

Namal University Transport Management System (NUTMS)

System Design Report – Milestone 3



- | | | |
|---|---------------|------------------|
| 1 | Altaf Hussain | NUM-BSCS-2024-11 |
| 2 | Rabia Ashraf | NUM-BSCS-2024-65 |
| 3 | Najia Nayab | NUM-BSCS-2024-62 |

Department of Computer Science
Namal University Mianwali

18th January, 2026

Contents

1	Introduction	3
1.1	Purpose of this Document	3
1.2	Scope	3
2	Design Assumptions and Constraints	3
2.1	Design Assumptions	3
2.1.1	Technical Assumptions	3
2.1.2	Operational Assumptions	4
2.1.3	Business Assumptions	4
2.2	Design Constraints	4
2.2.1	Technology Constraints	4
2.2.2	Performance Constraints	5
2.2.3	Security Constraints	5
2.2.4	Regulatory and Policy Constraints	6
3	Key Design Decisions	7
3.1	Breaking Down the System (DFD Decomposition)	7
3.1.1	Level 0 DFD	7
3.1.2	Level 1 DFD Processes	7
3.1.3	Level 2 DFD for Route & Schedule Management	7
3.1.4	Level 2 DFD for Authentication	8
3.1.5	Level 2 DFD for GPS Tracking	8
3.2	Class Diagram Design	8
3.2.1	User Inheritance	8
3.2.2	Key Relationships	9
4	System Design Diagrams	10
4.1	Use Case Diagram	10
4.2	Data Flow Diagrams	11
4.2.1	Level 0 : Context Diagram	11
4.2.2	Level 1	11
4.2.3	Level 2	12
4.2.4	Level 2	12
4.2.5	Level 2	13
4.3	Sequence Diagram	13
4.4	Activity Diagrams	14
4.4.1	Registration Activity	14
4.4.2	Login Activity	16
4.4.3	Notification Activity	17
4.4.4	Vehicle Booking Activity	18
4.4.5	Bus Tracking Activity	19

4.4.6	Admin Activity	20
4.5	Class Diagram	21
4.6	Component Diagram	22
5	Requirements–Design Traceability Table	23
6	Appendix	28
6.1	GitHub Repository	28
6.2	Meeting Minutes Reference	28
6.3	Figma Link	28

1 Introduction

1.1 Purpose of this Document

This Design Report presents the complete system design for the Namal University Transport Management System (NUTMS). It translates the Software Requirements Specification (SRS) into detailed design artifacts including UML diagrams, prototypes, and architectural decisions. This document serves as a bridge between the requirements and the actual implementation of the system.

1.2 Scope

This document covers the following areas:

- Complete UML diagrams representing system behavior and structure
- Design decisions and their justifications based on SRS requirements
- Traceability mapping from requirements to design elements
- Prototypes validated with stakeholders through meetings
- Design assumptions and constraints
- Component architecture and system organization

2 Design Assumptions and Constraints

2.1 Design Assumptions

2.1.1 Technical Assumptions

1. **GPS Availability:** All university buses have functional GPS devices capable of transmitting real-time location data every 10-30 seconds with accuracy within 10-20 meters.
2. **Network Connectivity:** Stable internet connectivity is available on campus and in most areas covered by bus routes. Campus WiFi infrastructure supports system access for students and staff.
3. **User Device Access:** Students, faculty, and staff have access to smartphones, tablets, or computers with modern web browsers (Chrome, Firefox, or Edge released within the last 3 years).
4. **Database Capacity:** PostgreSQL database can handle concurrent bookings and tracking data for multiple buses and users during peak hours without performance issues.

5. **Email Service:** University SMTP server or external email service is reliable and available for notification delivery with minimal delay.
6. **Map Services:** OpenStreetMap data is accurate and available for Mianwali region and university campus areas.

