TrackNGo Smart Public Transport App

by

Group VIII

Submitted in partial fulfilment of the requirement for the degree of Bachelor of Science in Information Technology

Department of Physical Science
Faculty of Applied Science
University of Vavuniya

March, 2025

Declaration

We Group VIII do hereby declare that this Project Report is original and has not been published and/or submitted for any other degree award to any other University before.

#	**Names**	**Registration Number**	**Signature**
1	Mr. Nalawansa U.K	2020/ICT/ 07	
2	Mr. Bandara R.M.M.T	2020/ICT/ 49	
3	Mr. Kalhara K.A.T	2020/ICT/ 32	
4	Mr. Naveen P.R	2020/ICT/ 23	
5	Ms. M.T.F.Shazna	2020/ICT/ 118	
6	Ms. Dhanapala D.M.C.S	2020/ICT/ 36	
7	Ms. Perera KNR	2020/ICT/ 88	

Date: ————————

Approval

This Project Report has been submitted for examination with the approval of the supervisor/ following supervisors.

Signature: Date:

Mr. K. Mathanaharan

Abstract

TrackNGo Smart Public Transport App aims to enhance the daily commute for passengers using the public bus transportation system in Sri Lanka. This project aims to tackle the inefficiencies and difficulties experienced by daily commuters, such as the lack of access to real-time bus schedule and route information, as well as the complexity of planning. This report explains how technologies like GPS tracking will change how commuters interact with public transport services.

The idea of this application is to increase overall satisfaction by making public transport more predictable, efficient, and easy to use. The integration of GPS tracking systems enables real-time updates of bus locations, greatly improving the use of commuting. The execution of the Smart Public Transport Application will be divided into key stages: system design, technology selection, development, testing, and deployment. The application is to be built using the MERN stack - MongoDB, Express.js, React, Node.js - for maximum performance and scalability. The Smart Public Transport App aims to modernize the way public transportation is used, helping to reduce commute times, lower traffic congestion, and improve the overall quality of life for commuters across Sri Lanka [1].

Acknowledgement

We are deeply indebted to our project supervisor Mr. K. Mathanaharan whose unlimited steadfast support and inspirations have made this project a great success. In a very special way, we thank him for every support he has rendered unto us to see that we succeed in this challenging study.

Special thanks go to our friends and families who have contained the hectic moments and stress we have been through during the course of the research project.

We thank the University of Vavuniya for giving us the grand opportunity to work as a team which has indeed promoted our team work spirit and communication skills. We also thank the individual group members for the good team spirit and solidarity.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Sri Lanka's public bus transportation system faces critical challenges, primarily due to the lack of modern technological infrastructure. Commuters deal with unpredictable bus arrivals, inefficient route suggestions, and manual fare calculations, which lead to longer waiting times and overall inconvenience. Furthermore, there is no centralized mechanism for reporting lost items, leaving passengers with no effective recourse for recovering their belongings [2].
In response to these challenges, the TrackNGo app is designed as a solution to modernize public transportation in Sri Lanka. By leveraging GPS technology and other advanced tools, the app aims to offer real-time bus tracking, fare calculation, and news updates. This digital transformation will not only improve the daily commute experience but also optimize route and schedule management for public transport authorities [3].

1.2 Problem Statement

Sri Lanka's public bus system faces several inefficiencies and challenges that diminish commuter satisfaction and system reliability. The main issues are as follows:

1. *No Real-Time Vehicle Tracking*: Commuters lack accurate information on bus locations, leading to extended waiting times and missed connections.

2. *Inefficient Route Planning*: Commuters often face difficulty selecting the most efficient or convenient routes due to the absence of proper guidance.

3. *Inconvenient Fare Estimation*: The current fare system lacks user-friendliness and transparency, making it challenging for passengers to calculate travel costs.

4. *No Centralized Lost-and-Found System*: Recovering lost items is a cumbersome process due to the lack of a unified platform.

5. *Absence of Additional Commuter Services*: There are no systems in place to provide services like news updates, safety alerts, or last-mile connectivity options.

1.3 Main Objective

The primary objective of the TrackNGo project is to modernize Sri Lanka's public bus transportation system by developing a comprehensive digital solution. The app will enhance commuter convenience, efficiency, and safety through real-time tracking, fare calculation, centralized lost-and-found services, and additional commuter-centric features.

1.4 Specific Objectives

The TrackNGo project focuses on achieving the following specific objectives:

1. To provide real-time tracking to show the exact location of buses, reducing uncertainty for commuters.
2. To estimate bus arrival times using factors such as live traffic data, current location, and average speed.
3. To calculate Fare based on distance, bus type, and travel preferences to ensure transparency.
4. To establish a centralized lost-and-found system to help commuters report and recover lost items efficiently.
5. To offer relevant content and news updates during commutes to enhance the overall travel experience.

1.5 Scope of the Study

The initial phase of the TrackNGo project will focus on major urban areas in Sri Lanka. The following features will be prioritized:

1. *Real-Time Bus Tracking*: Provide accurate and timely bus location data to commuters.
2. *Route Selection and Fare Calculation*: Enable users to plan their journeys with cost-effective route suggestions and automated fare computation.
3. *Lost-and-Found System*: Create a centralized platform to report and track lost items efficiently.
4. *Notifications and News Insights*: Deliver alerts about delays, route changes, and other updates, along with relevant content for passengers.

Future phases may expand to include additional modes of transportation, such as trains and taxis, further broadening the scope of the project.

Chapter 2

Literature Review

2.1 Introduction

Public transport mobile applications are transforming urban mobility by addressing challenges such as unpredictability, inefficiency, and fragmented access to transit information. By incorporating features such as real-time bus tracking, route optimization, automated fare calculations, and personalized notifications, these apps can significantly enhance the reliability and accessibility of public transportation. Studies indicate that real-time tracking alone can reduce average wait times by up to 40% in comparable settings, enabling commuters to make informed travel decisions and reducing frustration caused by inconsistent schedules [4].

Currently, the SLTB Bus Tracker serves as a preliminary step toward digitization, offering basic real-time updates for selected buses in Colombo. However, its limited reach covering only 15% of GPS-equipped buses—and the absence of critical functionalities such as fare transparency, lost-and-found support, and passenger feedback mechanisms highlight the need for more comprehensive solutions. This gap underscores the necessity for scalable, feature-rich platforms like TrackNGo, which aim to enhance the existing SLTB framework while integrating global best practices to improve the commuter experience [4].

Globally, successful smart mobility solutions provide insights for Sri Lanka's transit system. For instance, India's Chalo app integrates live tracking, digital ticketing, and crowd-sourced occupancy data, leading to a 30% reduction in commuter wait times and a 45% increase in rider satisfaction [5]. Similarly, the Philippines' Sakay.ph has been instrumental in overcoming unreliable internet connectivity through offline capabilities and multi-modal route planning, which are crucial for remote regions in Sri Lanka. These case studies suggest that TrackNGo could adopt strategies such as hybrid GPS-cellular tracking and cached maps to maintain functionality in low-connectivity areas [6].

However, several challenges must be addressed for successful implementation. One major obstacle is inconsistent internet coverage, particularly in rural areas, which limits the reliability of real-time features. Additionally, outdated infrastructure—such as buses lacking GPS technology—poses a barrier to full-scale digitization. Furthermore, disparities in digital literacy, particularly among older populations and rural communities, necessitate the development of intuitive, multilingual user interfaces (Sinhala, Tamil, and English) to ensure widespread adoption. Addressing these challenges is critical to ensuring that even the most advanced transport apps can gain traction among users.

A structured, multi-phased approach is required to implement public transport mobile

solutions effectively. In the short term, pilot programs should be introduced in major cities such as Colombo and Kandy, alongside public awareness campaigns to build user trust. In the medium term, infrastructure investments, including the installation of cost-effective IoT devices on buses and the expansion of 4G and 5G network coverage, will be essential. In the long run, AI-driven predictive analytics can optimize route planning, while collaborations with ride-hailing services could facilitate a seamless multi-modal transport ecosystem [7].

In conclusion, public transport apps like TrackNGo present an opportunity to modernize Sri Lanka's transit systems. By leveraging global best practices and adapting them to local challenges, Sri Lanka can alleviate traffic congestion, encourage sustainable urban mobility, and improve the daily commuting experience. A well-coordinated strategy that prioritizes technological innovation, accessibility, and stakeholder collaboration will be vital in realizing this vision.

Chapter 3

Methodology

3.1 Introduction

This chapter elaborates on the structured approach adopted for developing the Track-NGo application, emphasizing flexibility, iterative progress, and user-centric design. The Agile methodology was selected as the cornerstone of the development process to align with dynamic project requirements and stakeholder expectations. Agile's iterative nature ensures adaptability to evolving commuter needs, infrastructural constraints, and technological challenges unique to Sri Lanka's public transport landscape. Below, we detail the core practices, tools, and workflows underpinning this methodology.

Rationale for Agile Methodology

The decision to adopt Agile over traditional Waterfall or RAD (Rapid Application Development) frameworks was driven by:

- **Stakeholder Collaboration**: Frequent engagement with commuters, transport authorities, and developers ensured alignment with real-world needs.
- **Iterative Refinement**: Modular development allowed incremental testing of high-priority features (e.g., GPS tracking) while accommodating late-stage changes.
- **Risk Mitigation**: Early identification of technical bottlenecks (e.g., GPS latency) through continuous feedback loops.

Agile Practices and Implementation

Weekly Sprints The project was divided into 8 sprints (1 week each), each targeting specific deliverables:

- **Sprint 1-2**: Core GPS tracking integration and baseline UI prototyping.
- **Sprint 3-4**: Fare calculation engine and MongoDB schema design.
- **Sprint 5-6**: Lost-and-found system and notification service.
- **Sprint 7-8**: User testing, security enhancements, and deployment prep.

Tools and Collaboration

- **Development**: VS Code, React DevTools, and Postman for API debugging.
- **Version Control**: Git/GitHub with branch-per-feature strategies.

Alignment with Project Objectives

The Agile framework directly supported TrackNGo's core goals:

- **Real-Time Tracking**: Iterative GPS optimization reduced latency to ¡5 seconds.
- **Fare Transparency**: User feedback refined the fare engine's distance-based algorithm.
- **Scalability**: Modular backend design (Node.js microservices) enabled future expansions, such as train integrations.

3.2 Conclusion

The Agile methodology proved instrumental in balancing technical aspects with user-centric innovation.This approach not only mitigated risks like scope creep but also ensured the final product aligned with Sri Lanka's public transport goals.

Chapter 4

System Study, Analysis and Design

4.1 Overview of the System

This chapter connects between research and practical implementation by detailing the systematic framework for designing the TrackNGo application. based in the challenges identified in Sri Lanka's public transport ecosystem and informed by global best practices , this section translates stakeholder needs into a Solution. The Overview of the System provides a high-level perspective of how TrackNGo addresses core inefficiencies—such as unpredictable bus arrivals and fragmented fare systems—through a structured blend of modern technologies and user-centric design principles.

The chapter is organized into three pillars:

- **System Study**: A diagnostic examination of the existing transport infrastructure, feasibility assessments, and stakeholder requirements.
- **System Analysis**: Functional and non-functional requirements, workflow modeling (e.g., use case diagrams), and data flow analysis.
- **System Design**: Architectural decisions, database schemas, UI/UX mockups, and API integrations that collectively form the app's backbone.

