ul style="list-style-type: none"> I know bus drivers 	<ul style="list-style-type: none"> Produce may not satisfy the need of the users Development plan may have poor consideration
Visual Studio	<ul style="list-style-type: none"> Create user interface (code) Implement routes and prices Implement driver specific features Implement real time location Implement estimated arrival time All Testing 	<ul style="list-style-type: none"> IDE required to develop applications code 	<ul style="list-style-type: none"> Official Website- (Visual Studio Code, n.d) 	<ul style="list-style-type: none"> I would have to learn another IDE
React Native Directory	<ul style="list-style-type: none"> Create user interface (code) Implement routes and prices Implement driver specific features Implement real time location Implement estimated arrival time • 	<ul style="list-style-type: none"> To acquire libraries and tools 	<ul style="list-style-type: none"> Official Website- (React Native Directory, n.d) 	<ul style="list-style-type: none"> The entire project may need to be developed using a less familiar framework
Google maps API	<ul style="list-style-type: none"> Implement real time location Implement estimated arrival time 	<ul style="list-style-type: none"> To implement real time location 	<ul style="list-style-type: none"> Official Website- (Google, n.d) 	<ul style="list-style-type: none"> Would have to find another software which may have a harder learning curve
Mongo DB	<ul style="list-style-type: none"> Learn MongoDB Implement routes and prices per route 	<ul style="list-style-type: none"> To store data (prices, routes) 	<ul style="list-style-type: none"> Official Website- (MongoDB Inc, 2025a) 	<ul style="list-style-type: none"> Would have to search for an alternative Would not be able to store data
Node.js	<ul style="list-style-type: none"> Implement routes and prices per route Implement driver specific features 	<ul style="list-style-type: none"> To implement back-end resources 	<ul style="list-style-type: none"> Official Website- (OpenJS Foundation, 2024) 	<ul style="list-style-type: none"> An alternative would have to be found leading to time being wasted
Express.js	<ul style="list-style-type: none"> Implement routes and prices per route Implement driver specific features 	<ul style="list-style-type: none"> To implement back-end resources (alongside Node.js) 	<ul style="list-style-type: none"> Official Website- (OpenJS Foundation, 2025) 	<ul style="list-style-type: none"> An alternative would have to be found leading to time being wasted
Expo cli	<ul style="list-style-type: none"> Throughout development 	<ul style="list-style-type: none"> Allows the application to run swiftly without the need for numerous other software 	<ul style="list-style-type: none"> Official Website – (Expo Docs, n.d) 	<ul style="list-style-type: none"> I would have to use react native cli which would be more complicated to use.
Android Studio	<ul style="list-style-type: none"> When coding Test the user interface (self-test) Test the application (self-test) Integration & Final Testing 	<ul style="list-style-type: none"> For guidance throughout development Testing 	<ul style="list-style-type: none"> Official Website – (Developers, n.d) 	<ul style="list-style-type: none"> I would have to use another method of testing the application which may not provide a proper emulator like this one.
Android device	<ul style="list-style-type: none"> Test the user interface (self-test) Test the application (self-test) Integration & Final Testing 	<ul style="list-style-type: none"> Testing 	<ul style="list-style-type: none"> A Personal device 	<ul style="list-style-type: none"> I may not be able to have a driver or passenger full test the application
MongoDB atlas	<ul style="list-style-type: none"> Implement routes, prices per route and driver info 	<ul style="list-style-type: none"> To store data (prices, routes, driver info) 	<ul style="list-style-type: none"> Official Website – (MongoDB Inc, 2025a) 	<ul style="list-style-type: none"> I would have to use the GUI for MongoDB.
Postman	<ul style="list-style-type: none"> Implement routes and prices per route 	<ul style="list-style-type: none"> To test the backend functionality 	<ul style="list-style-type: none"> Official Website – (Postman, 2025) 	<ul style="list-style-type: none"> I would proceed with development an potentially encounter issues.
ChatGPT	<ul style="list-style-type: none"> Throughout Development 	<ul style="list-style-type: none"> Generate custom images Provided guidance in learning new tools To clarify coding issues 	<ul style="list-style-type: none"> OpenAI (2025) 	<ul style="list-style-type: none"> Slower learning process Slower problem solving Research and debugging time will be increased

Table 5: Resources (Mathurin, 2025)

[Previous Resource Table](#)**Key**

Original
Newly added

Reasoning for previously omitting the newly added resources

Expo CLI – A decision between this and React Native CLI was initially pending, but after further research, Expo CLI was determined to be the best option.

Android Studio – The initial plan was to test the application using the Google Chrome developer feature as done in TM352. Further research indicated that using the Android Studio emulator would be more effective for testing and realism, given the limitations of the Google Chrome developer feature.

Android Device – Testing the application using the Google Chrome developer feature was originally considered.

MongoDB Atlas – It was discovered that this is required to use MongoDB in a shell interface, which is preferred.

Postman – This was identified as a convenient tool to verify that successful progress is being made with the backend.

ChatGPT – Other means were considered to achieve the benefit of this resource, but it was identified as the most efficient method to achieve various needs.

Skill	When needed	Why needed	How to obtain / already got it	Problems if unavailable
Project Management	<ul style="list-style-type: none"> Throughout the project 	<ul style="list-style-type: none"> To effectively plan for the project within the available time frames Ability to make adjustment to meet deadlines Lowers possibility/impact of risks 	TM254	<ul style="list-style-type: none"> Disorganization especially given the projects complexity Missing deadlines Poor time management Failure of entire project
Referencing	<ul style="list-style-type: none"> When writing up reports 	<ul style="list-style-type: none"> To ensure no plagiarism in my work. 	Open university referencing content	<ul style="list-style-type: none"> At risk of plagiarism.
Reporting Writing	<ul style="list-style-type: none"> Working on all Assignments 	<ul style="list-style-type: none"> Documentation Improves clarity Improves organization 	Open University Essay and report writing skills course	<ul style="list-style-type: none"> Inability to show how practical work was completed effectively Wasting time documenting inaccurate information
React Native	<ul style="list-style-type: none"> Create user interface (code) Implement routes and prices Implement driver specific features Implement real time location Implement estimated arrival time 	<ul style="list-style-type: none"> Offer cross platform functionality Promote efficient development 	TM352	<ul style="list-style-type: none"> Application could be tailored to only specific devices Inability to meat scope of project
JavaScript	<ul style="list-style-type: none"> Implement routes and prices Implement driver specific features Implement real time location Implement estimated arrival time 	<ul style="list-style-type: none"> To implement functionality 	TM352	<ul style="list-style-type: none"> Application wouldn't be useable
TypeScript	<ul style="list-style-type: none"> Create user interface (code) Implement routes and prices Implement driver specific features Implement real time location Implement estimated arrival times 	Developing the application	TM352	<ul style="list-style-type: none"> Application wouldn't be useable
Using API	<ul style="list-style-type: none"> Implement routes and prices per route Implement driver specific features Implement real time location Implement estimated arrival time Expand knowledge in API usage 	<ul style="list-style-type: none"> To implement real time location Connection to backend resources 	TM352 / Further research	<ul style="list-style-type: none"> A crucial feature would not be implemented
Using node.js	<ul style="list-style-type: none"> Implement routes and prices per route Implement driver specific features 	<ul style="list-style-type: none"> Create backend resources 	TM352/Further research	<ul style="list-style-type: none"> Failure to complement frontend with complete functionality
MongoDB	<ul style="list-style-type: none"> Implement routes and prices per route Implement driver specific features Learn MongoDB 	<ul style="list-style-type: none"> Database (prices & routes) 	Online resources	<ul style="list-style-type: none"> Some key features would be neglected
Express.js	<ul style="list-style-type: none"> Implement routes and prices per route Implement driver specific features 	<ul style="list-style-type: none"> Implement backend resources (alongside node. js) 	TM352/Further research	<ul style="list-style-type: none"> Would be more difficult and time-consuming dealing with the backend resources
debugging	<ul style="list-style-type: none"> Create user interface (code) Implement routes and prices Implement driver specific features Implement real time location Implement estimated arrival time 	<ul style="list-style-type: none"> When coding bugs and issues are inevitable, so I need to know how to address them 	TM352/Further research	<ul style="list-style-type: none"> I could waste a lot of time if bug come up and I don't know how to address them.

Table 6: Skills (Mathurin, 2025)

[Previous skill table](#)**Key**

Original
Newly added

Reasoning for previously omitting the newly added skill

Debugging - After encountering numerous issues during coding and having to fix them by tracing the source of the problems, it became evident that debugging is not only a part of coding but a skill of its own, as improper debugging can allow major issues to persist during development.

Risk	Priority	Potential impact	How to reduce if occurs	Accept / avoid / reject / why / how
Working on two other modules until June 5 th	Medium	Medium	<ul style="list-style-type: none"> Ask for an extension 	Avoid Why- This would influence all concurrent models and reaching the project deadline. How – Effectively manage my time, with a priority base system.
Hardware issues	High	Medium	<ul style="list-style-type: none"> Ensure information is backup 	Reject Why - Lose of any project related information would affect the deadline drastically How – Ensure that information is backed up effectively.
Unable to get/cooperate with bus driver	Medium	Medium	<ul style="list-style-type: none"> Self-testing can be done 	Avoid Why- Although it is possible to self-test, feedback from an actual driver would be ideal. How – Find a cooperative bus driver in the early stages of the project
Unable to get/cooperate with passenger	Medium	Medium	<ul style="list-style-type: none"> Self-testing can be done 	Avoid Why- Although it is possible to self-test, feedback from an actual passenger would be ideal. How – Find a cooperative passenger in the early stages of the project
Unable to find the resources required to develop skills	High	High	<ul style="list-style-type: none"> Adjust the scope of the project to fit the skill available. 	Avoid Why – To produce the intended application within the time frame the identified skills are essential How – Identify the sources where each skill can be further developed at the very early stages
Personal Challenges	High	Medium	<ul style="list-style-type: none"> Ask for an extension Revert to time management skills 	Avoid Why – These personal challenges are not frequent but are possible so it is essential to understand what would be done if it happens. How – Ensure that I am within the recommended time utilizing my time management skill where required.
Final product does not function adequately	High	High	<ul style="list-style-type: none"> Write about the potential changes I would have made to solve the issues 	Avoid Why - It would be essential to show that the final product is working to display the technical aspects of my work How – Ensure that each increment is functioning adequately through testing.
Inability to effectively improve my debugging skills	Medium	High	<ul style="list-style-type: none"> Change my approach to the solution, using a simpler method if possible 	Avoid Why – I would have ample time to improve my debugging skill, and I know that I am a fast learner. How – Ensure that I set time aside to improve said skills.
Revisit/Backtrack	High	High	<ul style="list-style-type: none"> Make adjustment to my scope. Set a time to make the adjustments if available 	Avoid Why – having to back track would put a dent in the projects progress especially since I'm adhering to an incremental life cycle. How – Ensure that everything up to par before moving onto the next increment even if it causes a delay.
Poor user interface	High	Medium	<ul style="list-style-type: none"> Prioritized the areas which require most improvement, then modify accordingly 	Avoid Why – this application requires an interface which is simple to use give that some of the user may not be tech savvy. How – Ensure I check the credibility of the UI again a criterion.
The length of route chosen may complicate the final testing process	Low	Medium	<ul style="list-style-type: none"> The route can be changed to a shorter one which is on the same path as the one chosen 	Accept Why- Given that's where I reside it would be easier to get a cooperative driver from here How – Continue with the specified route and choose a driver from there
A software or tool I need, requires a payment	Low	Low	<ul style="list-style-type: none"> Find an alternative software 	Avoid Why – To avoid any time waiting How – ensure that the tools I identify do not require any payments.
Unavailability of ChatGPT	Medium	Medium	<ul style="list-style-type: none"> There are other options which could be utilized 	Accept Why - There are alternatives How – Identify an alternative option which could be swiftly utilized.

Table 7: Risks (Mathurin,2025)

[Previous risks table](#)**Reasoning for previously omitting the newly added risks**

The new risks outlined were identified during development with the complexity

Key

Original
Newly added

Literature Review

Alongside the TM470 course material (Open University, 2025a), guidance on selecting a suitable lifecycle model proved valuable in directing the early stages of this project. The resource highlighted the importance of considering aspects such as project duration, complexity, and the skills available. Despite being undated, its origin from the Open University gave it credibility, and it supported my decision to adopt an Incremental approach that balanced ambition with practicality.

The legal, social, ethical and professional issues guidance (Open University, 2025b) was also essential. This resource highlighted the importance of avoiding legal or ethical breaches, especially around privacy concerns for driver locations. While the material could have been more elaborate, its references provided a good foundation. Even though the publication date was not specified, it remained valid and ensured that my solution aligned with legal frameworks. Closely linked to this was the Open University's resource on embedding equality, diversity, and inclusion (Open University, 2025c). This made me consider how benefits for one stakeholder (e.g. passengers) might negatively affect others (e.g. drivers). It encouraged me to design with balance in mind, so that the project would promote fairness and inclusivity.

Alongside these academic sources, practical guidance for writing and structuring reports (Open University, 2025d) provided clarity on how to present the project. While not directly tied to software development, it ensured that documentation would meet professional standards, which is a critical component of TM470.

Design-related sources were also reviewed. Mane (2019) provided clear guidance on mobile app design, particularly in ensuring usability across both Android and iOS. Although there was a slight emphasis on iOS, the principles of good UI design were generalizable. This supported the Figma prototypes I created, ensuring that the user interface was consistent and intuitive. Similarly, Kravchenko (2025) presented a range of mobile UI testing checks. While not all applied directly to my project, the generalized criteria offered a practical framework for evaluating my interface.

Several technical libraries and frameworks were also consulted. Documentation for the React Native Vector Icons library (npm, n.d) and the React Native Bottom Sheet (Gorhom, n.d) provided detailed guidance on implementation. These sources were well-structured, regularly updated, and highly relevant, as icons and navigation elements formed a core part of the app's user interface. The Expo documentation (Expo Docs, n.d) and comparative evaluation of Expo CLI versus React Native CLI (Terekhov, 2025) also influenced the technical setup. Terekhov's discussion was particularly useful in consolidating my decision to use Expo CLI, which simplified deployment and testing during development.

For the backend, MongoDB documentation (MongoDB Inc, 2024) proved essential. It provided comprehensive, up-to-date instructions on setting up and managing the database. The clarity and frequent updates of this resource made it highly reliable.

In addition, several resources supported both backend development and user experience. The Node.js RESTful API tutorial (W3Schools, n.d.) provided a straightforward foundation for connecting the frontend with the backend. Although undated and lacking formal references, its clarity and frequent updates made it a reliable starting point for understanding CRUD operations. Closely related, the Express routing documentation (Express, n.d.) was invaluable for structuring server-side routes. As official documentation, it was authoritative and up to date, helping to ensure that the backend architecture remained scalable and maintainable.

Beyond the technical setup, design principles were further reinforced by guidance from the UX Design Institute (2024). This resource outlined seven fundamental UX design principles that could be applied to improve the usability of my application. Although framed within general UX industry standards, its provenance from a recognized training body and its recent publication made it a strong reference point for creating a user-centred design.

Shah (2022) also provided practical insight into Node.js application architecture. While based mainly on the author's personal expertise rather than external references, the structured advice was highly relevant. It offered best practices on organizing services and layers, which helped me shape a scalable and maintainable backend. Finally, Andrea (2024) supplied a practical usability testing checklist through Userbrain. Though slightly promotional, the checklist itself was objective and provided a structured process for testing the mobile app before deployment, ensuring that usability concerns were systematically addressed.

The Computer Misuse Act 1990 (legislation.gov.uk, 1990) highlighted the importance of protecting data and preventing unauthorized access or malicious use. Since this project involves storing routes, prices, and public GPS locations of drivers, the potential for misuse was significant. This prompted me to implement measures to secure stored data, use a trusted API (Google Maps) for real-time tracking, and acknowledge that, given time constraints, the final product might not fully address all security risks for real-world deployment.

The Copyright guidance (GOV.UK, 2023) reminded me to respect intellectual property when considering concepts from existing applications for the UI design. This prompted careful implementation to avoid copyright infringement and maintain originality, ensuring the project's outputs were legally defensible and ethically sound.

The Data Protection Act 2018 (legislation.gov.uk, 2018) stressed the need to protect personal information, which is particularly relevant given the app collects GPS locations from drivers. To address this, consent was requested, and information was anonymized so individual vehicles could not be directly identified, reducing surveillance risks and promoting privacy.