2.1.2 Operational Assumptions

1. **User Roles:** Users are pre-categorized as Student, Faculty, Driver, or Administrator based on university registration and email domain.
2. **Route Stability:** Bus routes and schedules remain relatively stable within a semester, with changes occurring primarily at semester boundaries or due to special circumstances.
3. **Peak Usage:** Maximum concurrent users during peak hours (morning and evening) is estimated based on total student and faculty population.
4. **GPS Accuracy:** GPS coordinates are accurate within 10-20 meters, sufficient for tracking bus progress along routes.
5. **Session Duration:** User sessions automatically timeout after 30 minutes of inactivity for security purposes.

2.1.3 Business Assumptions

1. **Approval Workflow:** Special vehicle booking requests from faculty follow a defined approval hierarchy with higher authority consent required.
2. **Capacity Limits:** Each bus has a fixed seating capacity that must be strictly enforced to prevent overcrowding and maintain safety.
3. **Cancellation Policy:** Users can cancel bookings up to a certain time before departure, freeing seats for other users.
4. **Academic Calendar:** System operates according to university academic calendar, with potential downtime during semester breaks.
5. **Record Retention:** Booking and trip records are maintained for a minimum of one academic year for reporting and audit purposes.

2.2 Design Constraints

2.2.1 Technology Constraints

1. **Platform:** Web-based application only, accessible through web browsers on various devices.

2. **Frontend Technology:** React.js framework for building responsive and interactive user interfaces.
3. **Backend Technology:** Node.js runtime with Express.js framework for server-side logic and API handling.
4. **Database:** PostgreSQL relational database management system for all data storage.
5. **Browser Support:** Must support Google Chrome, Mozilla Firefox, and Microsoft Edge (modern versions only).
6. **Storage Restriction:** No use of browser localStorage or sessionStorage due to artifact environment limitations.
7. **Real-time Communication:** WebSocket protocol (Socket.io) for real-time GPS updates and notifications.

2.2.2 Performance Constraints

1. **Page Load Time:** All pages must load within 3 seconds under normal network conditions.
2. **Booking Transaction:** Booking confirmation must complete within 2 seconds.
3. **GPS Update Frequency:** Real-time bus location updates every 10-30 seconds.
4. **Concurrent Users:** System must support multiple concurrent users, especially during peak booking periods.
5. **System Availability:** 99% uptime during academic semesters.

2.2.3 Security Constraints

1. **Authentication:** Secure user authentication with hashed password storage.
2. **Communication:** All data transmission over HTTPS with TLS encryption.
3. **Access Control:** Role-based access control (RBAC) to restrict functions based on user roles.
4. **Data Privacy:** Users can only view their own booking history and personal information.
5. **Input Validation:** Protection against SQL injection, XSS, and other common web vulnerabilities.

2.2.4 Regulatory and Policy Constraints

1. **University Policies:** Adherence to university transport policies regarding passenger limits and route allocations.
2. **Safety Regulations:** Strict enforcement of vehicle capacity limits as per safety regulations.
3. **Data Protection:** Compliance with university IT policies and data protection standards.

3 Key Design Decisions

This section summarizes the main design decisions made for the Namal University Transport Management System (NUTMS).

3.1 Breaking Down the System (DFD Decomposition)

3.1.1 Level 0 DFD

Decision: A Level 0 DFD (Context Diagram) was created to represent the entire system as a single process interacting with external entities.