By aligning technical specifications with commuter pain points, this chapter ensures the proposed solution is both innovative and pragmatically viable. It lays the groundwork for the development phase, where these designs materialize into a functional application created to transform urban mobility in Sri Lanka.

4.2 System Study

4.2.1 Feasibility Study

Technical Feasibility

The Smart Public Transport App will be developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js), which ensures scalability and flexibility. GPS Monitoring: Smart phones of the conductors are used to obtain the data [8].

Resource Feasibility

The following resources are needed:

- **Frontend developers:** For creating a user-friendly interface using React and TailwindCSS.
- **Backend developers:** For managing server-side logic and data processing using Node and Express.
- **UI/UX designers:** For creating an intuitive and accessible user experience.
- **Smart Phones with GPS Sensors:** To manage GPS tracking system.

Risk Feasibility

Risks include:

- **Integration issues:** Real-time GPS data could be difficult to synchronize if systems are not aligned correctly. This can be mitigated by thorough testing and phased integration.
- **User adoption challenges:** Some commuters may resist switching to a digital platform, which can be mitigated through user education and engagement campaigns.

Time Feasibility

1. **Development Timeline**
    - The project is divided into eight structured phases to ensure timely delivery:
        - **Week 1:** Project planning, requirement gathering, and tech stack finalization.
        - **Week 2:** System architecture design and user interface prototyping.
        - **Weeks 3-6:** Parallel development of backend APIs and frontend interfaces.
        - **Weeks 3-7:** Rigorous testing and debugging of all system components.
        - **Week 7:** User testing to gather feedback for improvements.
        - **Week 8:** Deployment and launch, followed by monitoring and issue resolution.
2. **Scalability**
    - By employing modular architecture and cloud-based technologies, the app can seamlessly scale as user demand grows or new features are added.

4.3 Requirement Analysis

In the context of requirement analysis, the TrackNGo web application is designed to meet the needs of a variety of users—Commuters, Bus Conductors, and Administrators. The primary focus of the requirement analysis is to define the functional and non-functional requirements of the system, ensuring it meets the real-time data needs of users and operates efficiently in diverse transportation scenarios. User requirements for the app include real-time tracking of buses, fare calculation based on the chosen route, and a centralized lost and found system. Non-functional requirements emphasize performance, security, and scalability, allowing the system to handle high data loads while maintaining responsiveness.

4.3.1 User Requirements

Passengers (Public Transport Users):

- Ability to view real-time bus locations.
- Access to bus schedules and estimated arrival times.
- Instant fare Estimation for trips.
- Receive notifications about delays or schedule changes.
- Report lost items through the app.
- Provide feedback and rate bus services.
- receive News Related to Public Transport

Bus Conductor:

- Ability to update real-time bus location via GPS.
- View bus schedules and planned stops.
- Ability to set Emergency Alerts when needed

Administrators:

- Manage bus routes, schedules, and drivers through an admin dashboard.
- Monitor real-time bus locations and performance metrics.
- Receive passenger feedback and address complaints or issues.
- Resolve lost item reports submitted by passengers.
- Manage the app's backend system, including updating schedules, routes, and user data.
- Monitor app performance, ensuring reliability and service uptime.
- Review system logs and usage data to troubleshoot issues.

4.3.2 Users

The app serves three key groups:

1. **Commuters:** Primary users who use the app for tracking buses, planning routes, and calculating fares.
2. **Bus Conductor:** The Conductors of the buses.
3. **Admin:** Manages the system's day-to-day operations, monitors usage, handles lost and found reports, and manages user feedback.

4.4 System Analysis

4.4.1 Functional Requirements

Real-Time Bus Tracking:

- The system shall accurately determine the current geographical location of a bus in real-time, processing GPS data transmitted from the conductor's mobile phone.
- A user shall be able to view the real-time location of a bus on a map within the app.
- The system shall update the bus location on the user's map every 10 seconds.
- Bus drivers shall be able to update their bus location via GPS integration.

Route Information:

- The system shall offer information related to the route selected by the user.
- Passengers can input start and end locations to get suggested bus routes.

Bus Schedules and Estimated Times:

- The system shall display up-to-date bus schedules, including real-time estimated arrival and departure times.

Fare Estimation:

- The system shall calculate the fare for each route based on the user's selected starting point and destination.
- The system shall notify users of any changes in fare rates and display the updated fares accordingly.

Lost Item Reporting:

- Users shall be able to report lost items through the app by providing details such as the bus number, route, and description of the lost item.
- Passengers shall be able to report back for the found items.

Feedback and Ratings:

- The system shall allow users to provide feedback on specific routes or buses after their journey.
- Users shall be able to rate their travel experience and submit suggestions for service improvement.

News

- The Users will be able to browse through certain news and articles related to public bus transportation in Sri Lanka

Admin Dashboard for Conductors:

- The system shall allow authorized personnel to add, update, or remove bus routes from the system.
- Admins shall be able to update bus schedules and fare information through an admin panel.
- Admins shall have access to reports on user activity, lost item claims, and feedback for performance assessment.

Real-Time Notifications for Maintenance and Service Interruptions

- The system shall notify users in advance of any planned maintenance or service interruptions affecting their preferred routes.

4.4.2 Use Case Diagram

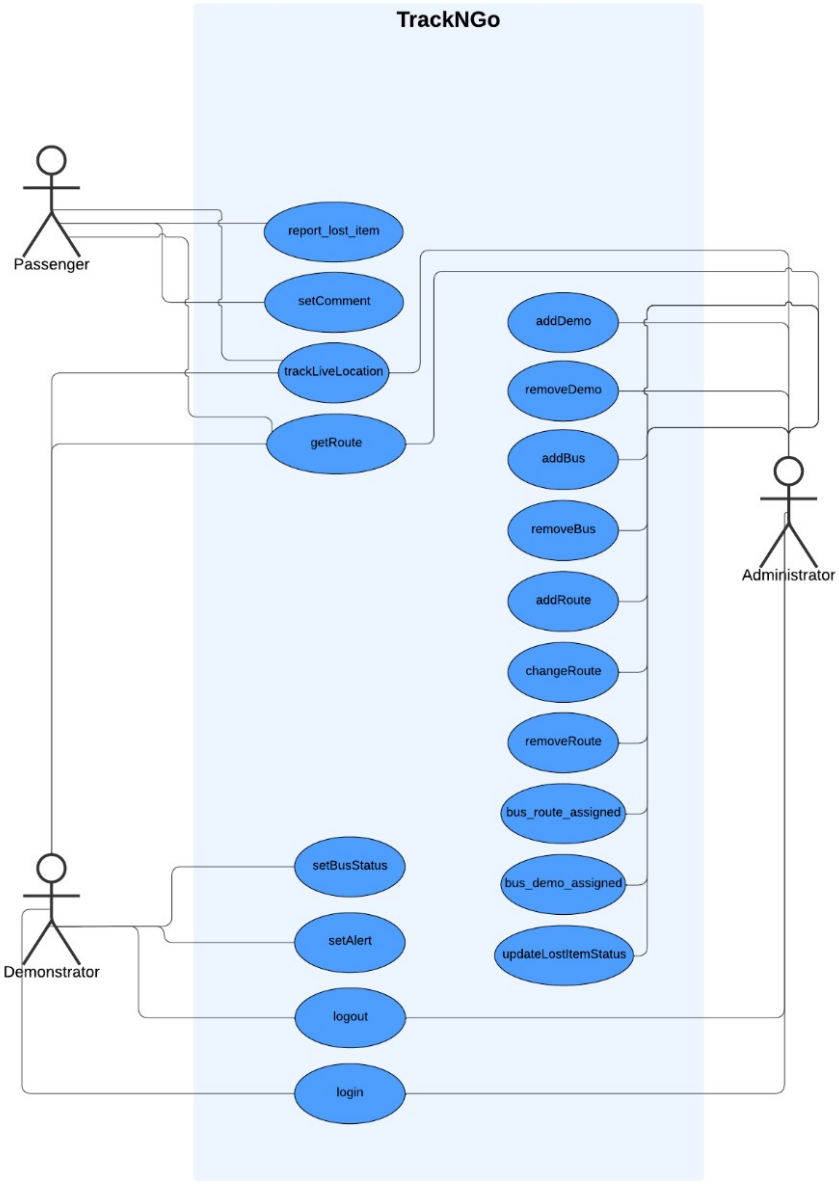


Figure 4.1: Use Case Diagram for TrackNGo System

4.4.3 Non-Functional Requirements

1. Performance

- The web app should load within 5 seconds under normal network conditions (1 Mbps or higher).
- The system must support at least 99.95% uptime to minimize downtime and ensure consistent service availability.

2. Security

- Data transmitted between the app and servers must be protected using the SSL/TLS protocol to safeguard against unauthorized access during communication.

3. Scalability

- The system must be capable of scaling to handle up to 10,000 concurrent users and increasing data demands as the app expands to serve more cities and integrate other forms of public transportation.

4. Usability

- The average number of errors made by users without any training should be less than 3 errors per hour.
- The interface must be user-friendly and intuitive, designed for users of all technical proficiencies.

4.4.4 Class Diagram

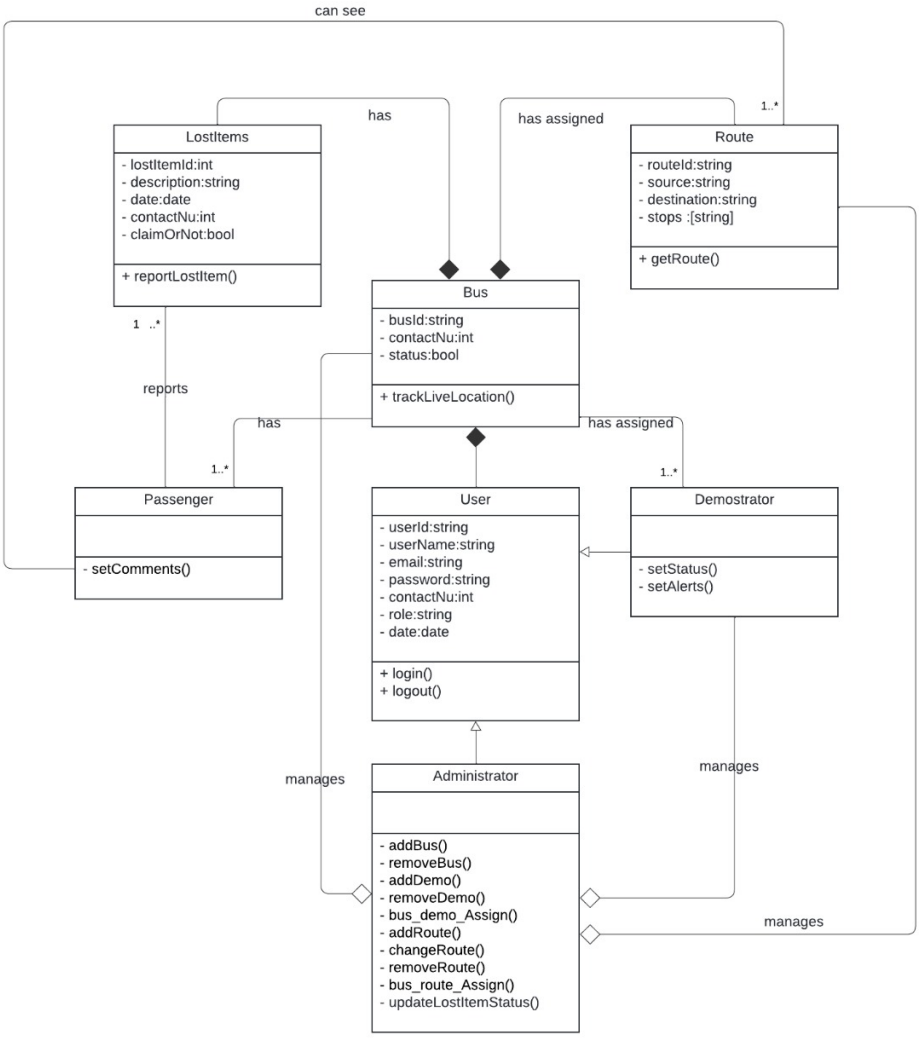


Figure 4.2: Class Diagram of TrackNGo System

4.5 System Design

4.6 Architectural Design

TrackNGo follows a layered **three-tier architecture** to ensure modularity, scalability, and maintainability:

- **Presentation Layer (Frontend)**: Built using React.js, this layer handles user interactions, including real-time bus tracking, fare calculation, and lost-and-found reporting. Bootstrap and Material-UI ensure responsive and accessible design across devices.
- **Application Layer (Backend)**: Node.js and Express.js manage API routing, business logic, and third-party integrations (e.g., Google Maps API). RESTful APIs enable seamless communication between frontend and backend.

- **Data Layer**: MongoDB stores dynamic data such as bus routes, user profiles, and lost-item records.—

4.7 Component Design

- **Real-Time Tracking Module**: Integrates Google Maps API to fetch GPS data from conductors' smartphones and visualize bus locations on a live map.
- **Fare Calculation Engine**: Uses distance-based algorithms and predefined fare matrices to automate fare estimates. Integrated with the `POST /api/fare` endpoint.
- **Lost-and-Found System**: A centralized MongoDB collection (`LostItem`) stores item descriptions, locations, and statuses. Users submit reports via a React form linked to `POST /api/lostitem`.

4.8 Database Design

- **Entity-Relationship Model (ERD)**:
  - `Bus`: Bus ID, route details, GPS coordinates, conductor contact.
  - `User`: Commuter profiles, travel history, notification preferences.
  - `LostItem`: Item description, report timestamp, status (resolved/pending).
- **Schema Design**:
  - `BusRoute`: Embeds sub-documents for stops, distances, and fare rules.

4.8.1 Entity Relations Diagram

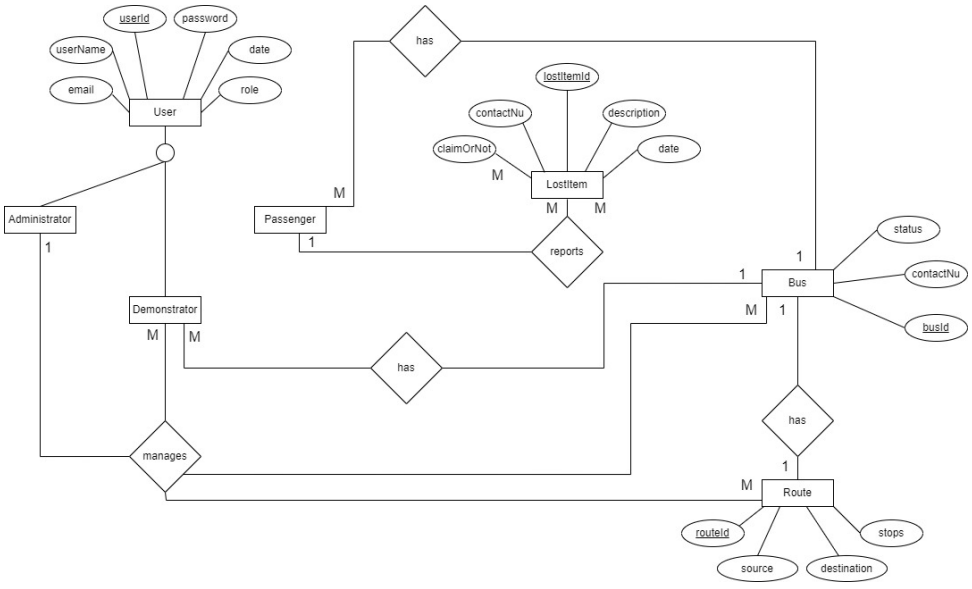


Figure 4.3: Entity Relationship Diagram

Chapter 5

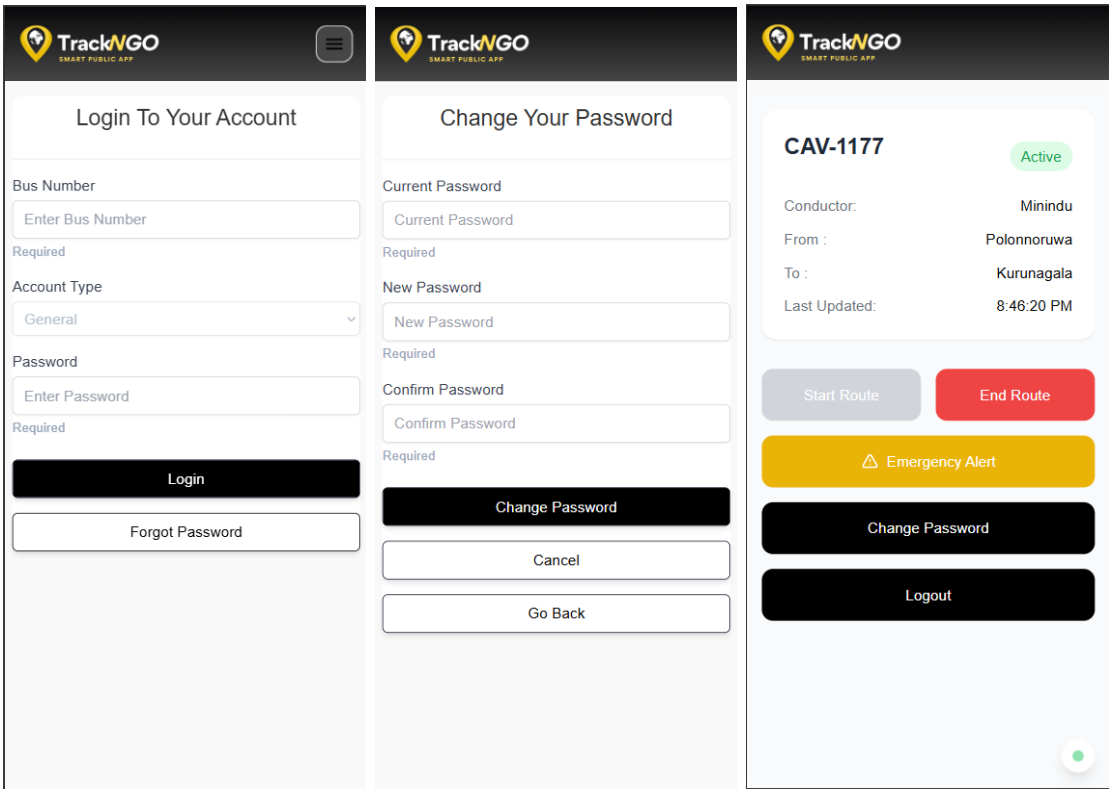
Presentation of Results

5.1 Introduction

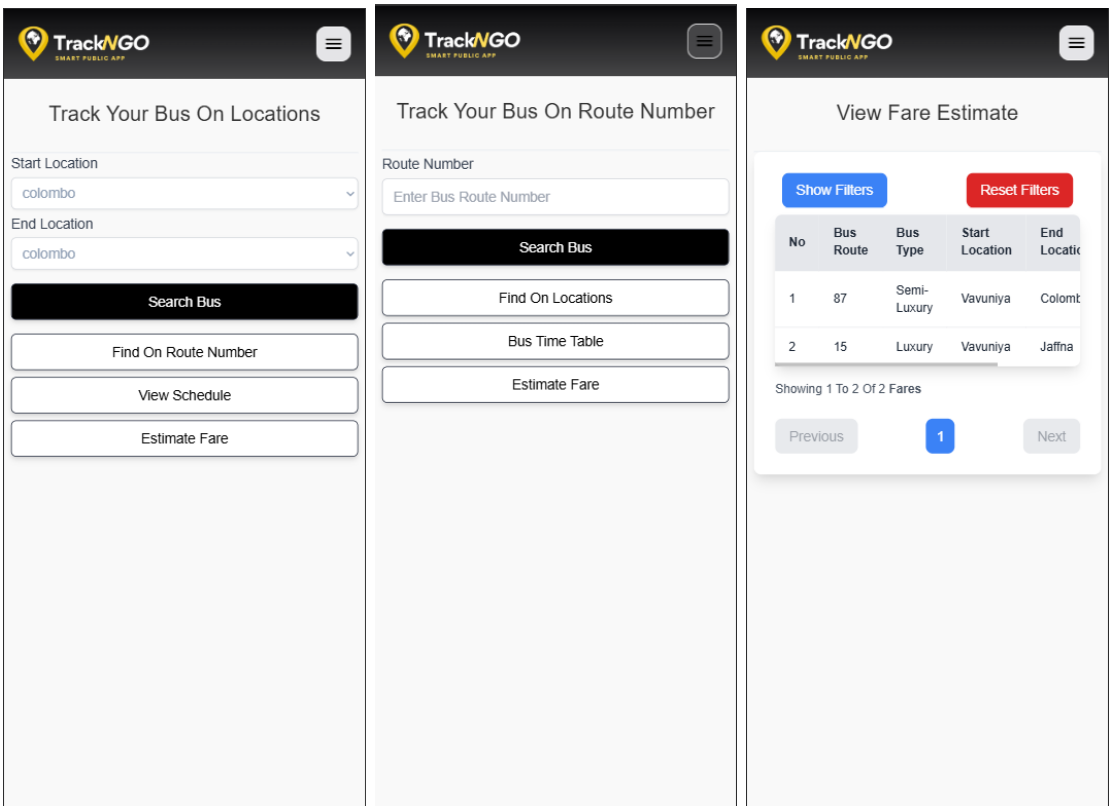
This chapter demonstrates the practical outcomes of the TrackNGo project, showcasing how the theoretical frameworks and methodologies outlined in earlier chapters translate into a functional, user-centric solution. Through a combination of visual evidence, performance metrics, and user feedback, the results validate the system's effectiveness in addressing Sri Lanka's public transport challenges. Key components include:

- **System Interfaces**: Screenshots of the app's core features, such as real-time bus tracking, fare calculation, and the lost-and-found portal, illustrate the user experience and design integrity.
- **Programming Environment**: A detailed overview of the tools and technologies.

