

# Final Report - Public Bus Tracking System App



**University:** The Open University of Sri Lanka

**Faculty & Department:** Faculty of Engineering Technology, Department of Electrical and Computer Engineering, Academic Year 2022/2023

**Course Name & Code:** EEI4369 - Mobile Application Development for Android

**Name:** A.L. SHARAF AHMED

**Registration:** 621436639

**supervisor's name:** Mr. Kasun Aratthanage.

**Date:**2023.06.05

## Contents

1. Project Overview:.....	3
2. Objectives:.....	4
3. Methodology: .....	5
4. System Design: .....	5
5. Features and Functionality: .....	6
6. Implementation: .....	7
7. Testing and Evaluation: .....	8
8. User Interface and User Experience: .....	11
9. Conclusion: .....	12
10. References: .....	15
11. Appendix: .....	16

## **01.Project Overview**

The Public Bus Tracking System app is an Android application developed to address the challenges faced by the public in tracking bus locations and accessing timely information. The app aims to provide a convenient and efficient solution for passengers to track the current location of buses and obtain relevant service information.

### **Purpose:**

The purpose of the Public Bus Tracking System app is to improve the overall bus commuting experience by enabling passengers to access real-time bus location information. By providing accurate and up-to-date data, the app helps users plan their journeys effectively, reducing waiting times and minimizing the risk of missing buses. Additionally, the app aims to enhance communication between bus operators, drivers, and passengers, facilitating better coordination and service delivery.

### **Key Features:**

1. **Bus Location Tracking:** The app allows users to track the real-time location of buses on their preferred routes. This feature enables passengers to have a clear understanding of the bus's current position.
2. **Service Information:** The app provides users with comprehensive service information, including the number of available buses, their schedules, and routes. This feature helps passengers plan their journeys by selecting the most convenient and suitable bus options.
3. **Alerts & Messages:** The app allows administrators to send Messages to passengers and bus drivers. Passengers receive important updates regarding route changes, delays, or other relevant information, ensuring a seamless travel experience. Bus drivers receive instructions or updates from the administration to facilitate efficient service operations.
4. **User Registration and Authentication:** The app offers user registration functionality, allowing passengers, bus drivers, and administrators to create their accounts. Registered users can log in securely, accessing personalized features and information based on their roles.
5. **Administrative Dashboard:** The app includes an administrative dashboard that enables authorized personnel to manage bus routes, update service information, and send alerts to passengers and drivers. This dashboard provides administrators with a centralized platform to monitor and control the system effectively.

### **Problem Statement:**

The current lack of a reliable bus tracking system creates various issues for passengers. Long waiting times, uncertainty about bus availability, and the risk of missing buses are common problems faced by commuters. The Public Bus Tracking System app aims to alleviate these

challenges by providing accurate bus location tracking, timely service information, and effective communication channels between passengers, bus drivers, and administrators.

By addressing these issues, the app seeks to enhance the overall efficiency and convenience of public bus transportation, improving the commuting experience for users and contributing to a more sustainable and organized public transportation system.

## **02.Introduction**

The Public Bus Tracking System app is a solution-driven project aimed at addressing the challenges faced by commuters in accessing reliable bus transportation information. The motivation behind choosing this specific application idea stems from the personal experiences of the project developer, as well as the need for an innovative and efficient system in the transportation industry.

### **Background:**

In many countries, including our own, public transportation plays a crucial role in the daily lives of millions of people. However, the lack of a robust and user-friendly bus tracking system often leads to inconvenience and frustration for commuters. Long waiting times, uncertainty about bus availability, and the risk of missing buses are persistent issues that negatively impact the overall commuting experience.

### **Context:**

Recognizing the need for a comprehensive and accessible solution, the Public Bus Tracking System app aims to provide an efficient and user-friendly platform for tracking bus locations and accessing real-time service information. By leveraging mobile technology and the power of data integration, the app seeks to revolutionize the way passengers interact with public bus transportation.

