**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**

This chapter explores existing research, technologies, and methodologies relevant to our University Buses Management System.

**2.1. Student Transportation Systems**

* **Traditional Systems:** 
  + Discuss limitations of traditional manual systems: Inefficiency, lack of real-time tracking, difficulty in managing routes and schedules, potential for fraud.
  + Mention examples of existing student transportation systems (if any) in your local context or similar institutions.
* **Modern Approaches:** 
  + **Intelligent Transportation Systems (ITS):** 
    - Highlight the role of ITS in improving efficiency and safety in transportation, including technologies like GPS, RFID, and telematics.
    - Discuss how ITS can be applied to university bus management, such as real-time tracking, route optimization, and predictive maintenance.
  + **Mobile Applications:** 
    - Discuss the growing trend of mobile applications in transportation management.
    - Emphasize the benefits of user-friendly mobile apps for students, parents, and drivers, such as real-time tracking, route information, and communication features.
  + **Cloud Computing:** 
    - Discuss the advantages of cloud-based solutions for data storage, processing, and analysis in transportation systems.
    - Mention the benefits of cloud computing for scalability, accessibility, and cost-effectiveness.

**2.2. Key Technologies**

* **RFID (Radio-Frequency Identification):** 
  + Explain the principles of RFID technology and its applications in various fields, including access control, inventory management, and transportation.
  + Discuss the advantages of RFID for student identification and fare collection in university buses.
* **GPS (Global Positioning System):** 
  + Describe the core principles of GPS technology and its role in real-time location tracking.
  + Discuss how GPS data can be used to track buses, monitor driver behavior, and optimize routes.
* **Flutter:** 
  + Briefly introduce Flutter as a cross-platform framework for building natively compiled applications from a single codebase.
  + Mention its advantages for developing mobile applications for both Android and iOS platforms.
* **Next.js:** 
  + Introduce Next.js as a React framework for building web applications, particularly those with server-side rendering.
  + Discuss its suitability for building the admin panel due to its performance and scalability.

**2.3. Related Work**

* Conduct a brief literature search for similar projects or research papers.
  + Look for examples of student transportation management systems, mobile applications for bus tracking, and the use of RFID and GPS in such systems.
* Analyze existing systems and identify their strengths and weaknesses.
* Highlight how your proposed system aims to address the limitations of existing solutions.

**2.4. Research Gaps and Objectives**

* Identify any gaps or limitations in existing research and systems.
  + For example, the lack of focus on student satisfaction, limited integration of real-time data with decision-making processes, or insufficient attention to user experience.
* Clearly state the research objectives of your project.
  + How will your system address the identified gaps and improve upon existing solutions?

**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