The Design Principles by Nielsen (2013) provided practical guidance on creating an intuitive and user-friendly interface. Since some users may not be tech-savvy, these principles

prompted testing with non-technical users and the simplification of interface elements to minimize learning curves and improve immediate usability.

The Freedom of Information guidance (GOV.UK, n.d.) was noted, but it had no direct relevance, as all required information for the project was sourced independently. No action was necessary in response to this guidance.

The British Computer Society (BSC) Code of Conduct (BSC, 2025) emphasized ethical, responsible, and professional behavior. This prompted careful consideration of how driver locations, pricing, and route data were accessed and used, ensuring that security, ethics, and professionalism were embedded in both development and the final product.

The Equality Act 2010 (legislation.gov.uk, 2010) reminded me to design a solution that benefits all stakeholders fairly. This prompted careful balancing of features so that passengers, drivers, and the public were not disadvantaged, and potential bias towards passengers was minimized.

The Health and Safety at Work Act 1974 (legislation.gov.uk, 1974) highlighted risks associated with driver behavior under passenger pressure. This prompted designing the system to avoid incentivizing unsafe driving, keeping driver inputs minimal, automating features where possible, and emphasizing safety over speed or competition.

The Equality and Accessibility Regulations (GOV.UK, 2024) highlighted the need for inclusivity. This prompted design choices that considered low digital literacy and disabilities, including high-contrast color schemes, simple navigation, screen reader compatibility, and adherence to WCAG accessibility standards.

The Consumer Rights Act 2015 (legislation.gov.uk, 2015) emphasized accurate and transparent pricing. This prompted careful calculation of trip costs, clear disclaimers for potential variations, and measures to ensure that passengers could trust the information presented.

Finally, the Environmental Principles Policy Statement (legislation.gov.uk, 2021) encouraged sustainable development. This prompted highlighting how the app could reduce unnecessary bus trips and waiting times, improve transport efficiency, and contribute positively to community sustainability.

Overall, the sources consulted included academic texts, Open University materials, industry documentation, practitioner blogs, and relevant legislation and guidance. Together, they provided theoretical grounding (e.g., lifecycle models, ethics, EDI, legal compliance), design guidance (UI/UX and testing), and practical technical support (libraries, APIs, databases). While some sources lacked publication dates, their provenance and relevance ensured they remained valuable. Collectively, these resources and regulatory frameworks prompted key decisions throughout the project, shaping the design, development, and evaluation of a solution that was practical, ethical, and inclusive.

**Danell Mathurin
PI: Z5162947**

TM470 EMA

Due date: 15/09/25

Work Done

Prior to starting any work on the actual project, the features required to produce an adequate application that considers both passengers and bus drivers were identified. These features can be seen in the Moscow table (Table 1).

The skills required to complete the project are summarized in Table 6, highlighting the current level of proficiency and the plan for further developing those skills.

Increment 1

Design user interface (sketch)

The user interface was first sketched using pen and paper, as this was the simplest approach at the time given the lack of familiarity with designer software. While this method was basic it ensured that progress could still be made without being delayed by the need to learn new tools. Showing the use of accessible methods to get going on the project even if the output lacks precision and professional quality provided by specialized software.

This initial sketch contained five screens including those utilized for logging in, utilizing unique name KOTE, which mean “where to” and a light blue themes colour which resonates with the colours and culture of St. Lucia

Considering Data protection Act 2018 (legislation.gov.uk, 2018) and Computer misuse Act 1990 (legislation.gov.uk, 1990), the login screen would require users to input their credentials. Since both drivers and passengers would use the same app, this approach reduced the need for extra development and made the application more accessible and transparent to both stakeholders. This also aligns with the ethical principles of British computer Society (bsc The Chartered Institute for IT, 2025) in ensuring responsible handling of information.

Initial sketch Part 1

The first screen when the app is opened which asked which user is there so that they could navigate to the respective login screen.

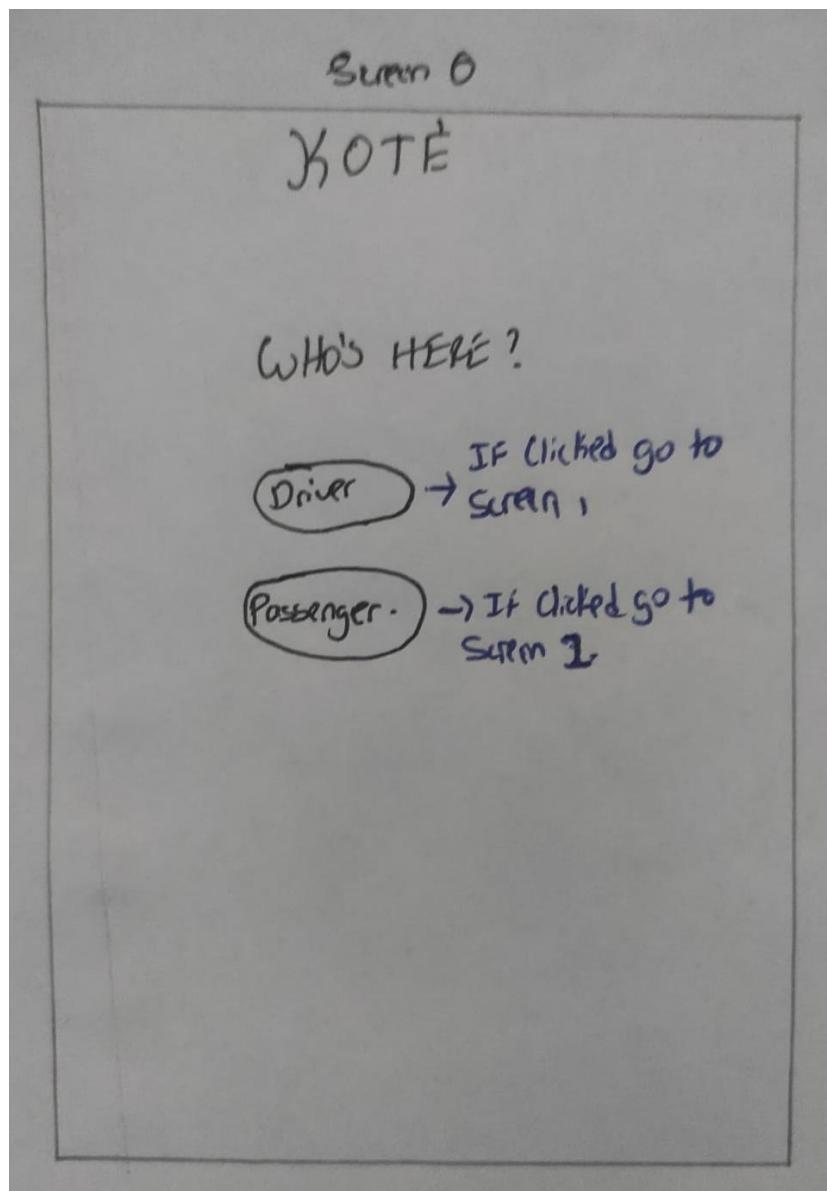


Figure 3:Initial sketch Part 1 (Mathurin, 2025)

Initial sketch Part 2

This was the projected layout for both login screens, which were similar promoting the concept of reusing code.

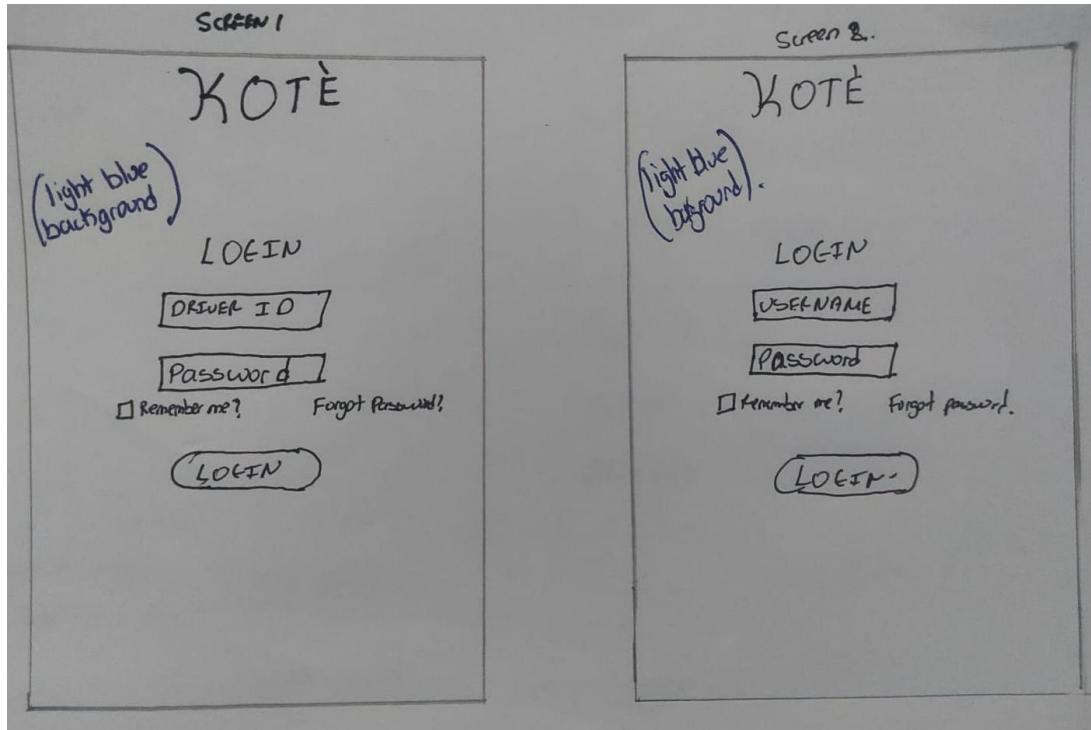


Figure 4: Initial sketch Part 2 (Mathurin, 2025)

Initial sketch Part 3

The final two screens were designed to display the available routes and provide details for each route, with Google Maps always shown in the background.

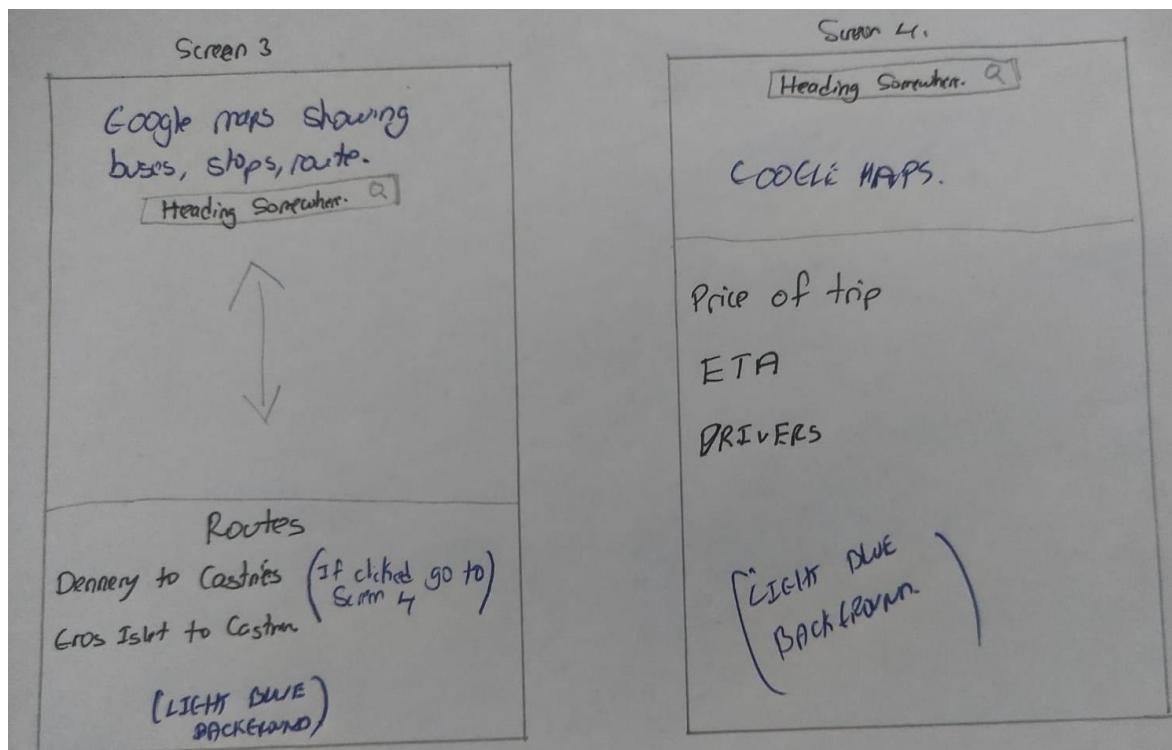


Figure 5: Initial Sketch Part 3 (Mathurin, 2025)

Further research on app development was conducted through a YouTube video, which demonstrated the practice of developing an app, demonstrating that professional workflows often involve creating a software design prior to any coding (JavaScript Mastery, 2025, 40:18). This highlighted the importance of digital prototyping as a step between sketching and implementation, ensuring that usability and layout decisions are addressed before development begins.

After obtaining basic guidance on how to use Figma through ChatGPT (OpenAI ChatGPT, 2025a), the initial paper sketches were digitalized as shown in figures one and two. The decision was made to use Figma because it is free, widely used in industry and provided tools for precise alignment, component reuse and collaborative edition. This transition to digital prototyping allowed for a clearer visualization of the interface and helped identify areas for improvement that were not obvious in the paper sketch.

Bus images used as icons were created using ChatGPT (OpenAI ChatGPT, 2025b). The decision to use AI-generated icons was inspired by its efficiency and accessibility, as it allowed unique visual assets to be produced quickly without relying on copyrighted material (GOV.UK, 2023) or external design resources. With this method refinements were highlighted such as the positioning of components and the choice of colour scheme which were less apparent in the earlier paper sketch.

A major design change was the removal of the login page, as it was determined to be unnecessary for the intended functionality of the application. This decision was guided by the principle of user-centered design and minimalism, where reducing unnecessary steps helps create a smoother experience for the users (Nielsen, 2013). Omitting the login page also reduced development complexity, allowing greater focus on features that directly benefit the users, such as real time tracking, and route information.