5.2 Interface Design

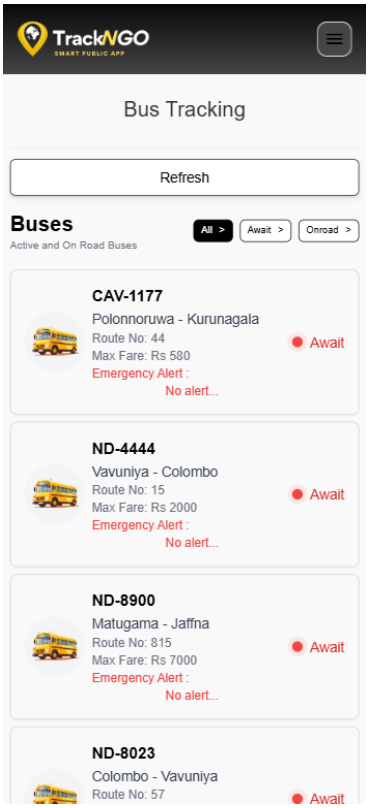


(a) Conductor Login (b) Change Password (c) Conductor Dashboard

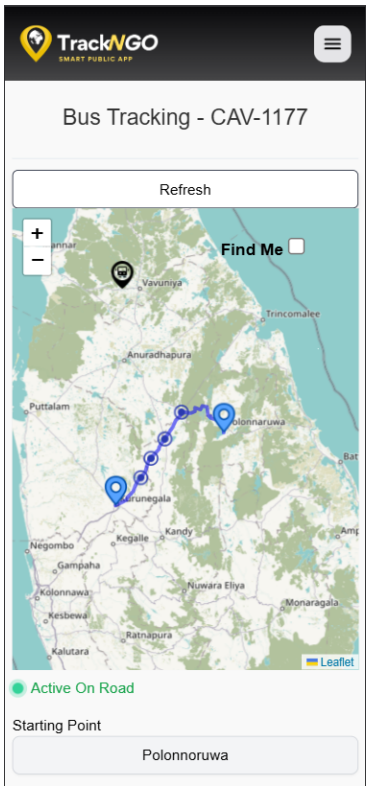


(d) User HomePage (e) Bus Search By Route (f) Bus Search By Route

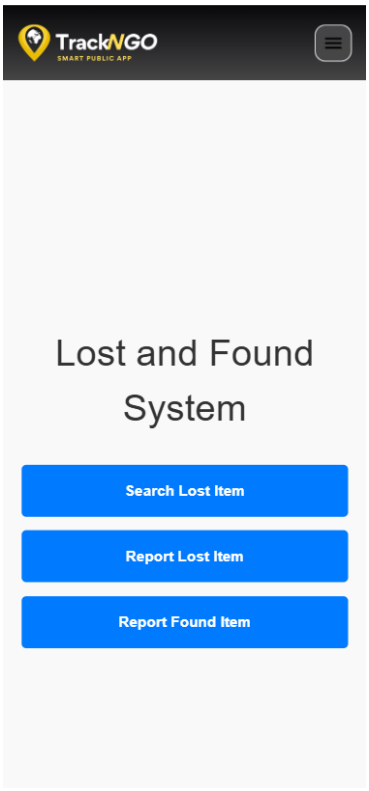
Figure 5.1: Interface Design - Conductor



