

Software Design 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

Page | 1

University Transport Tracking System (UTTS)

Version History

Version	Date	Author Description
1.0	05-May-2025	Shahzad Nawaz Vinod Kumar

Table of Contents

Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions & Acronyms
- 1.4 References

System Overview

- 2.1 System Architecture
- 2.2 Design Constraints

High-Level Design

- 3.1 System Context Diagram
- 3.2 System Flowchart
- 3.3 Data Flow Diagram

Detailed Design

4.1 Use Case Diagram

4.2 Sequence Diagrams

4.3 Activity Diagrams (Insert Diagram Here)

4.4 Class Diagram

4.5 Entity-Relationship (ER) Diagram

Database Design

5.1 Schema Description

5.2 Sample Queries

User Interface Design

6.1 Wireframes

Non-Functional Requirements

7.1 Performance

7.2 Security

Future Enhancements

Conclusion

1. Introduction

1.1 Purpose

This document outlines the design specifications for the University Transport Tracking System (UTTS), aligning with the requirements specified in the Software Requirements Specification (SRS). It covers system architecture, data flow, and component interactions.

1.2 Scope

The system includes:

- 1-Real-time bus tracking (Active/Delayed/Inactive).
- 2-Admin dashboard for managing buses, routes, and users.
- 3-Student/Staff portals for tracking and feedback.
- 4-Notifications for delays and schedule changes.

1.3 Definitions & Acronyms

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

1.4 References

- 1-SRS Document (Provided)
- 2-Flask & MySQL Documentation
- 3-Source code

2. System Overview

2.1 System Architecture

The system follows a 3-tier architecture:

- 1-Frontend: HTML/CSS, JavaScript (Flask templating).
- 2-Backend: Python (Flask framework).
- 3-Database: MySQL.

2.2 Design Constraints

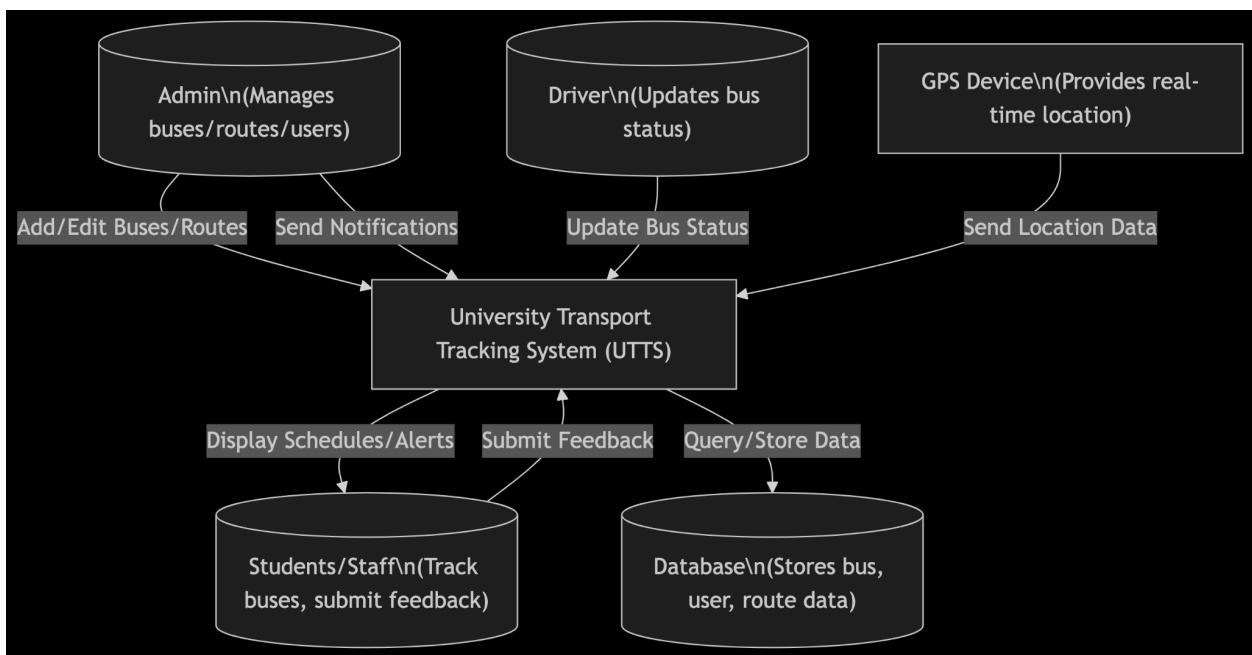
Dependencies: mysql-connector-python, Flask, Werkzeug.

Assumptions: GPS devices are simulated in the initial version.