Justification:

- Provides a high-level overview of the system
- Clearly shows system boundaries
- Easy for non-technical stakeholders to understand

3.1.2 Level 1 DFD Processes

Decision: The system was divided into five main processes:

- Authentication
- View Routes & Schedules
- Tracking
- GPS Tracking
- Notification Management

Justification:

- Each process performs a single clear function
- Matches the main features in the SRS
- Easy for users and developers to understand

3.1.3 Level 2 DFD for Route & Schedule Management

Decision: A Level 2 DFD was created for Route & Schedule Management, divided into:

- Maintain Routes & Stops
- Maintain Schedules
- Save Schedule

Justification:

- Route and schedule handling is more complex
- Clearly shows admin workflow
- Routes must be created before schedules

3.1.4 Level 2 DFD for Authentication

Decision: A Level 2 DFD was created for the Authentication process, decomposed into:

- Enter Credentials
- Verify User ID and Role
- Return Login Status

Justification:

- Authentication is a critical security function
- Clearly shows validation and decision flow
- Differentiates between successful and failed login outcomes

3.1.5 Level 2 DFD for GPS Tracking

Decision: A Level 2 DFD was created for GPS Tracking, divided into:

- Manage Location Data
- Calculate Progress and ETA
- Update Map Display

Justification:

- GPS tracking involves continuous data processing
- Improves clarity of real-time tracking workflow
- Separates data handling from user-facing map updates

3.2 Class Diagram Design

3.2.1 User Inheritance

Decision: User was designed as a parent class for Student, Faculty, Driver, and Administrator.

Justification:

- Common attributes are shared
- Supports role-based access

3.2.2 Key Relationships

- **Bus–Schedule (Composition):** Schedule depends on Bus
- **Driver–Bus (Association):** Temporary assignments
- **Student/Faculty–Schedule (Association):** View-only access

Decision: Same names and structures were used across all diagrams.

Justification:

- Improves traceability
- Ensures alignment between design and implementation

4 System Design Diagrams

4.1 Use Case Diagram

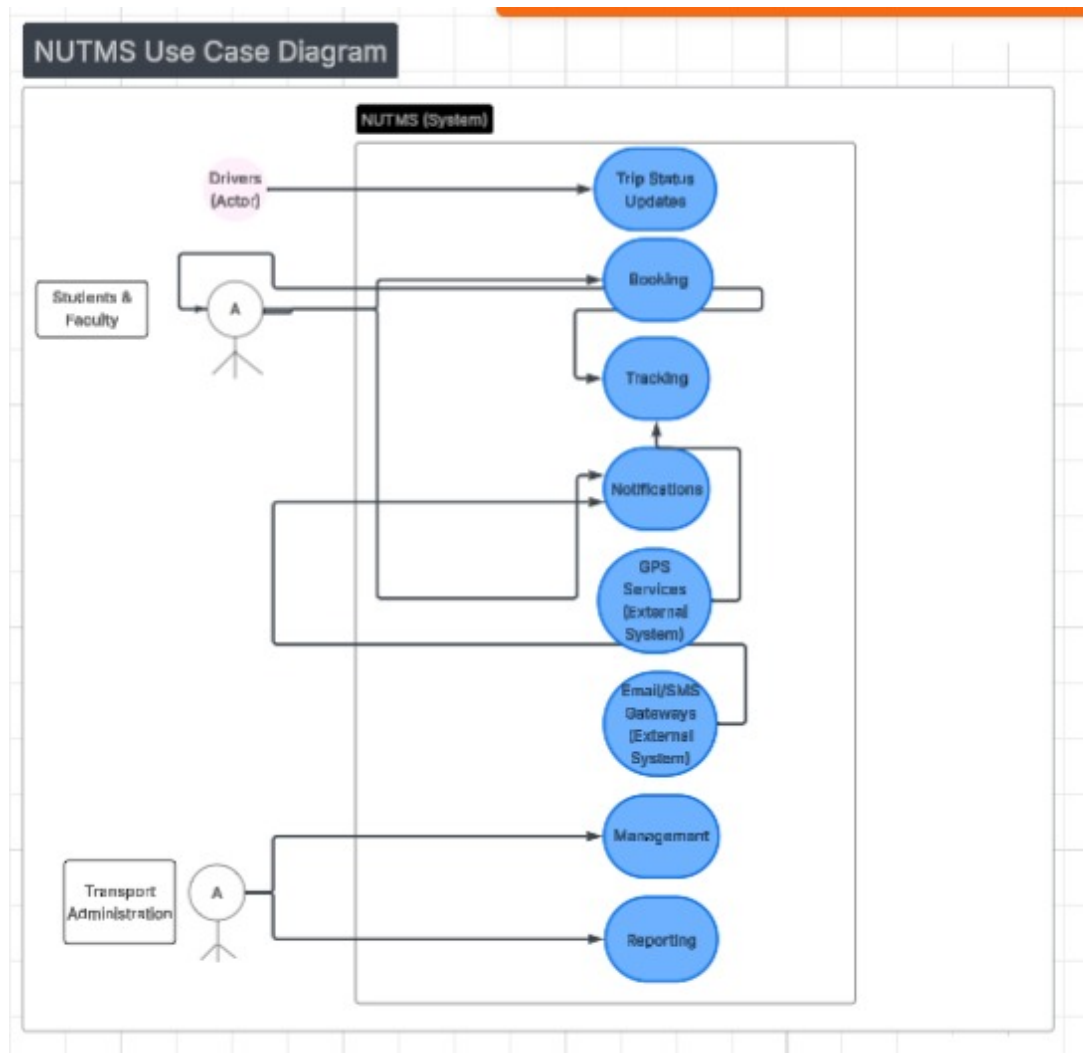


Figure 1: Use Case Diagram showing NUTMS actors and their interactions

