

# MyChildBus: Real-Time School Bus Monitoring and Rides Tracking Mobile Application for Parents

Fatin Nabilah Zulkaria<sup>1</sup>, Rozanawati Darman<sup>1\*</sup>

<sup>1</sup> Faculty of Computer Science and Information Technology,  
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

\*Corresponding Author: [zana@uthm.edu.my](mailto:zana@uthm.edu.my)

DOI: <https://doi.org/10.30880/aitcs.2025.06.02.007>

## Article Info

Received: 13 June 2025

Accepted: 3 November 2025

Available online: 30 November 2025

## Keywords

Live location tracking, School bus,  
Mobile application, Firebase, Google  
Maps API, Prototyping model

## Abstract

The absence of real-time school bus tracking systems in Malaysia raises significant concerns among parents and school bus operators regarding student safety and operational inefficiencies. This project aims to address these issues by developing "MyChildBus", a mobile application that enables parents to track their children's school bus location in real-time. The proposed application is designed for parents, school bus drivers, and administrator from a school bus company as users, with modules such as registration and login, communication, live location tracking, route navigation, incident reporting and handling, student, driver and bus management, and reporting. Using the Prototyping Model, the app integrates Firebase for real-time database management, and Google Maps API for live location tracking. The results of functional testing and User Acceptance Testing (UAT) have proven the achievement of this app of becoming a user-friendly location tracking application fulfilling both of its functional and non-functional requirements. Hence, future work could be done is to expand this application capabilities to support multiple bus operators rather than being used by a single bus company only at a time.

## 1. Introduction

School bus is considered as an easy solution for children transportation to school for parents [1]. However, the absence of real-time location tracking results in delays, uncertainty, and wasted time for parents, students, and drivers, as they struggle to predict bus arrival times at pick-up points [2]. Furthermore, the inability to confirm whether students have boarded the bus or arrived safely at their destination increases parental anxiety [3]. The current system deployed shows communication between parents and drivers relies on phone calls and messaging applications like WhatsApp, that are prone to missed updates while the manual administrative processes increase the likelihood of errors and inefficiencies, complicating the management of driver assignments, student records, routes, and schedules.

To address the issues, this study proposes the development of a mobile application that integrates real-time location tracking using Google Maps API, with objectives to design a user-friendly mobile application interface that enables parents to track their children's school bus location in real-time and to develop a centralized database system to store students, drivers, and route information data. Hence, MyChildBus is developed for parents, school bus drivers, and an administrator as users, with modules such as account registration and login, parents and child profile, communication channel, route planning and route navigation, driver and bus management, student management, student attendance, incident reporting and handling, and reporting. These

modules work together to provide reassurance to parents, improve communication and operational efficiency, and build trust between parents and school bus operators, creating a mutually beneficial outcome for all users.

## 2. Related Work

This section will present the technology used and study on existing systems related to live location tracking that have been conducted by doing comparison between three related existing systems with the proposed system.

### 2.1 Technology Used

The technologies used in developing the proposed application are mostly focusing on integrating the live location tracking feature into Android's mobile application by using Google Maps API with Dart as the programming language, Flutter as the framework, and Firebase for database storage. Software such as Visual Studio Code and Android Studio are highly utilized as code editor and Integrated Development Environment (IDE) for developing and testing the Android application.

### 2.2 Study of Existing Related Systems

The three similar existing systems that have been studied are MorningBus, KidBus.Tracker, and GPS2u by Explosoft. MorningBus is a mobile application designed to track school buses' live location with separate apps developed for parents, administrator, and drivers [4]. The next similar system is KidBus.Tracker which is a web-based school bus tracking system designed to monitor school children's commute to school with the employment of passive Radio Frequency Identification (RFID) technology [5]. Lastly, similar to KidBus.Tracker, GPS2u by Explosoft also have GPS hardware installed in vehicles, but as a mobile-based system [6]. The comparison between MorningBus, KidBus.Tracker, GPS2u by Explosoft, and the proposed application, MyChildBus, are identified and tabulated in Table 1.

**Table 1** Comparison between existing systems and proposed application

Feature	MorningBus [4]	KidBus.Tracker [5]	GPS2u by Explosoft [6]	MyChildBus (proposed system)
1. Mobile app/ Web-based	Mobile app	Web-based	Mobile app	Mobile app
2. Login and sign-up module	Yes	Yes	Login only	Yes
3. GPS hardware installation	No	Yes	Yes	No
4. Live location tracking module	Yes	Yes	Yes	Yes
5. Communication channel	Call only	No	No	Personal and group chat
6. Display driver and vehicle details	Yes	Vehicle details only	Vehicle details only	Yes
7. Route planning and navigation (driver interface)	No	No	No	Yes
8. Passenger management module (admin interface)	Yes	No	No	Yes
9. Driver and vehicle management module	Yes	No	No	Yes
10. Record passengers' attendance	No	No	No	Yes
11. Incident handling module	No	No	No	Yes
12. Report generation module	Yes	Yes	Yes	Yes

Based on the comparison between MyChildBus with the existing systems as tabulated in Table 1, MyChildBus surpassed the other existing systems with its unique functionalities such as route planning and navigation for drivers, recording passengers' attendance, and incident handling module that are not found in either MorningBus, KidBus.Tracker, or GPS2u by Explosoft. Furthermore, MyChildBus also have all of the features that the other systems have such as communication channel, report generation module, and passenger, driver and vehicle management module for admin, showing that MyChildBus could provide a cost-effective solution while still equipped with the standard functionalities should be owned by a bus live location tracking mobile application.

### 3. Methodology/Framework

This section discusses the methodology used in developing the proposed mobile application, which is the Prototyping Model, along with the system development workflow involved in the Prototyping Model.

#### 3.1 Prototyping Model

The methodology chosen for this project is the Prototyping Model, which is particularly useful when user requirements are unclear [7]. The process begins with gathering user requirements, followed by creating a quick design and developing a prototype. This prototype is then shared with users for feedback, and the feedback loop continues to refine the system iteratively until users are satisfied [8]. Once the prototype is finalized, the actual development of the software begins [8], implementing all the refined requirements and undergoing maintenance. The summarization of the phases involved when implementing the Prototyping Model is depicted in Figure 1.

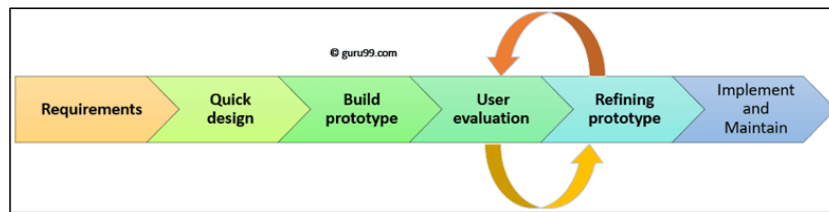


Fig. 1 Prototyping model [9]

#### 3.2 System Development Workflow

Each phase in the prototyping model employed has been dedicated to their own specific tasks and expected outputs throughout the system development workflow, as tabulated in Table 2. The system development workflow ensures that each phase, from requirements gathering to implementation, is systematically documented, aligning with the iterative nature of prototyping. This structured approach allows for continuous feedback and refinement, ensuring the final product meets user expectations.

Table 2 System development workflow

Phase	Tasks	Outputs
1. Requirements gathering	Study existing systems. Prepare interview questions and conduct interview with stakeholder. Define problem statement, objective, scope, and project significance in project proposal.	Project proposal. User and system requirements. Comparison table of the existing systems and proposed application.
2. Quick design	Design user interface. Create UML diagrams. Create database design.	Wireframes of user interface design. Use case diagram, sequence diagram, activity diagram, and class diagram.
3. Build prototype	Build user interface and mobile application modules.	Draft prototype.
4. User evaluation	Present module prototype to target users/ stakeholder. Functionality testing of prototype by target users/ stakeholder and collect feedback from users.	User feedback.
5. Refining prototype	Refine prototype based on user feedback.	New version of prototype.
6. Implementation and testing	Develop final application based on approved final prototype. Perform usability testing on the application. Identify and resolve issues during maintenance.	A complete application with overall functionality.

## 4. Analysis and Design

This section will document the deliverables gained from the analysis and design phase of this project that include the functional and non-functional requirements, use case diagram, sequence diagram, activity diagram, class diagram, and the interface design of the application.

### 4.1 System Requirement Analysis

System requirement analysis is one of the most important tasks to be conducted during analysis phase of the application development that identifies the functional and non-functional requirements of the application.

#### 4.1.1 Functional and Non-Functional Requirements

Functional requirements are the fundamental qualities that the system must provide as specifically requested by end-user [10]. Meanwhile, non-functional requirements define the essential criteria that a system must satisfy to fulfil the project contract's requirements [10] and represent the overall quality of a system [11]. The functional requirements and non-functional requirements of the application are tabulated in Table 3 and Table 4.

**Table 3** *Functional requirements of MyChildBus mobile application*

Modules	Functionalities
1. Registration and login module	The app should allow user to register account and log in to the application using the registered email and password.
2. Parents and child profile module	The app should allow parents to save profile and update details of parent and each child registered to the bus service.
3. Communication channel module	The app should enable chat between parents, drivers, and admin for inquiries and updates.
4. Bus live location tracking module	The app should allow parents and admin to track the live location of bus through a map view.
5. Student attendance module	The app should allow parents to notify child absence to drivers and update drivers with students attending and absent on current or future dates.
6. Route planning and route navigation module	The app should allow drivers to arrange starting points, order of student homes as stops, and destination personalized for every day's route and receive route navigation of the saved routes.
7. Driver and bus management module	The app should allow admin to add buses, add drivers, and manage specialized driver assignments to buses.
8. Student management module	The app should enable admin to manage registered students by assigning them to the available driver and bus and update available quota for each bus.
9. Incident reporting and handling module	The app should allow drivers to report incidents to admin, and admin should be able to respond to the incident.
10. Reporting module	The app should allow admin to generate reports related to bus operations, especially monthly and yearly student subscription record.