Figma Design Part 1

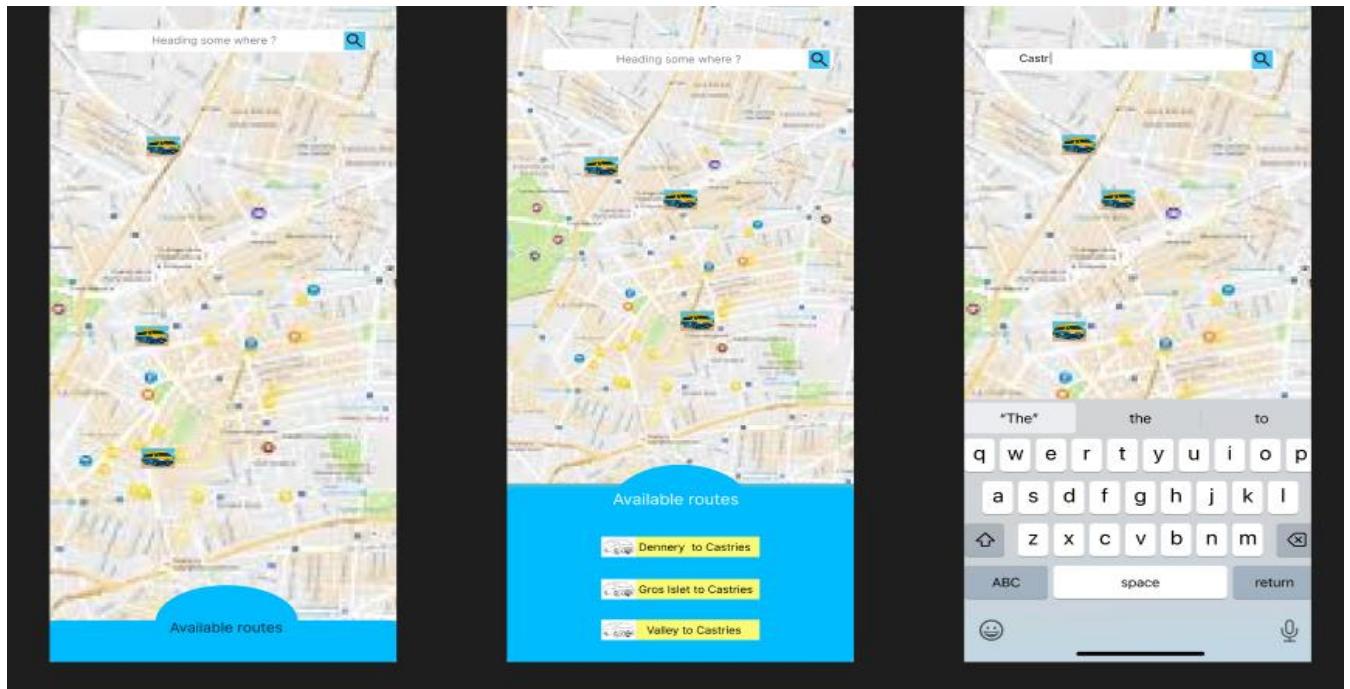


Figure 6:Figma Design Part 1 (Mathurin, 2025)

Figma Design Part 2

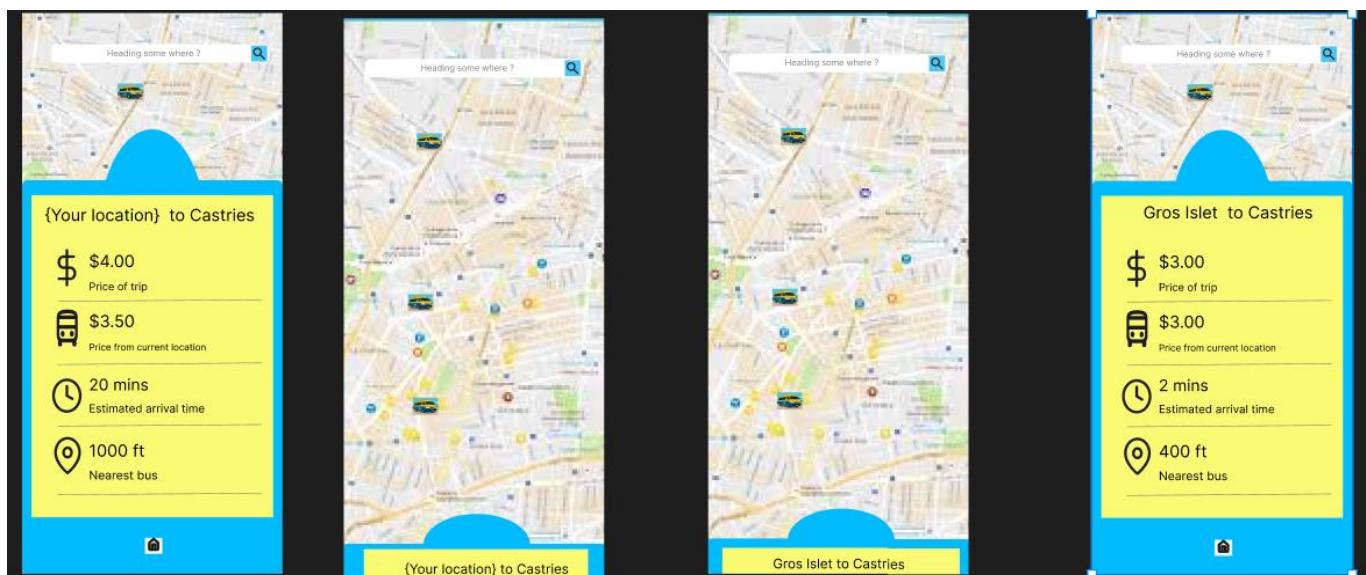


Figure 7: Figma Design Part 2 (Mathurin. 2025)

Creating the user interface (code)

From the previously referenced YouTube video (JavaScript Mastery, 2025, 19:30), it was observed that Expo CLI was used. After carrying out further research (Terekhov, 2025), the reasoning behind this choice was clarified, concluding that Expo CLI would be the better option compared to React Native CLI, which was mainly used in the TM352 module. Expo CLI is more beginner-friendly in terms of setup and makes it simpler to test the application on either an emulator or a physical device. This approach is considered more suitable, as in the TM352 module, work is generally done on pre-existing code, with applications rarely created from scratch, and testing is typically conducted through the Chrome developer tools feature.

Set up application

The first step was to create the application in Visual Studio using instructions from the Expo documentation, which also provided guidance on how to connect the application to the emulator (Expo Docs, n.d), ensuring reliability minimizing configuration issues.

Once the application was set up, the main components required for the interface were identified based on the Figma design. These included the search bar, a custom button for the routes, and the bottom sheet. The bottom sheet was designated for screen changes, while the map remained visible in the background. This layout provides a key usability feature, allowing real-time tracking to be readily available from the moment the application is accessed, ensuring users can obtain essential information quickly. Each decision considered usability and accessibility reflection Equality Act 2010 (legislation.gov.uk, 2010) and meet the requirements of equality and accessibility regulation (GOV.UK, 2024).

Create search bar.

The search bar component was created based on the Figma design, ensuring consistency between the prototype and the actual application. The icon for the search button was implemented using the library react-native-vector-icons/Ionicons (Obenauer, 2024), chosen for its wide range of ready-to-use icons and ease of integration into React Native projects. This approach allowed for a visually accurate and functional search feature.

```
    return (
      <SafeAreaView style={styles.container}>

        //Text input
        <TextInput style={styles.input}
          value={search}
          onChangeText={updateSearch}
          placeholder={placeholder}
          autoCapitalize="words"
          autoCorrect={false}

        />

        // Create button, With search Icon
        <TouchableOpacity style={styles.button} onPress={updateSearch}>
          <Icon name="search" size={24} color="" />
        </TouchableOpacity>

      </SafeAreaView>
    )
  }
```

Figure 8: Code Snippet from search bar (Mathurin,2025)

Implementing Google Maps place holder

A placeholder for the Google Maps background was added using the ImageBackground component, while also ensuring that the map remained visible transparently through the status bar by adjusting the background colour parameter of the StatusBar component. This approach provided a better layout, allowing for more screen space for the map, particularly on smaller devices.

```
<SafeAreaProvider>
  <ImageBackground style={styles.image} source={require('@/assets/images/GOOGLE MAPS.png')} resizeMode="cover">
    <StatusBar translucent backgroundColor="transparent" />
  <SafeAreaView style={styles.container} >
```

Figure 9:Code snippet for Google maps place holder (Mathurin, 2025)

Create bottom sheet (home)

The bottom sheet, which displays the available routes, was implemented using the gorhom/bottom-sheet library (Mo Gorhom, n.d), relying heavily on the official documentation. This choice was made after an attempt to use an alternative library, React Native Reanimated Bottom Sheet, proved unsuccessful.

The snap points of the bottom sheet were configured according to the design: when the application is initially opened, the bottom sheet remains closed, and it can then be opened to approximately half the screen to display the available routes. This aligns with the design principle of keeping the main map visible as much as possible supporting user interaction and access to relevant route information.

```
// Ref for bottom sheet
const bottomSheetRef = React.useRef<BottomSheet>(null);

// Snappoints
let index = 0;
const snapPoints = React.useMemo(() => ['20%', '85%'], []);

<BottomSheet
  ref={bottomSheetRef}
  onChange={handleSheetChanges}
  snapPoints={snapPoints}
  index={0}
  backgroundStyle={{backgroundColor: '#00BCFF'}}
  handleIndicatorStyle={{backgroundColor: '#545454', padding: 1}}
>
```

Figure 10: Code snippet from bottom sheet(home) (Mathurin, 2025)

Create custom button

To create this icon, the bus icon generated through ChatGPT (OpenAI ChatGPT, 2025b) was imported and added to a TouchableOpacity component for the button. This simple approach allowed for customization, permitting the easy integration of the icon with interactive features

```
<View>
  <TouchableOpacity onPress={onPress}
    style={styles.button}
  >

    <View>
      <Image
        style={styles.smallLogo}
        source={require('@/assets/images/bus_drawing.png')}
      />
    </View>

    <View>
      <Text style={styles.buttonText}> {title} </Text>
    </View>

  </TouchableOpacity>

</View>
```

Figure 11: Code snippet from custom button (Mathurin, 2025)

Addin custom button to bottom sheet(home)

For demonstrative purposes, three buttons were added (representing three routes), although only one is fully functional. This approach allowed the user interface to be designed realistically with multiple routes while leaving full functionality to be implemented at a later stage

```
<BottomSheetView style={styles.contentContainer}>

  <View >
    <Text style={styles.bottomSheetTitle}> Available Routes</Text>
  </View>

  <CustomButton
    title='Dendney to Castries'
    onPress={() => setStage("detail")}
  />

  <CustomButton
    title='Gros Islet to Castries'
    onPress={() => setStage("detail")}
  />

  <CustomButton
    title='Valley to Castries'
    onPress={() => ('Details')}
  />

</BottomSheetView>
```

Figure 12: Code snippet for adding button to bottom sheet (Mathurin, 2025)

Create bottom sheet (details)

This bottom sheet displays the details for a specific route, meaning it is navigated to from the home bottom sheet and requires a back button. The code for the home bottom sheet was reused, with the addition of the back button and placeholders for the route details. In the Figma design, the back button was placed at the bottom of the bottom sheet however, this was considered a poor choice due to potential usability issues. Therefore, it was planned to be positioned at the top-left corner of the bottom sheet, improving usability and allowing users can easily return to the previous screen without confusion.

Details

```
<BottomSheetView style={styles.contentContainer}>
    <View style={styles.yellow} >

        <View style={styles.container2}>
            <Text style={styles.bottomSheetText}> Dennery To Castries </Text>
        </View>

        <View style={styles.container3}>
            <View style={styles.details}>
                <Icon name="bus" size={46} color="" style={{paddingRight:10}} />
                <View >
                    <Text style={styles.detailsText1}> $7.00 </Text>
                    <Text style={styles.detailsText2}> Price of trip </Text>
                </View>
            </View>
        </View>

        <View style={styles.details}>
            <Icon name="bus" size={46} color="" style={{paddingRight:10}} />
            <View >
                <Text style={styles.detailsText1}> $5.00 </Text>
                <Text style={styles.detailsText2}> Price from current location </Text>
            </View>
        </View>

        <View style={styles.details}>
            <Icon name="time" size={46} color="" style={{paddingRight:10}} />
            <View >
                <Text style={styles.detailsText1}> 10 mins </Text>
                <Text style={styles.detailsText2}> Estimated arrival time </Text>
            </View>
        </View>
    </View>
</BottomSheetView>
```

Figure 13: Code snippet from bottom sheet (details (Mathurin, 2025))

Back button

```
<TouchableOpacity
    onPress = {goToHome}

    style={styles.buttonConfig}
    >
    <Icon name="arrow-back" size={30} color="grey" />
</TouchableOpacity>
```

Figure 14: Code snippet from back button (Mathurin, 2025)

Navigation Bottom sheet(home) to Bottom sheet (details)

An initial attempt to implement this feature was made using the react-native-bottom-sheet library (npm, 2024), however issues arose, resulting in wasted time. Since the purpose of this increment was to develop the user interface without full functionality, it was decided to use a manual navigation method temporarily, to be improved later. The manual method, learned from the TM352 module, was applied using code from previous assignments as a guideline. Code from both bottom sheets was combined into a single file using distinct variables for each, providing a simple and functional navigation process. The initial stage was set to the home page, ensuring that the interface could be tested effectively without implementing full navigation logic.

```
const [stage, setStage] = React.useState("home"); // set initial stage to home
```

Figure 15: Code snippet for initializing initial stage (Mathurin, 2025)

An asynchronous function was used to set the initial stage to the home page and to ensure that the bottom sheet opens automatically when the stage is loaded. This provided a predictable and consistent interface behaviours enhancing the overall user experience.

```
// Go back to home bottoms sheet is open
const goToHome = async() => {
    bottomSheetRef.current?.snapToIndex(1);
    setStage('home');
}

// Go back to home bottoms sheet is open
const goToHome = async() => {
    bottomSheetRef.current?.snapToIndex(1);
    setStage('home');
}
```

Figure 16: Code snippet for async function (Mathurin, 2025)

The same asynchronous function was called when the back button was pressed, allowing the application to return to the home stage seamlessly. This method ensured that navigation between screens remained simple and reliable, without requiring the full implementation of a navigation library at this stage.

```
<TouchableOpacity  
    onPress = {goToHome}
```

Figure 17: Code snippet for calling async function (Mathurin, 2025)

The complete User interface

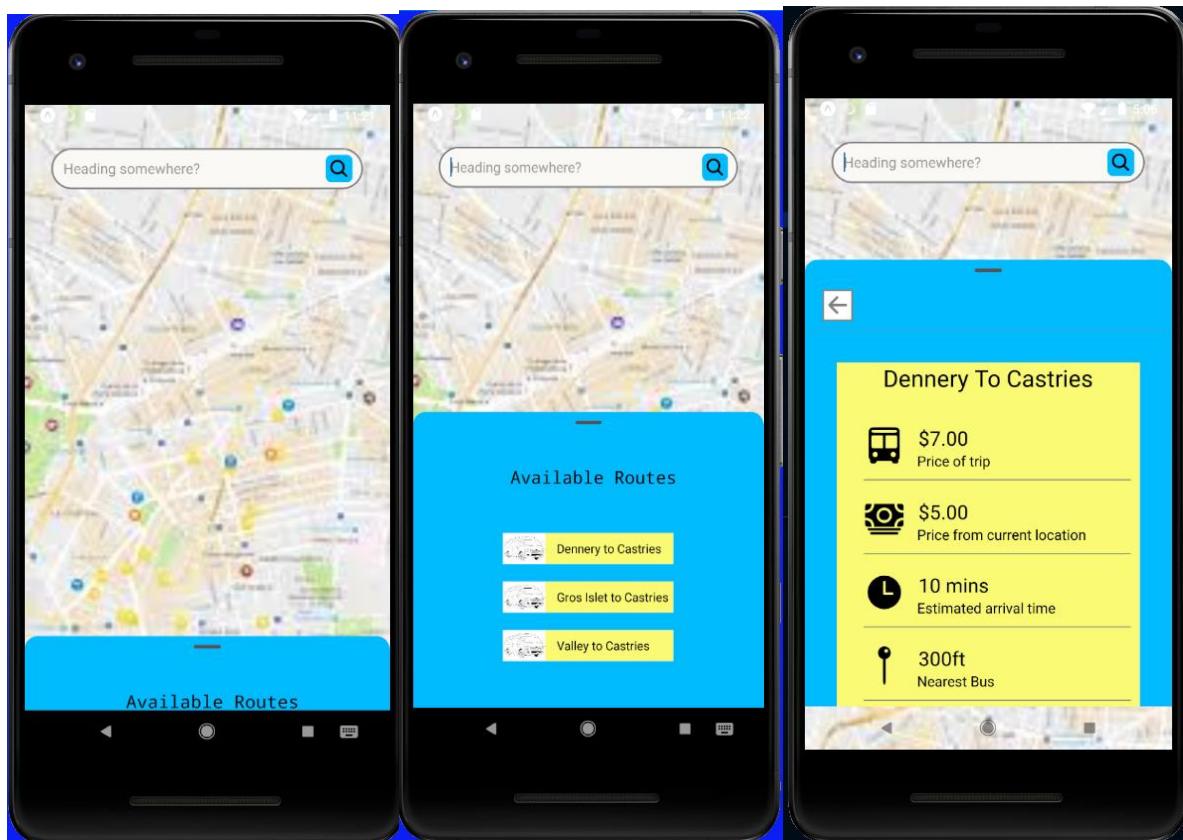


Figure 18: Completed user interface (Mathurin, 2025)

Test the user interface (self-test)

To perform this self-test, the application was run on an Android phone using the Expo CLI app (Expo Docs, n.d.). Testing simulated the perspective of a non-tech-savvy user to ensure accessibility and usability for a wide range of users, following user-centered design principles. Navigation between screens, visibility of key elements, and responsiveness were evaluated. Prior to testing, research was conducted to understand best practices for mobile application testing (Andrea, 2024.) A mobile UI testing checklist was also used to systematically evaluate usability and interface functionality (see Appendix 5).

It should be noted that, as the testing was conducted by myself, there may be some bias, and certain usability issues could have been overlooked compared to testing with independent users. Also, to help mitigate this, ethical considerations were applied following BCS guidelines to ensure fairness and accessibility (bsc The Chartered Institute for IT, 2025).

