



EcoTransit: A Lean Framework for Efficient Public Transport Pooling Applications

1. Problem Statement

The complexity of pooling public transport applications often leads to high resource consumption and slow performance. This creates a need for an optimized, lean framework that can efficiently manage resources and improve the overall user experience.

2. Purpose, Scope, and Objectives

- **Purpose:** Develop a lean framework to streamline the sharing of pooling public transport applications, reducing resource usage and enhancing performance.
- **Scope:** The project will focus on pooling public transport applications built using modern frameworks and technologies, incorporating Agile methodologies within the Software Development Life Cycle (SDLC).
- **Objectives:**
 - Achieve a 30% reduction in resource utilization.
 - Attain a 20% increase in performance metrics.

3. Abstract (System Overview)

This project aims to develop a lean framework for pooling public transport applications, utilizing Agile methodologies throughout the SDLC. By optimizing resource usage and enhancing performance, the framework will offer a sustainable solution for developers, focusing on iterative development, user feedback, and continuous integration.

4. Introduction

With the increasing reliance on pooling public transport, the need for efficient resource management and high performance is critical. This project addresses these challenges by creating a lean framework that minimizes resource consumption while improving the overall performance of the application.

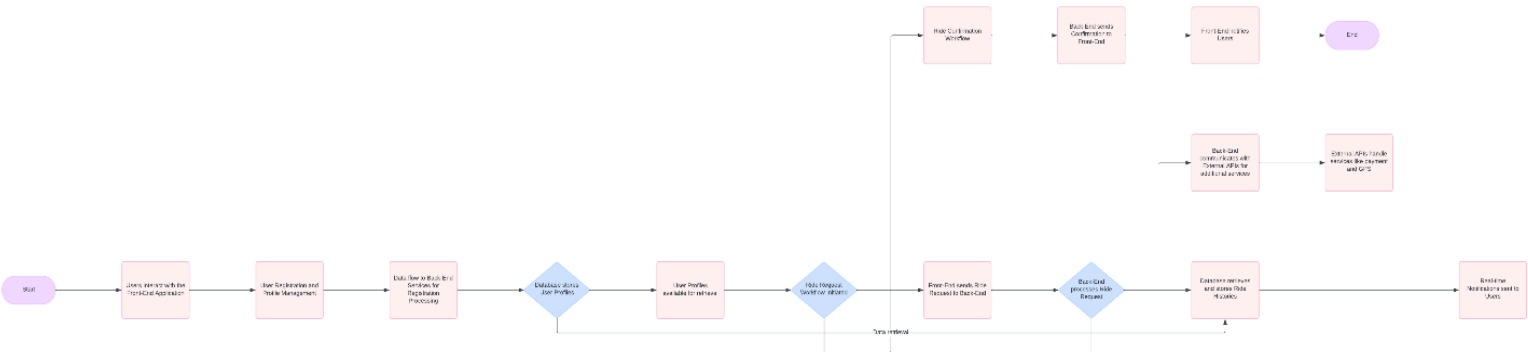
5. Literature Survey

Conduct a review of recent literature focusing on:

- Lean methodologies in software development.
- Performance optimization in pooling public transport applications.
- Resource management strategies.
- Agile practices in framework development.

6. System Architecture

- **Architecture Overview:** The system will consist of a client-side application and a server-side application, connected through RESTful APIs.
- **Components:**
 - **Front-end:** Built using React.js.
 - **Back-end:** Developed with Node.js and Express.js.
 - **Database:** Utilizes MongoDB for managing dynamic user and transport data.