**Table 4** *Non-functional requirements of MyChildBus mobile application*

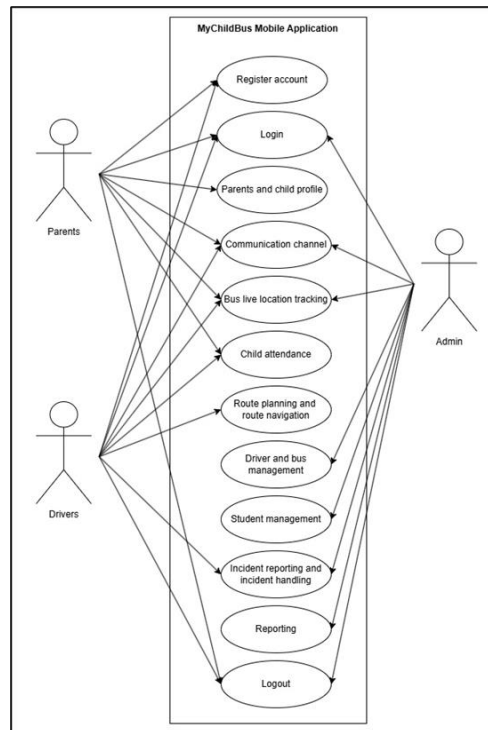
Requirements	Descriptions
1. Performance	The app should display accurate live location updates to users as long as it is connected with stable internet connection.
2. Operational	The Android app should run smoothly on Android devices.
3. Security	The app should only allow user login using registered email and correct password, encrypt all sensitive data, and implement access control based on user type.

## 4.2 Unified Modelling Language (UML)

UML diagrams that include use case diagram, sequence diagram, and activity diagram have been designed to visualize the proposed application architecture, providing clearer visualization of the system flow.

### 4.2.1 Use Case Diagram

A use case diagram has been designed as in Figure 2, showing parents, drivers, and admin are the actors or entities that will interact with the application while the use cases are the modules contained in the application.



**Fig. 2** Use case diagram for MyChildBus mobile application

### 4.2.2 Sequence Diagram

Sequence diagrams also have been used to illustrate the sequences of events involving different objects contained within the application. For the MyChildBus application, three separate sequence diagrams have been designed based on different user types as shown in Appendix A, Appendix B, and Appendix C.

### 4.2.3 Activity Diagram

Activity diagrams have been designed to provide the visual representation of the control flow of activities involved when user is using the MyChildBus mobile application. As MyChildBus mobile application is developed for three types of users with different interfaces and module access, an activity diagram is designed for each type of user as shown in Appendix D, Appendix E, and Appendix F.

## 4.3 Database Design

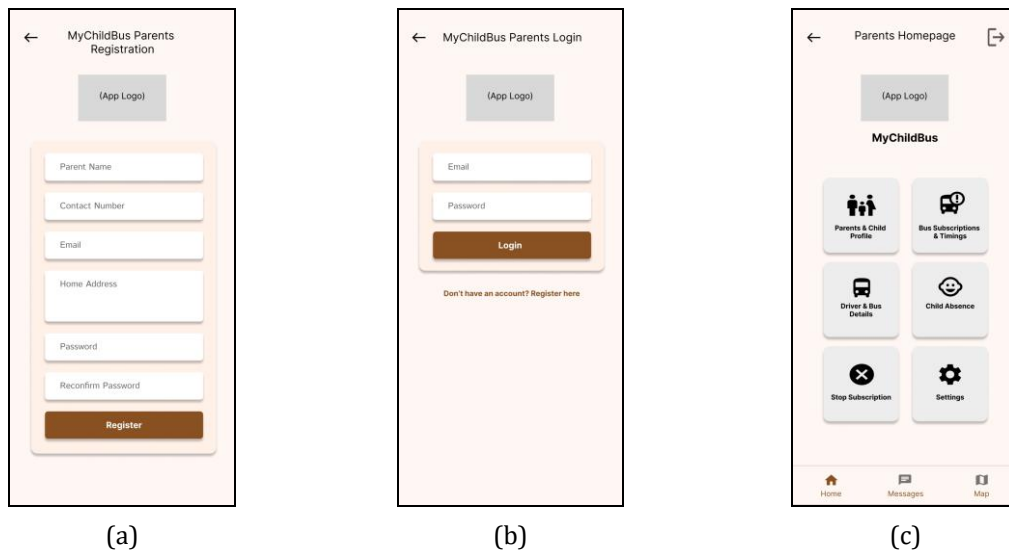
Designing database for MyChildBus mobile application involves designing a class diagram to provide the visual representation of the connection between different classes and attributes of this application.

### 4.3.1 Class Diagram

The designed class diagram for MyChildBus app consists of several classes, each representing a different entity in the system, is shown in Appendix G.

## 4.4 User Interface Design

The user interface design of pages at parents' interface have been designed, such as registration page as in Figure 3(a), login page as in Figure 3(b), and parent's homepage as in Figure 3(c).



**Fig. 3** Parent's (a)Registration Page; (b) Login Page; (c)Homepage

## 5. Results and Discussion

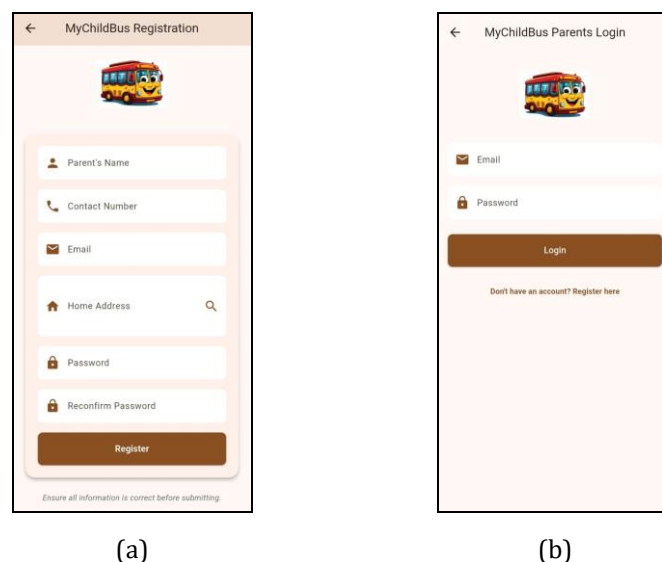
This section will present the implementation of the crucial modules inside MyChildBus app. The functionalities of these crucial modules are then tested, and the results of the testing will also be discussed in this section.

### 5.1 Implementation

The implementation of MyChildBus's modules utilized Visual Studio Code, Android Studio, Dart programming language, Flutter framework, Firebase for database storage, Google Maps API for live location tracking integration, and is deployed on an Android device.

#### 5.1.1 User Registration and Login Module

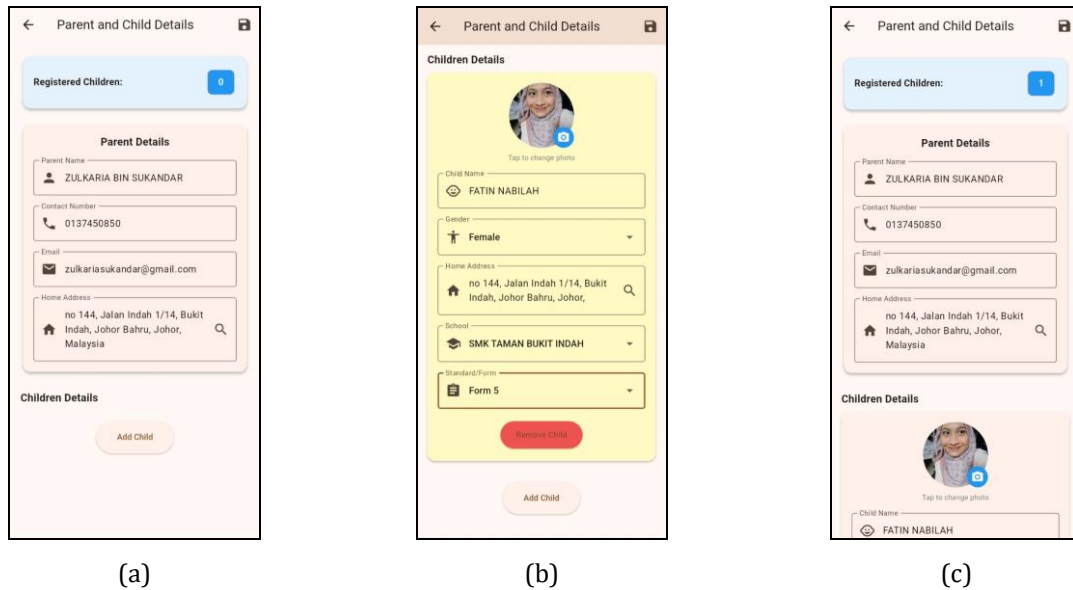
Parent's registration page as in Figure 4(a) require parent's details such as name, contact number, email, home address, and password. Input validation such as requiring valid Malaysian phone number with 9 to 10 digits for the field phone number, valid email address for the field email, password at least 8 characters long that must include alphanumerical, uppercase, lowercase and special character, and Google Maps API's autodetection and autocomplete address integration for the field home address are implemented to minimize human input error and safeguard user account with strong password. After successful account creation, parents could login to the application by entering the registered email and password on the login page as in Figure 4(b).