### **Relevance in the Industry:**

The relevance of this project in the transportation industry is undeniable. With the rapid advancement of mobile technology and the increasing dependence on smartphones, mobile applications have become integral tools for enhancing various aspects of our daily lives. However, the bus tracking system in our country is still in its infancy, lacking a user-centric approach and failing to meet the evolving needs of commuters.

By introducing the Public Bus Tracking System app, we aim to bridge this gap and provide a comprehensive solution that significantly improves the commuting experience. This app aligns with the growing trend of digitizing transportation services and empowers passengers with accurate bus location data, timely service information, and effective communication channels.

Furthermore, the project's relevance extends beyond the individual commuter, as it has the potential to contribute to a more organized and efficient public transportation system. By

reducing waiting times, minimizing the risk of missed buses, and enabling better coordination between passengers and bus operators, the app can enhance the overall efficiency and sustainability of public transportation.

In conclusion, the Public Bus Tracking System app addresses a pressing problem in the transportation industry by providing a solution that leverages mobile technology and data integration. With its potential to improve the commuting experience and contribute to a more organized public transportation system, the app holds significant relevance in our country's evolving transportation landscape.

### **03.Objectives**

The objectives of the Public Bus Tracking System Android application are as follows:

1. **Real-Time Bus Tracking:** The app aims to provide users with real-time information about the location of buses on their preferred routes. By utilizing GPS technology the app will display the approximate location of buses on a map, allowing users to track their expected arrival times approximately.
2. **Efficient Journey Planning:** The app will enable users to plan their bus journeys more efficiently by getting bus location and bus schedules that allowing them to optimize their travel plans and reduce waiting times.
3. **Reliable and Timely Information:** The app will strive to provide reliable and up-to-date information about bus locations, routes, and schedules.
4. **User-Friendly Interface:** The app will feature a user-friendly interface that is easy to navigate and understand. It will prioritize simplicity and intuitive design to ensure that users can access the desired information quickly and easily.

### **04.Methodology**

The development of the Public Bus Tracking System Android application followed a structured approach using various development tools, programming languages, frameworks, and libraries. The chosen methodology ensured efficient development, code maintainability, and adherence to industry best practices. The key components of the methodology are outlined below:

1. **Development Tools:** Android Studio, the official integrated development environment (IDE) for Android app development, was used as the primary development tool. Android Studio provides a comprehensive set of features and tools for designing, coding, debugging, and testing Android applications.
2. **Programming Language:** Java was used for this app. The Android application was developed using Java as the primary programming language. Java is a widely adopted and well-established language in the Android development community, offering a robust set of libraries and frameworks for building scalable and performant applications.

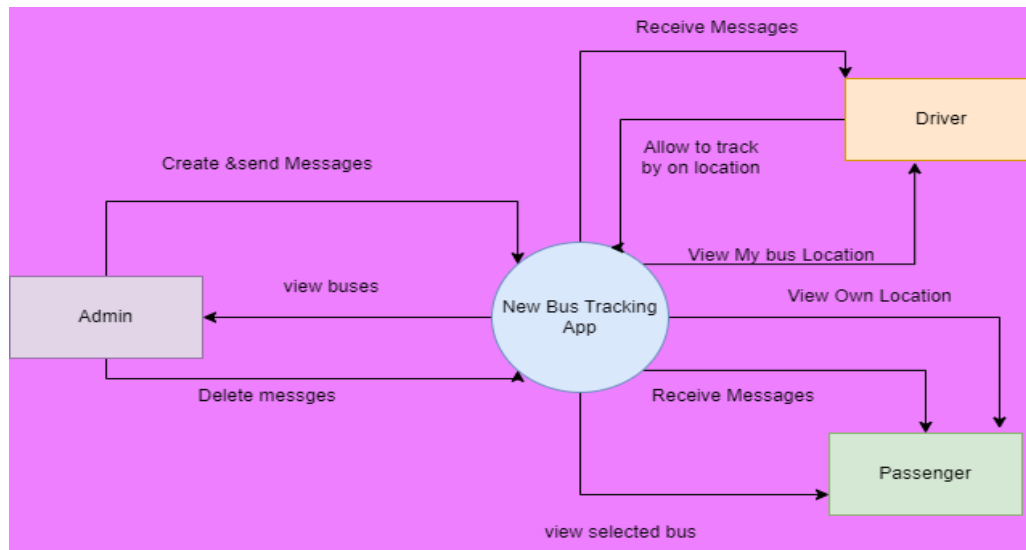
### 3. Frameworks and Libraries:

- **Firestore:** Firestore, a comprehensive mobile development platform, was utilized for various backend functionalities. Firestore Authentication was used for user authentication and registration processes. Firestore Realtime Database was employed for real-time data synchronization, allowing seamless updates and retrieval of bus location and user-related information.
- **Google Maps API:** The Google Maps API was integrated into the application to provide accurate map-based functionalities, such as current location and real-time bus tracking on the map.

By employing this methodology and leveraging the appropriate tools, languages, frameworks, and libraries, the Android application was developed efficiently, meeting the project requirements and industry standards.

## 05. System Design

The Public Bus Tracking System Android application was designed with a scalable and modular architecture to ensure flexibility, maintainability, and extensibility. The system architecture consists of multiple components that interact with each other to provide a seamless user experience.



## **06.Features and Functionality**

The Public Bus Tracking System Android application offers several key features and functionalities to enhance the experience of users, including passengers, drivers, and administrators. Each feature contributes to achieving the project objectives of providing real-time bus tracking, bus information, and improving the efficiency of public bus transportation. Here are the key features and their functionalities:

### **1. Real-Time Bus Tracking:**

- Passengers can track the real-time location of buses on a map.
- Drivers' locations are continuously updated and displayed on the map.
- This feature allows passengers to plan their trips and reduce waiting time.

### **2. Bus Information:**

- Passengers can access information about available buses, their routes, and schedules.
- Passengers can choose the most convenient bus based on their destination and schedule.

### **3. Alerts and Notifications (Messages):**

- Administrators can send alerts and notifications to passengers and drivers.
- Alerts can include important information such as delays, route changes, or service disruptions.
- Passengers and drivers receive notifications in real-time, ensuring they are well informed.

### **4. User Registration and Authentication:**

- Users, including passengers, drivers, and administrators, can register and create their accounts.
- Authentication mechanisms ensure secure access to the application and its features.

### **5. Driver Dashboard:**

- Drivers have their dedicated dashboard to view alerts and notifications sent by administrators.
- They can toggle their location status to indicate their availability and start tracking their bus's location.

### **6. Passenger Dashboard:**

- Passengers have their dashboard to view alerts, notifications, and current bus locations.
- They can select their preferred bus route and view the real-time location of the bus they intend to board.
- The dashboard offers a user-friendly interface for passengers to access essential information.

### **7. Administrator Dashboard:**

- Administrators have a comprehensive dashboard to manage the system.
- They can send alerts and notifications to passengers and drivers.
- The dashboard allows administrators to view all buses' current moving locations and manage alerts efficiently.

These features collectively aim to improve the overall experience of public bus transportation by providing real-time information, reducing waiting time, and enhancing communication between passengers, drivers, and administrators.

## 07.Implementation

The implementation of the Android application involved various components and functionalities to achieve the desired objectives. Here, I'll provide an overview of the implementation process and highlight some code snippets to illustrate important aspects.

### 1. Real-Time Bus Tracking:

- The application utilizes Firebase Realtime Database to store and retrieve bus location data.
- The code snippet shows how the current user's location is retrieved and updated in the Firebase database:

```
FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Current
Location1").child(user.getEmail().replace('.', ','));

// Retrieve user's location and update in the database
ref.addListenerForSingleValueEvent(new ValueEventListener() {

    @Override

    public void onDataChange(@NonNull DataSnapshot snapshot) {

        // Retrieve and process the location data

        if (snapshot.exists()) {

            MyLocation location = snapshot.getValue(MyLocation.class);

            // Update the marker on the map with the retrieved location

            // ...

        }

    }

})
```



```

@Override

public void onCancelled(@NonNull DatabaseError error) {

    // Handle database error if needed

}

});