Increment 2

Learn MongoDB

A basic understanding of databases (both SQL and NoSQL) had already been gained from a level 2 course (TM254). To further develop MongoDB skills, a combination of resources was used, including YouTube tutorials, AI tools (such as ChatGPT), and the official MongoDB documentation

To avoid spending unnecessary time on unrelated topics, the features required for the application were first identified, narrowing down the scope of what needed to be learned. Based on this, ChatGPT was used to provide a tailored study plan focused on acquiring the knowledge necessary to complete the required tasks. (see Appendix H Response 5)

As learning progressed, concepts were applied immediately, helping to reinforce understanding and reduce overall learning time. Using the MongoDB documentation, the required software was installed. Initially, MongoDB Shell (MongoSH) was used, but after further consideration, MongoDB Atlas was chosen for its more user-friendly interface with a built-in GUI, making development and testing more efficient.

Creating the Database

After learning the basics of MongoDB, the database was created. Initially, the MongoDB Shell was used locally, but MongoDB Atlas was later adopted due to its web-based GUI, which simplified managing and visualizing data, improving efficiency during development.

An Atlas account was created and a new database named “bus_management” was set up, with “drivers” and “routes” collections added (MongoDB, 2022, 10:02). These collections were essential for storing and retrieving the core data used in the application ensuring that the app could access required information efficiently and consistently.

The screenshot shows the MongoDB Atlas interface. At the top, it displays "DATABASES: 1" and "COLLECTIONS: 2". Below this is a search bar labeled "Search Namespaces". A button labeled "+ Create Database" is visible. The main area shows a single database named "bus_management". Underneath the database name, there are two collection names: "drivers" and "routes".

Figure 19: MongoDB database (Mathurin, 2025)

Expand knowledge in API usage.

Knowledge of API usage was significantly expanded during this increment. From prior experience in TM352, it was clear that this was initially a weaker area, as integrating APIs had previously required additional effort to fully understand. Although manageable during TM352 coursework, additional effort was required to become comfortable. Working independently reduced the complexity of navigating unfamiliar structures as was the learning method in that TM352.

Node.js and Express.js were chosen as the core technologies for API implementation due to their efficiency in handling server-side logic and RESTful endpoints. Their official documentation (OpenJS Foundation, 2025), OpenJS Foundation, 2024) was heavily relied upon, supplemented by YouTube tutorials (MongoDB, 2022) and AI-assisted resources such as ChatGPT (OpenAI ChatGPT, 2025c). These provided structured examples and guidance that supported problem-solving when documentation alone was insufficient.

In addition, prior TM352 coursework remained beneficial, particularly as a foundation for understanding fundamental API principles Open University (2025e). The combination of official documentation, tutorials, and AI support facilitated steady progress, ultimately strengthening confidence in implementing and debugging APIs. This had a direct impact on development efficiency, as the APIs were central to connecting the mobile application with the database, ensuring that user requests and responses were handled reliably.

Implementing the Routes

Route Collection

For this project, a single route was implemented due to time constraints, as this represents the first version of the application. The route collection was designed following a YouTube tutorial (MongoDB, 2022, 6:12), but its structure required careful consideration. Initially, a separate prices collection was considered to store fare information. However, fares were integrated directly into the route collection, as they are inherently related, and this approach proved more efficient within the available timeframe.

The route collection contains fields such as the route name, ID, stop locations along the route, fare details, and driver information, including the driver's current location. Each stop is defined by its name, a unique code, and its coordinates

Route collection Part 1

QUERY RESULTS: 1-1 OF 1

```
_id: ObjectId('68669f40953ef1b96b890d90')
route_id : "R001"
name : "Valley to Castries"
▼ locations : Array (6)
  ▼ 0: Object
    code : "L0"
    name : "Bus stop in Dennery"
    ▼ coordinates : Object
      lat : 13.953428
      lng : -60.927178
  ▼ 1: Object
    code : "L1"
    name : "La Resource Gap"
    ▼ coordinates : Object
      lat : 13.936703
      lng : -60.915417
  ▼ 2: Object
    code : "L2"
    name : "Tomazo Gap"
    ▼ coordinates : Object
      lat : 13.92782
      lng : -60.943595
```

Figure 20: Route collection part 1(Mathurin, 2025)

Route collection Part 2

```
  ▾ 3: Object
    code : "L3"
    name : "Bexon Gas Station"
  ▾ coordinates : Object
    lat : 13.95497
    lng : -60.97655
  ▾ 4: Object
    code : "L4"
    name : "Cul de Sac Bus Stop"
  ▾ coordinates : Object
    lat : 13.984785
    lng : -60.997685
  ▾ 5: Object
    code : "L5"
    name : "Bus Stop in Castries"
  ▾ coordinates : Object
    lat : 14.009459
    lng : -60.992526
  ▶ fares : Array (7)
  ▾ active_buses : Array (1)
    ▾ 0: Object
      driver_id : "DRV001"
      bus_number_plate : "ABC123"
    ▾ current_location : Object
      lat : 13.953428
      lng : -60.927178
      last_updated : "2025-07-01T18:30:00Z"
    status : "active"
```

Figure 21: Route collection part 2 (Mathurin, 2025)

Add prices to the route collection

As previously noted, the option of creating a separate collection for fares was rejected. Instead, fare information was incorporated within the fares field of the route collection.

The fare data is structured according to the sequence of locations along the route. Because fares vary depending on the start and end points of a journey, predetermined prices are assigned for travel between each stop and the final destination. For example, traveling from Location 0 to Location 1 may cost \$2. These prices are stored in the database, and if a passenger disembarks before reaching the final stop, the application dynamically calculates the fare based on the distance travelled. The prices were verified to be accurate through a regular user of the route ensuring alignment with the Consumer Rights Act 2015(legislation.gov.uk, 2015).

Prices in database

```
  ▼ fares : Array (7)
    ▼ 0: Object
      from : "L0"
      to : "L1"
      price : 2
    ▼ 1: Object
      from : "L0"
      to : "L2"
      price : 2.5
    ▼ 2: Object
      from : "L0"
      to : "L3"
      price : 3
    ▼ 3: Object
      from : "L0"
      to : "L4"
      price : 4
    ▼ 4: Object
      from : "L0"
      to : "L5"
      price : 5
    ▼ 5: Object
      from : "L5"
      to : "L2"
      price : 4
    ▼ 6: Object
      from : "L5"
      to : "L1"
      price : 5
```

Figure 22:Prices in database (Mathurin, 2025)

Create driver collection

Since a specific driver was not yet available during development, the driver collection was initialized using placeholder data. This allowed the application to be tested and the database schema to be validated without waiting for real user data, a common practice in software development and database prototyping.

Also, when creating the database considerations were made to limit the amount of personal information for the driver would be stored or displayed through the application adhering to the Data Protection Act 2018(legislation.gov.uk, 2018).

Driver collection



The screenshot shows a MongoDB query results interface. At the top, it says "QUERY RESULTS: 1-1 OF 1". Below that is a single document representation:

```
_id: ObjectId('68646ff7953ef1b96b852446')
driver_id : "DRV001"
name : "John Daniel"
contact_number : "+1758345212"
work_status : "active"
number_plate : "M222"
current_route : null
```

Figure 23: Driver collection (Mathurin, 2025)

Connecting the database to the app

The required MongoDB URI was located using MongoDB Atlas, guided by a YouTube tutorial (Aviroxi, 2020, 1:00), and stored in a .env file to securely manage environment variables. This follows best practices for keeping sensitive information out of the codebase. Additionally, the application's port was configured to ensure it ran on the desired network endpoint, which is essential for consistent local testing and deployment.

Environment Variables

```
MONGODB_URI= mongodb+srv://danell:Tigerblood722@cluster0.29yqdzp.mongodb.net/bus_management?retryWrites=true&w=majority&appName=Cluster0
PORT=3000
```

Figure 24: Environment Variables (Mathurin, 2025)

Using the URI in server.js

```
// Connect to MongoDB
mongoose.connect(process.env.MONGODB_URI, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
})
.then(() => console.log(' Connected to MongoDB'))
.catch((err) => console.error(' MongoDB connection error:', err));
```

Figure 25: Using URI in server.js (Mathurin, 2025)

Using the port in server.js

```
// Start the server
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(` Server is running on port ${PORT}`);
});
```

Figure 26: Using the port in server.js (Mathurin, 2025)

The route files driverRoutes and routeRoutes were created to manage HTTP requests between the application and the database. These files encapsulate the API endpoints, enabling organized handling of CRUD operations for drivers and routes, which improves code maintainability and clarity (W3Schools, n.d.; Express, n.d.).

```
server > routes > js driverRoutes.js > ...
1  const express = require('express');
2  const Driver = require('../models/Driver');
3
4  const router = express.Router();
5
6  // Get all drivers
7  router.get('/', async (_req, res) => {
8    try {
9      const drivers = await Driver.find();
10     res.json(drivers);
11   } catch (err) {
12     res.status(500).json({ message: err.message });
13   }
14 });
15
16 // Add a new driver (optional)
17 router.post('/', async (req, res) => {
18   const { name, phone, route } = req.body;
19
20   const driver = new Driver({ name, phone, route });
21   try {
22     const newDriver = await driver.save();
23     res.status(201).json(newDriver);
24   } catch (err) {
25     res.status(400).json({ message: err.message });
26   }
27 });
28
29 module.exports = router;
```

Figure 27:Code snippet for driverRoutes (Mathurin, 2025)

```
1  const express = require('express');
2  const Route = require('../models/Route');
3
4  const router = express.Router();
5
6  // Get all routes
7  router.get('/', async (req, res) => {
8    try {
9      const routes = await Route.find();
10     res.json(routes);
11   } catch (err) {
12     if (err instanceof Error) {
13       res.status(500).json({ message: err.message });
14     } else {
15       res.status(500).json({ message: 'An unknown error occurred' });
16     }
17   }
18 });
19
20 // Get a route by route_id
21 router.get('/:route_id', async (req, res) => {
22   try {
23     const route = await Route.findOne({ route_id: req.params.route_id });
24     if (!route) return res.status(404).json({ message: 'Route not found' });
25     res.json(route);
26   } catch (err) {
27     if (err instanceof Error) {
28       res.status(500).json({ message: err.message });
29     } else {
30       res.status(500).json({ message: 'An unknown error occurred' });
31     }
32   }
33 });
34
35 // Add a new route
36 router.post('/', async (req, res) => {
37   const route = new Route(req.body);
38   try {
39     const newRoute = await route.save();
40     res.status(201).json(newRoute);
41   } catch (err) {
42     if (err instanceof Error) {
43       res.status(400).json({ message: err.message });
44     } else {
45       res.status(400).json({ message: 'An unknown error occurred' });
46     }
47   }
48 });
49
50 module.exports = router;
51
```

Figure 28: Code snippet for routeRoutes (Mathurin, 2025)

Display the data

The driverService file was created to handle fetching driver data from the database. By centralizing the data access logic in a service file, the application maintains separation of concerns, making the code easier to maintain, test, and scale Shah, H. (2022).

```
1 const apibase = "http://192.168.1.117:3000/api/drivers/";
2
3 // Define the driver type you expect from backend
4 export interface Driver {
5   driver_id: string;
6   name?: string;
7   contact_number?: string;
8   work_status?: string;
9   number_plate?: string;
0   current_route?: string | null;
1 }
2
3 export async function getAllDrivers(): Promise<Driver[]> {
4   const res = await fetch(apibase);
5   if (!res.ok) throw new Error("Error " + res.status);
6   return res.json();
7 }
8
9 export async function getDriverByDriverId(driverId: string): Promise<Driver> {
0   const res = await fetch(` ${apibase}driver-id/${driverId}`);
1   if (!res.ok) throw new Error("Driver not found");
2   return res.json();
3 }
```

Figure 29:Code snippet for driverService (Mathurin, 2025)

Data retrieved from the database was displayed in the application to provide passengers with clear and accessible driver information. To further enhance user experience and support quick assessment of operational status, a visual indicator will be implemented in the next increment. This aligns with best practices in user experience design (UX Design Institute, 2024) and accessibility standards under the Equality Act 2010 (legislation.gov.uk, 2010).

```
<View >
<Text style={styles.driverText1}>Drivers</Text>

[drivers.map((driver) => (
  <View key={driver.driver_id} style={styles.driverDetails}>
    <Text style={styles.driverText3}>ID: {driver.driver_id}</Text>

    <View>
      <Text style={styles.driverText2}>Plate Number: {driver.number_plate}</Text>
      <Text style={styles.driverText2}>Number: {driver.contact_number}</Text>
    </View>
  </View>
))]
```

Figure 30:Code snippet to display driver details (Mathurin, 2025)

Driver details in UI

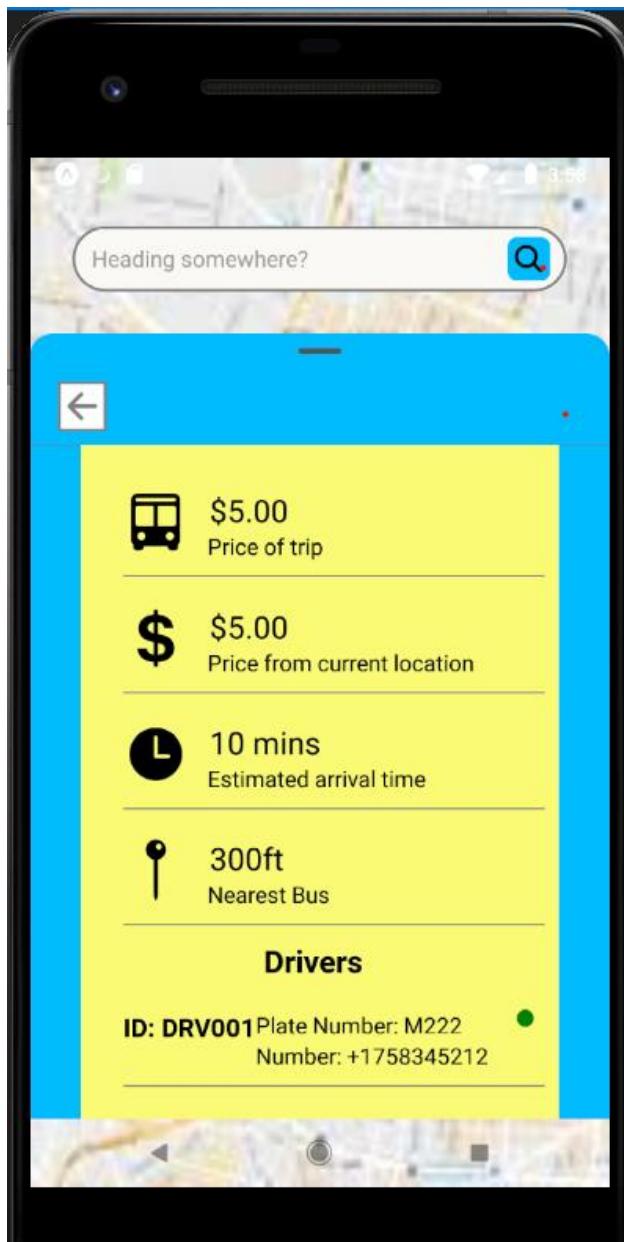


Figure 31: Driver details in UI (Mathurin, 2025)

Testing the Application

At this stage of development, a self-test was conducted to verify the application's functionality. Since the main frontend update was only the visibility of driver information, most of the development effort in this increment focused on the backend. The application was run on an Android device using the Expo CLI app (Expo Docs, n.d.), ensuring that the database connection remained stable and responded correctly to updates.

Certain features, such as displaying prices from the current location, could not yet be evaluated because they rely on the Google Maps API, which will be integrated in the final increment. Despite this, the application functioned as expected at this stage, demonstrating a stable backend connection and readiness for subsequent frontend enhancements. Ethical and accessibility considerations were applied following BCS guidance similarly to the testing at increment 1 (bsc The Chartered Institute for IT, 2025).