**Fig. 4** (a)Parent's Registration Page; (b) Parent's Login Page

### 5.1.2 Parent and Child Profile Module

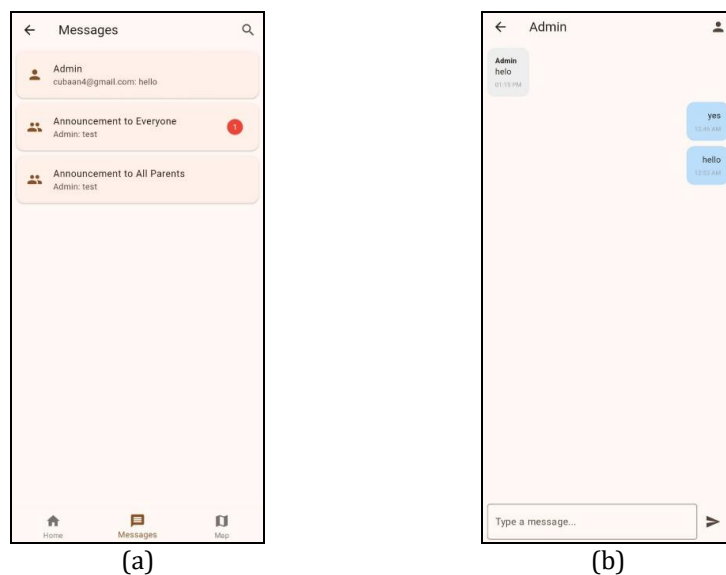
Figure 5(a) depicts the Parents and Child Details page, that display and allow edits on the parent details. On this page, parents also able to add and update details of child to be registered to the school bus service (refer Figure 5(b)). The number of registered children displayed will be updated automatically as in Figure 5(c) after adding any child.



**Fig. 5** (a) Parent and Child Profile Page displaying parent’s details; (b) Adding child information to the Parent and Child Profile Page; (c) Child’s information saved successfully.

### 5.1.3 Communication Channel Module

Figure 6(a) and (b) shows communication channel module that is equipped with personal chat rooms and announcement rooms, creating platform for more efficient communication between parents, admin, and drivers.

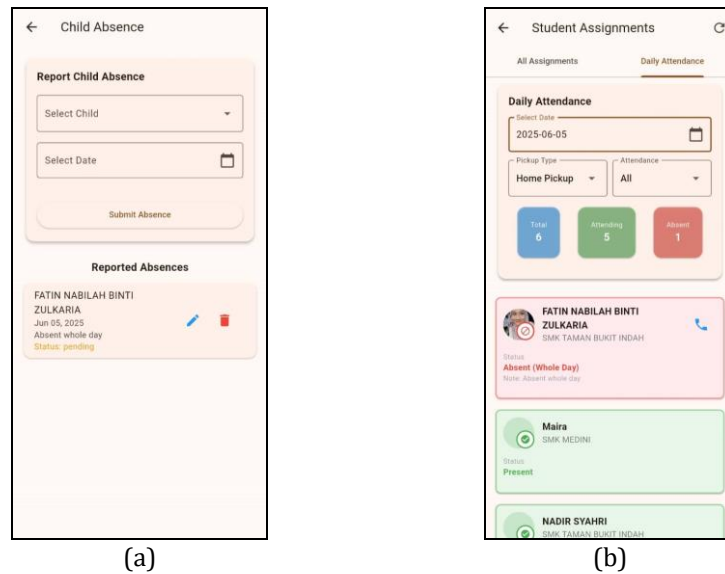


**Fig. 6** (a) Chat rooms in communication channel; (b) Personal chat room

### 5.1.4 Student Attendance Module

Figure 7(a) shows Child Absence page at parent’s interface and Figure 7(b) shows “Daily Attendance” tab at driver’s Student Assignments page that build up the student attendance module in the application. Through the Child Absence page, parents could report any child’s absence by selecting child, date of attendance, and type of

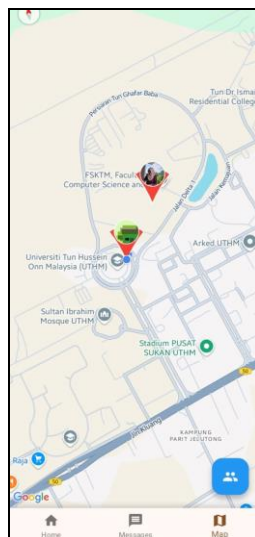
pickup the child would be absent on that day. The data will then be sent to the driver's Student Assignment page that allows driver to filter assigned students on any specific date based on students' attendance status for any pickup type through the "Daily Attendance" tab. Through this tab, the assigned students are displayed with clear indication of their attendance status either present or absent on that day (refer Figure 7(b)).



**Fig. 7** (a) Parent's Child Absence page; (b) "Daily Attendance" tab at driver's Student Assignments page

### 5.1.5 Bus's Live Location Tracking Module

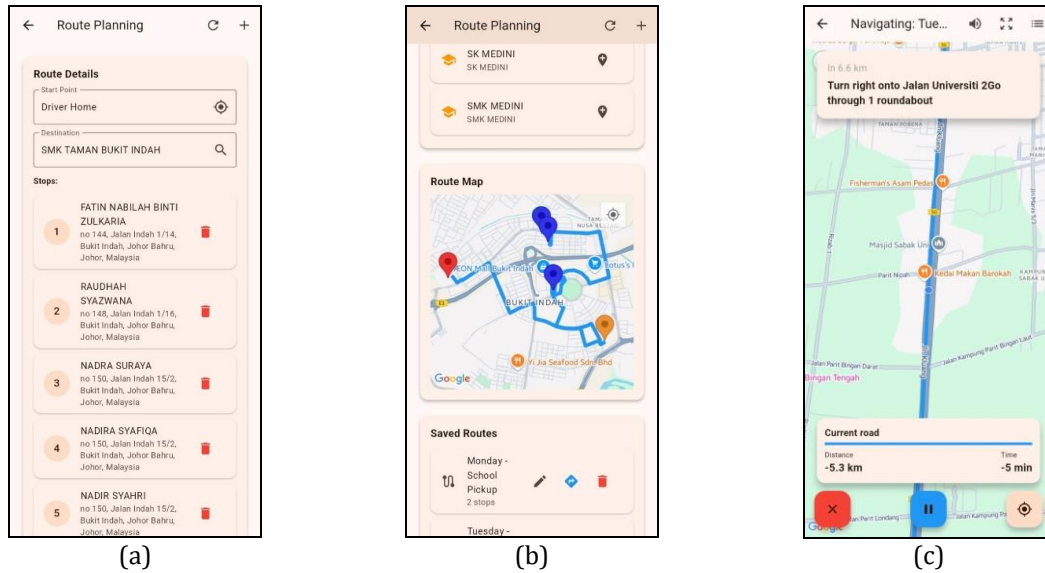
Figure 8 shows the page that updates the live location of school buses in real time with integration of Google Maps API. Parents will be able to track the live location of drivers assigned to their children through this page with the profile photo of each driver assigned as markers on the map.



**Fig. 8** Bus live location tracking page.

### 5.1.6 Route Planning and Route Navigation Module

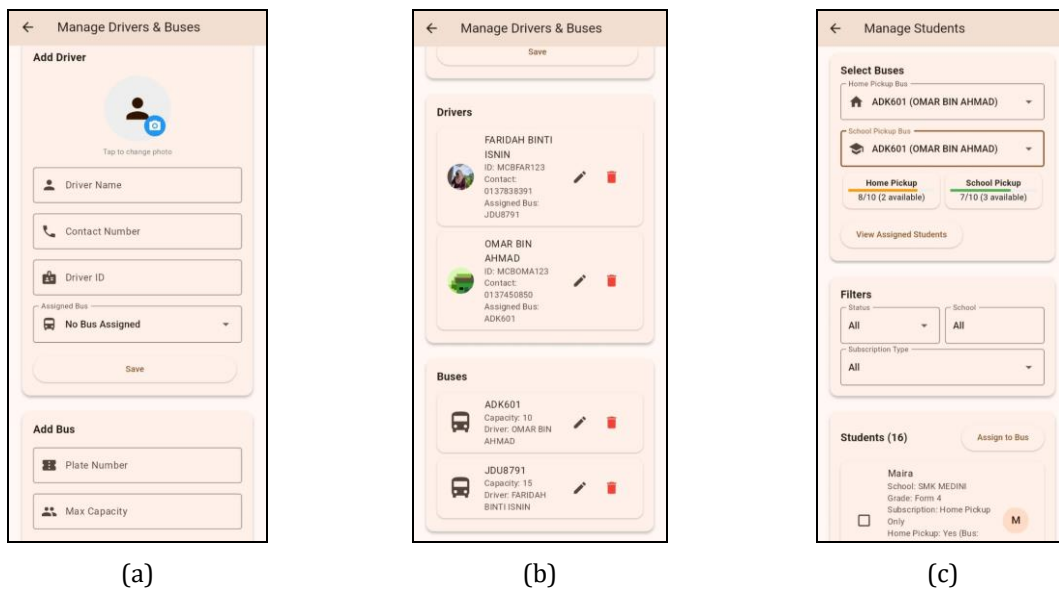
The route planning and route navigation module for school bus drivers enable drivers to plan daily route by choosing the starting point and destination and arranging the order of stops along the route that consists of the students' homes and schools (refer Figure 9(a)). The polylines that consist of all the starting point, stops, and destination will be drawn on the map at this page to enable driver plans the best route for each day (refer Figure 9(b)). Route navigation feature also be included in all saved routes to facilitate driver navigating to each student's home, as depicted in Figure 9(c).