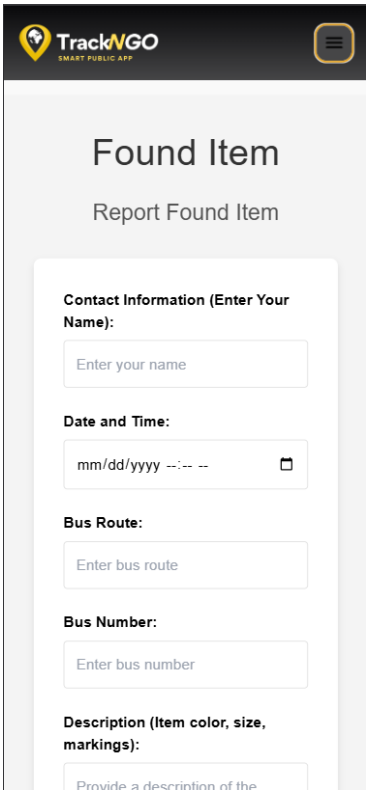
(a) Bus Tracking 01



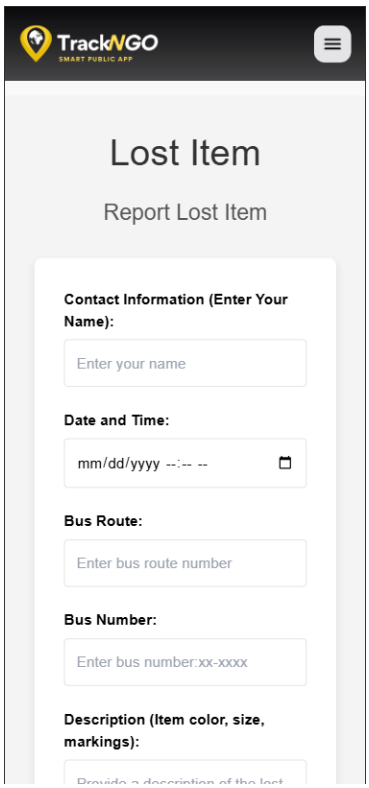
(b) Bus Tracking 02



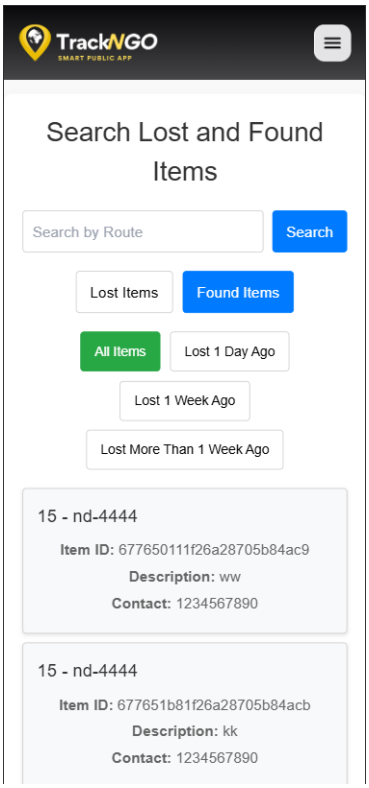
(c) Lost and Found



(d) Report Found Item

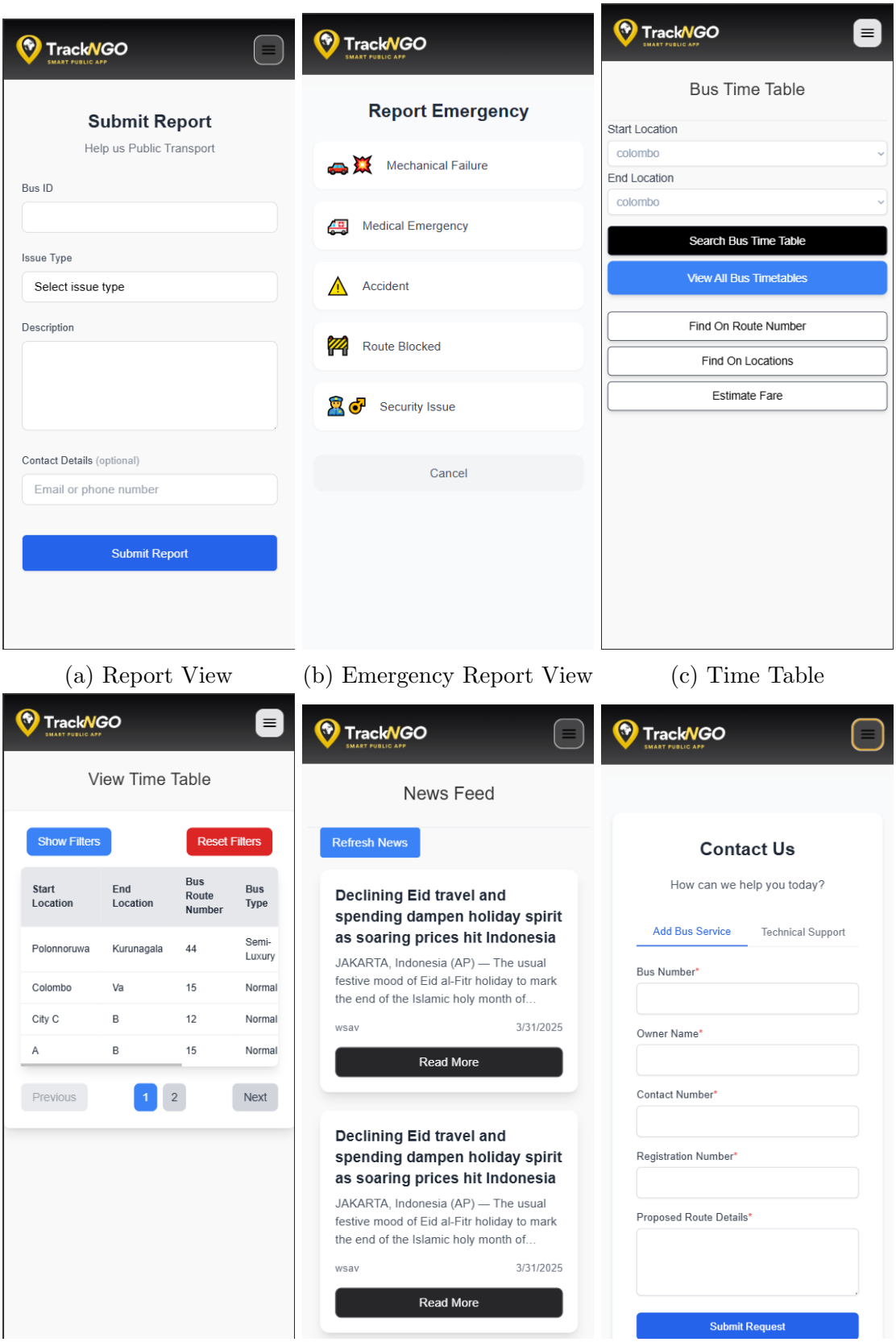


(e) Report Lost Item



(f) All Lost and Found Items

Figure 5.2: Interface Design - Conductor Part 2



(a) Report View (b) Emergency Report View (c) Time Table

(d) Time Table View (e) News View (f) Contact Us

Figure 5.3: Interface Design Conductor Part 3

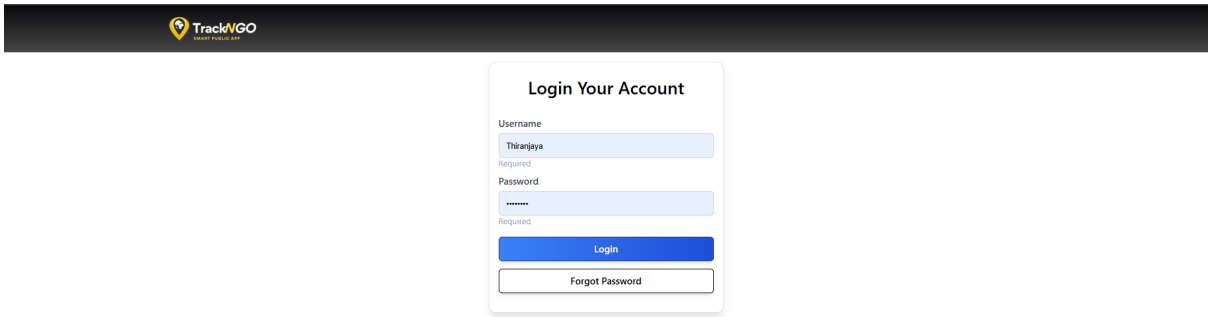


Figure 5.4: Admin Login

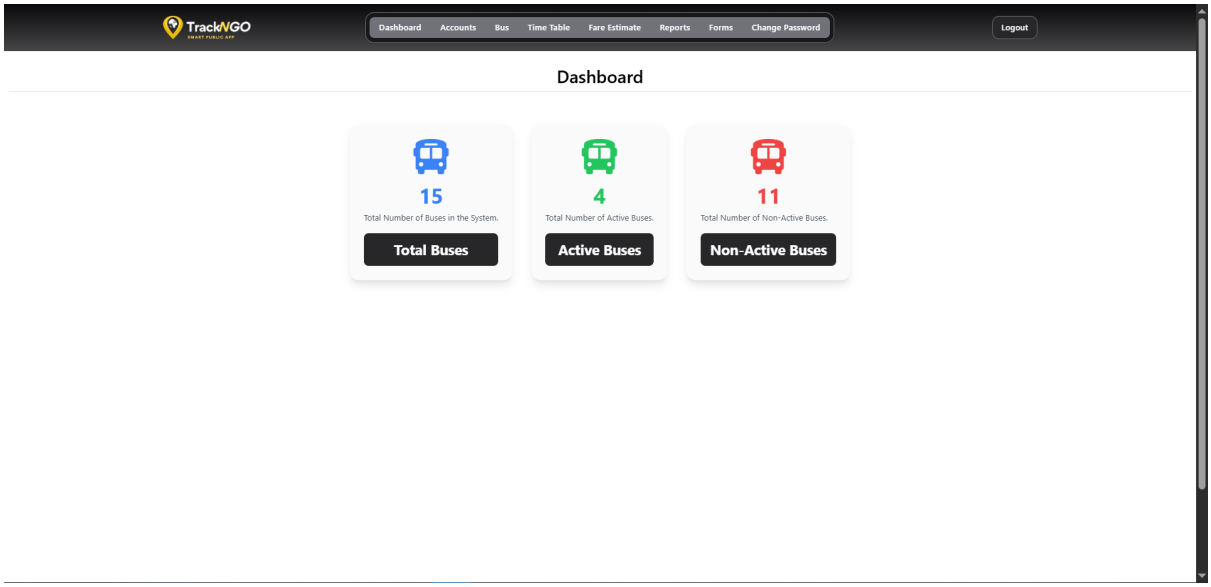


Figure 5.5: Admin Dashboard

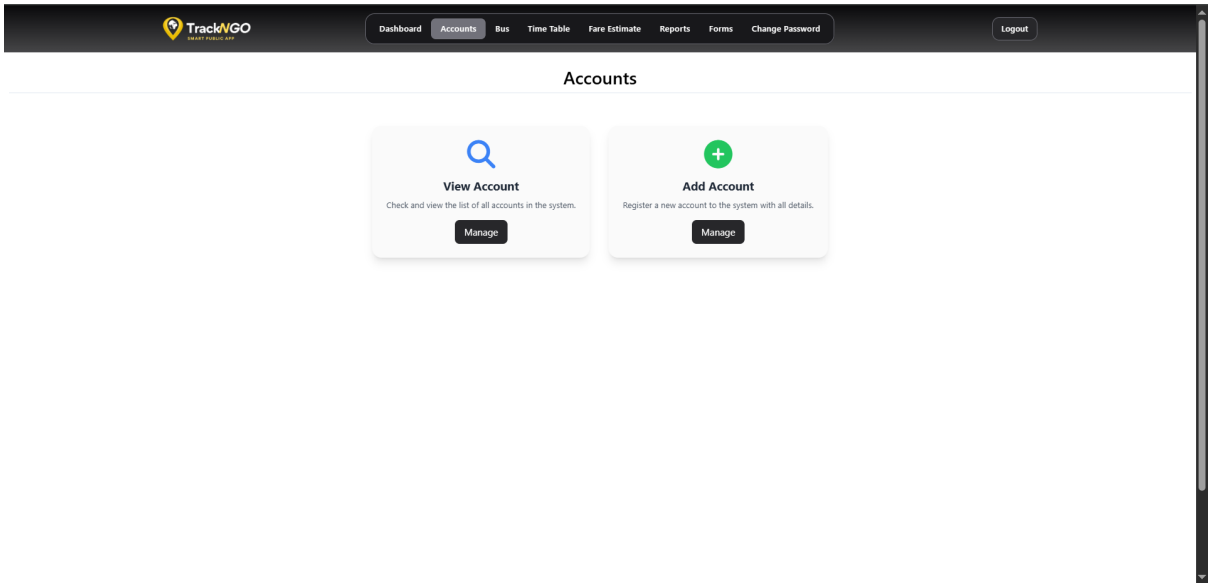


Figure 5.6: Admin Accounts

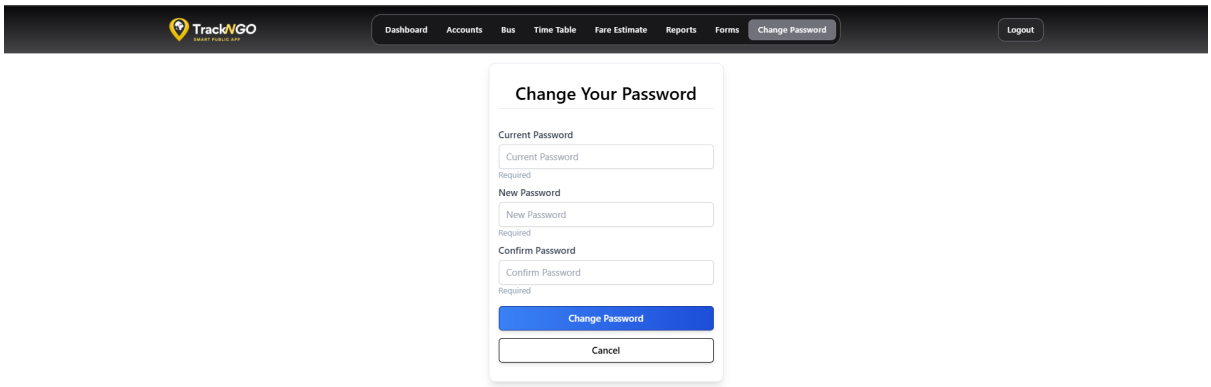


Figure 5.7: Admin Change Password

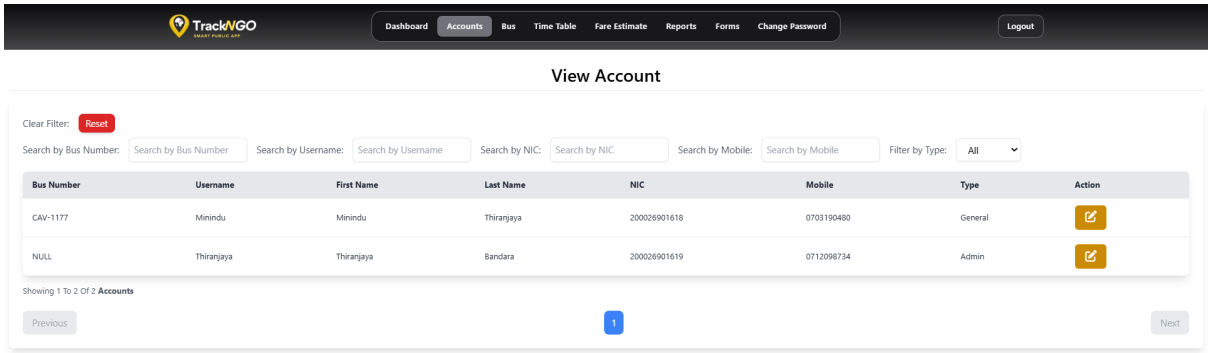


Figure 5.8: Admin All Accounts

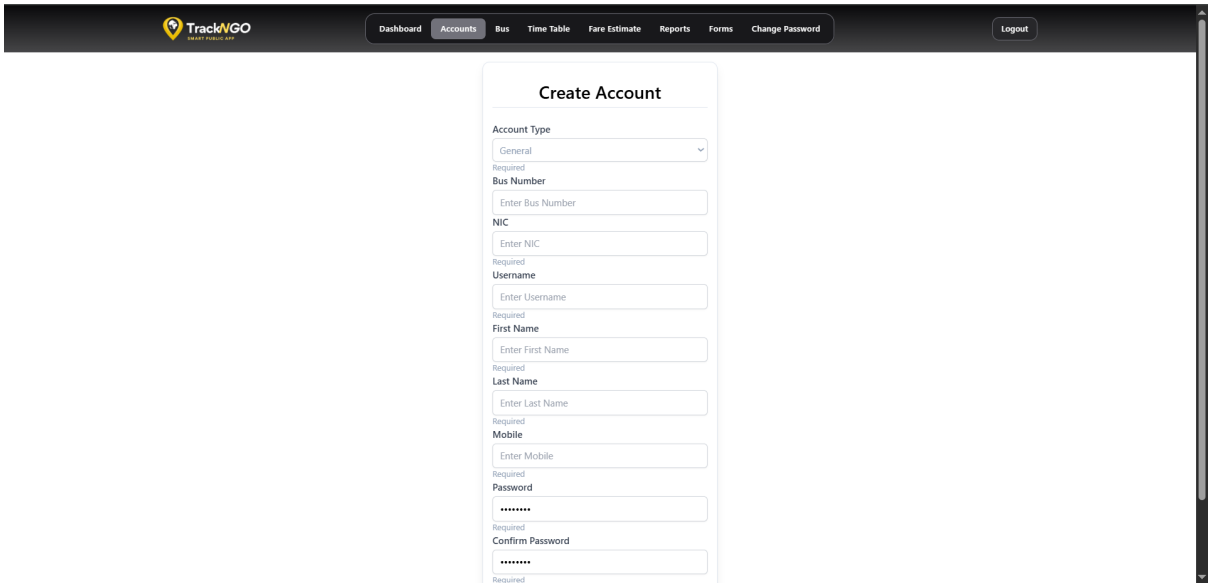


Figure 5.9: Admin Create Account

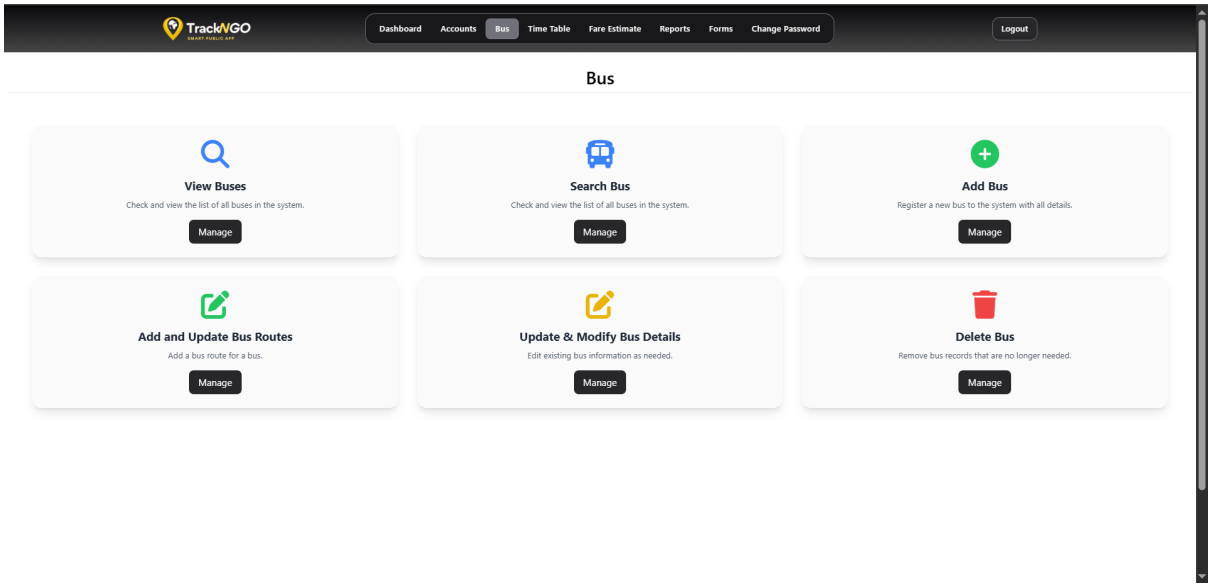


Figure 5.10: Bus Dashboard

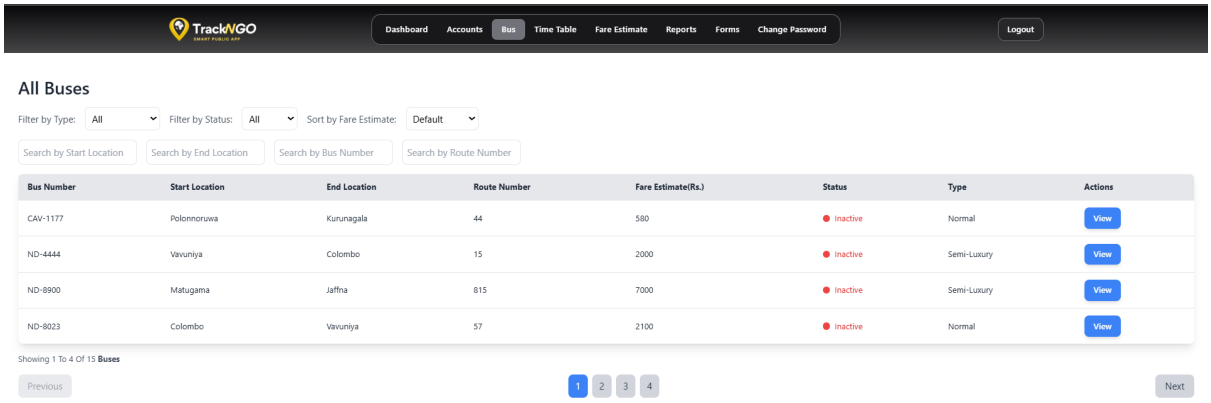


Figure 5.11: Admin Bus View

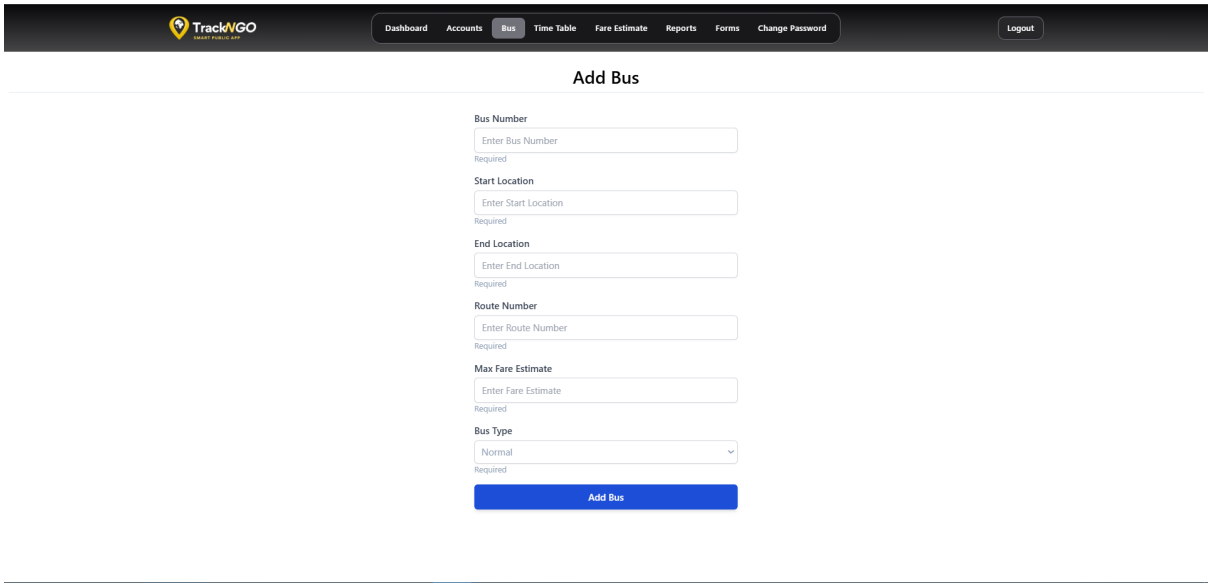


Figure 5.12: Admin Add Bus

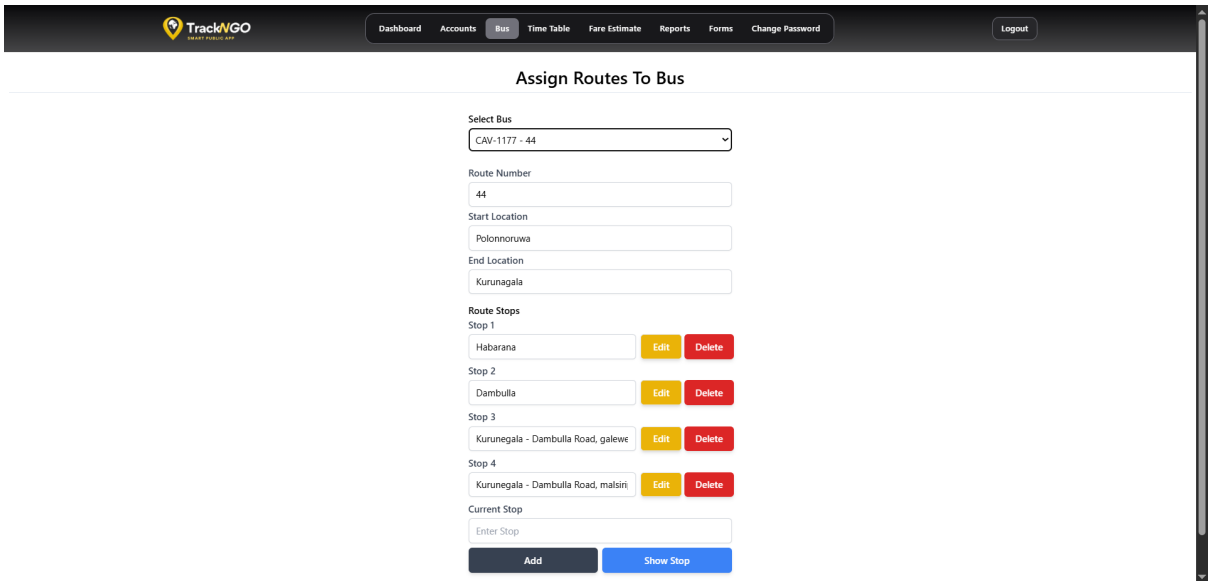


Figure 5.13: Admin Add Bus Route

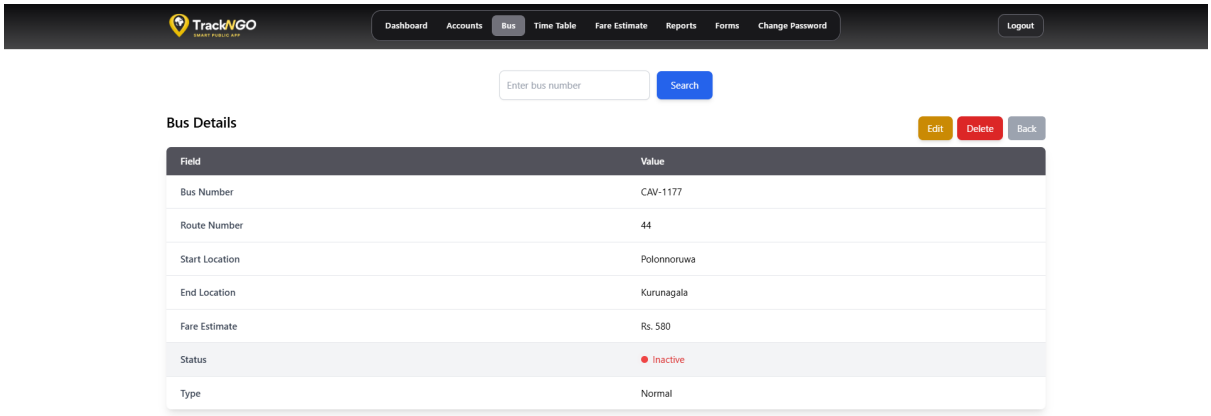


Figure 5.14: Bus Details Edit

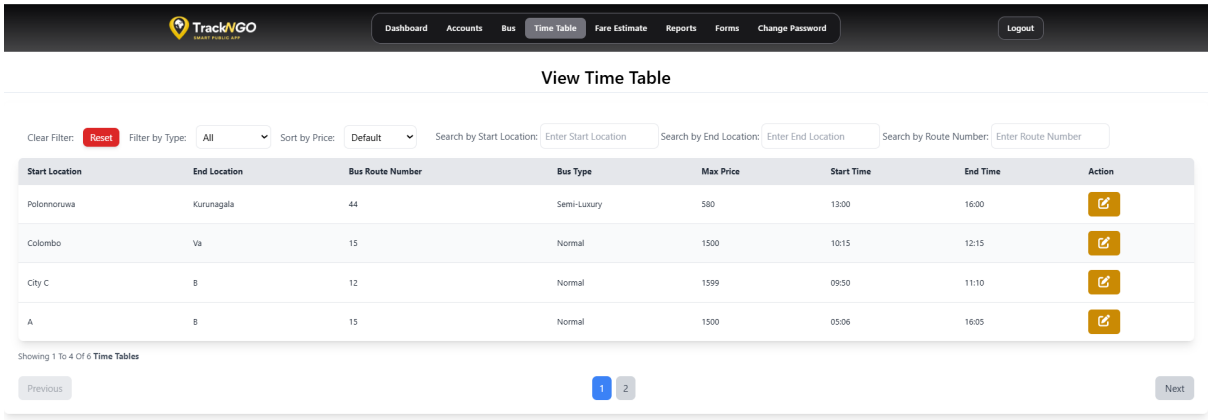


Figure 5.15: Admin Timetable

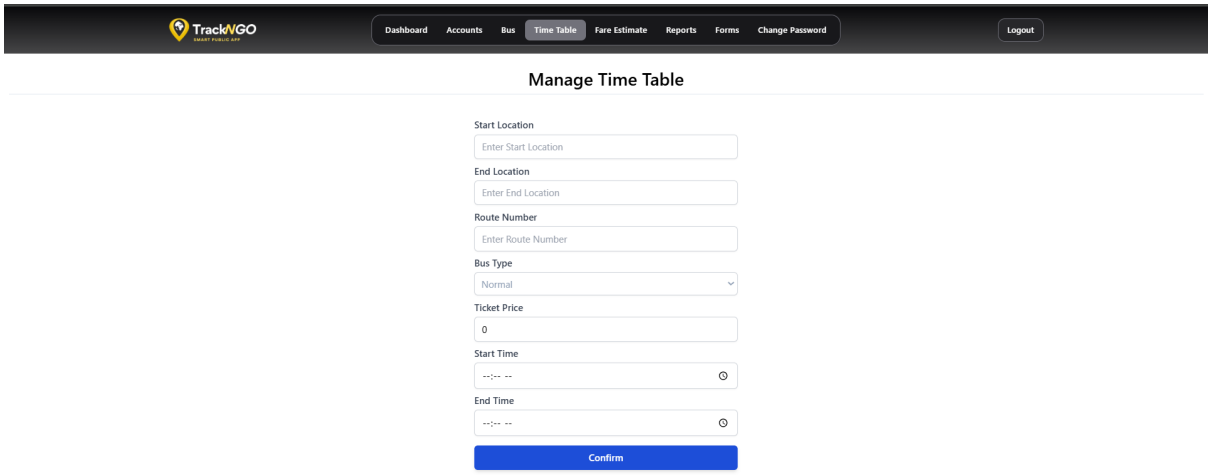


Figure 5.16: Add Timetable

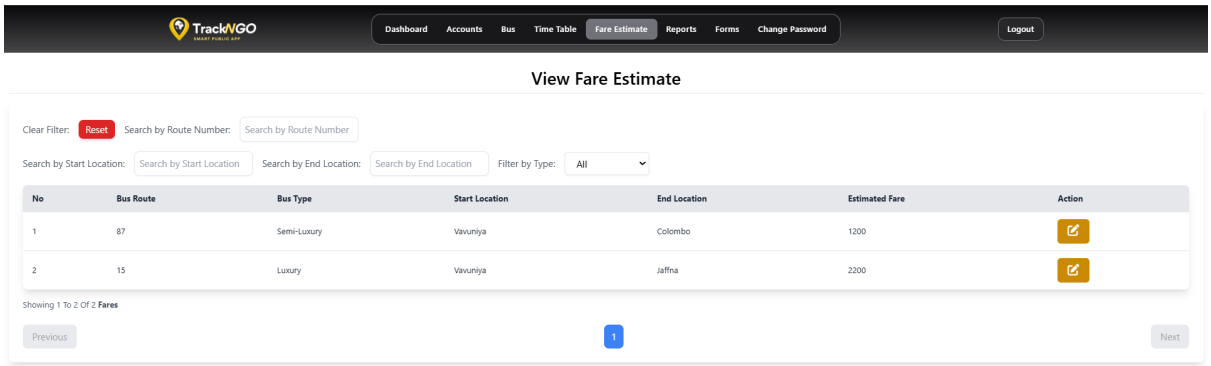


Figure 5.17: Fare Estimate View

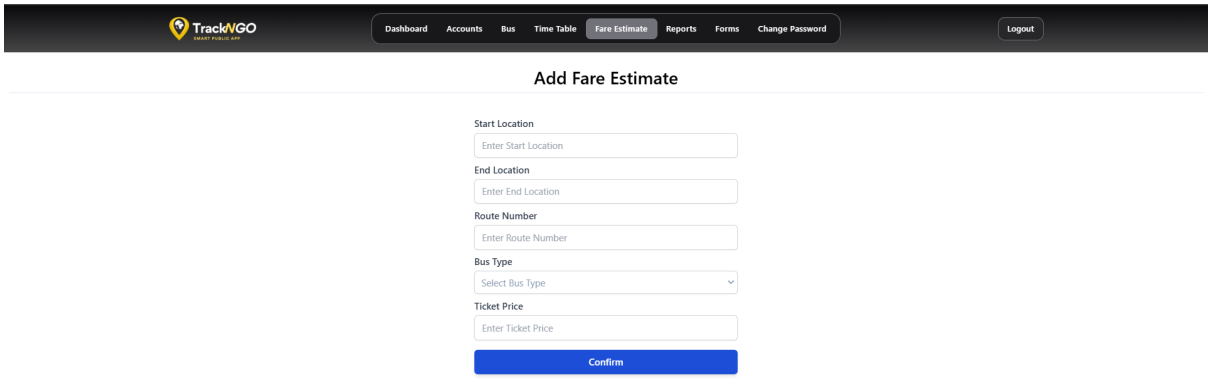


Figure 5.18: Add Fare Estimate

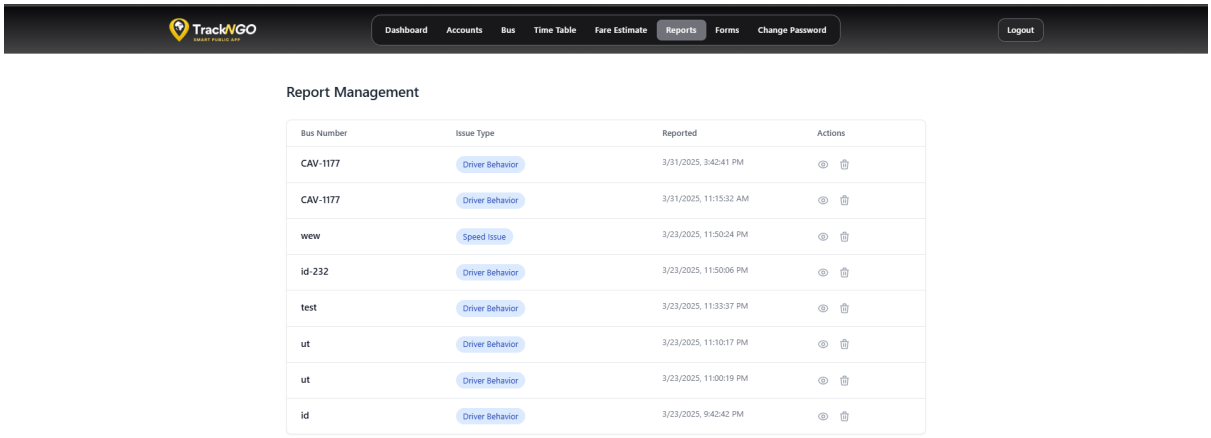


Figure 5.19: Reports View

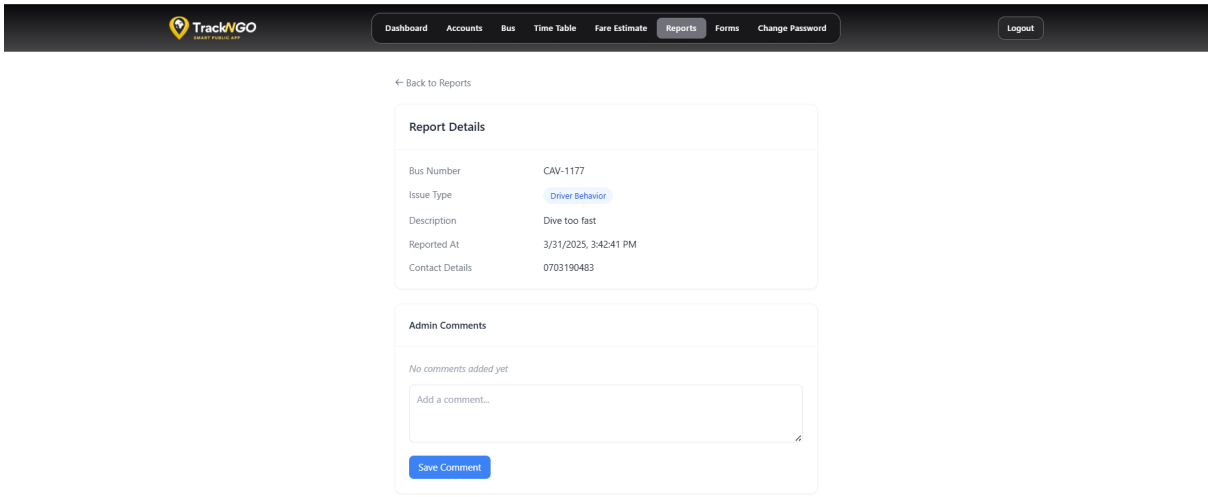


Figure 5.20: Report View

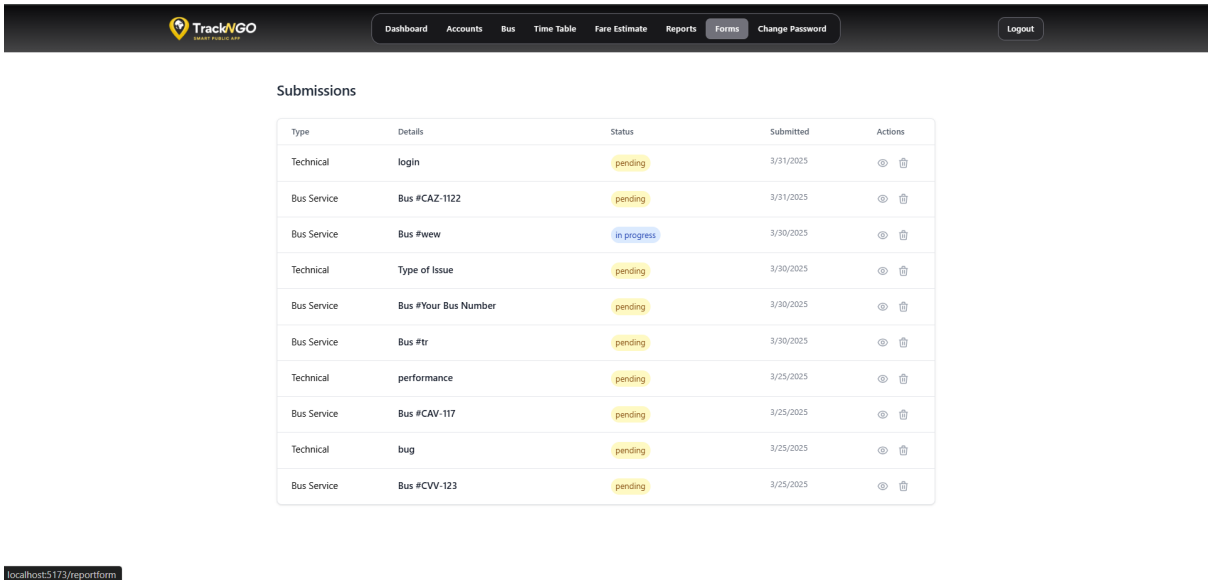


Figure 5.21: Technical and Requests

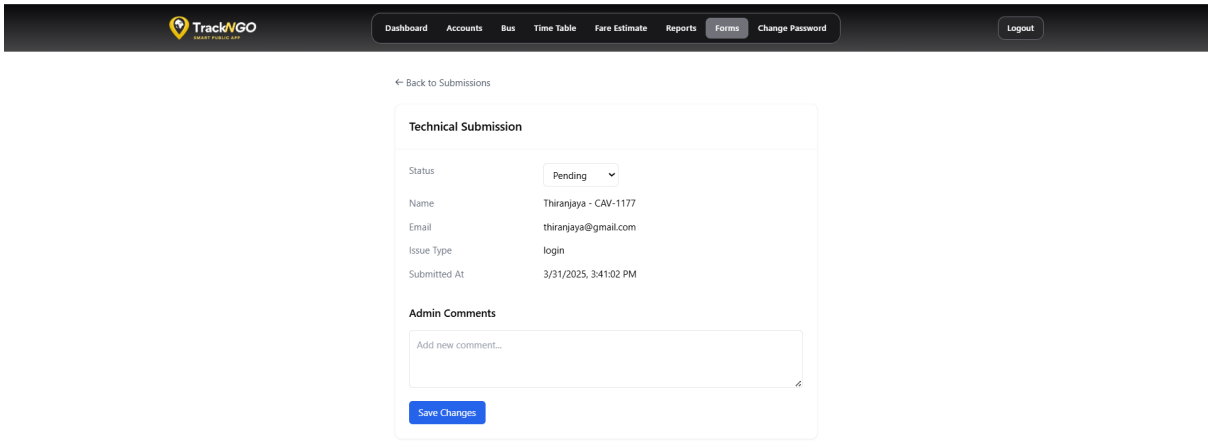


Figure 5.22: View Submission

5.3 Programming Environment

The development process leveraged several tools to enhance efficiency and effectiveness:

- **Frontend Tools:**
  - React for dynamic user interfaces.
- **Backend Tools:**
  - Node.js with Express for creating APIs.
  - MongoDB for database management.
- **API Integration Tools:**
  - Google Maps API for GPS tracking and route navigation.
  - News Api from NewsData.io

Chapter 6

Limitation,Recommendation and Conclusion

6.1 Introduction

This chapter critically evaluates the TrackNGo project by reflecting on its scope, constraints, and broader implications. While the system addresses key challenges in Sri Lanka's public transport ecosystem, it is essential to acknowledge its limitations and propose actionable recommendations for future enhancements. The chapter concludes by synthesizing the project's contributions to modernizing urban mobility and improving commuter experiences. Key areas of focus include:

- **Limitations**: Technical and operational constraints encountered during development, such as dependency on GPS accuracy, infrastructural challenges, and phased implementation scope.
- **Recommendations**: Strategic proposals for expanding features (e.g., multi-modal transport integration), improving scalability, and fostering stakeholder collaboration.