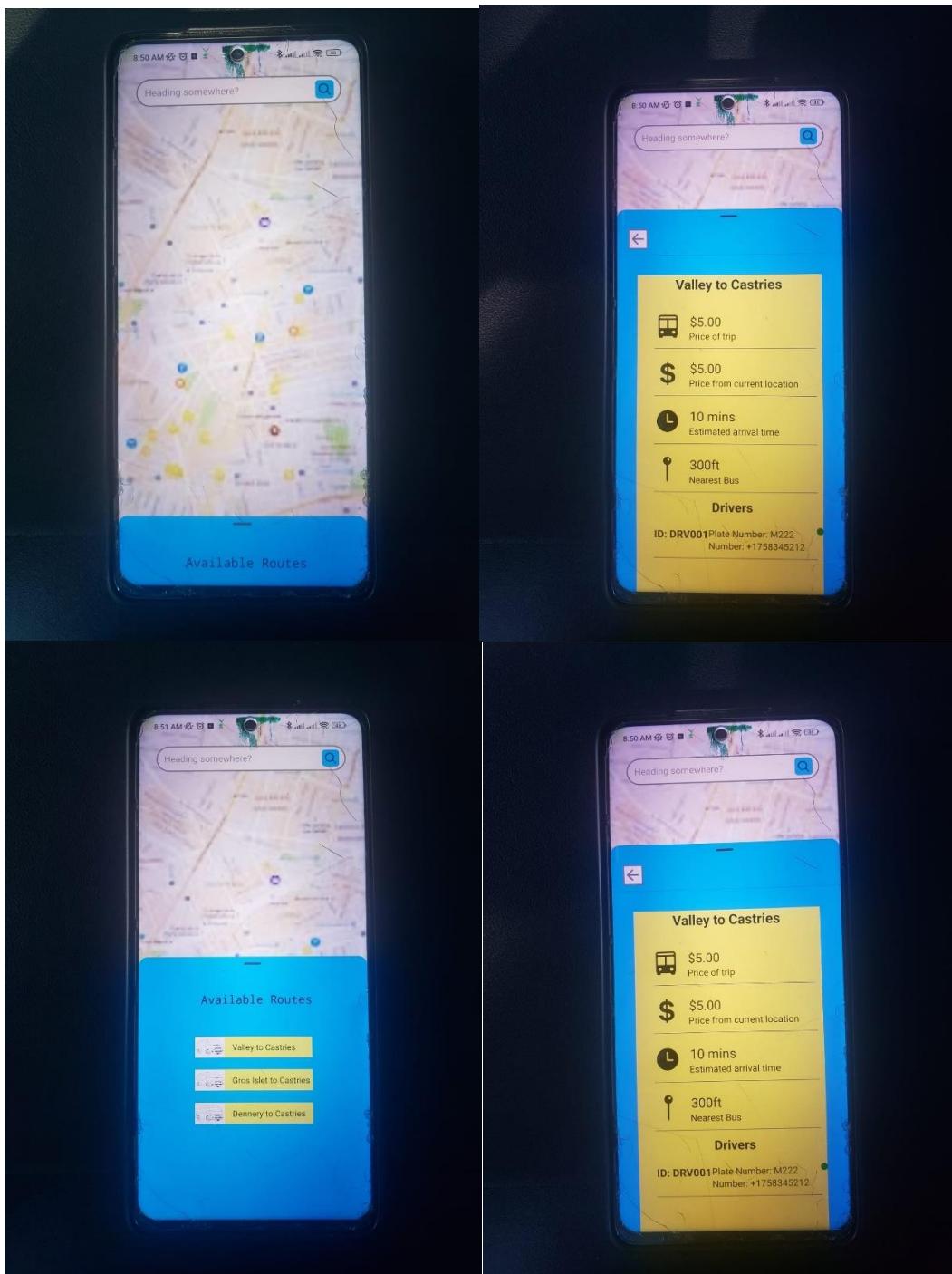


Figure 32: App on physical android device. (Mathurin, 2025)

c

```
<View style={styles.statusIndicator}>
  <View
    style={[
      styles.statusDot,
      {
        backgroundColor: driver.work_status === 'active' ? 'green' : 'red',
      },
    ]}
  />
</View>
</View>
```

Figure 33: Work status indicator code snippet. (Mathurin, 2025)

Work status indicator

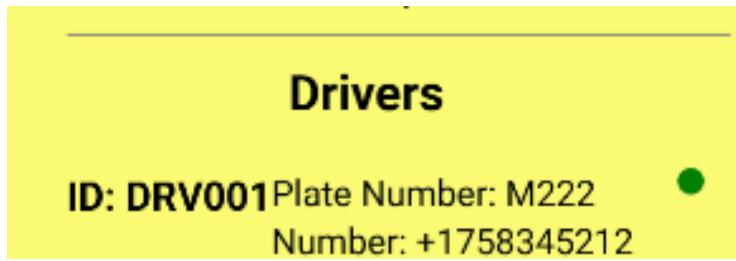


Figure 34: Image of the work status indication. (Mathurin, 2025)

Research on Google Maps API and usage

To gain an understanding of how to implement the Google Maps API, I used a layered approach with multiple sources of guidance. Initially, AI tools were consulted for general setup guidelines in React Native. To complement this, a YouTube tutorial provided a practical demonstration of integration in a mobile context. Finally, the official Google Maps API documentation served as the authoritative source, ensuring correctness and currency.

This approach demonstrates several principles:

- Authoritative vs. supplementary sources: Official documentation ensured accuracy, while AI tools and tutorials clarified practical usage.
- Incremental learning and validation: Cross-checking across sources reduced the risk of implementation errors.
- Efficiency in learning: Combining conceptual explanations with visual examples and authoritative confirmation accelerated understanding.

This combination of sources built a clear picture of how to implement the API while highlighting potential integration challenges.

Implementing real-time location

Replaced place holder map with Google map

The initial image, which was a static place holder map was replaced with the actual google maps. This was done using the react native maps library which permitted the import of the google provider.

Code to implement the map

```
<MapView
  provider={PROVIDER_GOOGLE}
  style={styles.mapBackground}
  initialRegion={DEFAULT_REGION}
  showsUserLocation={true}
>
```

Figure 35: Code snippet for implementing Google map. (Mathurin, 2025)

Bus icon

The bus icon, previously created using ChatGPT(see Appendix, Response 3), was placed at the start point of the route on the map. AI-generated icons were chosen because they provide unique, copyright-safe assets without relying on external designers (GOV.UK, 2023). The icon was imported using the Image component which provided essential flexibility.

Code for bus icon

```
<Image
  source={require("@/assets/images/bus_marker.png")}
  style={styles.busMarker}
  resizeMode="contain"
/>
```

Figure 36: Code snippet to display the bus icon. (Mathurin, 2025)

Route through a polyline

The specific route from start to end was displayed using a Google Maps polyline. This was selected because it provides an efficient method for rendering custom paths, with flexible styling options that ensure clarity for passengers.

Code to implement the polyline

```
// Fetch bus route from Google Directions API
useEffect(() => {
  const fetchRoute = async () => {
    try {
      const origin = "13.953428,-60.927178";
      const destination = "14.009459,-60.992526";
      const waypoints = [
        "13.936703,-60.915417",
        "13.92782,-60.943595",
        "13.95497,-60.97655",
        "13.984785,-60.997685",
      ].join("|");

      const response = await fetch(
        `https://maps.googleapis.com/maps/api/directions/json?origin=${origin}&destination=${destination}&waypoints=${waypoints}&mode=driving&key=AIzaSyAHOCnme4wpOcNI0Aso516jd8P5y3Bw9wE`
      );

      const json = await response.json();

      if (json.routes.length) {
        const points = polyline.decode(json.routes[0].overview_polyline.points);
        const coords = points.map(([lat, lng]) => ({
          latitude: lat,
          longitude: lng,
        }));
        setRoutePoints(coords);
      } else {
        console.warn("No routes found");
      }
    } catch (err) {
      console.error("Error fetching directions:", err);
    }
  };
  fetchRoute();
}, []);

/* Polyline from Google Directions */
{routePoints.length > 0 && (
  <Polyline
    coordinates={routePoints}
    strokeColor="#05D8F2"
    strokeWidth={5}
    zIndex={1}
  />
)}
```

Figure 37: Code for implementing the polyline. (Mathurin, 2025)

Adding bus stops

Custom bus stop icons were also generated through ChatGPT (see Appendix H, Response 6) and placed along the route also aligning with the Copyright laws (GOV.UK, 2023). This ensured consistency in style with the bus icon while avoiding copyright restrictions. The stops which were in my database (see Figures 20 & 21) were manually added to an array, which were then referenced as a marker using the custom icon.

Code implementing bus stops

```
const STOPS = [
  { id: "stop-0", latitude: 13.953428, longitude: -60.927178, title: "Main Stop" },
  { id: "stop-1", latitude: 13.936703, longitude: -60.915417, title: "Stop 1" },
  { id: "stop-2", latitude: 13.92782, longitude: -60.943595, title: "Stop 2" },
  { id: "stop-3", latitude: 13.95497, longitude: -60.97655, title: "Stop 3" },
  { id: "stop-4", latitude: 13.984785, longitude: -60.997685, title: "Stop 4" },
  { id: "stop-5", latitude: 14.009459, longitude: -60.992526, title: "Final Stop" },
];

/* Stops Markers */
{STOPS.map((stop) => (
  <Marker
    key={stop.id}
    coordinate={{ latitude: stop.latitude, longitude: stop.longitude }}
    title={stop.title}
    pinColor="green"
  >
    <Image
      source={require("@/assets/images/bus_stop.png")}
      style={styles.busMarker}
      resizeMode="contain"
    />
  </Marker>
))}
```

Figure 38: Code snippet for implementing the bus stops. (Mathurin, 2025)

Simulating the GPS location

Due to limited driver availability, a JavaScript script was created to simulate GPS data by generating points along the map that allowed the bus icon to move.

The first script generated arbitrary points that did not align with the intended route. This was later refined so that the simulated points followed the actual polyline in a realistic manner.

First simulator created

```
1  /**
2   * simulatorGPS - Simulates real time GPS input to the database
3   */
4
5  const axios = require('axios');
6
7  // Predefined GPS coordinates along the route
8  const routePoints = [
9    { lat: 14.00500, lng: -60.99000 }, // Segment A start
10   { lat: 14.00600, lng: -60.98900 },
11   { lat: 14.00700, lng: -60.98800 }, // Segment B
12   { lat: 14.00800, lng: -60.98700 },
13   { lat: 14.00900, lng: -60.98600 } // Segment C end
14 ];
15
16 // Index to track where we are in the route
17 let currentPointIndex = 0;
18
19 // Send location every 5 seconds
20 setInterval(() => {
21   const point = routePoints[currentPointIndex];
22   axios.post('http://localhost:3000/api/drivers/update-location', {
23     driver_id: 'DRV001',
24     lat: point.lat,
25     lng: point.lng
26   })
27   .then(() => {
28     console.log(`Sent location: ${point.lat.toFixed(5)}, ${point.lng.toFixed(5)} `);
29   })
30   .catch((err) => {
31     console.error('Failed to send location:', err.message);
32   });
33 }
34
35 // Move to next point, loop back to start when done
36 currentPointIndex = (currentPointIndex + 1) % routePoints.length;
37 }, 5000); // every 5 seconds
```

Figure 39: Code for first simulator created. (Mathurin, 2025)

Second simulator created

```
1  const axios = require("axios");
2  const polyline = require("@mapbox/polyline"); // npm install @mapbox/polyline
3
4  async function generateRoute() {
5    const origin = "13.953428,-60.927178";
6    const destination = "14.009459,-60.992526";
7    const waypoints = [
8      "13.936703,-60.915417",
9      "13.92782,-60.943595",
10     "13.95497,-60.97655",
11     "13.984785,-60.997685",
12   ].join("|");
13
14  const url = `https://maps.googleapis.com/maps/api/directions/json?origin=${origin}&destination=${destination}&waypoints=${waypoints}&mode=driving&key=AIzaSyAH0Cnme4wp0cNI0Aso5l6jd8P5y3Bw9wE`;
15
16
17  const res = await axios.get(url);
18  const points = polyline.decode(res.data.routes[0].overview_polyline.points);
19
20  // Convert to objects { lat, lng }
21  return points.map(([lat, lng]) => ({ lat, lng }));
22}
23
24  async function startSimulator() {
25    const smoothRoute = await generateRoute();
26    let index = 0;
27
28    setInterval(() => {
29      const point = smoothRoute[index];
30
31      axios
32        .post("http://localhost:3000/api/drivers/update-location", {
33          driver_id: "DRV001",
34          lat: point.lat,
35          lng: point.lng,
36        })
37        .then(() => {
38          console.log(`Sent location: ${point.lat}, ${point.lng}`);
39        })
40        .catch(console.error);
41
42      index++;
43      if (index >= smoothRoute.length) index = 0; // loop
44    }, 1000); // 1 update per sec
45  }
46
47  startSimulator();
```

Figure 40: Code for second simulator created. (Mathurin, 2025)

Simulated GPS is a standard development practice, allowing testing of real-time features without reliance on live drivers or hardware. This aligns with agile prototyping principles, where realistic stand-ins are used until production-ready data sources become available. In a real-world deployment, the location would be obtained either by installing GPS hardware on the bus or by utilising the driver's personal smartphone. Each approach has distinct pros and cons, such as cost, maintenance, and privacy implications, which would need to be balanced in future development.

Implementing Estimated arrival times

So the initial UI design for this was to display the estimated arrival time to the main stop only, this would not favour the passengers as not everyone would be going to the main stop. So the UI design was modified to account for the times for each stop. Originally, the nearest bus stop was designed to display the distance to the bus. This has been changed to show the time instead, as it provides a more user-friendly experience for users who may not accurately perceive distances. The previous UI design is illustrated in Figure 31.

Modified UI design

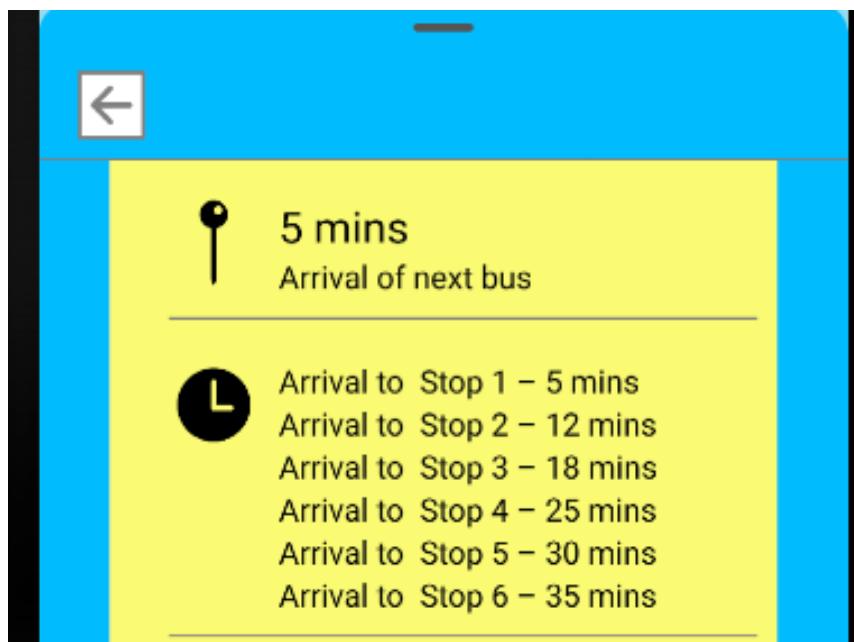


Figure 41: Image of modified UI design. (Mathurin, 2025)

User's current location

Implementing this feature would require the location of the user, this was also simulated, but with just one static location which would be amended manually for testing. This was another marker using a pin icon with a distinctive blue colour. This feature would need to be securely implemented as someone's location is very sensitive information (legislation.gov.uk, 2018).

Code to simulate the user's current location

```
const SIMULATED_USER_LOCATION = {  
  latitude: 13.934306643469037,  
  longitude: -60.934880657006254,  
};  
  
const [userLocation, setUserLocation] = useState(SIMULATED_USER_LOCATION);  
<Marker  
  coordinate={userLocation}  
  title="You"  
  pinColor="blue"  
/>
```

Figure 42: Code to simulate user current location. (Mathurin, 2025)

Integration and testing

The final step in this increment was to ensure the features were integrated and functioning cohesively. For evaluation, the application was self-tested by the developer and tested with one passenger to gather feedback from the user perspective. At this stage, a driver was not available for participation, meaning feedback from the driver's perspective could not yet be obtained. While this limited the scope of testing, the passenger feedback still provided valuable insights into clarity and usability, and future testing with drivers will be essential to validate ease of use from their side.