**Fig. 9** (a) Selecting start point, order of stops, and destination for route planning; (b) Route plan displayed on map; (c) Route navigation for saved routes.

### 5.1.7 Driver, Bus, and Student Management Module

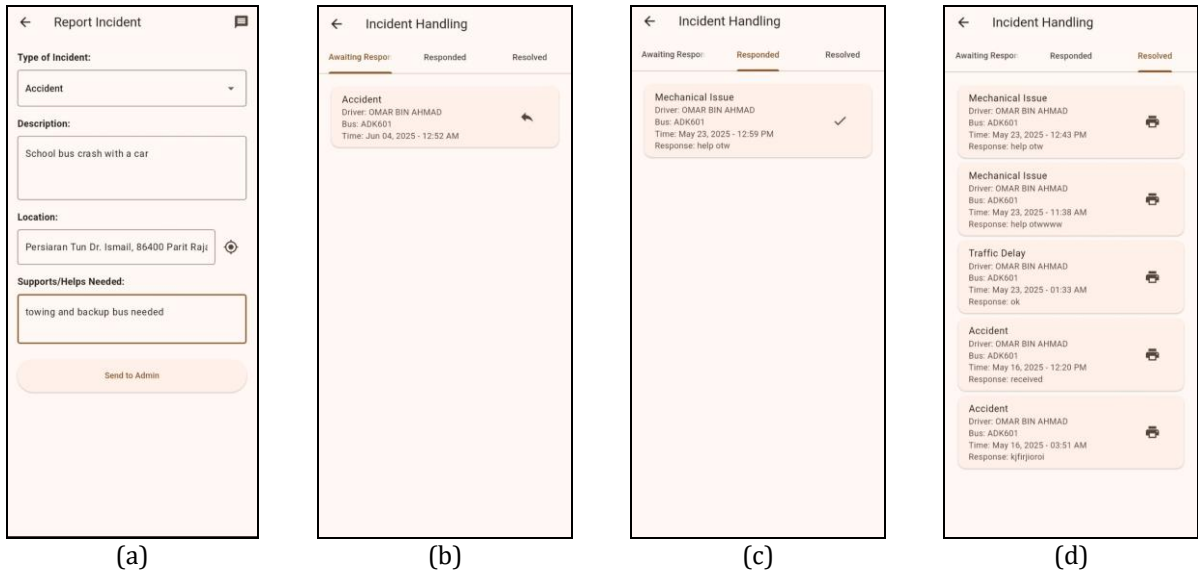
Figure 10(a) and (b) shows the driver and bus management module that allow admin to add driver, add bus, and assign driver to specific bus. Student management module as in Figure 10(c) enable admin to assign registered students to the available driver and bus for each pickup type based on the available quota available for the bus.



**Fig. 10** (a) Adding driver or bus in driver and bus management page; (b) List of drivers and buses displayed; (c) Managing students' assignment to bus drivers through student management module.

### 5.1.8 Incident Reporting and Handling Module

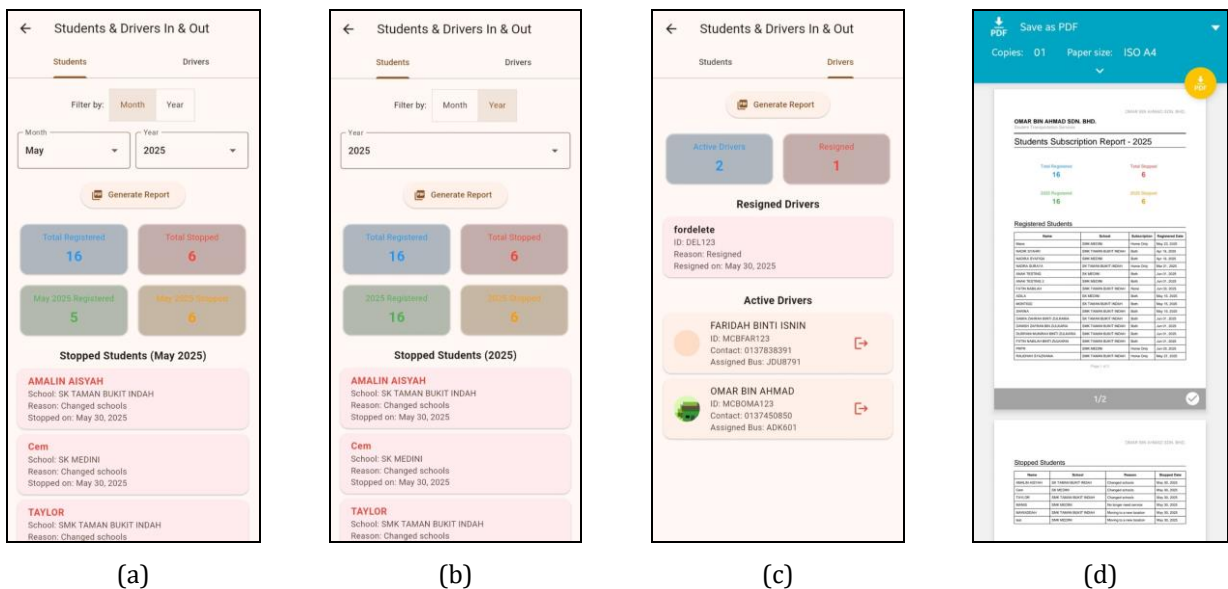
Figure 11 (a) shows the incident reporting page for driver to report any incident encountered during the journey to admin and Figure 11 (b), (c), and (d) shows incident handling page for admin to receive and respond to the incident reported by drivers. Admin can respond to the incident reported by drivers, mark any incident as resolved, and generate and download or print the report of the incident.



**Fig. 11** (a) Driver's incident reporting page; (b) Admin's incident handling page (incident awaiting response tab); (c) Admin's incident handling page (incident responded tab); (d) Admin's incident handling page (incident resolved tab)

### 5.1.9 Reporting Module

Reporting module is implemented in the admin's 'Students & Drivers In & Out' page that summarize the report on the lists and total number of students registered and stopped the subscription of the school bus service in each month (refer Figure 12(a)) and year (refer Figure 12(b)) in the tab 'Students', and summarize report on active and resigned drivers in tab 'Drivers' (refer Figure 12(c)). The report generated can be downloaded and printed by admin as in Figure 12(d).



**Fig. 12** (a) Monthly students' subscription report; (b) Annual students subscription report; (c) Drivers' status report; (d) PDF report generated could be downloaded and printed.

## 5.2 Testing

In this section, testing including functionality testing and user acceptance testing are done to ensure that the app developed has fulfilled all the requirements, serve all its functionalities and ready to be launched to target users.

### 5.2.1 Test Plan

Functionality testing on the MyChildBus’s modules are conducted by undergoing all the test cases dedicated to each module in the test plan and observing the result of each test case tested. The test plan results for each module are tabulated in Table 5.

**Table 5** Test Plan Results

Module	Test Cases	Expected Output	Result
1. Registration and Login Module	(i) First-time user registers an account with valid information.	Account created successfully.	Pass
	(ii) User logs in with registered email and password.	User logs in successfully and redirects to the homepage.	Pass
	(iii) Parent logs in with incorrect credentials.	Error message displayed indicating invalid credentials.	Pass
2. Communication Channel Module	(i) User sends text messages in personal chat with any other type of user.	Message appears in the chat room and received by the receiver.	Pass
	(ii) Admin sends text messages in the announcement room.	Message appears in the chat room and received by the announcement room’s participants.	Pass
3. Bus Live Location Tracking Module	(i) Driver toggles location sharing.	Driver’s location updated in real-time as location sharing enabled.	Pass
	(ii) Parent views assigned driver’s live location on the map.	All assigned drivers’ live locations are displayed on the map.	Pass
4. Parents and Child Profile Module	(i) Parent edits any parent’s detail and save.	The updated information saved successfully.	Pass
	(ii) Parent adds new child profile by filling in all required information.	Child profile created and saved successfully. Registered children counter automatically updated.	Pass
5. Student Attendance Module	(i) Parent reports any child’s absence through Child Absence page.	Absence recorded and updated to the assigned driver.	Pass
	(ii) Driver opens “Daily Attendance” tab in the Driver Assignments page and filter assigned students on any specific date based on attendance status.	The assigned students are displayed with clear indication of their attendance status either present or absent on that day.	Pass
6. Driver and Bus Management Module	(i) Admin add new driver.	New driver added successfully.	Pass
	(ii) Admin edits any existing driver’s details.	Driver information updated successfully.	Pass
	(iii) Admin deletes any driver.	Driver deleted successfully.	Pass
	(iv) Admin add new bus.	New bus added successfully.	Pass
	(v) Admin edits any existing bus’s details.	Bus information updated successfully.	Pass