3. High-Level Design

3.1 System Context Diagram

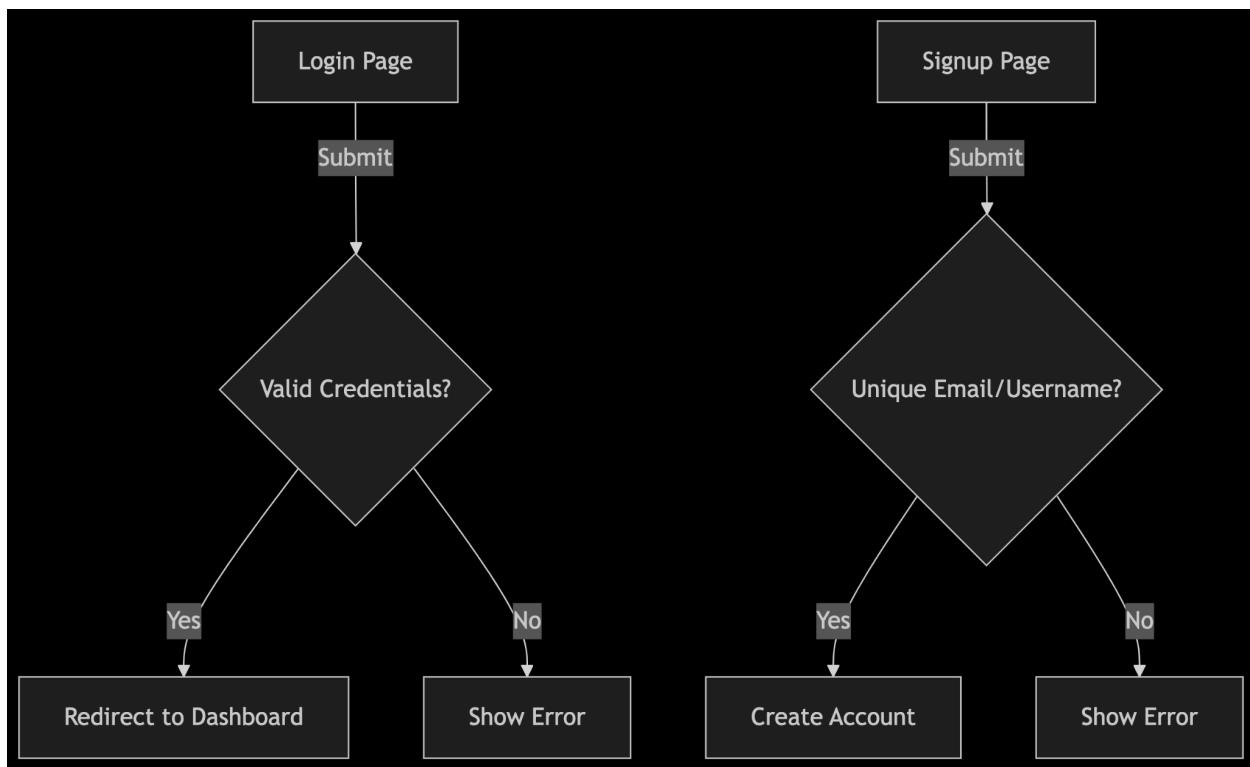
This Context Diagram section in the Software Design Document (SDD) provides a high-level overview of the system's interactions with external entities (users, devices, or other systems). It defines the system's boundaries by illustrating input/output data flows without delving into internal processes, serving as a foundational reference for stakeholders. This diagram ensures clarity on the system's scope, interfaces, and dependencies early in the design phase. By simplifying complex relationships into a single view, it bridges communication between technical and non-technical audiences. The context diagram sets the stage for detailed DFDs or use cases, ensuring alignment with business goals.



3.2 System Flowchart

This Flowchart section in the Software Design Document (SDD) visually represents the system's processes, decision points, and workflow logic using standardized symbols (e.g., rectangles for steps, diamonds for decisions). It clarifies complex algorithms, user interactions, or backend operations in a step-by-step manner, ensuring developers and stakeholders share a common understanding of the control flow. This section often complements use cases or pseudocode, bridging high-level requirements with technical implementation. Flowcharts are particularly useful for illustrating error handling, loops, and conditional branching, making them critical for debugging and maintenance. By documenting these processes, the SDD ensures consistency in logic execution across the development lifecycle.