```

## 2. User Messages:

- The application allows users to send and receive messages using Firebase Realtime Database.
- The code snippet demonstrates how messages are sent and retrieved from the database:

```

// Send message
private void sendMessage(String message) {

    // Get current user's email
    String userEmail = auth.getCurrentUser().getEmail();

    // Generate a unique message ID
    String messageId = messagesRef.push().getKey();

    // Create a new UserMessage object
    UserMessage newMessage = new UserMessage("Admin", message);

    // Save the new message in the database
    messagesRef.child(messageId).setValue(newMessage);

    // Show a toast message indicating successful message sent
    Toast.makeText(AdminDashboard.this, "Message sent", Toast.LENGTH_SHORT).show();

    // Clear the message input field
    etMessage.setText("");

}

```

```
// Retrieve messages
messagesRef.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot snapshot, String previousChildName) {
        // Retrieve and process the new message
        UserMessage message = snapshot.getValue(UserMessage.class);
        if (message != null) {
            // Add the message to the message list and update the UI
            // ...
        }
    }
});

// Other event listeners for message updates, removal, etc.
// ...
});
```

### 3. Driver Location Updates:

- The application continuously updates the driver's location using the device's GPS and stores it in the Firebase database.
- The code snippet showcases how the driver's location is retrieved and updated in the database:

```
// Request location updates
LocationServices.getFusedLocationProviderClient(DriverDashboard.this)
    .requestLocationUpdates(locationRequest, new LocationCallback() {
        @Override
        public void onLocationResult(@NonNull LocationResult locationResult) {
```

```

// Process the updated location result
if (locationResult != null && locationResult.getLocations().size() > 0) {
    // Retrieve the latest location and update in the database
    double latitude = locationResult.getLocations().get(index).getLatitude();
    double longitude = locationResult.getLocations().get(index).getLongitude();

    // Update the driver's location in the database
    // ...
}
}

// Other methods and error handling
// ...

}, Looper.getMainLooper());

```

These codes are conclusion codes of this app. These code snippets illustrate the implementation of specific functionalities within the Android application. The implementation process involved integrating Firebase Realtime Database, handling location updates, and managing message sending and retrieval.

Challenges faced during the implementation of handling database errors, ensuring proper permission management for location access, and maintaining smooth real-time updates. I got Solutions for connecting firebase from documentation.

## 08. Testing and Evaluation

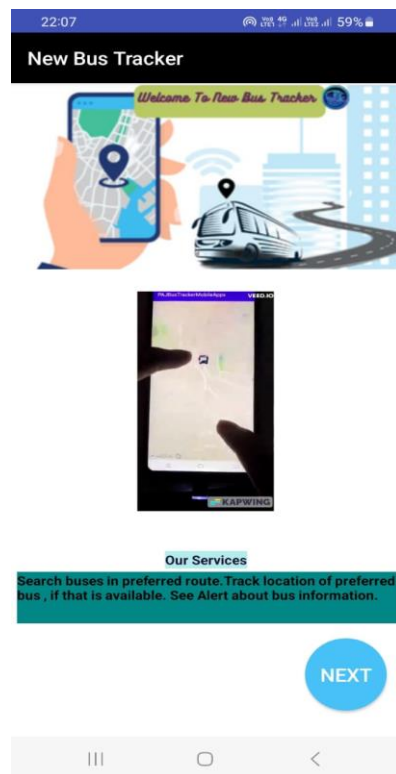
During the testing phase, I employed various methods to ensure the functionality and reliability of my bus tracking app. I conducted unit testing to check individual components, integration testing to verify seamless interaction between different modules, and functional testing to validate the app's features. Additionally, I performed performance testing to assess its performance under different conditions. To gather user feedback, I organized a user acceptance testing phase with beta testers. Based on the feedback received, I prioritized and addressed identified issues, including enabling multiple drivers to share their locations. Continuous testing and feedback loops were crucial in improving the app's reliability and user experience.