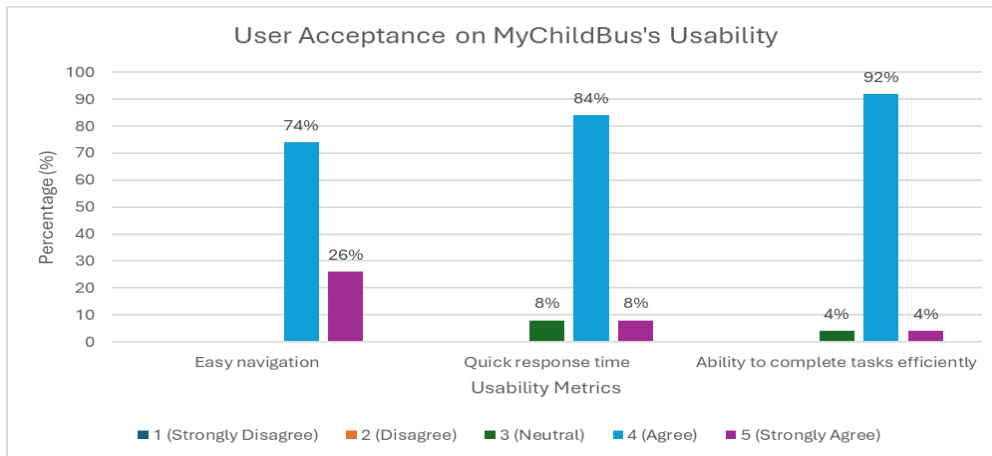
**Table 5** Test Plan Results (cont.)

	(vi)	Admin deletes any bus.	Bus deleted successfully.	Pass
	(vii)	Admin assigns driver to a bus.	Driver's bus assignment saved successfully.	Pass
7. Student Management Module	(i)	Admin assigns students to any available bus for home or school pickup.	Students successfully assigned to the bus and available quota of the bus automatically updated.	Pass
	(ii)	Admin attempts to assign students to any full bus.	Error message displayed indicated that the bus already full and the student could not be assigned to the bus.	Pass
	(iii)	Admin unassign student from bus.	Student removed from the bus assignment.	Pass
8. Route Planning and Route Navigation Module	(i)	Driver plans a route for any day and pickup type with start point, stops, and destination.	Route is saved and polylines are displayed on the map.	Pass
	(ii)	Driver reorders stops in the route.	Stop order and route is updated.	Pass
	(iii)	Driver starts navigation for any saved route.	Driver redirected to the route navigation page and assisted with turn-by-turn directions.	Pass
9. Incident Reporting and Handling Module	(i)	Driver reports any incident.	Incident record submitted successfully to admin.	Pass
	(ii)	Admin views pending incidents and responds.	Responses sent to driver and driver able to mark any responded incident as resolved	Pass
	(iii)	Admin generates incident report.	Incident report generated and could be printed or downloaded as PDF.	Pass
10. Reporting Module	(i)	Admin filters students' subscription report data by month or year.	Reports displayed based on the applied month or year.	Pass
	(ii)	Admin generates a report on monthly or yearly student subscriptions data.	Report generated as PDF and could be downloaded or printed.	Pass
	(iii)	Admin generates a report on drivers' data.	Report of active and resigned drivers generated as PDF and could be downloaded or printed.	Pass

### 5.2.2 User Acceptance Testing

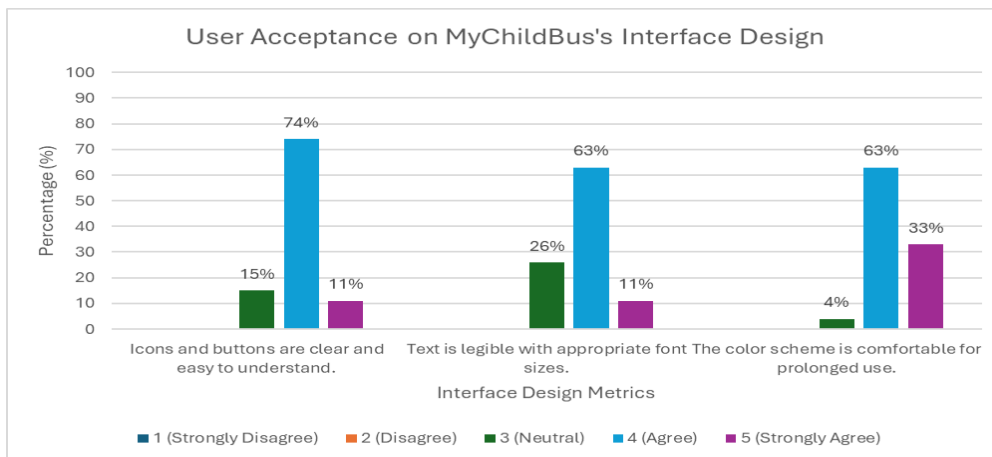
User Acceptance Testing on application's usability and interface design has been conducted by collecting feedback from respondents consists of 23 individuals among parents, 3 school drivers, and 1 administrator through Microsoft form for parents, as attached in Appendix H and printed forms for admin and drivers, as attached in Appendix I. Each section, usability and interface design in the form contains 5 metrics that respondent should rate from the scale 1 (strongly disagree) to 5 (strongly agree). From the responses received, three metrics that gain the highest satisfaction level from users are extracted into graph. Figure 13 and Figure 14 illustrate the summarization of user acceptance results on application's usability and interface design.

The result of user acceptance on MyChildBus's usability as in Figure 13 shows that majority of respondents are satisfied with the application's usability from the aspect of navigation, response time, and ability to complete tasks. 74% from the total 27 respondents agree that the application is easy to navigate, 84% of them agree that the app respond quickly to their actions, and 92% of them agree that the app helps them to complete tasks efficiently.



**Fig. 13** User acceptance results on application's usability.

The result of user acceptance on MyChildBus's interface design as in Figure 14 shows good implementation of icons, buttons, text, and color scheme in creating user-friendly interface as 74% of respondents agree that the icons and buttons are clear and easy to understand, 63% of them agree that the text is legible with appropriate font sizes, and 63% of them agree that the color scheme is comfortable for prolonged use.



**Fig. 14** User acceptance results on application's interface design.

Hence, through the results gained, it shows that the app not only proven through the functional testing test plan results to fulfil the functional requirements, but also fulfilling the non-functional requirements especially from the aspect of performance and operational as proven in the result of user acceptance on the app's usability. Moreover, the good result of user acceptance on the app's interface design shows that the app has achieved its objective to design a user-friendly mobile application interface that enables parents to track their children's school bus location in real-time.

## 6. Conclusion

MyChildBus mobile application successfully achieves its objectives by providing a user-friendly mobile application interface through integration of Google Maps API that enables intuitive interface with clear visual indicators for bus live locations, further justified by result gained from user acceptance testing on application's interface design, developing a centralized database system to store student, driver and route information through the implementation of Firebase for real-time data management, and testing the application's performance and accuracy in location tracking through extensive functionality testing on the bus live location tracking module that confirmed the application's live location tracking feature operates as intended, with no critical errors affecting performance.

Key advantages of the application include real-time bus location tracking functionality through integration of Google Maps API as a cost-effective solution rather than expensive GPS hardware, automated attendance system that allows parents to notify drivers of student absences for improved operational efficiency and dynamic daily route adjustments, built-in messaging channel that facilitates secure and efficient communication between parents, drivers and admin, advanced route planning and route navigation feature for drivers that reduce time, route distance and fuel consumptions, and lastly a comprehensive incident reporting and handling mechanism enables immediate emergencies reported by drivers and appropriate responses coordinated by admin.

Despite these achievements, the application also portrays some limitations, such as dependency on continuous internet connectivity for real-time location tracking, restricted potential for multi-operator deployment as the current application's architecture support is limited for a single bus company at a time, and lack of multi-factor authentication for advanced security and protection. In order to overcome these limitations, suggested future work and development recommendations include implementation of offline functionality through location data caching to maintain basic tracking functionalities during internet service interruption, architectural modifications to support multiple bus operators, and integration of advanced security features through multi-factor authentications such as SMS or email verification codes and biometric authentication.

In conclusion, MyChildBus successfully addresses critical concerns in school bus transportation, offering a cost-effective and efficient solution for parents, drivers, and school bus operators.

## Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

## Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

## Author Contribution

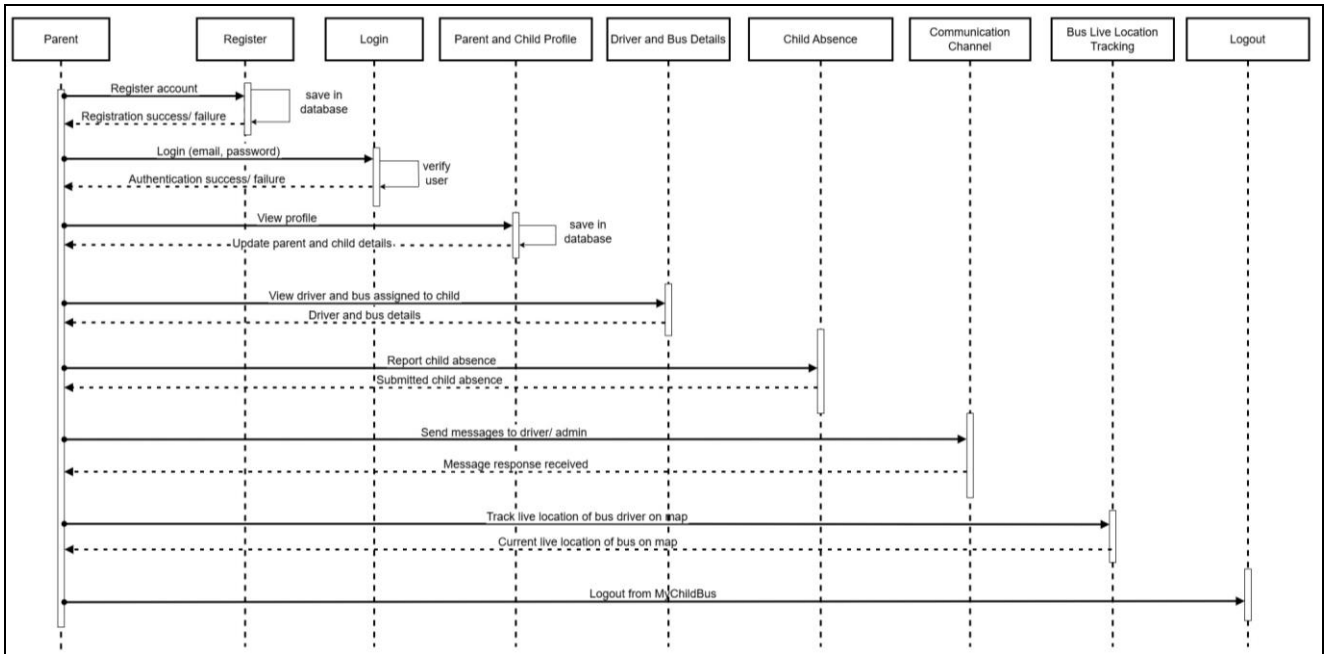
The authors confirm contribution to the paper as follows: **study conception and design:** Fatin Nabilah Zulkaria, Rozanawati Darman; **data collection:** Fatin Nabilah Zulkaria, Rozanawati Darman; **analysis and interpretation of results:** Fatin Nabilah Zulkaria, Rozanawati Darman; **draft manuscript preparation:** Fatin Nabilah Zulkaria, Rozanawati Darman; All authors reviewed the results and approved the final version of the manuscript.

## References

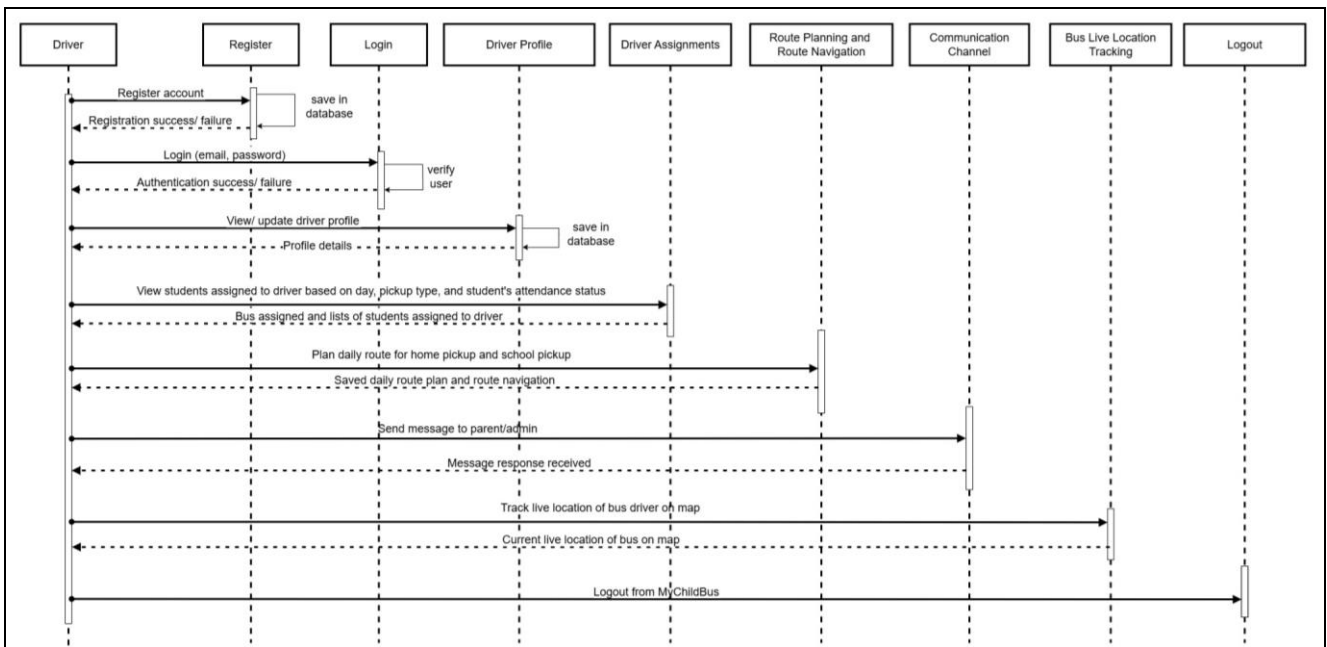
- [1] Gull, H., Aljohar, D., Alutaibi, R., Alqahtani, D., Alarfaj, M., & Alqahtani, R. (2021). Smart School Bus Tracking: Requirements and Design of an IoT based School Bus Tracking System. *Proceedings of the 5th International Conference on Trends in Electronics and Informatics, ICOEI 2021*. <https://doi.org/10.1109/ICOEI51242.2021.9452818>
- [2] Sarosa, M., Ningrum, M. T. P., & Mas'udia, P. E. (2023). Design and implementation of school bus information and tracking system application. *Indonesian Journal of Electrical Engineering and Computer Science*, 30(2). <https://doi.org/10.11591/ijeecs.v30.i2.pp1047-1058>
- [3] Azhar, A., Mustapha, B., & Mansor, Z. (n.d.). *SISTEM PEMANTAUAN PENUMPANG BAS SEKOLAH (BUS-T)*.
- [4] MorningBus - School bus tracking system. (n.d.). <https://morningbus.com/>
- [5] Kalid, K. S., & Rosli, N. (2017). The design of a schoolchildren identification and transportation tracking system. *International Conference on Research and Innovation in Information Systems, ICRIS*. <https://doi.org/10.1109/ICRIIS.2017.8002454>
- [6] *GPS2u - Apps on Google Play*. (n.d.). <https://play.google.com/store/apps/details?id=com.gps2u.my&hl=en-US>
- [7] Sim, Y. H. R., Lua, Z. Z., Kelaver, K. K., Chua, J. Q., Lim, I. Z. J., Cao, Q., Keoh, S. L., & Lim, L. H. I. (2023). Experiences and Lessons Learned from Real-World Projects in Software Engineering Subject. *Software Engineering Education Conference, Proceedings, 2023-August*. <https://doi.org/10.1109/CSEET58097.2023.00036>
- [8] Garg, A., Kumar Kaliyar, R., & Goswami, A. (2022). PDRSD-A systematic review on plan-driven SDLC models for software development. *8th International Conference on Advanced Computing and Communication Systems, ICACCS 2022*. <https://doi.org/10.1109/ICACCS54159.2022.9785261>
- [9] Bennet, L. (2024). Prototyping Model in Software Engineering: Methodology, Process, Approach. *Guru99.com*. <https://www.guru99.com/software-engineering-prototyping-model.html>

- [10] Saroja, S., & Haseena, S. (2023). Functional and Non-Functional Requirements in Agile Software Development. In *Agile Software Development: Trends, Challenges and Applications*.  
<https://doi.org/10.1002/9781119896838.ch5>
- [11] Muhammad, A., Siddique, A., Mubasher, M., Aldweesh, A., & Naveed, Q. N. (2023). Prioritizing Non-Functional Requirements in Agile Process Using Multi Criteria Decision Making Analysis. *IEEE Access*, 11.  
<https://doi.org/10.1109/ACCESS.2023.3253771>

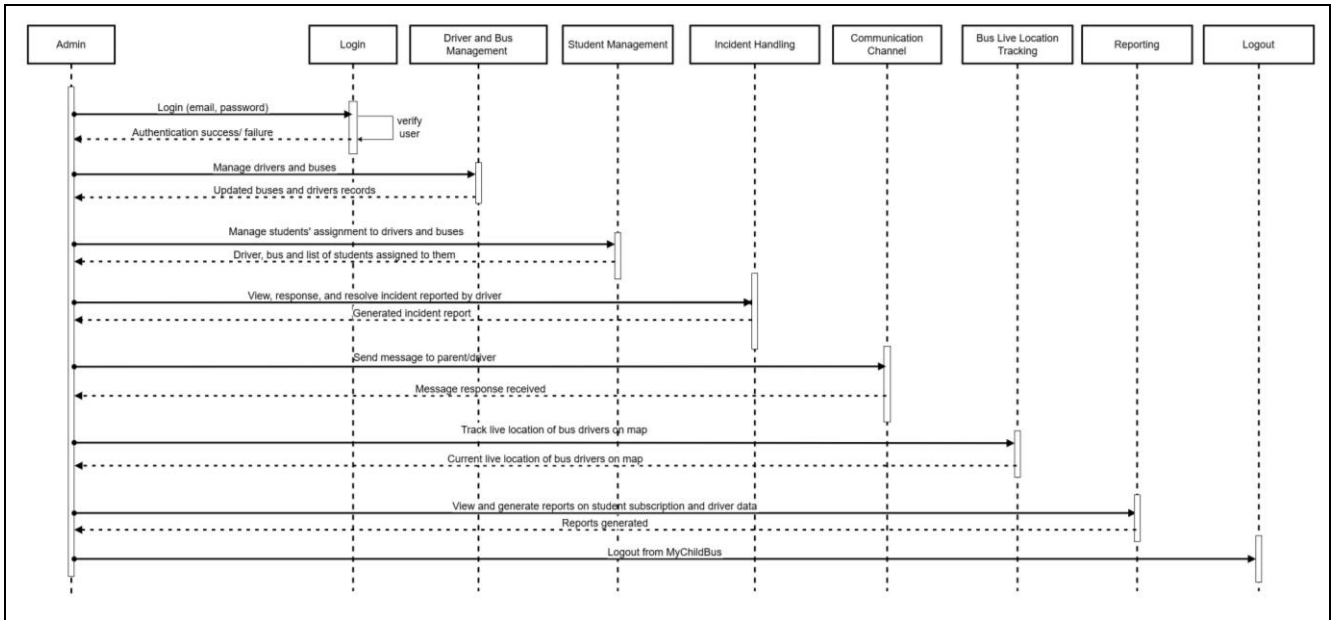
### Appendix A: Sequence Diagram for Parents