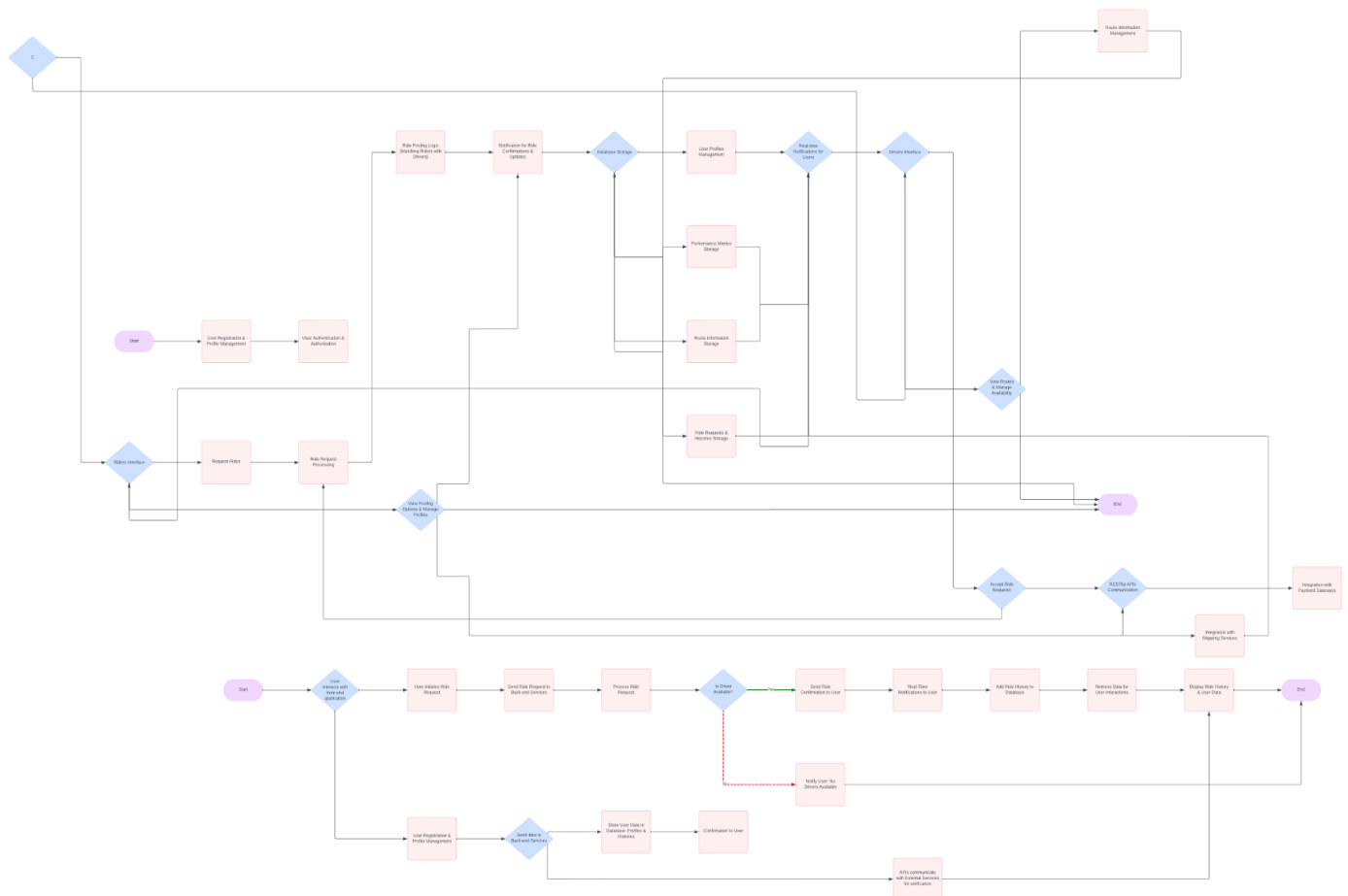
7. Dataset Preparation

- **Data Collection:** Gather data relevant to pooling public transport, including user interactions, route optimizations, and real-time tracking.
- **Data Cleaning:** Ensure the data is clean and reliable for effective analysis and reporting.

8. Research Methodology

- **Approach:** Employ an iterative Agile approach, focusing on sprints to develop features and gather user feedback.
- **Techniques:** Conduct data analysis using statistical methods to evaluate resource utilization and performance metrics.

9. Conceptual Framework



10. Technology Stack

- **Languages:**
 - **Front-end:** JavaScript, HTML, CSS.
 - **Back-end:** JavaScript (Node.js).
- **Frameworks:**
 - **Front-end:** React.js.
 - **Back-end:** Express.js.
 - **Database:** MongoDB.
- **Tools:** Git for version control, JIRA for project management, Jenkins for CI/CD.

11. Performance Analysis

- **Metrics:** Define key performance indicators (KPIs) such as response time, resource utilization, and throughput.
- **Testing:** Conduct performance tests using tools like JMeter to assess the framework's efficiency.

12. Conclusion (Implications)

The developed framework shows potential for significantly improving the performance and resource management of pooling public transport applications. By adopting lean principles, developers can create more sustainable and efficient solutions that better meet the needs of users.

13. Future Research

Future research could explore the integration of predictive analytics using machine learning to enhance route optimization and resource management. Additionally, testing the framework across different public transport models would provide insights into its adaptability.