Future work

Estimated Arrival Times

Building on the foundation already implemented, the arrival time feature would be enhanced to calculate estimates for each bus stop along the route, not just the main stop. This would make the application more equitable for all passengers, since not everyone is traveling to a terminal point. Implementation would leverage Google Maps' built-in travel time estimation functionality, combined with the driver's real-time GPS location. This would allow passengers to see accurate, stop-specific arrival times. Future iterations could also include predictive adjustments that account for traffic conditions, delays, or bus driver schedules.

Search Bar

The inclusion of a search bar would significantly improve usability by allowing passengers to simply type their destination and receive the most relevant route information. Technically, this would be implemented by querying the database of stops and routes against the passenger's input. For improved accuracy, integration with Google Maps' Places API could be considered, enabling searches for well-known landmarks as well as bus stops. This feature would particularly benefit less tech-savvy users by streamlining route discovery into a single, familiar interaction.

Price of Trip

At present, the trip price feature only contains placeholder data. A future version would calculate fares dynamically, using the passenger's current location as the starting point and their chosen destination as the endpoint. This would involve retrieving distance-based fare information from the database and displaying the calculated cost within the interface. To improve transparency, the app could also show a fare breakdown (e.g., base fare plus distance rate). This would provide passengers with greater clarity and allow for potential future integration with digital payment systems.

Additional Enhancements

Further developments could include:

- Accessibility features such as voice-assisted navigation or haptic feedback for visually impaired users.
- Driver-focused tools, like route optimization suggestions or earnings tracking.
- Passenger notifications, such as push alerts for approaching buses or route changes. These additions would extend the functionality of the application, making it more robust, inclusive, and practical for real-world deployment.

Review and Reflection

Description

During the early stages of the first increment, the foundation for the project was laid out identifying aspects such as the scope, plan, life cycle and the resources to be used.

Pivoting from this, the project has now entered a more complex phase as I begin learning new software, improving skills, and writing code. Progressively improving my skills while using them to develop the application has proven to be a very effective way of learning for me. It resonates strongly, as I see this method of learning and applying immediately helps me form better connections with the content. From learning to use Figma to digitize the sketch of the user interface, to developing skills in React Native, particularly in using different libraries and making effective use of the documentation provided, all of this has contributed to my development.

Building on this, I created the database using MongoDB Atlas after learning the basics. I followed the official MongoDB documentation closely and created the collections while still in the learning process. This hands-on approach helped reinforce the concepts. I also set up the back end of the application using Node.js and Express.js. Most of the guidance for this part of the project came from the knowledge and practice I gained during TM352, particularly from the applications used in the assignments. This gave me a solid starting point to structure and develop the server-side logic.

During this increment, most of the work was focused on the back end. As a result, the front end saw limited visible progress, with the only notable addition being the display of driver information. However, the groundwork laid through the back-end development is essential for integrating more features later.

Increment 3 introduced driver-specific features and real-time functionality, including work status indicators, Google Maps integration, polyline route visualization, bus stops, GPS simulation, and initial estimated arrival time functionality (see Section Z). Limited driver availability meant I could not test an actual driver, which was slightly frustrating, but I was content with simulating the driver using alternative methods. I was able to use the same passenger previously engaged in Increment 2 for final testing, ensuring some continuity in user feedback. While some features remain to be fully implemented, increment 3 provided a working demonstration of real-time tracking capabilities.

Feelings

When I started to set up the project, I had a lot of issues doing so, some deriving from previous settings on my laptop and because the documentation I was using was for an older version of Android Studio. I spent countless hours trying to figure this out, and I was already feeling overwhelmed, considering this was just the start of the project.

Frustration was fuelled later when I wanted to implement the bottom sheet, getting constant issues, even though I was following the instructions and it just was not working. I simply needed a break at one point, considering I tried multiple libraries. As I have noticed, for those situations, taking a break drastically helps. When I am stuck, it is like I am repeating the same ineffective process. When I return with a different strategy and a refreshed mind, things go a lot smoother.

At the start of Increment 2, I was highly motivated, but after a while, I just felt drained and completely tired, which led to me taking an intentional break for 10 days. It was a bit stressful having to code the back end, especially because I was implementing while learning, and it was difficult to debug, which was a very common issue. The constant debugging without being entirely sure of what was going wrong took a toll mentally. It started to feel like trial and error most of the time, which became exhausting.

It felt good when I started coding and seeing my design materialize. This was further emphasized when I loaded the application on my physical device and saw that it was functional, which was a wonderful feeling. I am slightly worried that the full requirements of the project may not be met due to the lack of a proper driver, but I am hopeful that I can still pull through.

Starting Increment 3 brought a mix of anticipation and pressure. Delays in starting this increment and the technical complexity of real-time GPS simulation caused initial concern. I also felt anxious as it became clear that not all features could be completed due to time constraints and earlier delays from Increment 2. The fact that a bus driver was still unavailable was a bit annoying, but using simulated data provided a workable alternative. Seeing the bus icon move along the polyline in real time was motivating and validated the earlier planning and learning in Increments 1 and 2. Using multiple sources AI guidance, tutorials, and official documentation helped manage uncertainty and gave confidence in implementation. Personal scheduling constraints occasionally impacted progress, but prioritization of essential features ensured meaningful outcomes.

Evaluation

Communication with stakeholders has not been up to par. I currently have a passenger, but not a driver, due to time constraints. I aim to have one by the completion of Increment 2 or decide on whether I will self-test while posing as a bus driver. Not having this role active in the testing stage is a concern and something I know is affecting the project's overall testing process.

There wasn't any improvement in communication with the stakeholder, more specifically the driver. At this point, it is hinting that I may have to self-test the entire application. This is not the ideal route, but given the circumstances, it may be the most practical option moving forward.

Also, while my tutor suggested a feature where a passenger could request a bus, I now realize this is not feasible because I did not include a design that distinguishes between the driver and the passenger. This oversight has impacted how much control users have and make the suggested feature unworkable.

On a more positive note, during a conversation with my tutor, I was reminded that I do not need to go in-depth with every feature or aspect of the application. The key is to ensure that the principal concept is clearly demonstrated. This helped relieve some pressure I was placing on myself, and it has given me a better sense of direction on how to approach the remaining work. It was reassuring to know that I don't have to overcomplicate the implementation but rather focus on getting the main idea across effectively.

The incremental methodology proved effective for continuous learning and testing. Positive outcomes in Increment 3 included clear visual cues for driver activity, real-time GPS simulation allowing testing without live drivers, realistic route representation with polylines and bus stops, and the initial estimated arrival times enhancing passenger usability.

Challenges included partial implementation of certain features due to time constraints, limited stakeholder feedback, and simulation limitations compared to live driver data. Attempting overly realistic implementation in Increment 2 slowed progress and contributed to initial anxiety (include reference), but also taught valuable lessons in balancing perfection with achievable goals. Despite the absence of a live driver, testing with the same passenger as before provided continuity and useful feedback for usability improvements. These outcomes highlight the trade-offs between realistic testing and available resources, while still achieving core project goals.

Analysis

From early issues like setting up Android Studio using outdated documentation, to trying and failing with different bottom sheet libraries, I have learned the importance of using up-to-date tools and clear documentation. Each roadblock taught me something, and stepping away to reset my mind worked every time.

The incremental life cycle I am following has really helped break the work down into smaller tasks, which has been essential given that I have had heavy workloads from other modules. From a development perspective, I can say the user interface is currently simple, and from self-testing, it does appear user-friendly, which aligns with one of my initial goals.

The use of AI played a huge role in helping me apply the skills I was learning while developing. It acted as a quick reference point and helped provide the basics when I needed a starting point or some clarity. Having this kind of support made it easier to keep moving forward without getting stuck for too long.

Although I am using an incremental life cycle, in Increment 2 I made some unplanned changes to the UI. This was not initially part of the scope for this stage, but I felt it was necessary to adjust based on how the app was shaping up. While this did affect the timeline a

bit, I believe it was a worthwhile change to improve the overall user experience and align more closely with how I envisioned the final product.

Scope changes included modifying features such as the nearest bus distance indicator to better suit user needs (see Section Z). Stakeholder interaction was limited, but simulated data allowed for meaningful evaluation and incremental improvements. LSEPI/EDI considerations were reflected in accessibility and inclusivity through color indicators, and future iterations could include alternative visual cues. Professionalism was maintained through thorough documentation, adherence to coding standards, and systematic testing. Increment 3 enhanced skills in Google Maps API integration, React Native component handling, and AI-assisted asset creation, while experience from Increment 2 reinforced the need to balance ambition with practical deliverables. Risk management was addressed by adjusting timelines and prioritizing essential features, mitigating the impact of delays. Development and testing issues, such as refining GPS simulation scripts to follow polyline routes, demonstrate iterative problem-solving (see Appendix C).

Action Plan

Going forward, I plan to use the extra time from completing my other two modules to reassess and possibly revise some parts of the scope. If I ever take on a similar project again, I will be more specific and certain about user roles and authentication from the start. I will also continue using the intentional break when I feel stuck, but with a better balance to avoid wasting time, and I will aim to communicate more frequently with stakeholders to ensure testing and feedback remain meaningful.

Right now, I will focus on clarifying the unclear aspects of the application and take bold steps to fix what I can so that I can bring this project to successful completion. For the remainder of the project, I will be working to ensure that the essential features are developed and in place to deliver a functioning application by the end. This means prioritizing key areas and making sure I stay focused on what really matters to meet the core goals. I also plan to secure a passenger for proper testing, and if I am unable to get a driver, I will settle for self-testing in that role. While not ideal, this will still allow me to demonstrate how the application is intended to work.

Future work will focus on completing the estimated arrival time feature for each bus stop, implementing a search bar for improved passenger usability, dynamic fare calculation and potential digital payment integration, accessibility improvements such as voice navigation or haptic feedback, and enhanced driver tools and passenger notifications. These improvements aim to extend the app's functionality, inclusivity, and real-world applicability while building on the lessons learned throughout all three increments.

Conclusion

Looking back, I probably should have included login functionality early on. This would have helped me create distinct roles for the passenger and driver, which is now proving to be a limitation. I also think I should have secured a driver stakeholder earlier instead of relying

heavily on self-testing. Better initial planning might have avoided the current uncertainty I feel about some features.

The idea of the final application, especially in terms of its complexity, in hindsight would not be possible within the time and scope available. While the original vision was ambitious, I now see that focusing on the core concept and demonstrating the main functionality is far more realistic and achievable. This realization has helped me adjust my expectations and focus on delivering a working solution that still reflects the overall idea, just in a simpler and more manageable way.

Increment 3 demonstrated the feasibility of real-time bus tracking, highlighting both strengths and limitations of the application. The combination of UI, backend, and real-time features provided a solid foundation for future enhancements, including full driver integration, dynamic fare calculation, and expanded accessibility features. Feedback from tutors (see Appendix D) informed technical and design decisions, reinforcing the value of iterative reflection.

(8809 Words)

Reference List

ACAS Working for Everyone (n.d.) *Equality, diversity and inclusion: What they are.* Available at: <https://www.acas.org.uk/improving-equality-diversity-and-inclusion> (Accessed: 12 April 2025).

Andrea (2024) *Mobile App Usability Testing Checklist.* Available at: <http://userbrain.com/blog/mobile-app-usability-testing-checklist/> (Accessed: 23 May 2025).

Aviroxi (2020) *How to find mongoURI inn mongodb* 9 November. Available at: <https://www.youtube.com/watch?v=tGciYnyQBE8> (Accessed: 8 July 2025).

BSC The Chartered Institute for IT (2025) *Articles, opinions & research.* Available at: <https://www.bcs.org/> (Accessed: 12 April 2025).

Citymapper Limited (2013) *Citymapper* (Version 11.35) [Mobile app]. Available at: <https://play.google.com/store/apps/details?id=com.citymapper.app.release&hl=en> (Accessed: 12 April 2025).

Expo Docs (n.d.) *Getting started.* Available at: <https://docs.expo.dev/> (Accessed: 1 May 2025).

Express (n.d.) *Express routing.* Available at: <https://expressjs.com/en/guide/routing.html> (Accessed: 11 September 2025).

Google LLC (2025) *Android Studio Narwhal* (Version 2025.1.1 Canary 9) [Computer program]. Available at: <https://developer.android.com/studio> (Accessed: 23 May 2025).

Google Maps Platform (n.d.) *Make the world your platform.* Available at: <https://cloud.google.com/maps-platform> (Accessed: 11 April 2025).

Gorhom, M. (n.d.) *React Native Bottom Sheet.* Available at: <https://gorhom.dev/react-native-bottom-sheet/> (Accessed: 6 May 2025).

GOV.UK (n.d.) *How to make a freedom of information (FOI) request.* Available at: <https://www.gov.uk/make-a-freedom-of-information-request> (Accessed: 12 April 2025).

GOV.UK (2023) *Intellectual property: Copyright.* Available at: <https://www.gov.uk/government/collections/intellectual-property-copyright> (Accessed: 12 April 2025).

GOV.UK (2024) *Meet the requirements of equality and accessibility regulations.* Available at: <https://www.gov.uk/guidance/meet-the-requirements-of-equality-and-accessibility-regulations> (Accessed: 4 August 2025).

JavaScript Mastery (2025) *React Native Course for Beginners in 2025 / Build a Full Stack React Native App.* 7 March. Available at: <https://www.youtube.com/watch?v=f8Z9JyB2EIE&t=4108s> (Accessed: 26 April 2025).

Kravchenko, A. (2025) *Mobile UI Testing: How to Do It Right.* Available at: <https://luxequality.com/blog/mobile-ui-testing/> (Accessed: 23 May 2025).

legislation.gov.uk (1974) *Health and Safety at Work etc. Act 1974.* Available at: <https://www.legislation.gov.uk/ukpga/1974/37/contents> (Accessed: 4 August 2025).

legislation.gov.uk (1990) *Computer Misuse Act 1990.* Available at: <https://www.legislation.gov.uk/ukpga/1990/18/contents> (Accessed: 12 April 2025).

legislation.gov.uk (2010) *Equality Act 2010.* Available at: <https://www.legislation.gov.uk/ukpga/2010/15/contents> (Accessed: 4 August 2025).

legislation.gov.uk (2015) *Consumer Rights Act 2015.* Available at: <https://www.legislation.gov.uk/ukpga/2015/15/contents> (Accessed: 6 August 2025).

legislation.gov.uk (2018) *Data Protection Act 2018.* Available at: <https://www.gov.uk/data-protection> (Accessed: 12 April 2025).

legislation.gov.uk (2021) *Environmental Principles Policy Statement.* Available at: <https://www.gov.uk/government/publications/environmental-principles-policy-statement/environmental-principles-policy-statement> (Accessed: 6 August 2025).

Mane, R. (2019) *App Design: How to Create the Perfect Design for Mobile Apps.* Available at: <https://duck.design/mobile-app-design/> (Accessed: 29 April 2025).

Microsoft Corporation (n.d.) *Visual Studio Code* (Version 1.99.3) [Computer Program]. Available at: <https://code.visualstudio.com/Download> (Accessed: 11 April 2025).

MongoDB (2022) *Getting Started with MongoDB Atlas- A Modern Database!* 20 September. Available at: <https://www.youtube.com/watch?v=bBA9rUdqmgY> (Accessed: 3 July 2025).

MongoDB Inc (2025a) *MongoDB* (Version 8.0.10). Available at: <https://www.mongodb.com/> (Accessed: 11 April 2025).

MongoDB Inc (2025b) *MongoDB Documentation.* Available at: <https://www.mongodb.com/docs/> (Accessed: 11 April 2025).

Moovit (2014) *Moovit: Bus & Public Transit* (Version 5.169.0.1719) [Mobile app]. Available at: <https://play.google.com/store/apps/details?id=com.tranzmate&hl=en> (Accessed: 12 April 2025).

Nielsen, J. (2013) *10 usability heuristics for user interface design.* Available at: <https://www.designprinciplesftw.com/collections/10-usability-heuristics-for-user-interface-design> (Accessed: 12 April 2025).

npm (2024) *react-native-bottom-sheet* (Version 3.0.0) [Software]. Available at: <https://www.npmjs.com/package/react-native-bottom-sheet> (Accessed: 3 May 2025).