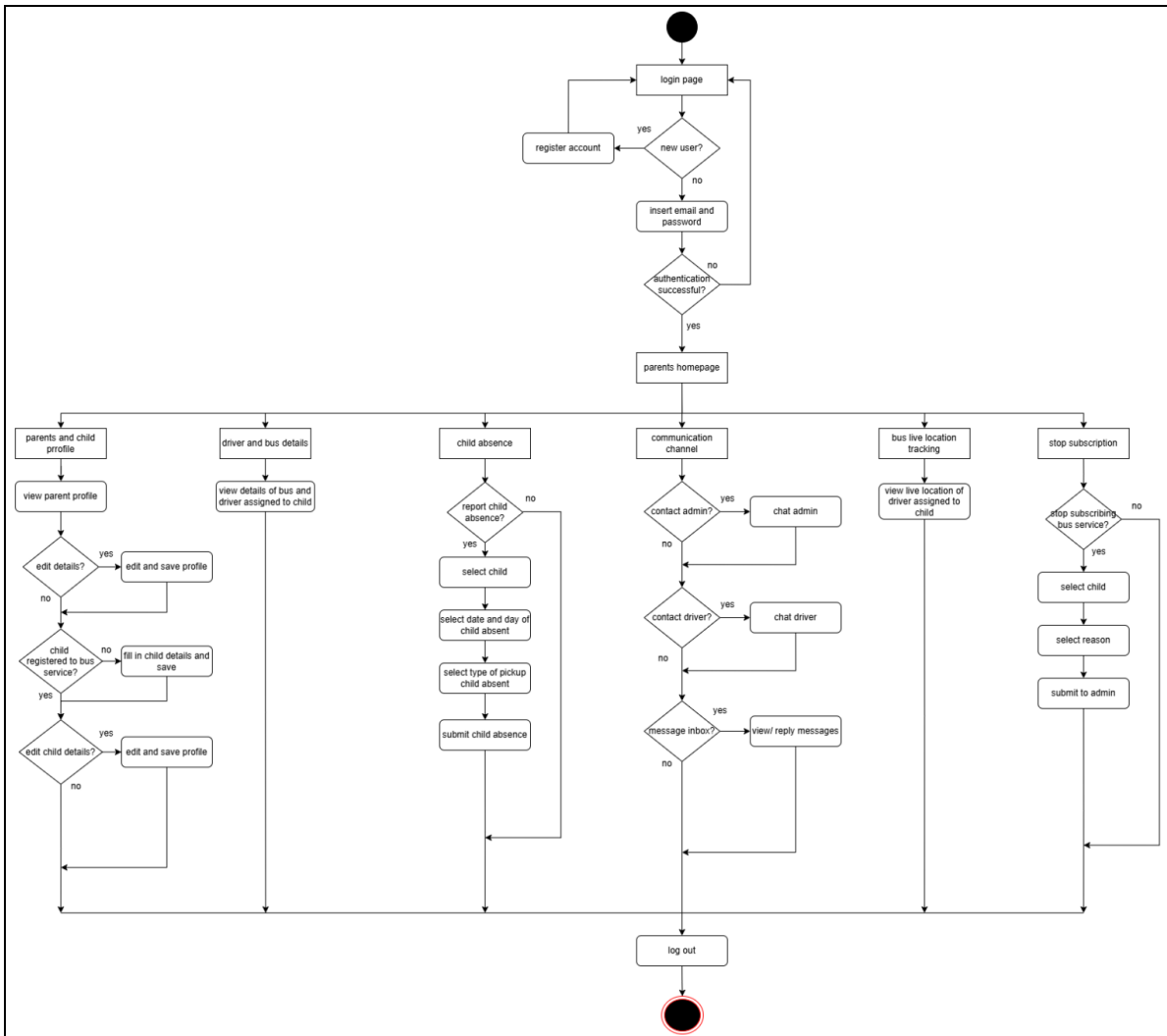
### Appendix B: Sequence Diagram for Drivers



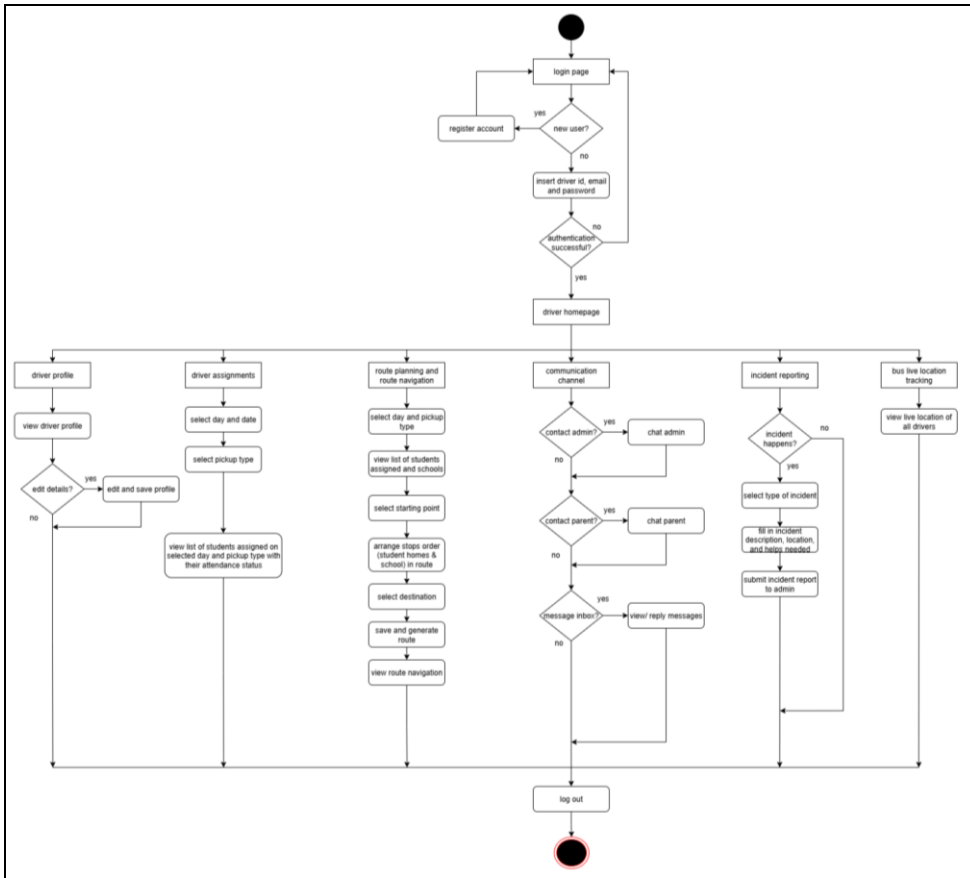
### Appendix C: Sequence Diagram for Admin