- **Conclusion**: A summary of the project's achievements in aligning with its core objectives, alongside its potential to inspire sustainable, tech-driven transportation reforms in Sri Lanka and beyond.

By addressing these dimensions, the chapter provides a perspective on the project's impact while making the way for future innovation in public transport systems.

6.2 Limitations

The TrackNGo project, while innovative in addressing Sri Lanka's public transport challenges, faces several limitations that warrant consideration:

6.2.1 Technical Limitations

- **GPS Dependency and Accuracy**: Real-time tracking relies on GPS signals, which may be inconsistent in densely built urban areas or remote regions. Signal delays or inaccuracies could lead to misleading arrival time estimates.

- **Internet Connectivity**: The app requires stable internet access for real-time updates. In areas with poor network coverage, users may experience limited functionality, undermining the app's reliability.
- **Device Compatibility**: The system assumes widespread smartphone adoption among conductors and commuters. Outdated devices or incompatible operating systems may restrict access to critical features.

6.2.2 Infrastructural Constraints

- **Legacy Bus Systems**: Many buses lack modern hardware (e.g., GPS-enabled devices), forcing reliance on conductors' personal smartphones, which introduces variability in data quality and availability.
- **Power Supply Issues**: Frequent power outages in Sri Lanka could disrupt backend server operations or charging infrastructure for GPS devices, affecting service continuity.

6.2.3 User Adoption Challenges

- **Digital Literacy Gap**: Elderly or rural commuters may struggle with app navigation, limiting its usability and inclusivity.
- **Resistance to Change**: Commuters accustomed to traditional methods might hesitate to adopt the app, slowing initial uptake and engagement.

6.2.4 Operational and Scalability Limitations

- **Phased Implementation**: The current scope focuses on major cities, leaving rural areas underserved. Expanding coverage requires significant investment in infrastructure and stakeholder collaboration.
- **Multi-Modal Integration**: Future expansion to trains or taxis would necessitate partnerships with disparate transport providers, complicating system interoperability.

6.2.5 Financial and Regulatory Challenges

- **Sustained Funding**: Long-term maintenance costs (e.g., server hosting, app updates) pose financial risks if revenue streams (e.g., ads, partnerships) underperform.
- **Policy Barriers**: Bureaucratic delays in obtaining government approvals or aligning with existing transport regulations could hinder scalability.

6.2.6 Data Privacy Concerns

- **Sensitive Data Handling**: Collecting user location data raises privacy issues. Ensuring compliance with local data protection laws (e.g., Sri Lanka's Data Protection Act) requires robust encryption and transparency, which may strain technical resources.

These limitations highlight the need for adaptive strategies to mitigate risks while advancing the project's core mission of modernizing public transport in Sri Lanka.