4.2 Data Flow Diagrams

4.2.1 Level 0 : Context Diagram

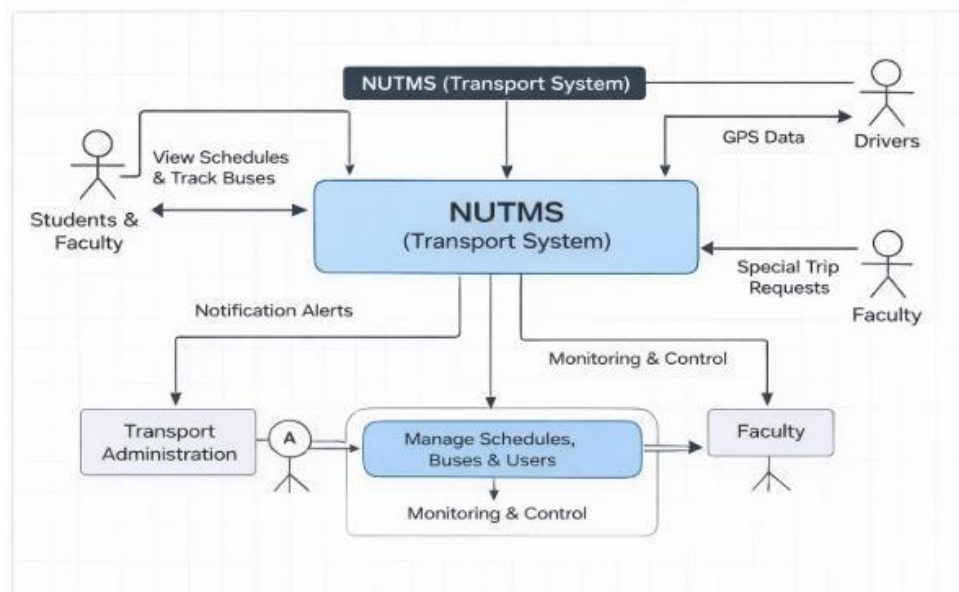


Figure 2: NUTMS Context Diagram - System interactions with external actors

4.2.2 Level 1