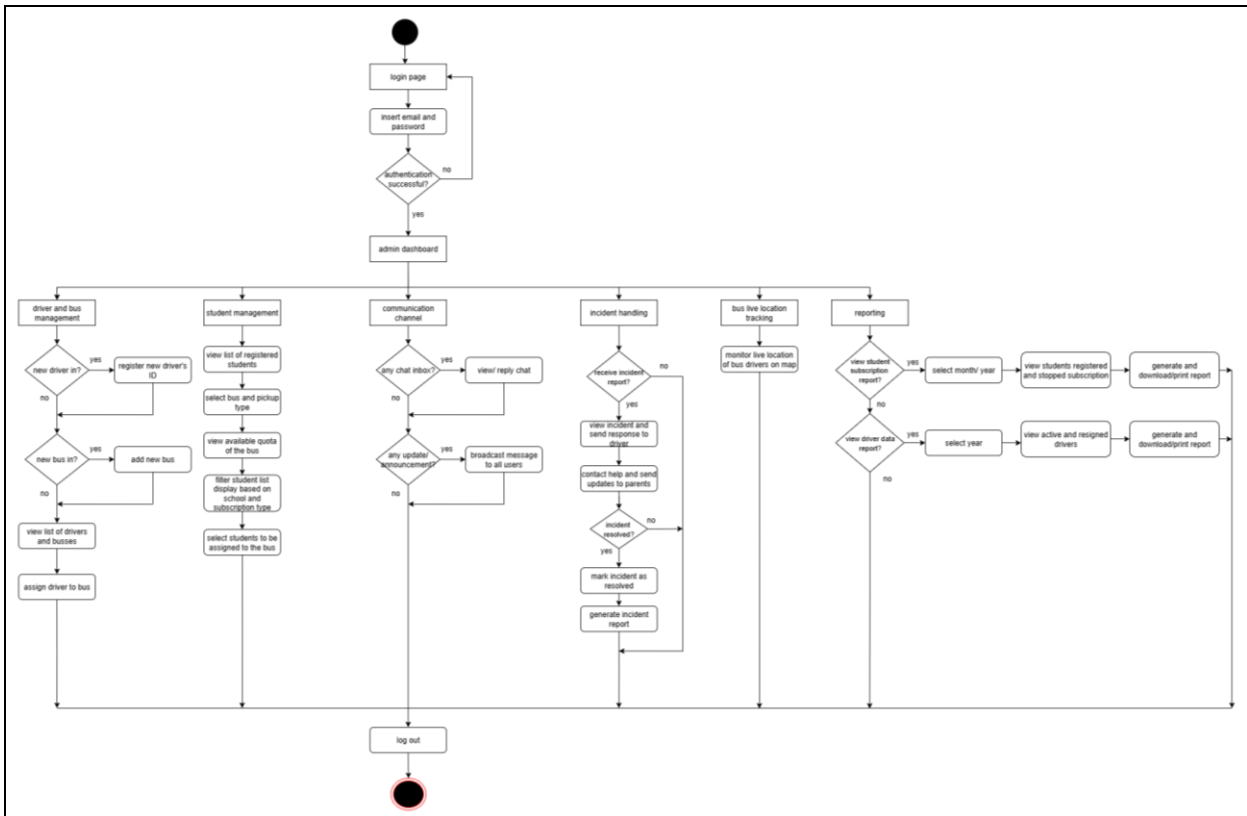
### Appendix D: Activity Diagram for Parents



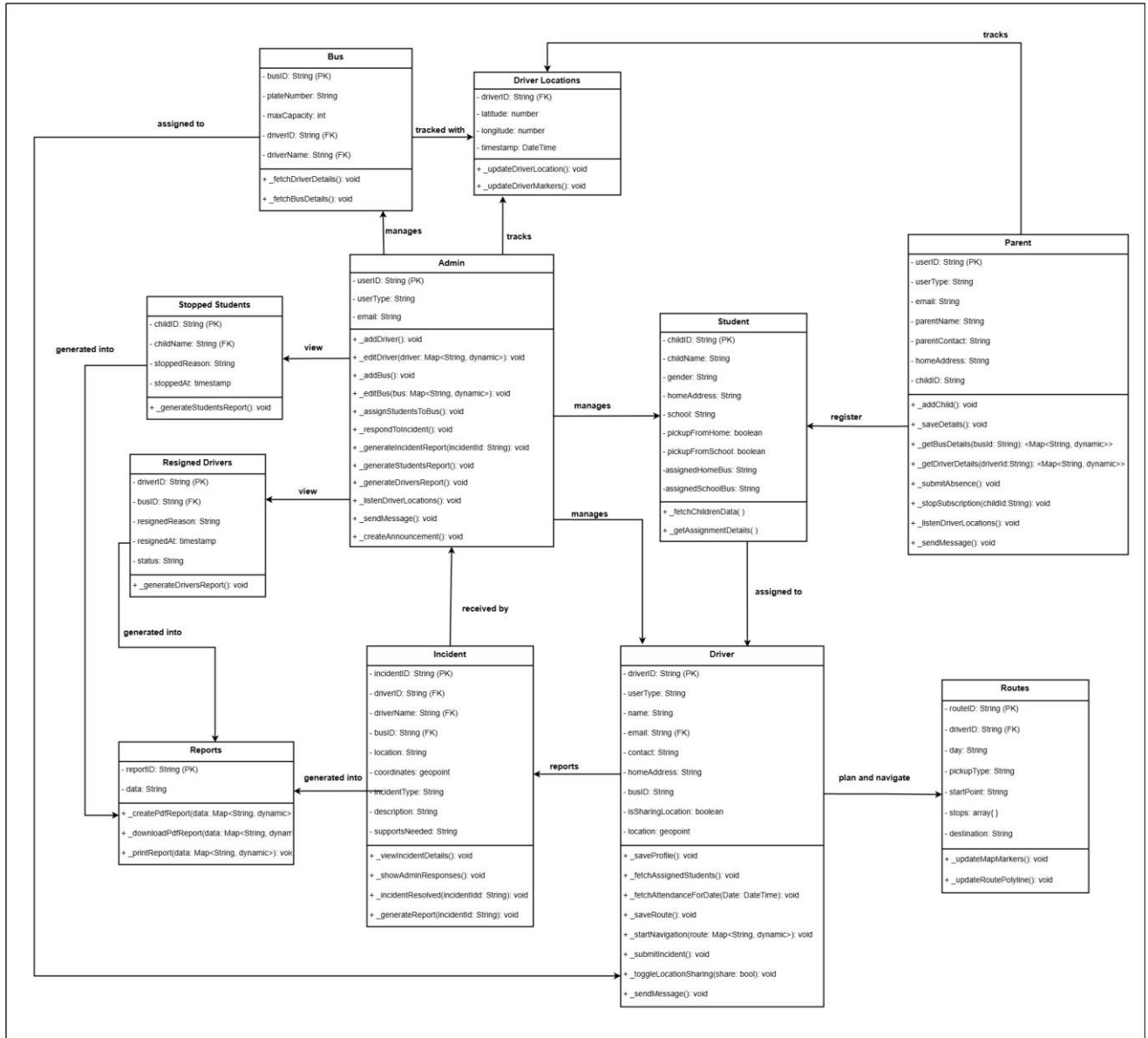
### Appendix E: Activity Diagram for Drivers



### Appendix F: Activity Diagram for Admin



### Appendix G: Class Diagram of MyChildBus Mobile Application



### Appendix H: User Acceptance Testing Form (Microsoft Form) for Parents

### Appendix I: User Acceptance Testing Form (Printed) for Admin and Drivers

USER ACCEPTANCE TESTING (UAT) FOR ADMIN						USER ACCEPTANCE TESTING (UAT) FOR DRIVERS							
MyChildBus: Real-Time School Bus Monitoring and Rides Tracking Mobile Application for Parents						MyChildBus: Real-Time School Bus Monitoring and Rides Tracking Mobile Application for Parents							
Name : <u>OMAR BIN AHMAD</u> Company Name : <u>OMAR BIN AHMAD SDN. BHD.</u> Position : <u>EMPLOYER / DRIVER</u>						Name : <u>FARIDAH BINTI ISNIN</u> Company Name : <u>OMAR BIN AHMAD SDN. BHD.</u> Position : <u>DRIVER</u>							
Acceptance Criteria		Acceptance Rate					Acceptance Criteria		Acceptance Rate				
		1 (Strongly Disagree)	2 (Disagree)	3 (Neutral)	4 (Agree)	5 (Strongly Agree)			1 (Strongly Disagree)	2 (Disagree)	3 (Neutral)	4 (Agree)	5 (Strongly Agree)
<b>Section 1: Application's Functionality</b>						<b>Section 1: Application's Functionality</b>							
1. The Communication Channel Module allows me to communicate effectively with parents and drivers.						✓	1. The Communication Channel Module allows me to communicate effectively with parents and admin.						✓
2. The Bus Live Location Tracking Module provides accurate real-time data for all drivers.					✓		2. The Bus Live Location Tracking Module updates my location accurately in real-time.						✓
3. The Driver and Bus Management Module helps me manage drivers and buses efficiently.						✓	3. The Student Attendance Module helps me track student attendance easily.						✓
4. The Student Management Module allows me to assign students to buses easily.					✓		4. The Route Planning and Route Navigation Module assists me in planning and navigating routes efficiently.						✓
5. The Incident Handling Module enables me to respond to and resolve incidents promptly.					✓		5. Reporting incidents through the Incident Reporting Module is simple and effective.						✓
<b>Section 2: Application's Usability</b>						<b>Section 2: Application's Usability</b>							
1. The app is easy to navigate and user-friendly.						✓	1. The app is easy to navigate and user-friendly.						✓
2. The features are intuitive and require minimal guidance to use.				✓			2. The features are intuitive and require minimal guidance to use.				✓		
3. The app responds quickly to my actions (e.g., loading data, generating reports).					✓		3. The app responds quickly to my actions (e.g., loading maps, sending messages).					✓	
4. The layout is logical and well-organized.					✓		4. The layout is logical and well-organized.					✓	
5. I can complete tasks efficiently.					✓		5. I can complete tasks efficiently.					✓	
<b>Section 3: Application's Interface Design</b>						<b>Section 3: Application's Interface Design</b>							
1. The layout of the app is visually appealing.						✓	1. The layout of the app is visually appealing.						✓
2. The icons and buttons are clear and easy to understand.					✓		2. The icons and buttons are clear and easy to understand.				✓		
3. Text is legible with appropriate font sizes.				✓			3. Text is legible with appropriate font sizes.					✓	
4. The color scheme is comfortable for prolonged use.					✓		4. The color scheme is comfortable for prolonged use.					✓	
5. The app's design is consistent across all modules.					✓		5. The app's design is consistent across all modules.					✓	
Signature: Name: <u>OMAR BIN AHMAD</u> Date: <u>1/6/25</u>						Signature: Name: <u>FARIDAH BINTI ISNIN</u> Date: <u>1/6/2025</u>							