**FYP Project on University Buses Management**

**Table of Contents**

**Chapter 2: Literature Review**

1. Introduction
2. Student Transportation Systems

2.1. Traditional Systems

2.2. Modern Approaches

2.2.1. Intelligent Transportation Systems (ITS)

2.2.2. Mobile Applications

2.2.3. Cloud Computing

1. Key Technologies

3.1. RFID (Radio-Frequency Identification)

3.2. GPS (Global Positioning System)

3.3. Flutter

3.4. Next.js

1. Related Work
2. Research Gaps and Objectives

**Chapter 3: System Requirements**

1. Introduction
2. Functional Requirements

2.1. Driver App

2.1.1. Student Validation

2.1.2. Bus Tracking

2.1.3. Communication

2.1.4. Bus Load Monitoring

2.2. Student/Parent App

2.2.1. Bus Tracking

2.2.2. Route Information

2.2.3. Fare Payment Information

2.2.4. Communication

2.3. Admin Panel

2.3.1. User Management

2.3.2. Route Management

2.3.3. Fare Management

2.3.4. Communication

2.3.5. System Monitoring

1. Non-Functional Requirements

3.1. Performance

3.2. Usability

3.3. Security

3.4. Reliability

3.5. Maintainability

3.6. Scalability

3.7. Compatibility

1. User Interface (UI) Requirements

4.1. Driver App

4.2. Student/Parent App

4.3. Admin Panel

**Chapter 2: Literature Review**

**2.1 Introduction**

**The University Buses Management System (UBMS) aims to enhance transportation services for students and parents by integrating technologies such as RFID scanning, GPS tracking, real-time data analytics, and mapping services. This chapter reviews existing literature and technologies pertinent to the project, including RFID technology in transportation systems, GPS-based bus tracking, load management in public transportation, predictive arrival systems, and the utilization of mapping APIs like Google Maps and Mapbox. Additionally, it examines existing student transportation management systems to identify current solutions and gaps.**

**2.2 RFID Technology in Transportation Systems**

**Radio Frequency Identification (RFID) is extensively used in transportation for access control and user validation. Embedding RFID tags in student cards facilitates automated scanning and validation, ensuring user authenticity. Studies highlight RFID’s reliability, speed, and accuracy in real-time scenarios, making it suitable for monitoring passenger eligibility in bus systems.**

**For instance, research demonstrates RFID’s effectiveness in reducing manual labor and minimizing errors in public transit systems. Additionally, RFID plays a crucial role in ensuring fare compliance, aligning with UBMS’s functionality of validating fee payments.**

**Incorporating time constraints in RFID validation—such as scanning within specific intervals—adds an extra layer of security. Research supports the effectiveness of time-based checks in preventing service misuse.**

**2.3 GPS-Based Bus Tracking**

**Global Positioning System (GPS) technology has transformed public transportation by enabling real-time vehicle tracking. This allows passengers and administrators to monitor bus locations, enhancing operational transparency and service quality. Studies indicate that GPS tracking systems improve user satisfaction by providing accurate location updates and estimated arrival times.**

**Incorporating GPS into UBMS assists parents and students in effective journey planning. Literature also emphasizes GPS’s importance in ensuring driver accountability and optimizing routes. Combining GPS data with predictive algorithms can enhance Estimated Time of Arrival (ETA) accuracy, a feature critical for user convenience.**

**2.4 Load Management in Public Transportation**

**Load management is vital in public transportation, with overloading leading to safety concerns and underutilization causing inefficiencies. Techniques like weight sensors and real-time passenger counting are proposed to monitor bus capacity. Studies show that integrating load management systems in buses can improve operational decisions, such as rerouting or scheduling additional buses during peak hours.**

**UBMS’s real-time load analysis ensures passenger safety and provides administrators with data to optimize bus fleet utilization. Research suggests that combining load data with historical patterns can forecast demand, a feature for future system enhancements.**

**2.5 Predictive Arrival Systems**

**Predictive algorithms for arrival times are essential in modern transportation systems, utilizing GPS data, traffic patterns, and historical records to provide accurate ETAs. Studies highlight that predictive systems enhance user trust and satisfaction. Integrating machine learning algorithms can significantly improve prediction accuracy.**

**For UBMS, predictive arrival features are crucial for reducing waiting times and improving convenience for students and parents. Research emphasizes the importance of presenting this data through intuitive interfaces, such as mobile apps, to enhance user experience.**

**2.6 Mapping APIs: Google Maps and Mapbox**

**Mapping services are integral to transportation management systems, providing visualization and geolocation functionalities. Google Maps API offers comprehensive mapping solutions, including real-time traffic data and route planning, which are essential for accurate ETA calculations. Its extensive database and reliability make it a popular choice for developers.**

**Mapbox, as a customizable map SDK, allows developers to design tailored map experiences. It offers features like dynamic theming, offline maps, and integration capabilities with various data sources. Mapbox’s flexibility and performance make it suitable for applications requiring specialized mapping solutions.**

**Both Google Maps API and Mapbox have their advantages. Google Maps is known for its extensive database and routing capabilities, while Mapbox offers greater customization and potentially more favorable pricing for large-scale applications. The choice between them should consider factors like specific project requirements, budget, and desired user experience.**

**2.7 Existing Student Transportation Management Systems**

**Several transportation management systems cater to student transit needs, offering features like GPS tracking, route optimization, and parent communication. Notable examples include:**