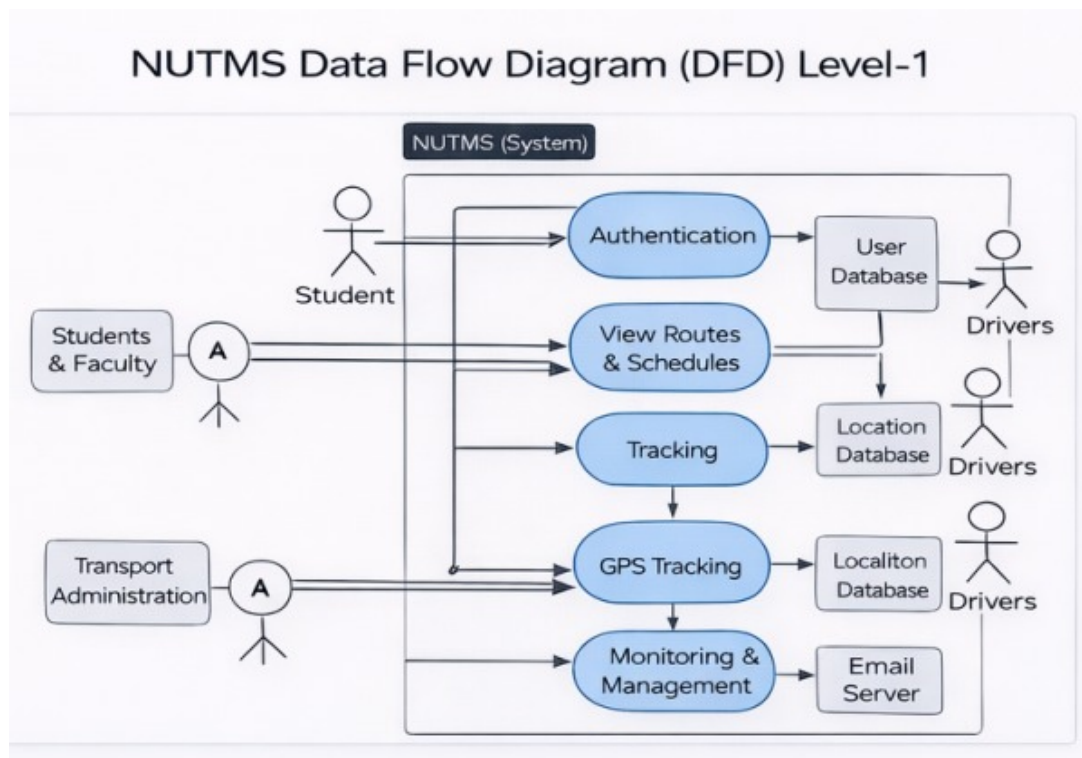


Figure 3: Level 1 DFD

4.2.3 Level 2

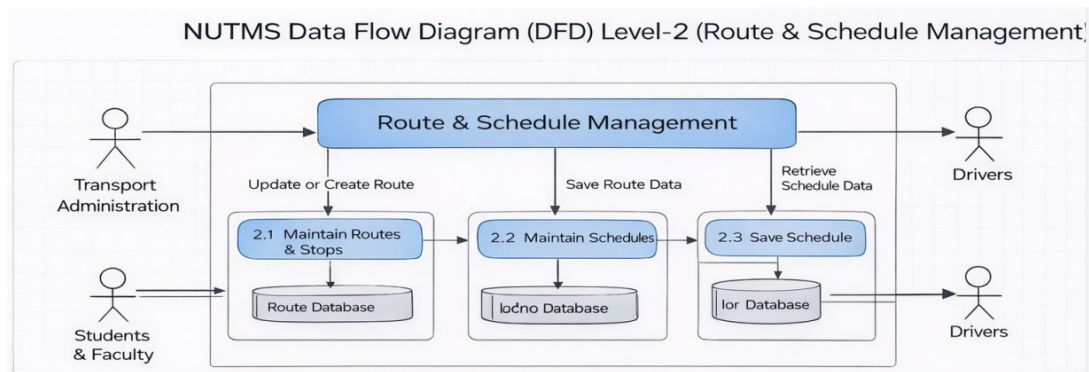


Figure 4: Level 2 DFD

4.2.4 Level 2

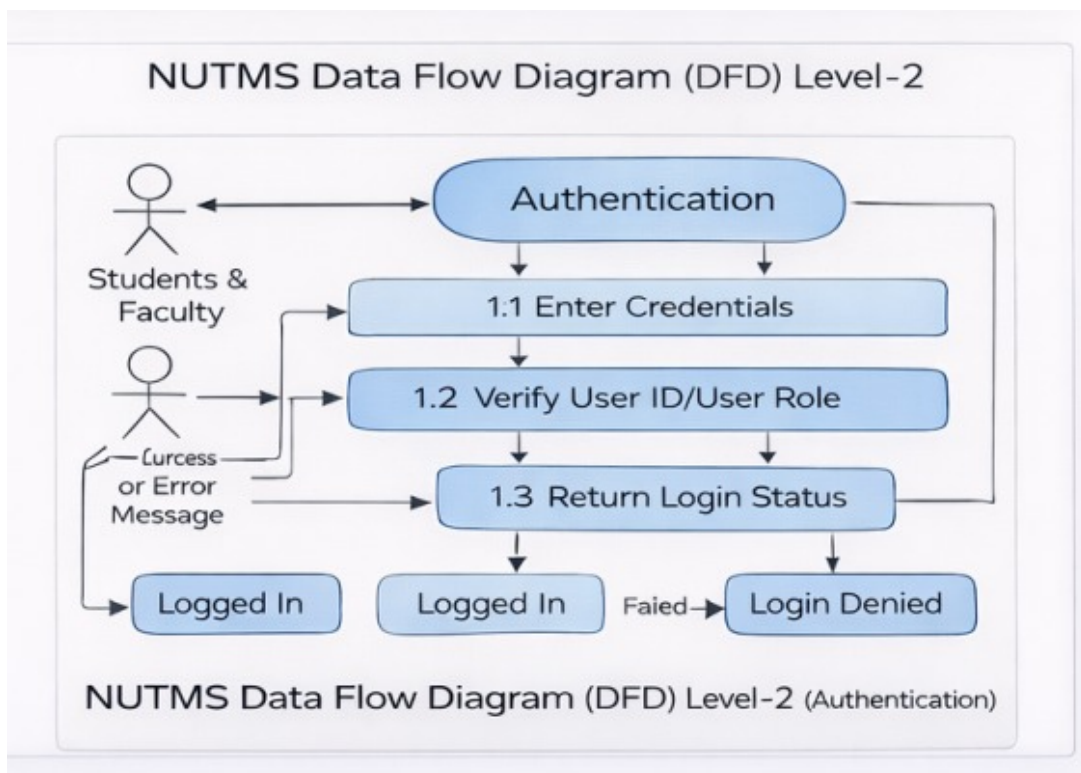


Figure 5: Level 2 DFD