Obenauer, O. (2024) *react-native-vector-icons* (Version 10.2.0) [Software]. Available at: <https://www.npmjs.com/package/react-native-vector-icons> (Accessed: 3 May 2025).

OpenAI (2025) *ChatGPT*. Available at: <https://chat.openai.com> (Accessed: 4 August 2025).

OpenAI ChatGPT (2025a) *ChatGPT response to Danell Mathurin*, 26 April.

OpenAI ChatGPT (2025b) *ChatGPT response to Danell Mathurin*, 29 April

OpenAI ChatGPT (2025c) *ChatGPT response to Danell Mathurin*, 23 May.

OpenAI ChatGPT (2025d) *ChatGPT response to Danell Mathurin*, 8 July.

OpenAI ChatGPT (2025e) *ChatGPT response to Danell Mathurin*, 10 July.

OpenAI ChatGPT (2025f) *ChatGPT response to Danell Mathurin*, 8 August.

OpenJS Foundation (2024) *Node.js* (Version 22.14.0) [Computer Program]. Available at: <https://nodejs.org/en> (Accessed: 11 April 2025).

OpenJS Foundation (2025) *Express.js* (Version 5.1.0) [Computer Program]. Available at: <https://expressjs.com/> (Accessed: 11 April 2025).

Open University (2025a) *Choosing a Lifecycle Model. TM470: The computing and IT project*. Available at: <https://learn2.open.ac.uk/mod/oucontent/view.php?id=2456473> (Accessed: 12 April 2025).

Open University (2025b) *Legal, Social, Ethical and Professional Issues. TM470: The computing and IT project*. Available at: <https://learn2.open.ac.uk/mod/oucontent/view.php?id=2456502> (Accessed: 12 April 2025).

Open University (2025c) *Embedding Equality, Diversity and Inclusion in Your Project. TM470: The computing and IT project*. Available at: <https://learn2.open.ac.uk/mod/oucontent/view.php?id=2456504> (Accessed: 12 April 2025).

Open University (2025d) *Writing, Structuring, Styling and Editing Reports. TM470: The computing and IT project*. Available at: <https://learn2.open.ac.uk/mod/oucontent/view.php?id=2456508> (Accessed: 12 April 2025).

Open University (2025e) *Designing mobile apps 1 TM352: Web mobile and cloud technologies* Available

at: <https://learn2.open.ac.uk/mod/oucontent/view.php?id=2353284§ion=4> (Accessed: 4 July 2025).

Postman (2025). *Postman – The World's Leading API Platform. (Version 11.53.0)* Available at: <https://www.postman.com/> (Accessed: 10 July 2025).

React Native Directory (n.d.) *React Native Directory*. Available at: <https://reactnative.directory/> (Accessed: 11 April 2025).

Shah, H. (2022) *Best Practices for Node.js Application Architecture*. Available at: <https://hardiks.medium.com/best-practices-for-node-js-application-architecture-8ccb1f94452a> (Accessed: 10 July 2025).

Terekhov, V. (2025) *Expo vs React Native CLI: 10 Key Differences You Didn't Know*. Available at: <https://attractgroup.com/blog/expo-vs-react-native-cli-10-key-differences-you-didnt-know/> (Accessed: 1 May 2025).

Transit Inc (2013) *Transit: Subway & Bus Time* (Version 5.17.19) [Mobile app]. Available at: <https://play.google.com/store/apps/details?id=com.thetransitapp.droid&hl=en> (Accessed: 12 April 2025).

UX Design Institute (2024) *7 fundamental UX design principles all designers should know*. Available at: <https://www.uxdesigninstitute.com/blog/ux-design-principles/> (Accessed: 12 July 2025).

W3Schools (n.d.) *Node.js RESTful API*. Available at: https://www.w3schools.com/nodejs/nodejs_rest_api.asp (Accessed: 6 August 2025).

Appendices

Appendix A - Ethics Form



Judith.Tope<judith.tope@open.ac.uk>

To: You

[Reply](#) [Reply all](#) [Forward](#)

Fri 4/4

Dear Danell

Thank you for submitting your project proposal.

I have reviewed the ethical approach to your project with the following outcome: Approved

My comments are below and have been attached to your submission record:

better, well done

Details of your submission are included at the end of this email (please note that some of the questions are dependent on earlier answers, so you may see 'N/A' next to any that were not relevant to your proposal).

Please take a screen shot of this email or save as a PDF to upload as Appendix 1 to your TMA01.

We expect your project scope to develop and evolve as you work through the module. If the response to any of the questions you have answered in this questionnaire changes, you must submit a new ethics questionnaire.

If you have any queries, please get in touch with me.

Your Project Proposal:

Submitted: April 4, 2025 9:46 PM

Submission ID: 1113

Please select the modules that your project proposal relates to (select all that apply):

[TM352 - Web, mobile and cloud technologies"]

Proposed Project Title (Draft):

Mini-Bus Guidance App

Describe your project in brief:

1. The under developed mini-bus system in St. Lucia.
2. This is considered a problem as notable complaints are frequently raised by passengers due to the limited mini-bus availability and inability to plan their trips adequately.
3. Solving this issue would provide many benefits such as.
 - Increased passenger satisfaction
 - An organized system to be utilized by the mini-bus drivers

Does your project involve, or have any effect on human subjects?

Yes

If you are able, please give further details of your reasoning for involving human subjects:

The stakeholders in this project are the passengers, mini-bus drivers and the general population.
Getting their views and concerns would be ideal in providing the best application possible.

Are all participants able to give their informed consent? (e.g. signed declaration)

Yes

Is the study likely to involve vulnerable people? (e.g. children, disabled people)

No

What safeguarding measures have you considered?

N/A

Will any subjects be observed covertly?

No

For participants involved longer than an hour at any time, has due regard been made to their health, safety and well-being in a suitable plan and discussed with the participants in advance?

No

Will the subjects be asked about sensitive topics? e.g. money, appearance, relationships, hygiene, politics, religion, protected characteristics, alcohol and drug use, etc.

No

Does your proposed research involve any action on, or action that might affect any animals and/or plants?

No

Your tutor will need to discuss this with you, please provide further details:

N/A

Does your research involve the collection of data and information?

No

Please share how you plan to store and manage this data/information?

N/A

Could the storage, computations, or data analysis, whether through AI or standard coding, potentially harm individuals or organisations through data breaches, misuse, or unintended computational outcomes?
N/A

For AI projects, is it possible that either the outcomes, or the analysis, will have any impact on human decision making?

Not applicable

Cheers

Ju

Figure A1: Ethics form (Mathurin, 2025)

[Return to impact of LSEPI/EDI](#)

Appendix B - LESPI Form



LSEPI Form

I confirm that I have read and understood the resources on the TM470 module website on working with human participants.

[Legal, Social, Ethical and Professional issues: 3 Working with human participants | OU online](#)

[Legal, Social, Ethical and Professional issues: Appendix A Guidelines for conducting research with human participants | OU online](#)

I understand that The Open University is unable to offer liability insurance to cover any negative consequences of my actions in carrying out research as part of this module.

Name: Danell Mathurin

Date: 8th June 2025

Appendix C -Consent Form



CONSENT FORM

**INFORMED CONSENT FOR DEVELOPMENT AND
EVALUATION OF A BUS GUIDANCER APPLICATION
FOR ST. LUCIA**

Name: Danell Mathurin
Position: Final year student, BSc Computing and IT, The Open University
Email: Danell722@outlook.com
Phone: +1(758) 726-3534

Please highlight your choice by clicking inside the appropriate box.

1. Taking part in the study

I have read and understood the information sheet for the following study: <i>Development and Evaluation of a Bus Guidance Application for St. Lucia</i> , or it has been read to me. I have been able to ask questions about my participation and my questions have been answered to my satisfaction.	YES <input type="checkbox"/>	NO <input type="checkbox"/>
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions I am not comfortable with and that I can withdraw from the study at any time by contacting Danell Mathurin (danell722@outlook.com), up until 30th August 2025, without having to give a reason.	YES <input type="checkbox"/>	NO <input type="checkbox"/>
I understand that taking part in the study involves providing feedback on usability and experience of the bus guidance application (approximately 5–15 minutes per session).	YES <input type="checkbox"/>	NO <input type="checkbox"/>

2. Use of the information in the study

I understand that information I provide will be used in the researcher's final year project report, presentations, or academic publications, in anonymised form.	YES <input type="checkbox"/>	NO <input type="checkbox"/>
I understand that personal information collected about me that can identify me, such as my name or where I live, will not be shared beyond the study team.	YES <input type="checkbox"/>	NO <input type="checkbox"/>
I understand that my data will be stored securely on the researcher's password-protected personal computer (accessible only to the researcher) for 12 months, after which it will be securely deleted.	YES <input type="checkbox"/>	NO <input type="checkbox"/>
I agree to being quoted anonymously in the final project report or related outputs.	YES <input type="checkbox"/>	NO <input type="checkbox"/>

3. Future use and reuse of the information by others

I give permission for my anonymised feedback to be used for academic purposes only. It will not be shared with third parties and will be securely deleted after 12 months	YES <input type="checkbox"/>	NO <input type="checkbox"/>
I would like to receive a summary of the findings of this study.		
If YES, please provide your email address:	YES <input type="checkbox"/>	NO <input type="checkbox"/>
	YES <input type="checkbox"/>	NO <input type="checkbox"/>
	YES <input type="checkbox"/>	NO <input type="checkbox"/>

4. Signatures

Name of participant [in CAPITALS]	Signature	Date
(electronic signatures may be accepted)		

For participants unable to sign their name, please mark the box instead of signing

This research project conforms to and complies with the OU Human Research Ethics Committee's conditions for exemption from formal review.

Appendix D - Participation sheet



RESEARCH STUDY PARTICIPANT INFORMATION SHEET

Development and evaluation of a bus guidance application for St. Lucia

Name: Danell Mathurin
Email : danell722@outlook.com
Phone : +1 (758) 726-3534

Tutor email : judith.tope@open.ac.uk

Invitation to Participate

You are being invited to take part in a research study. Before you decide whether to take part, it is important for you to understand why the research is being done and what it will involve. Please take time to read the following information carefully.

You have been chosen because you are a user of public buses in St. Lucia on the Dennery to Castries route, and your input is valuable to help improve the use of public transportation in the country. Participation is entirely voluntary, and whether you take part will not affect any rights or services available to you.

Purpose of study and overview

The purpose of this research is to develop and test a mobile application that allows users to track public buses in real time in St. Lucia. The study involves user testing and feedback collection via short interviews. The study will run over a period of 6 months.

What you will be asked to do

Participants will be asked to:

- Provide feedback on usability and experience (approximately 5–15 minutes per session).

Participation is entirely voluntary. You can choose whether to take part and may withdraw at any time. By participating, you will help improve public transport services and provide valuable feedback for future application development.

Disadvantages and risks:

There are minimal risks involved. Participation requires only a short commitment of your time. No personal financial costs are involved.

Data usage and confidentiality

- Data will be stored securely on the researcher's personal computer, which is password-protected and accessible only to the researcher. All data will be anonymised before analysis.
- Personal identifiers will not be shared, and survey responses will be anonymised.
- Data may be used in the final project report, presentations, or academic publications but will not identify individual participants.
- Data will be retained for 12 months and then securely deleted.

Your right to withdraw from the study

You may withdraw at any time by contacting the researcher at danell722@outlook.com. You may also request removal of your data until 30th August 2025.

How to participate

To participate, you can complete the consent form provided by the researcher after confirmation of your participation.

Closing remarks

Thank you for taking the time to read this information sheet and for considering participation in this study. Your feedback will help improve public transportation in St. Lucia.

Appendix E - Log- only what is spoken about in reflection and review is required

Date	Work done	Issues encountered	How overcome	Sources of info used	Plan for next time
02/04/25	Looked at similar application	I needed to log in to access some of them	Continues searching for does which didn't require log in looked at pictures of those which required log in	(Moovit, 2014) (Transit Inc, 2013) (Citymapper Limited,2013)	Delve deeper into the functionality
05/04/25	Create a potential sketch of the user interface	Not familiar with drawing software	Decided to do it physically (Pen and paper)	None	Briefly learn a sketching software and place the sketch there. (easier to adjusted)
06/04/25	Had a conversation with two passengers	Potential bias involved from their perspective	Asked them to be as honest a possible	None	Have question in mind to ask rather than a causal conversation
08/04/25	Searched for online resources with information related to the current bus system	There was not much information available online	Had to use my own experience and find out information manually.	https://www.youtube.com/watch?v=o40UsQ6I30U https://www.facebook.com/100024153592606/posts/1736743986365463/	Have the mind set of find my own information rather than search online, it will most likely lead to a waste of time
12/04/25	Chose a project life cycle	Understanding which one would be the best for this project.	Utilized online sources along with material for Open University	https://learn2.open.ac.uk/mod/oucontent/view.php?id=2456473&section=5 https://www.projectsmart.co.uk/agile-project-management/which-life-cycle-is-best-for-your-project.php#google_vignette	Deeper comparison
26/04/25	Thought – if the system goes live a collaboration can be made with the government to provide the driver and their vehicle details. So, it would not be necessary to implement a login screen especially since payment would not be made through the application	---	---	---	---
1/05/25	Set up android studio	-the instructions given on expo documents were for an older version -I got the emulator to work but it was very slow -I tried another one but it crashed	-I found some help on YouTube -Also through expo documents - I downloaded an older version - I lowered the configurations of the emulated device	https://www.youtube.com/watch?v=y6DwGxe2Ek https://docs.expo.dev/get-started/set-up-your-environment/ https://developer.android.com/studio	Start implementing the design.
6/5/25	Implementing bottom sheet	-I was getting an error related to gesture handling	Decided to utilize a different library	https://docs.swmansion.com/react-native-reanimated/examples/bottomsheet/	Start using gorhom bottom sheet.
17/6/25	Decided to take an intentional break from work	Could potentially setback the entire project	Ask for an extension		
3/7/25	I contacted my tutor concerning how in depth I should do my project	I was indeed trying to do too much	I was told to focus on showing the principal concept rather than going in full depth		

Appendix F- Keeping In Touch – only what is spoken about in reflection and review is required

The screenshot shows two emails exchanged between Danell Mathurin and Judith.Tope.

Email 1 (From Danell Mathurin to Judith.Tope):

- From:** Danell Mathurin <danell722@outlook.com>
- To:** Judith.Tope <judith.tope@open.ac.uk>
- Subject:** Re: TM470 Danell Mathurin
- Date:** Mon 5/5/2025 7:39 PM
- Message:** Hi Ju, This is my progress update, it was supposed to be sent on Saturday the 2nd of May.
...
- Attachment:** update.docx (28 KB)

Email 2 (From Judith.Tope to Danell Mathurin):

- From:** Judith.Tope <judith.tope@open.ac.uk>
- To:** You
- Subject:** Re: TM470 Danell Mathurin
- Date:** Tue 5/6/2025 3:02 AM
- Message:** better late than never
- Cheers**
Ju

Message Headers:

```
From: Danell Mathurin <danell722@outlook.com>
Sent: 06 May 2025 00:39
To: Judith.Tope <judith.tope@open.ac.uk>
Subject: Re: TM470 Danell Mathurin
```

Report Phishing: External email: if the sender or content looks suspicious, please click the Report Message icon, or forward it to report-phishing

Figure F1: Email extract 1 (Mathurin, 2025)

Danell Mathurin
To: Judith.Tope

update.docx 26 KB

Good day Ju,
My 2-week update.

J Judith.Tope <judith.tope@open.ac.uk>
To: You

why leave out login page? remember to justify decision when you do next tma
Cheers
Ju

From: Danell Mathurin <danell722@outlook.com>
Sent: 21 May 2025 15:15
To: Judith.Tope <judith.tope@open.ac.uk>
Subject: Re: TM470 Danell Mathurin

External email: if the sender or content looks suspicious, please click the Report Message icon, or forward it to report-phishing

Figure F2: Email extract 2. (Mathurin, 2025)

DM Danell Mathurin
To: Judith.Tope
okay will do.

1. Did you want a reason for leaving out the login page now as well?

2 . Also, from Tma01 you mentioned that I didn't specify what was I was doing in each increment.
should I make the adjustment to the plan diagram or is creating a table which shows what would happen an option as well?

3. Also, in Tma01 there was an approach to solution section in relation to the Evaluation criteria I did it completely wrong; to verify I should have written my approach to the solution for each criteria ?