6.3 Further Development

6.3.1 Introduction

The initial implementation of TrackNGo focuses on providing core functionalities such as real-time bus tracking, fare calculations, route planning, and lost item reporting. However, to ensure the long-term success and relevance of the app, it is crucial to plan for future enhancements and scalability. This section outlines the potential areas for further development, addressing both technical advancements and user-centric features.

6.3.2 Integration of AI for Predictive Analysis

Artificial Intelligence (AI) can enhance the user experience and operational efficiency of TrackNGo:

- **Predictive Arrival Times:** AI algorithms can analyze historical traffic patterns and real-time data to provide accurate predictions of bus arrival times.
- **Route Optimization:** Dynamic route optimization for buses based on traffic conditions and passenger density.
- **User Behavior Insights:** Personalized recommendations based on user travel history and preferences.

6.3.3 Enhanced Multi-Modal Transportation Support

Expanding TrackNGo to support additional transportation modes can increase its utility:

- **Railway Integration:** Real-time updates for train schedules and delays.
- **Ride-Sharing Services:** Partnering with ride-hailing platforms to offer first-mile and last-mile connectivity.
- **Cycling Routes:** Including bike-friendly routes and integration with bike-sharing services.

6.3.4 Offline Functionality

Introducing offline capabilities can improve accessibility:

- **Offline Route Maps:** Allow users to download route maps and schedules for use without internet connectivity.
- **Cached Data:** Store recent tracking and notification data locally for quick access.

6.3.5 Scalability Enhancements

As user adoption grows, scalability becomes a critical factor:

- **Cloud-Based Microservices:** Transitioning to a microservices architecture to handle increasing user demands efficiently.
- **Load Balancing:** Implementing load balancers to distribute traffic evenly across servers.
- **Global Expansion:** Adapting the app to support multilingual interfaces and regional transport networks.

6.3.6 Advanced Security Measures

As data privacy becomes increasingly important, TrackNGo should implement advanced security features:

- **Biometric Authentication:** Enable fingerprint or facial recognition for secure logins.
- **Regular Security Audits:** Conduct periodic audits to identify and mitigate vulnerabilities.

6.3.7 Community Engagement Features

Building a community around TrackNGo can foster loyalty and active user participation:

- **Feedback Portal:** Provide users with a platform to share feedback and suggest new features.
- **Social Integration:** Allow users to share their travel experiences and updates on social media platforms.

6.3.8 Green Initiatives

Promoting sustainable transportation aligns with global environmental goals:

- **Carbon Footprint Tracker:** Enable users to track their carbon savings by choosing public transport over private vehicles.
- **Eco-Friendly Notifications:** Provide tips and insights on sustainable travel practices.