1. User Authentication flow chart

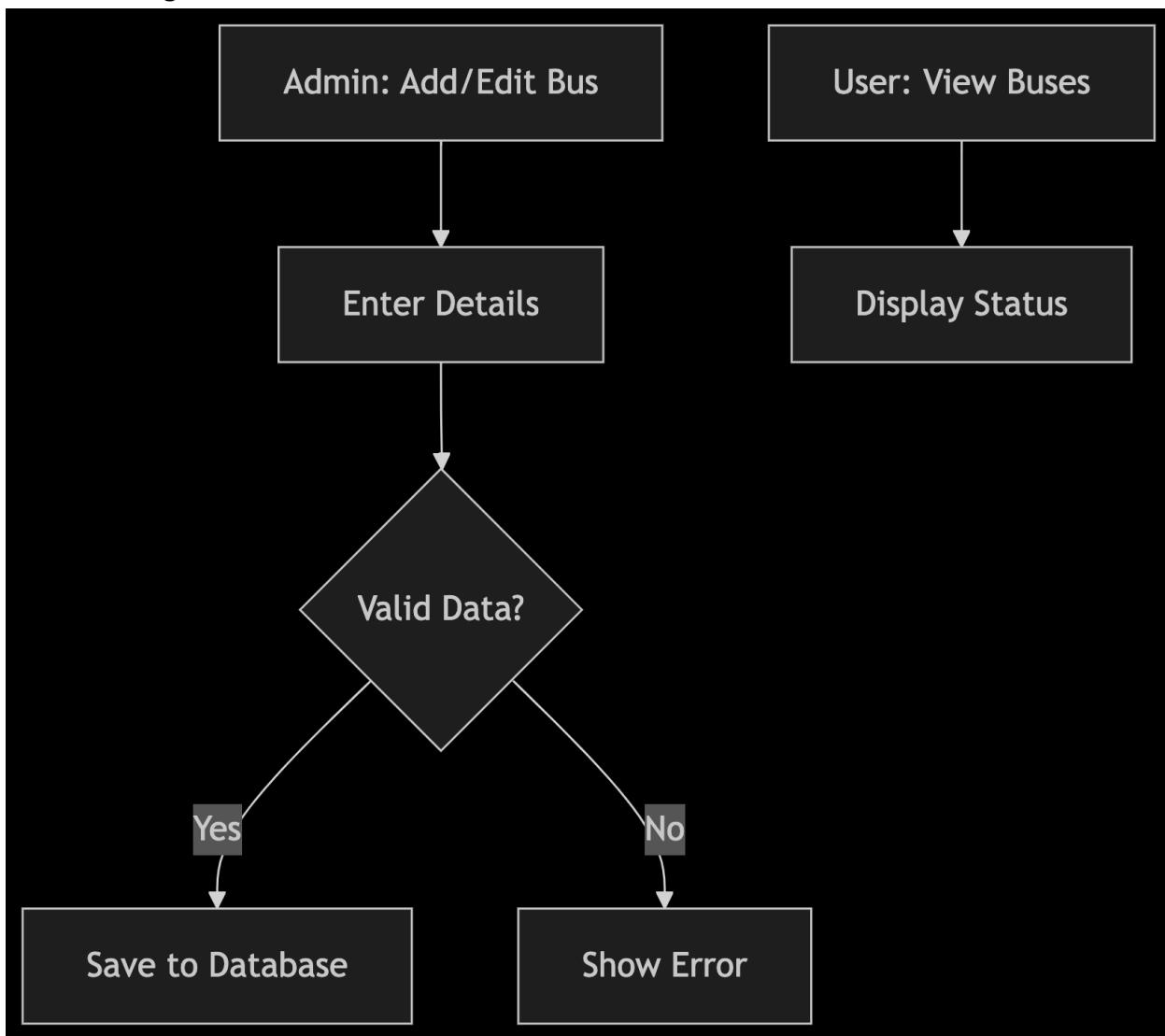


Start: User enters credentials (username/password) or selects social login.

Verification: System checks credentials against the database; if valid, grants access.

End: Successful login grants entry; failed attempts prompt retry or password reset.

2. Bus Management



Start

→ Admin/Driver logs in (Authentication)

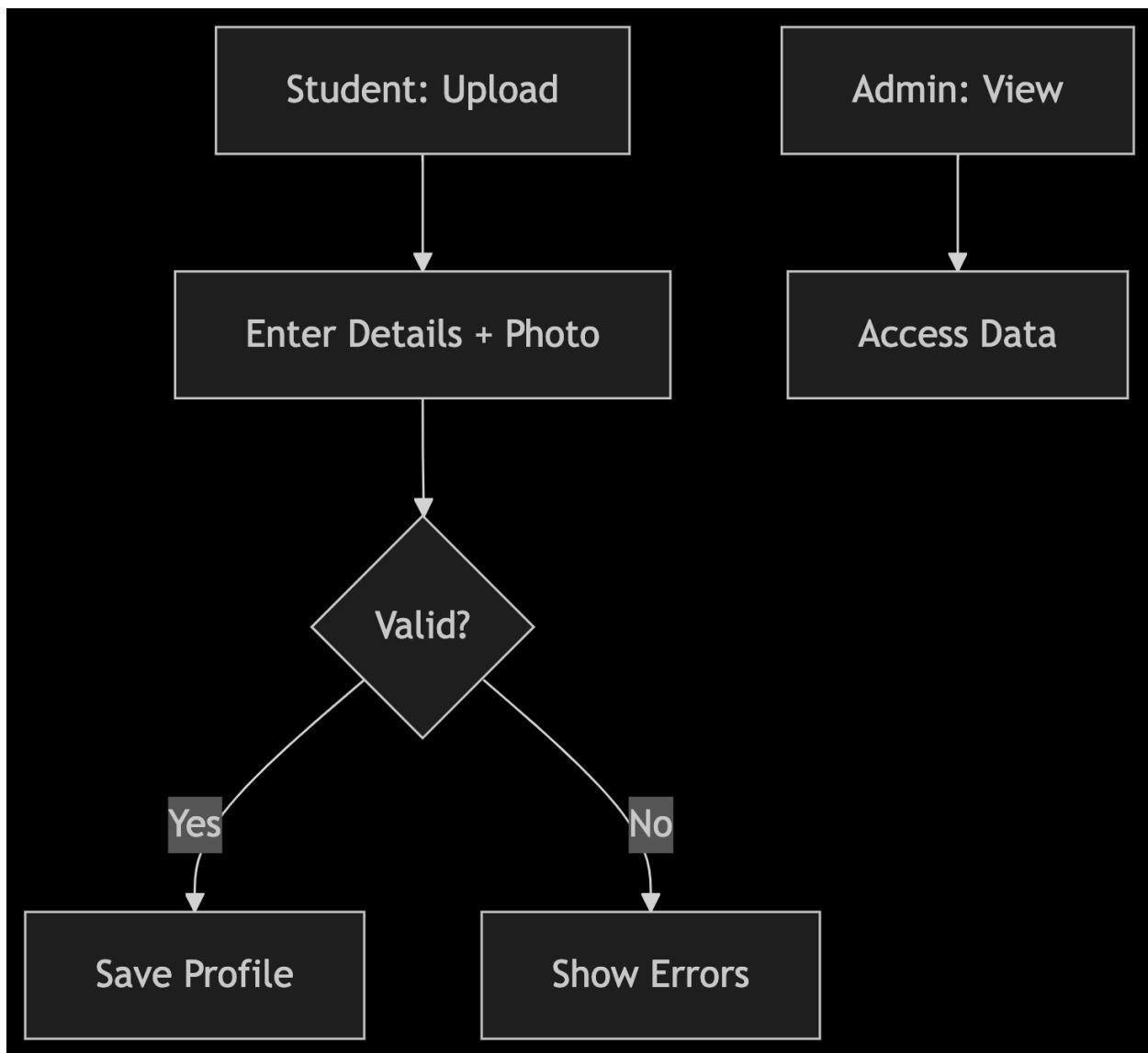
Admin Actions (If role = Admin)