Danell.

...

DM Danell Mathurin
To: Judith.Tope
Continuation from previous email

4. The work done should go into detail of all what have done so far, coding, research, skill development and communication with stake holders?

...

J Judith.Tope<judith.tope@open.ac.uk>
To: You

1 yes
2 better clearly in plan, make sure I can read it; you could put a table of activities in an appendix but have it all in pplan
3 yes
4 yes

Cheers
Ju

From: Danell Mathurin <danell722@outlook.com>
Sent: 21 May 2025 16:46
To: Judith.Tope <judith.tope@open.ac.uk>

Figure F3: Email extract 3 (Mathurin, 2025)

 Danell Mathurin
To: Judith.Tope

[Reply](#) [Reply all](#) [Forward](#) | [Print](#) | ...
Wed 5/21/2025 12:09 PM

Why I omitted the login page

If the application were to be deployed, it would most likely be in collaboration with the transportation department so there would not be a need for a driver to log in and add themselves to the system if all records are already available from the transportation department.

Also, the main reason why authentication would be required for the passengers was if payment would be implemented which it would not be for this version of the application.

So given the required features it can be achieved without a login page.

...

 Judith.Tope<judith.tope@open.ac.uk>
To: You

[Reply](#) [Reply all](#) [Forward](#) | [Print](#) | ...
Wed 5/21/2025 1:24 PM

ok, explain this in 'changes to scope' in main report

Cheers
Ju

From: Danell Mathurin <danell722@outlook.com>
Sent: 21 May 2025 17:09
To: Judith.Tope <judith.tope@open.ac.uk>
Subject: Re: TM470 Danell Mathurin

External email: if the sender or content looks suspicious, please click the Report Message icon, or forward it to report-phishing

...

 Danell Mathurin
To: Judith.Tope

[Reply](#) [Reply all](#) [Forward](#) | [Print](#) | ...
Wed 5/21/2025 1:41 PM

Ok, will do. Thanks!

...

Figure F4: Email extract 4. (Mathurin, 2025)



Danell Mathurin
To: Judith.Tope

Reply | Reply all | Forward | | ...
Thu 7/3/2025 8:36 AM

Hi Ju,

I feel like I've been trying to develop this application with precise accuracy to the real-world situation, but I'm starting to wonder if that level of detail is actually necessary.

For example:

- I chose the route I'm using because I'm familiar with it and I know the general pricing. But I've noticed there are a lot of factors in the real world that make things more complex for example, the stops aren't clearly defined, and passengers can request to stop almost anywhere.
- Should I be accounting for things like route obstructions and the use of alternative routes, or would it be acceptable to simplify those areas for the sake of the project?

Also, is it okay if I make certain assumptions and simply explain how I would have handled more complex situations? If so, I'd like to know how many assumptions are considered acceptable before it becomes too much or starts to weaken the project.

I just want to make sure I'm focusing on the right areas and not overcomplicating the project beyond what's expected.

Thanks.

...



Judith.Tope<judith.tope@open.ac.uk>
To: You

Reply | Reply all | Forward | | ...
Thu 7/3/2025 8:55 AM

just stress you are doing one route as a proof of concept

do not even mention that obstructions or diversions might occur, assume all will be fine

just make out there is one stop per road in the route (or two on a long road) then passengers will be able to judge how long before next bus will come even if bus flagged down randomly

try not to make many assumptions otherwise the proof of concept will disappear

Cheers
Ju

Figure F5: Email extract 5 (Mathurin, 2025)



Danell Mathurin
To: Judith.Tope

Okay thanks you ,

Also, when you mention one route should I consider points A to B as one route.
Or would A to B and B to A be considered the one route?

...



Judith.Tope<judith.tope@open.ac.uk>
To: You

out and back. so A > B > A

Cheers
Ju

From: Danell Mathurin <danell722@outlook.com>
Sent: 03 July 2025 14:26
To: Judith.Tope <judith.tope@open.ac.uk>
Subject: Re: TM470 Danell Mathurin

External email: if the sender or content looks suspicious, please click the Report Message icon, or forward it to report-phishing

...



Danell Mathurin
To: Judith.Tope

Got it, thanks!

Figure F6: Email extract 6 (Mathurin, 2025)

Appendix G - Increment 1 testing check list (OpenAI ChatGPT, 2025c)

1. Navigation & Structure

- Can users find key features easily?
- Is the navigation flow intuitive (no one gets lost)?
- Are similar functions grouped logically?

2. User Interface & Design

- Are fonts, colors, and buttons consistent across screens?
- Does the app support accessibility (contrast, text size, screen reader)?

3. Functionality

- Do all features work without bugs or glitches?
- Are feedback indicators (loading, success messages, errors) shown clearly?

4. Content & Information

- Is text clear, concise, and easy to understand?
- Is information presented in a logical order?

5. Interaction & Engagement

- Do all buttons, links, and gestures respond correctly?
- Are there engagement hooks (personalized content, rewards)?

6. Performance & Speed

- Does the app load quickly on different network conditions?
- Do animations and transitions run smoothly without lag?

Appendix H - Generative AI

Prompt 1 (OpenAI ChatGPT, 2025a)

I want to use Figma to design a UI for the first time I want tips and trick which would be essential to help complete this.

Response 1

Great! If you're designing a UI in **Figma for the first time**, here are **essential tips and tricks** to help you work smarter and finish your UI successfully:

🎨 **Getting Started**

1. **Start with a Frame (Artboard)**

- * Press `F` to choose a frame size (like iPhone, Desktop, etc.).
- * Frames are your screen canvases.

2. **Use Pre-built UI Kits**

- * Go to **Community** tab in Figma and search “UI kit” or “Material Design.”
- * These save tons of time—just drag and edit components.

☑ **Efficient Design Practices**

3. **Use Auto Layout (Shift + A)**

- * Lets your buttons, cards, and sections resize automatically.
- * Acts like Flexbox in CSS—helps make responsive UIs.

4. **Create Components**

- * Turn repeated elements (like buttons, navbars) into components.
- * Reuse and update them everywhere instantly.

5. **Use Grids & Layouts**

- * Add layout grids to frames (for alignment and consistency).
- * Ideal for web and mobile screen structure.

Design Tips

6. **Stick to a Design System**

- * Use consistent font sizes, colours, and spacing.
- * Create styles for text (`T`) and colours for reusability.

7. **Use Icons from the Figma Community**

- * Search for free icon packs and drag them into your project.

8. **Keep It Simple**

- * Design one screen at a time. Start with wireframes (simple boxes), then add visuals later.

Finishing Touches

9. **Use Prototyping (Present Mode)**

- * Link screens using the `Prototype` tab.
- * Press `Play`  to test your UI flow.

10. **Export Assets**

- * Right-click → `Export` (e.g., PNG, SVG) for developers or use **Figma Inspect** for code hints.

Bonus: Quick Shortcuts

Action	Shortcut
Frame	`F`
Text Tool	`T`
Rectangle	`R`
Duplicate	`Cmd/Ctrl + D`
Quick Action Search	`Cmd/Ctrl + /`

Would you like a simple step-by-step checklist for designing your first screen (like a login page)?

Prompt 2 (OpenAI ChatGPT, 2025b)

Using this image create a cartoon style bus image in black and white.



Response 2



Prompt 3 (OpenAI ChatGPT, 2025b)

Now add colour to the image with a St. Lucia theme.

Response 3 (OpenAI ChatGPT, 2025b)



Prompt 4 (OpenAI ChatGPT, 2025d)

I want to create a simple database in mongodb can you give me a study plan to learn the essentials to complete this

Response 4

Great! Here's a MongoDB Atlas–focused study plan to help you learn the essentials and create a simple database with two collections. This plan assumes you're new to MongoDB but want quick, focused learning.

31] 3-Day MongoDB Atlas Study Plan (Essentials Only)

 **Goal:** Create a simple MongoDB Atlas database with 2 collections and understand how to use it through the GUI and basic queries.

Day 1: Intro & Setup

Objectives:

- Understand what MongoDB Atlas is
- Set up your cloud database

Steps:

1. Create a MongoDB Atlas account at <https://www.mongodb.com/cloud/atlas>
2. Create a Free-tier cluster (choose AWS, region closest to you)
3. Create your first database (e.g., schoolDB)
4. Add two collections (e.g., students and courses)
5. Learn basic MongoDB concepts:
 - Database → Collection → Document
 - JSON/BSON format

Practice:

- Add documents manually using the Atlas UI

Resources:

- MongoDB official [Quickstart Guide](#)
-

Day 2: CRUD Operations via GUI

Objectives:

- Learn to manage data inside your database (without code)

Topics:

- Create: Insert Document
- Read: Find, use filters
- Update: Modify fields
- Delete: Remove documents

Practice:

- Insert sample student records:

json

CopyEdit

{ "name": "John", "age": 20, "course": "Math" }

- Query to find all students in "Math"
- Update a student's age
- Delete one document

Resources:

- [Atlas UI Docs](#)
-

Day 3: Data Relationships & Export/Import

Objectives:

- Understand basic structure for multiple collections

- Know how to connect data (embedded or referenced)

Topics:

- Difference between embedded docs vs references
- How to model relationships (e.g., student → course)
- Use Export/Import tool from Atlas UI to back up or move data

Practice:

- Add a few documents to courses

json

CopyEdit

```
{ "course_name": "Math", "course_code": "MTH101" }
```

- Link students to courses by course code or embedded info
-

Final Goal:

You'll have a working cloud database in Atlas with two collections and some basic skills to:

- Add, edit, delete, and query documents
- Model simple relationships
- Use MongoDB Atlas's GUI effectively

Prompt 5 (OpenAI ChatGPT, 2025d)

I have a mongodb database and I want to use node.js and express.js to connect it to my application can you give me the step by step assistance

Response 5

Absolutely! Here's a **step-by-step guide** to help you connect your MongoDB Atlas database to a Node.js + Express.js application.

Prerequisites

- Node.js installed (node -v)
 - MongoDB Atlas database created
 - Your database connection string (from Atlas)
-

Step 1: Initialize Your Node.js Project

bash

CopyEdit

mkdir my-app

cd my-app

npm init -y

Step 2: Install Required Packages

bash

CopyEdit

npm install express mongoose dotenv

- express: for your web server
- mongoose: to interact with MongoDB

- dotenv: to manage your database URI securely
-

Step 3: Project Structure

pgsql

CopyEdit

my-app/

```
|── .env  
|── index.js  
└── models/  
    └── User.js
```

Step 4: Create .env File

env

CopyEdit

MONGODB_URI=mongodb+srv://<username>:<password>@<cluster>.mongodb.net/<dbname>?retryWrites=true&w=majority

Replace the <username>, <password>, <cluster>, and <dbname> with your real values from MongoDB Atlas.

Step 5: Setup Mongoose & Express (index.js)

js

CopyEdit

```
require('dotenv').config();  
  
const express = require('express');  
  
const mongoose = require('mongoose');
```

```
const app = express();

app.use(express.json()) // for parsing JSON

// Connect to MongoDB
mongoose.connect(process.env.MONGODB_URI, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
})

.then(() => console.log("✓ Connected to MongoDB"))

.catch(err => console.error("✗ MongoDB connection error:", err));

// Sample route
app.get("/", (req, res) => {
  res.send("MongoDB + Node.js API is working!");
});

const PORT = process.env.PORT || 3000;

app.listen(PORT, () => console.log(`📡 Server running on port ${PORT}`));
```

📋 Step 6: Create a Model (models/User.js)

js

CopyEdit

```
const mongoose = require('mongoose');
```

```
const userSchema = new mongoose.Schema({  
  name: String,  
  email: String,  
});  
  
module.exports = mongoose.model('User', userSchema);
```

Step 7: Add a Route to Use the Model

Update index.js:

js

CopyEdit

```
const User = require('./models/User');
```

```
app.post("/users", async (req, res) => {  
  try {  
    const user = new User(req.body);  
    await user.save();  
    res.status(201).send(user);  
  } catch (err) {  
    res.status(400).send({ error: err.message });  
  }  
});
```

```
app.get("/users", async (req, res) => {
```

```
const users = await User.find();
res.send(users);
});
```

Step 8: Run Your App

bash

CopyEdit

node index.js

Open <http://localhost:3000/> — you should see "MongoDB + Node.js API is working!".

You can use **Postman** or **cURL** to test your /users POST and GET endpoints.

Prompt 6 (OpenAI ChatGPT, 2025f)

Can you create an icon which represents a bus stop for a mobile app using the colours blue and white?

Response 6



Appendix I -Previous Plan

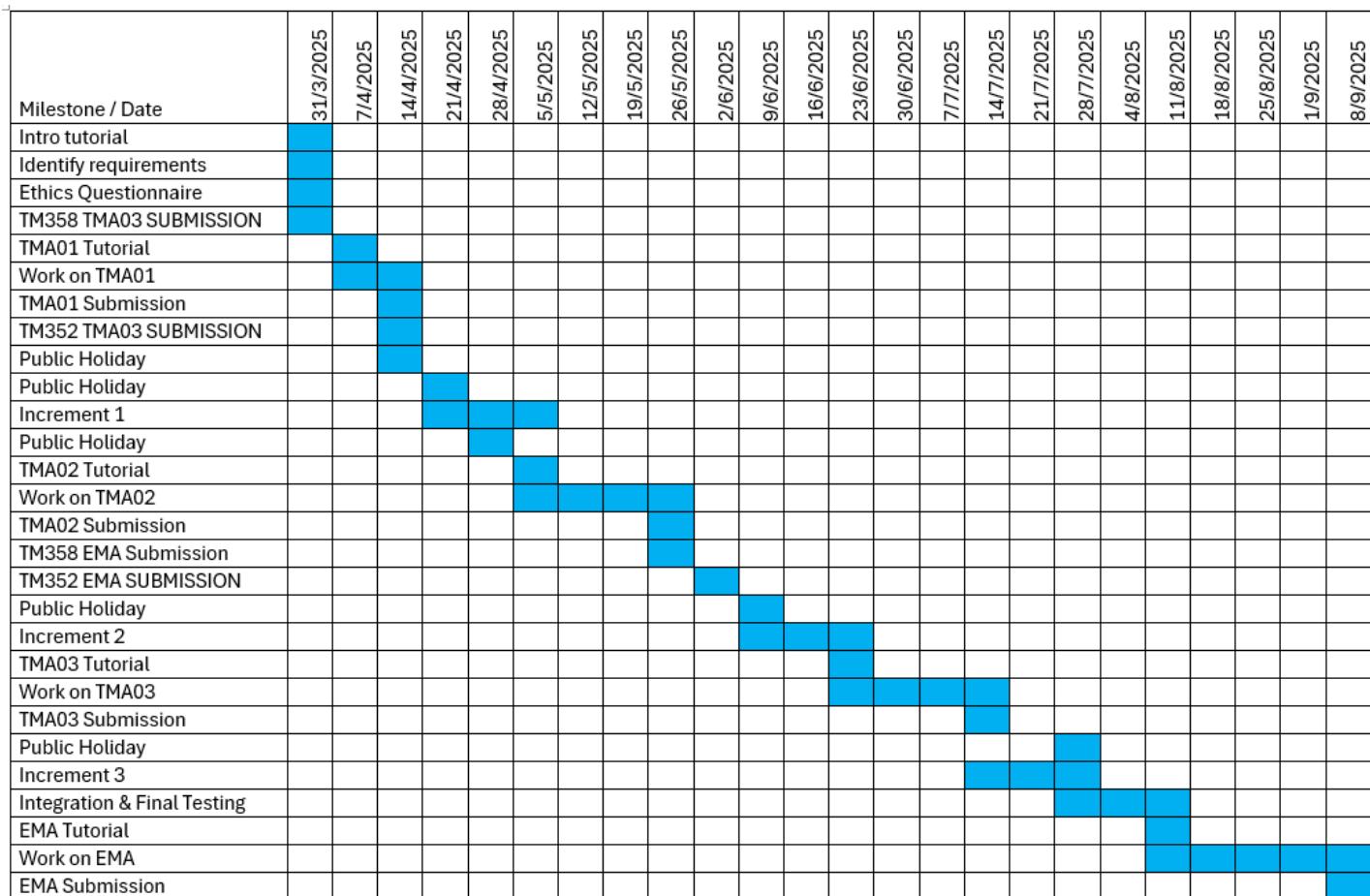


Figure II: Previous plan. (Mathurin, 2025)