6.3.9 Conclusion

Further development of TrackNGo will ensure its relevance and adaptability in the evolving public transportation landscape. By incorporating advanced technologies, expanding service offerings, and addressing user needs, TrackNGo can become a comprehensive, future-ready solution for smart transportation.

6.4 Conclusion

The Smart Public Transport App represents a significant step forward in modernizing and improving public transportation in Sri Lanka, with a particular focus on bus services. As commuters increasingly demand more accessible, efficient, and real-time transport information, the necessity for such a solution has become evident. By addressing critical issues like poor scheduling, the absence of real-time tracking, and inefficient routes, the app provides a robust solution benefiting both transport operators and users.

This initiative has established a comprehensive plan for the creation and deployment of the Smart Public Transport App, leveraging a cutting-edge tech stack, including the MERN (MongoDB, Express, React, Node.js) framework, to ensure seamless backend and frontend integration.

The app offers an array of features—real-time tracking, fare calculation, route planning, push notifications, lost item reporting, and safety measures—that significantly enhance the commuter experience. These functionalities reduce waiting times, improve safety, and provide additional services such as personalized news updates and last-mile connectivity through partnerships with ride-hailing services.

References

[1] J. Edirisinghe,Department of Engineering and Faculty of Engineering, University of Peradeniya. "Sri Lanka's Transport Systems: Embracing New Technologies for a Better Future". In: *Daily News* (2023). URL: https://archives1.dailynews.lk/2023/05/23/features/303906/sri-lanka%E2%80%99s-transport-systems-embracing-new-technologies-better-future (visited on 01/21/2024).

[2] *Sri Lanka Toward an Urban Transport Strategy for Colombo. A Technical Note Energy and Infrastructure Department South Asia Region Work in Progress*. Technical Report. World Bank, 2001. URL: https://documents1.worldbank.org/curated/pt/476731468101389366/pdf/262970Colombo1UT1Strategy1final.pdf (visited on 01/21/2024).

[3] L. dell'Olio, A. Ibeas, and P. Cecín. "The quality of service desired by public transport users". In: *Transport Policy* 18.1 (2011), pp. 217–227. DOI: 10.1016/j.tranpol.2010.08.005.

[4] *SLTB Bus Tracker*. Eseat.lk. 2024. URL: https://tracker.eseat.lk/sltb/bus-tracker/all (visited on 01/21/2024).

[5] *Chalo App*. URL: https://chalo.com/app/ (visited on 01/21/2024).

[6] *Sakay.ph — The best commute directions in Metro Manila*. URL: https://www.sakay.ph/ (visited on 01/21/2024).

[7] Rusul Abduljabbar et al. "Applications of Artificial Intelligence in Transport: An Overview". In: *Sustainability* 11.1 (2019). ISSN: 2071-1050. DOI: 10.3390/su11010189. URL: https://www.mdpi.com/2071-1050/11/1/189.

[8] Y. Zhuravlova. *IoT App Development: Tools, Platforms, and Tech Stack*. 2024. URL: https://www.eliftech.com/insights/iot-app-development/ (visited on 01/21/2024).