- Add/Edit Routes & Schedules
- Assign Buses & Drivers
- Generate Reports

End

→ Logout or System Sleep

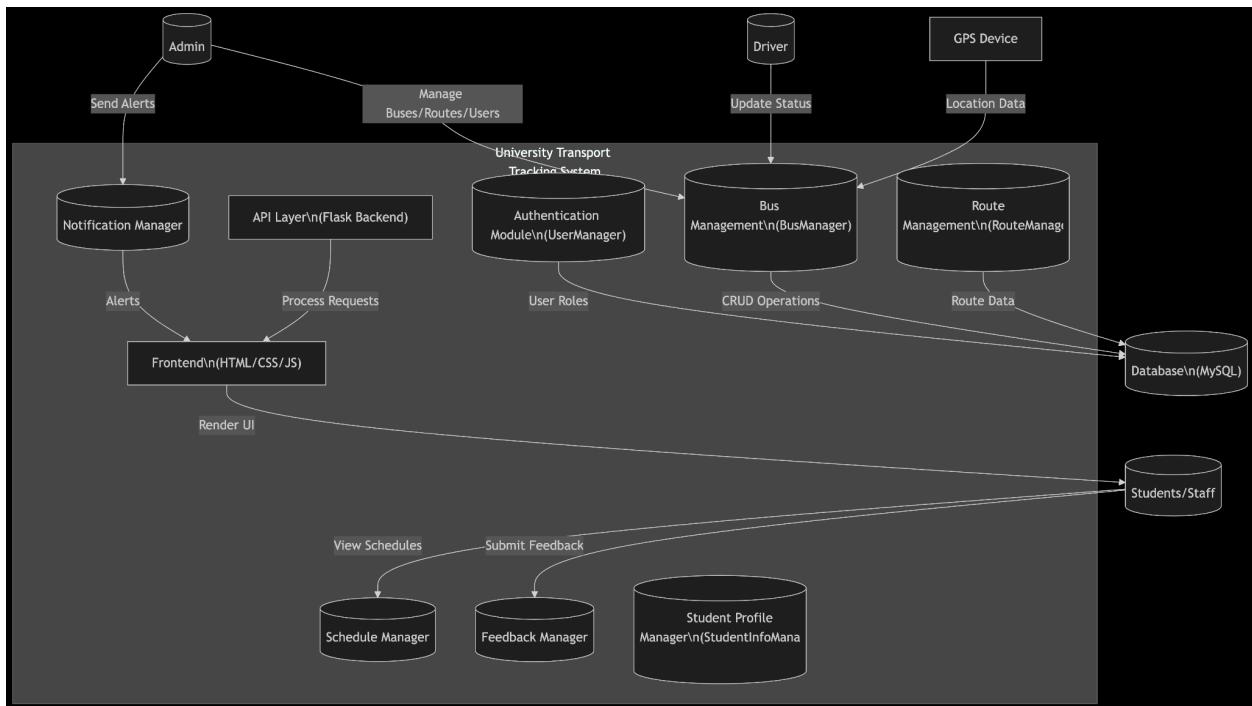
3. Student Profiles



Description: Flow chart diagram shows the step-by-step flow of user actions (e.g., login → view bus status).

3.3 Data Flow Diagram (DFD)

This Data Flow Diagram (DFD) section in the Software Design Document (SDD) illustrates how data moves through the system, highlighting processes, data stores, external entities, and data flows. It provides a visual representation of system functionality at various levels of abstraction (e.g., Level 0 for context, Level 1+ for granular processes), ensuring alignment with business requirements. DFDs help identify inefficiencies, redundancies, or security gaps in data handling by mapping inputs, transformations, and outputs. This section bridges stakeholder expectations with technical design, clarifying interactions between users, systems, and databases. By standardizing data flow logic, the DFD ensures scalable and maintainable architecture.



Description: Depicts how data moves between processes (e.g., bus status updates → notifications).

Processes: Shows how data is processed (e.g., "Update Bus Status" or "Send Passenger Notifications").

Data Flows: Arrows depict movement (e.g., GPS coordinates → "Tracking System" → "Passenger App").

Data Stores: Databases/tables hold info (e.g., "Schedules DB" or "User Profiles").

External Entities: Sources/destinations (e.g., Drivers input delays; Passengers receive alerts).