## 09. User Interface and User Experience

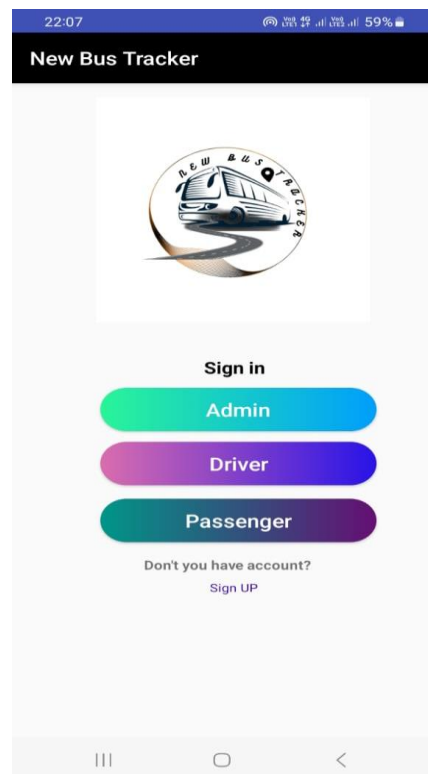
When evaluating the user interface (UI) design and user experience (UX) of my bus tracking application, I focused on aspects such as usability, visual appeal, and intuitive navigation to create an engaging and user-friendly experience.

- **Usability:** I aimed to make the app's functionality easily accessible to users. I designed a simple and intuitive layout with prominently placed buttons that allow users to navigate to different sections of the app. By keeping the interface clean and clutter-free, users can quickly understand how to interact with the app and access its features.
- **Visual Appeal:** I used a cohesive color scheme, visually pleasing graphics, and appropriate iconography to enhance the overall look and feel of the app. The use of visually appealing elements helps to attract and retain users' attention.
- **Intuitive Navigation:** Although I did not include a navigation bar, I ensured that the buttons within the app were self-explanatory and provided clear indications of their purpose. Each button was labeled appropriately, and their placement followed a logical flow, allowing users to navigate seamlessly between different screens and functionalities.