* **Tyler Technologies’ Student Transportation Software: Provides integrated solutions for bus routing, fleet maintenance, and parent communication, connecting various aspects of transportation management. citeturn0search0**
* **Transfinder: Offers school bus tracking and routing solutions, emphasizing efficient route planning and real-time tracking to enhance safety and operational efficiency. citeturn0search8**
* **Edulog: Combines school bus routing, GPS fleet tracking, student ridership management, and parent communication apps into a single platform, aiming to streamline transportation operations. citeturn0search14**
* **Loqqat: Provides a smart real-time school bus tracker and management software, enabling route scheduling and live tracking to ensure student safety. citeturn0search10**

**While these systems offer comprehensive features, they may not fully address specific requirements such as time-based validations or detailed load management tailored for university settings. UBMS aims to fill these gaps by integrating RFID-based validation with time constraints, real-time load monitoring, and predictive arrival features into a cohesive solution customized for university transportation.**

**2.8 Conclusion**

**The literature review underscores the potential of integrating technologies like RFID, GPS, predictive algorithms, and advanced mapping APIs in modernizing transportation systems. By combining these technologies with functionalities such as time-based validations, load monitoring, and customizable mapping solutions, the University Buses Management System aspires to set a new standard in university transportation. Future enhancements could explore AI-based route optimization and expanded analytics to further improve efficiency and user satisfaction.**

**This chapter establishes a foundation for understanding the technological landscape and informs the design and implementation decisions discussed in subsequent chapters of the SRS document.**

**Chapter 3: System Requirements**

**3.1 Functional Requirements**

* **Driver App:**
  + **Student Validation:** 
    - Scan student RFID cards using the device's camera or an integrated RFID reader.
    - Verify student card validity (i.e., whether the student has paid the necessary fees).
    - Check if the student has been scanned within the allowed time frame (e.g., two hours).
    - Record student scan data (timestamp, student ID, etc.) on the system.
    - Display relevant information to the driver (e.g., student name, route, seat availability).
  + **Bus Tracking:** 
    - Continuously track the bus's location using GPS.
    - Transmit real-time location data to the server.
    - Display current location, speed, and other relevant information on the driver's app.
    - Receive and display route guidance and navigation instructions.
  + **Communication:** 
    - Receive notifications and alerts from the admin panel (e.g., route changes, emergencies).
    - Send notifications to the admin panel (e.g., bus arrival at stops, incidents).
  + **Bus Load Monitoring:** 
    - Monitor and display the current occupancy level of the bus (e.g., through manual input or sensor data).
    - Alert the driver if the bus is overcrowded or undercrowded.
* **Student/Parent App:**
  + **Bus Tracking:** 
    - View real-time location of the assigned bus.
    - Track the bus's progress along the route.
    - Estimate the next stop arrival time.
    - View historical trip data (e.g., past routes, arrival times).
  + **Route Information:** 
    - View bus schedules, routes, and stops.
    - Plan and view travel routes.
    - Receive notifications about delays, cancellations, and other service disruptions.
  + **Fare Payment Information:** 
    - View payment history and status.
    - Receive reminders for upcoming fee payments.
  + **Communication:** 
    - Send feedback and inquiries to the university administration.
    - Receive notifications and alerts from the system (e.g., bus arrival alerts, emergency messages).
* **Admin Panel:**
  + **User Management:** 
    - Manage user accounts (students, parents, drivers, administrators).
    - Assign roles and permissions to users.
  + **Route Management:** 
    - Define and manage bus routes, stops, and schedules.
    - Assign buses and drivers to routes.
    - Monitor bus locations in real-time.
    - Generate reports on bus utilization and passenger traffic.
  + **Fare Management:** 
    - Manage student fees and payment records.
    - Generate reports on fare collection and revenue.
  + **Communication:** 
    - Send notifications and alerts to drivers and users.
    - Manage communication channels (e.g., in-app messaging, email).
  + **System Monitoring:** 
    - Monitor system performance and identify any issues.
    - Generate system logs and reports.

**3.2 Non-Functional Requirements**

* **Performance:** 
  + Real-time tracking with minimal latency.
  + Fast response times for user interactions.
  + High availability and reliability of the system.
* **Usability:** 
  + User-friendly and intuitive interfaces for all applications.
  + Easy navigation and accessibility for all users.
  + Clear and concise information presentation.
* **Security:** 
  + Secure data storage and transmission.
  + Strong authentication and authorization mechanisms.
  + Protection against cyber threats and data breaches.
* **Reliability:** 
  + High system availability and fault tolerance.
  + Robust error handling and recovery mechanisms.
* **Maintainability:** 
  + Easy to maintain, update, and upgrade the system.
  + Well-documented code and system architecture.
* **Scalability:** 
  + Ability to handle increasing user demand and data volume.
* **Compatibility:** 
  + Compatible with various mobile devices and operating systems.
  + Compatible with different browsers and devices for the admin panel.

**3.3 User Interface (UI) Requirements**

* **Driver App:** 
  + Clear and concise display of student information and bus status.
  + Large, easy-to-read buttons and controls.
  + Minimal distractions and clutter on the screen.
* **Student/Parent App:** 
  + Intuitive and easy-to-navigate interface.
  + Interactive maps and visualizations.
  + Clear and concise display of information.
* **Admin Panel:** 
  + User-friendly dashboard with key performance indicators.
  + Customizable reports and visualizations.
  + Efficient data entry and management tools