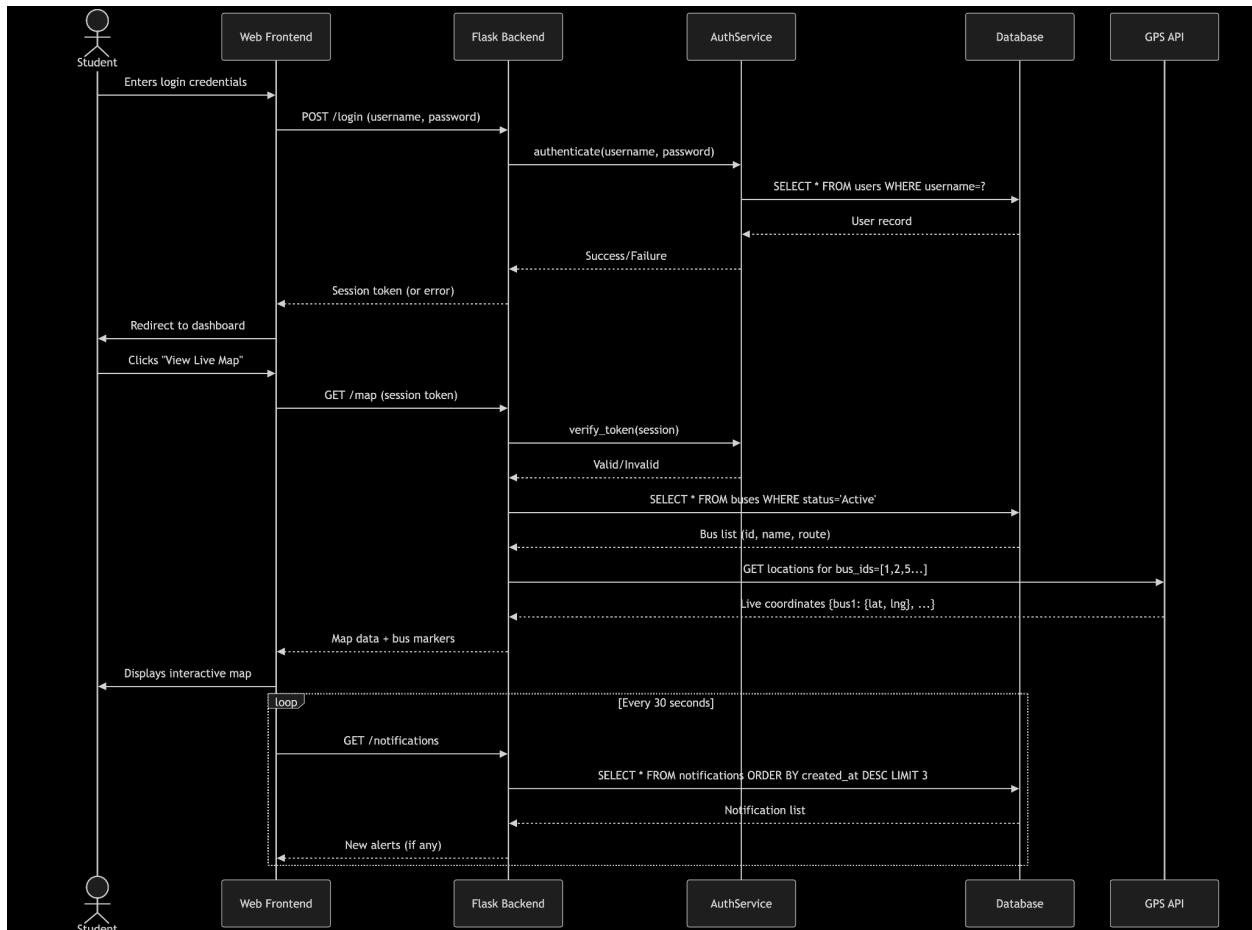
Levels: Level 0 (overview) to Level 2+ (detailed workflows, like fare calculation).

Example:

Driver → (Submits Delay) → "Trip Mgmt." → (Updates) → "Notifications" → Passenger

2. Sequence Diagrams

This Sequence Diagram section in the Software Design Document (SDD) captures dynamic interactions between system components or actors over time, using lifelines, messages, and activation bars to visualize the flow of operations. It clarifies the order of events, method calls, and responses in specific scenarios (e.g., user login, API calls), ensuring alignment with use cases and business logic. This diagram highlights synchronous/asynchronous communication, error handling, and timing constraints, aiding developers in understanding runtime behavior. By documenting these interactions, the SDD reduces ambiguity in implementation and facilitates efficient debugging. Sequence diagrams are especially critical for complex workflows or distributed systems, where timing and collaboration are paramount.



Student (Actor): Initiates the flow by requesting bus status on the UI (e.g., clicks "Refresh").

UI → Backend: Sends an API call (e.g., GET /bus-status?route=101) to the backend server.

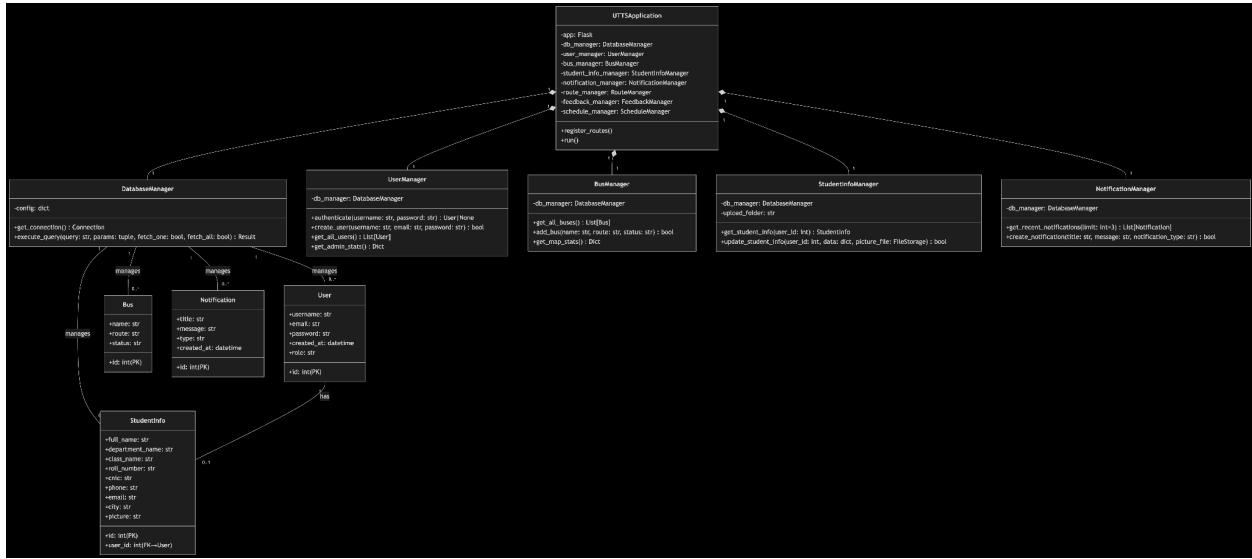
Backend → Database: Queries the database for real-time bus location and schedule data.

Database → Backend: Returns raw data (e.g., GPS coordinates, ETA) to the backend.

Backend → UI: Processes data (e.g., formats ETA) and updates the UI display for the student.