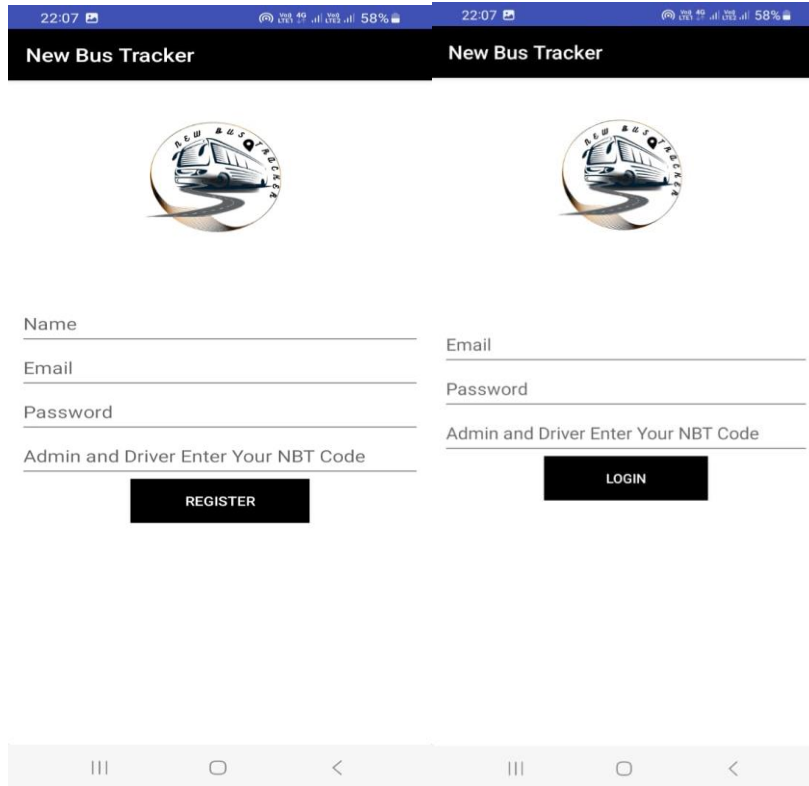
### Screenshot of User Interface



S-01

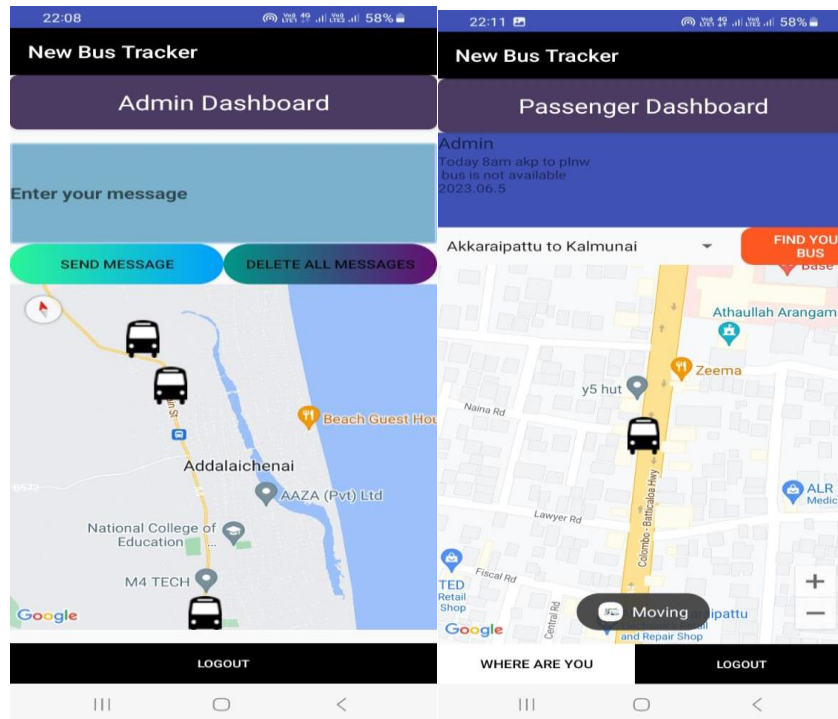


S-02



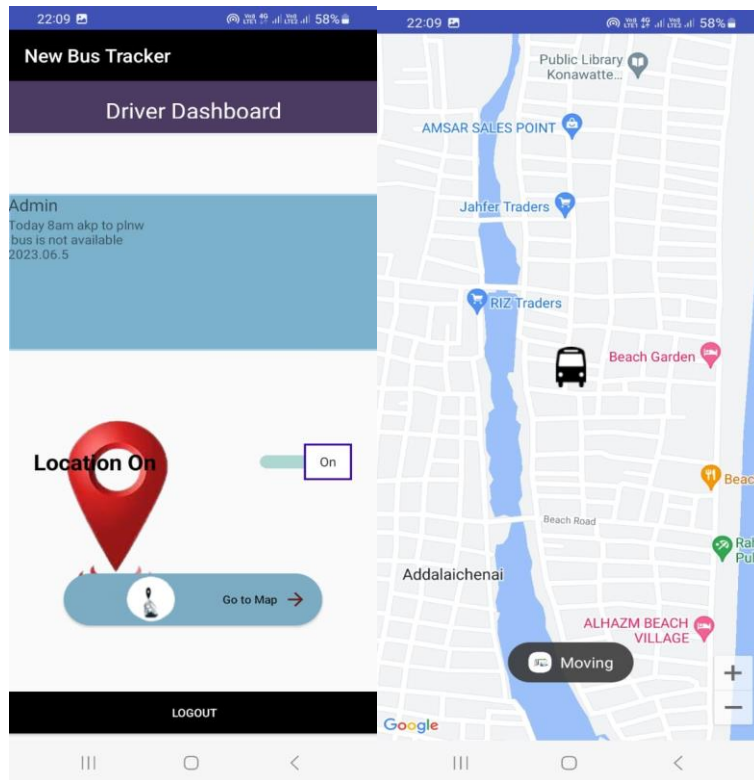
S-03

S-04



S-04

S-06



S-07

S-07.1

#### Description of UI:

- S-01 is Home Page: The home page serves as the entry point for users and provides visual instructions on how to navigate the app. It includes a video help option on accessing the app's features.
- S-02 is Main Dashboard: The main dashboard acts as a hub, allowing users to select their role (admin, driver, or passenger) and navigate to their respective dashboards after login. It also provides a registration link for new users to sign up.
- S-03 is Registration Page: New users can register at this page by providing name email, password and driver and admin must enter their Specific NBT code when they are registering but passenger no need but fill the registering process based on instruction.
- S-04 is Login Page: The login page enables authorized users (passenger, admin and drivers) to log in using their email and password(driver and admin must include their NBT code that separated for this system user). Authentication is handled through Firebase Authentication.
- S-05 is Admin Dashboard: The admin dashboard provides features for sending alerts or information to passengers and drivers. It also displays the current locations of all buses and allows the admin to delete alerts using Firebase Realtime Database and logout button to leave this page.

- S-06 is Passenger Dashboard: The passenger dashboard includes a display field for alerts and an option to select a preferred bus based on the route. It provides a button to view the passenger's current location. Logout button to leave this page.
- S-07 is Driver Dashboard: The driver dashboard displays received alerts or information from the admin. It includes a toggle button to enable location tracking when the driver starts the bus. It also provides a button to navigate to the driver's current location Logout button to leave this page. S-07.1 that page is bus current location app shows that.

Overall, the UI design and UX of my bus tracking application prioritize usability, visual appeal, and intuitive navigation. By providing a clean and intuitive interface with clearly labeled buttons, users can easily navigate through the app and access its functionalities. The visually appealing design elements enhance the overall user experience and contribute to a positive impression of the app.

