Software requirement Document

Student

Shahzad Nawaz	Vinod Kumar
S23BSEEN1M01116	S23BSEEN1M01102

Course Name

Software Construction

Course Teacher

Dr. MALIK DALER ALI AWAN

(Lecturer)

DEPARTMENT OF SOFTWARE ENGINEERING
FACULTY OF COMPUTING
THE ISLAMIA UNIVERSITY OF BAHAWALPUR

05-May-2025

Table of Contents

1. Introduction

1.1 Purpose

The University Transport Tracking System (UTTS) is designed to provide real-time tracking and management of university buses, schedules, and routes. It aims to improve transportation efficiency for students, faculty, and staff by offering live updates, notifications, and an admin dashboard for bus management.

1.2 Document Conventions

Bold Text: Key functionalities and system components.

Italic Text: Important notes and assumptions.

Code Blocks: Database schemas and technical details.

1.3 Intended Audience

University Administrators: Manage buses, routes, and notifications.

Students/Staff: Track bus locations, schedules, and receive alerts.

Developers: Understand system architecture for maintenance/updates.

1.4 Project Scope

Real-time bus tracking (status: Active/Delayed/Inactive).

Admin dashboard for bus, route, and user management.

Student profiles with personal and transport-related details.

Notifications & feedback system for updates and user input.

2. Overall Description

2.1 Product Perspective

UTTS is a web-based application with:

Frontend: HTML/CSS, JavaScript (Flask templating).

Backend: Python (Flask framework).

Database: MySQL (Tables: buses, users, routes, notifications, feedback, schedules).

2.2 User Classes

Role	**Permissions**
Admin	Manage buses, routes, users, send notifications
Driver	Update bus status (Limited access in future versions)
Student/Staff	View schedules, track buses, submit feedback

2.3 Operating Environment

Server: Linux/Windows with Python 3.8+, MySQL 5.7+.

Client: Web browsers (Chrome, Firefox, Edge).

2.4 Assumptions & Dependencies

Assumption: Buses have GPS devices for real-time tracking (simulated in current version).

Dependency: mysql-connector-python, Flask, Werkzeug for file uploads.

3. System Features & Requirements

3.1 Functional Requirements

3.1.1 User Authentication

ID	Requirement
FR1	Users can log in with username/password (UserManager).
FR2	New users can register (student role by default)

.

3.1.2 Bus Management

ID	Requirement
FR3	Admin can add/edit buses (BusManager).
FR4	Users view bus status (Active/Delayed/Inactive).

3.1.3 Route & Schedule Management

ID	Requirement
FR5	Admin defines routes with stops (RouteManager).
FR6	Users view schedules (ScheduleManager).

3.1.4 Notifications & Feedback

ID	Requirement
FR7	Admin sends alerts (e.g., delays) (NotificationManager).
FR8	Users submit feedback (FeedbackManager).

3.1.5 Student Profiles

ID	Requirement
FR9	Students upload profiles with photos (StudentInfoManager).

3.2 Non-Functional Requirements

ID	**Requirement**
NFR1	System responds within 2 seconds for 100+ users.
NFR2	Data encrypted in transit (HTTPS recommended).
NFR3	MySQL backups daily.

4. Database Schema

4.1 Tables Overview

buses: id, name, route, status

users: id, username, email, password, role (student/driver/admin)

routes: id, name, stops (text)

notifications: id, title, message, type (info/warning/alert)

feedback: id, name, email, message

schedules: Time-based routes (e.g., time_fg_to_bjc, day_type).

4.2 Sample Queries

```
-- Get active buses
SELECT * FROM buses WHERE status = 'Active';
-- Count delayed buses
SELECT COUNT(id) FROM buses WHERE status = 'Delayed';
```

5. User Interfaces

Login/Signup Pages: Credential input.

Dashboard: Displays buses, notifications, and schedules.

Admin Panel: CRUD operations for buses/routes/users.

Map View: Visualizes bus locations (future integration with Google Maps API).

6. Future Enhancements

GPS Integration: Real-time bus tracking via IoT devices.

Mobile App: Android/iOS companion app.

Predictive Analytics: Estimate arrival times using historical data.

7. Conclusion

UTTS streamlines university transport management with a scalable, role-based system. Current features focus on core functionality, with room for expansion via APIs and real-time tracking modules.

Appendices

Appendix A: Glossary of Terms

Term	**Definition**
CRUD	Create, Read, Update, Delete operations
UAT	User Acceptance Testing
GPS	Global Positioning System