4.4 Class Diagram

This Class Diagram section in the Software Design Document (SDD) provides a structural overview of the system's object-oriented design, depicting classes, their attributes, methods, and relationships (e.g., inheritance, associations, dependencies). It serves as a blueprint for developers, clarifying responsibilities, hierarchies, and interactions between components to ensure modularity and maintainability. Key elements include visibility modifiers (public/private), multiplicity notations (1.. *), and interfaces to define contracts between classes. This diagram aligns with business logic and use cases, bridging requirements and implementation. By standardizing class structures, it reduces ambiguity and facilitates efficient coding, testing, and future scalability.



Key Classes:

User (Admin, Student, Driver).

Bus, Route, Notification.

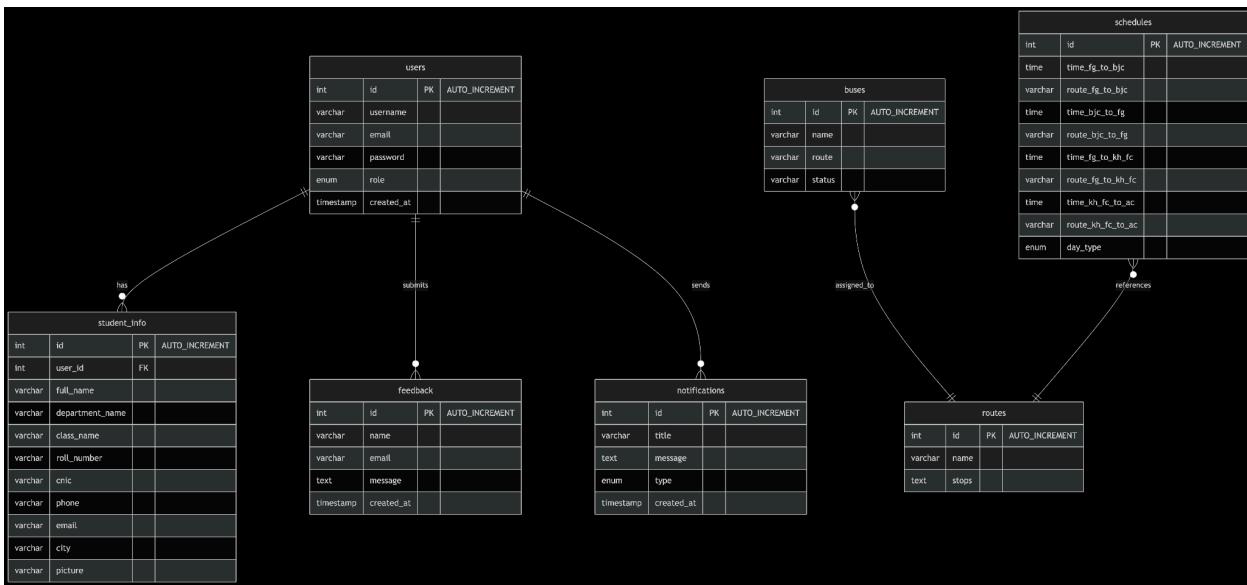
4.5 ER Diagram

Purpose: An ER diagram visually represents the data model, showing entities (tables), attributes (fields), and relationships (diamonds), and cardinality (one-to-one, one-to-many, many-to-many).

Components: Includes entities (rectangles), attributes (ovals), relationships (diamonds), and cardinality (one-to-one, one-to-many, many-to-many).

SDD Role: Helps developers and stakeholders understand database structure before implementation.

Clarity: Simplifies complex data relationships, ensuring efficient database design.



Entities: users, buses, routes, feedback.

5. Database Design

The database design section provides a detailed blueprint of the system's data architecture, including tables, fields, primary/foreign keys, and relationships (illustrated via ER diagrams). It specifies normalization techniques (e.g., 3NF) to minimize redundancy, indexing strategies for performance optimization, and security measures like encryption or access controls. The SDD also documents stored procedures, triggers, and data validation rules to enforce business logic at the database level. This section ensures alignment between the application layer and database schema, serving as a critical reference for developers, testers, and maintainers to ensure scalability, integrity, and efficiency throughout the system's life cycle.

5.1 Schema Description

Buses:	id, name, route, status
Users:	id, username, email, password, role (student/teacher/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_bj, day_type).

5.2 Sample Queries

-- Get delayed buses

```
SELECT * FROM buses WHERE status = 'Delayed';
```

6. User Interface Design

The User Interface (UI) Design section of the Software Design Document (SDD) outlines the structure, behavior, and visual presentation of the system's interfaces to ensure usability and consistency. It includes:

Layout & Navigation – Wireframes or mockups illustrating screen layouts, menus, buttons, and flow between screens.

UI Components – Detailed descriptions of reusable elements (forms, tables, modals) with states (active, disabled, hover).