## **10. Conclusion**

In conclusion, the Public Bus Tracking System Android application aims to provide a comprehensive solution to address the challenges faced by commuters in accessing reliable bus transportation information. By leveraging mobile technology and data integration, the app offers real-time bus tracking, bus information, and effective communication channels between passengers, drivers, and administrators.

Throughout the development process, a structured methodology was followed, utilizing tools such as Android Studio, Java programming language, Firebase, and Google Maps API. The app was designed with a scalable and modular architecture to ensure flexibility, maintainability, and extensibility.

Key features and functionalities were implemented, including real-time bus tracking, bus information, alerts and notifications, user registration and authentication, driver dashboard, passenger dashboard, and administrator dashboard. These features collectively aimed to enhance the overall experience of public bus transportation, providing users with accurate information, reducing waiting times, and facilitating better communication.

Testing and evaluation were conducted to ensure the app's functionality, reliability, and performance.

The user interface (UI) design and user experience (UX) were prioritized, focusing on usability, visual appeal, and intuitive navigation. The app was designed to be user-friendly, allowing users to easily navigate through different sections and access the desired information.

Overall, the Public Bus Tracking System Android application has the potential to significantly improve the commuting experience for passengers, drivers, and administrators. By providing real-time information, reducing waiting times, and enhancing communication, the app contributes to a more efficient and organized public transportation system.

## 11. References

During the development of the Public Bus Tracking System Android application, various resources and references were consulted to ensure the adherence to industry best practices and the utilization of appropriate tools and technologies. Some of the key references used include:

1. Android Developer Documentation: <https://developer.android.com/docs>
2. Firebase Documentation: <https://firebase.google.com/docs>
3. Google Maps Platform Documentation: <https://developers.google.com/maps/documentation>
4. Java Documentation: <https://docs.oracle.com/en/java/>
5. Stack Overflow: <https://stackoverflow.com/>
6. Android Studio Documentation: <https://developer.android.com/studio/intro>
7. OUSL Guidebook
8. ChatGPT

These references provided valuable insights and guidance throughout the development process, helping to overcome challenges and ensure the implementation of a robust and user-friendly Android application.