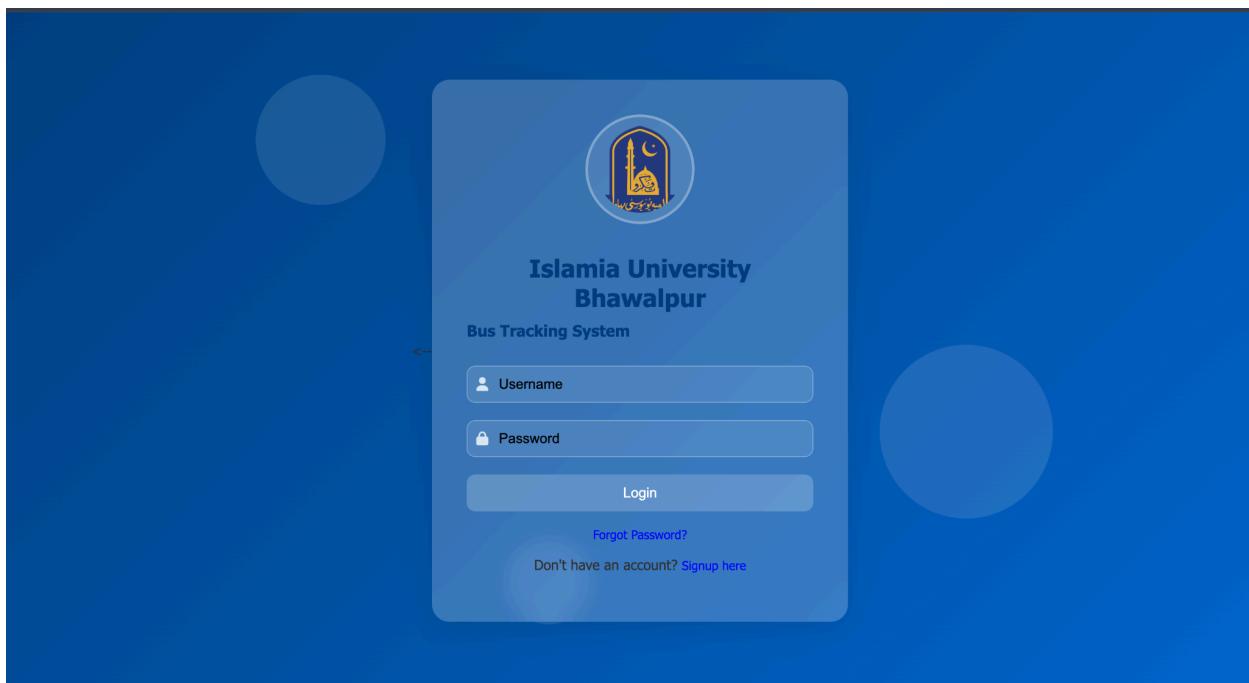
User Interactions – Expected behaviors for clicks, inputs, validations, error handling, and feedback mechanisms (e.g., toast notifications).

Responsiveness & Accessibility – Adaptability across devices (desktop, mobile) and compliance with standards (WCAG) for users with disabilities.

Style Guide – Typography, color schemes, icons, and branding aligned with corporate identity (referencing a separate style guide if available).

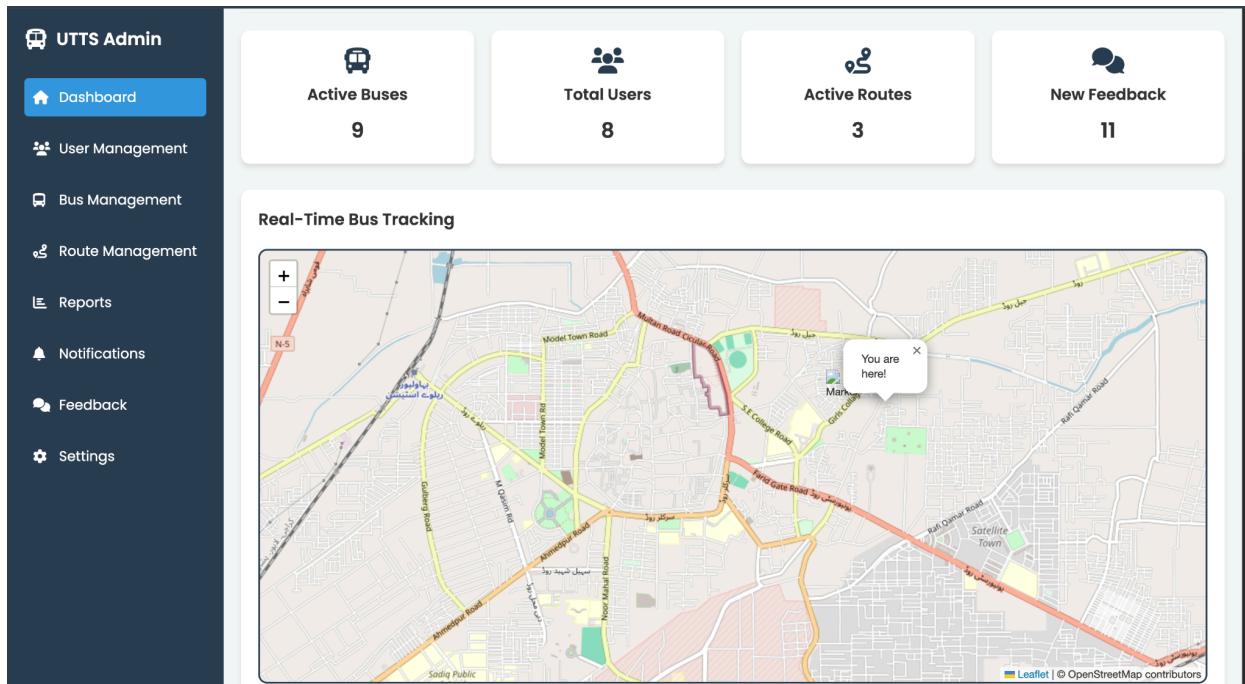
This section ensures developers and designers adhere to a unified vision, reducing ambiguity during implementation.

1-Login Page



This wireframe shows the login page of the system.

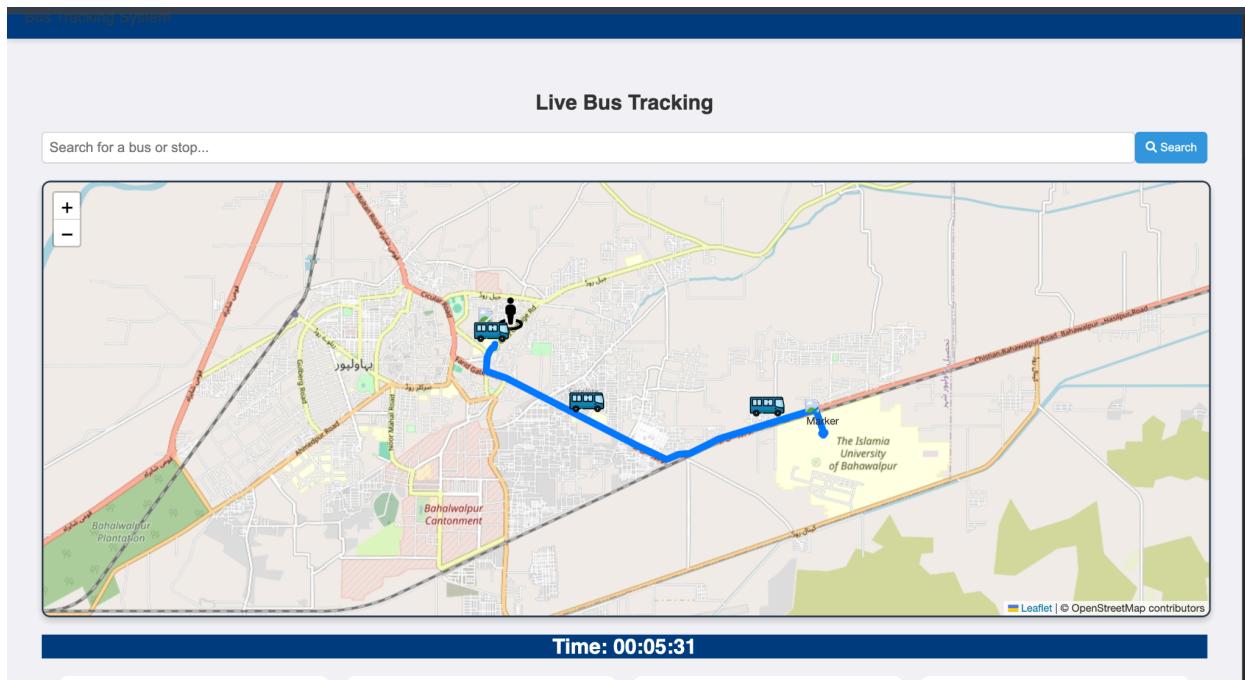
2-Admin Dashboard



The wireframe of the Admin Dashboard for UTTS Admin. It features a dark sidebar on the left with a navigation menu containing links for Dashboard, User Management, Bus Management, Route Management, Reports, Notifications, Feedback, and Settings. The main area has four cards: Active Buses (9), Total Users (8), Active Routes (3), and New Feedback (11). Below these is a map titled "Real-Time Bus Tracking" showing a city layout with roads and landmarks like Model Town Road, Multan Road, and S.E. College Road. A callout bubble says "You are here!" with a marker icon. The map is powered by Leaflet and OpenStreetMap contributors.

This wireframe shows the admin portal for system.

3-Live Bus Tracking Map



The wireframe of the Live Bus Tracking System. It has a header bar with the title "Bus Tracking System". The main content area is titled "Live Bus Tracking" and includes a search bar with placeholder text "Search for a bus or stop...". Below the search bar is a map showing bus routes. One bus route is highlighted in blue, showing its path from a starting point to a destination marked "Marker". The map also shows landmarks like "The Islamia University of Bahawalpur" and "Bahawalpur Cantonment". A timestamp at the bottom of the map indicates "Time: 00:05:31". The map is powered by Leaflet and OpenStreetMap contributors.

This wireframe shows the live map of the routes where student can track buses easily

3-Student Dashboard



Islamia University Bhawalpur
Bus Tracking System

Dashboard Live Map Bus Schedule Logout

Welcome, SHAHZAD

Search for buses or routes... 🔍

Weather in Bhawalpur

Temperature: 30.43°C
Condition: overcast clouds

Active Buses

- Bus-1-Abasia to bhagdad-Active
- Bus-2-Abasia to bhagdad-Active
- Bus-3-Abasia to bhagdad-Active

Quick Links

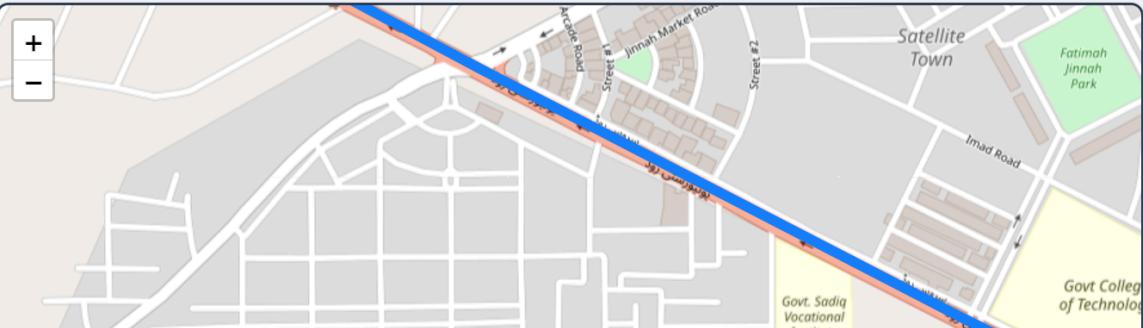
- Live Bus Tracking
- Bus Schedule
- Contact Support

Notifications

Warning Date 11:58

Title: hello frineds
Today bus-1 will be dealys 10 minute.

Live Bus Tracking Map



This wireframe shows the student dashboard where they can see basic information about the system.

7. Non-Functional Requirements

7.1 Performance

- 1-Response time < 2 sec for 100+ users.
- 2-Data encrypted in transit (HTTPS recommended).
- 3-MySQL backups daily.

7.2 Security

- 1-Data encryption (HTTPS).
- 2-Role-based access control.

8. Future Enhancements

- 1-GPS Integration (IoT devices).
- 2-Mobile App (Android/iOS).

9. Conclusion

This SDD provides a blueprint for UTTS development, ensuring alignment with SRS requirements.

Appendices

Appendix A: Full SQL Schema

Appendix B: Revision History

=====