4.2.5 Level 2

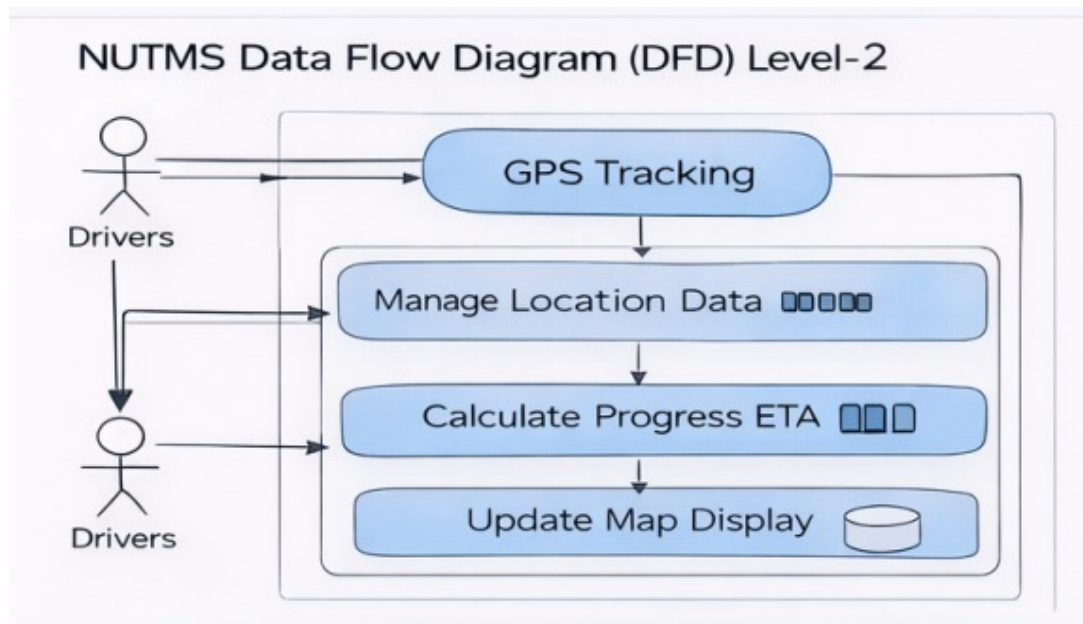


Figure 6: Level 2 DFD

4.3 Sequence Diagram

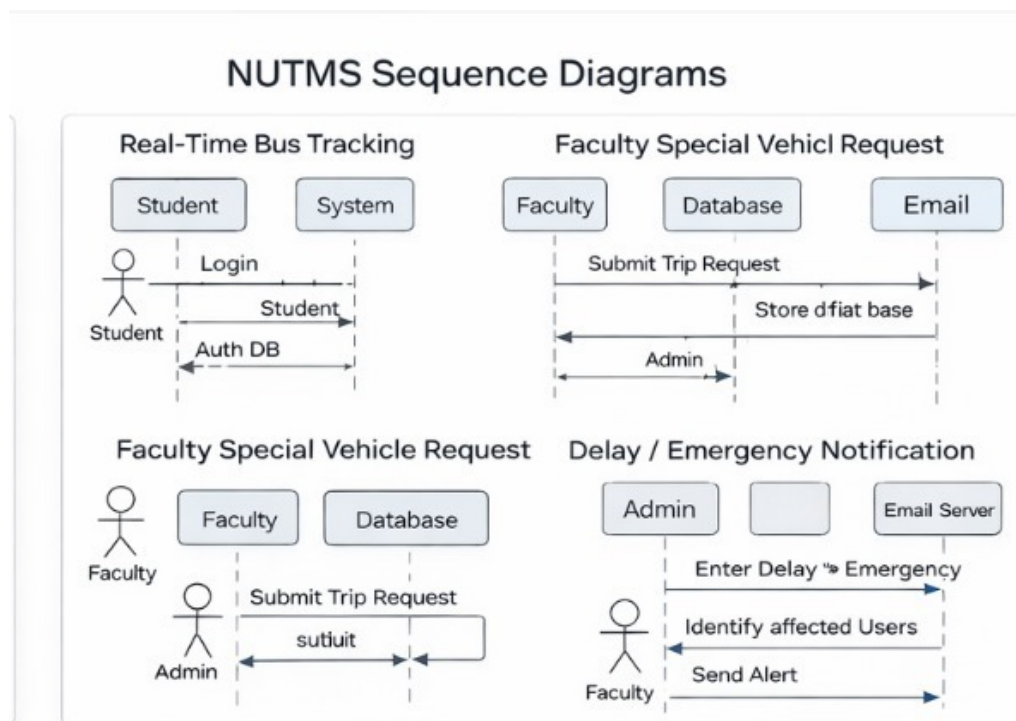


Figure 7: Sequence diagram

4.4 Activity Diagrams

4.4.1 Registration Activity

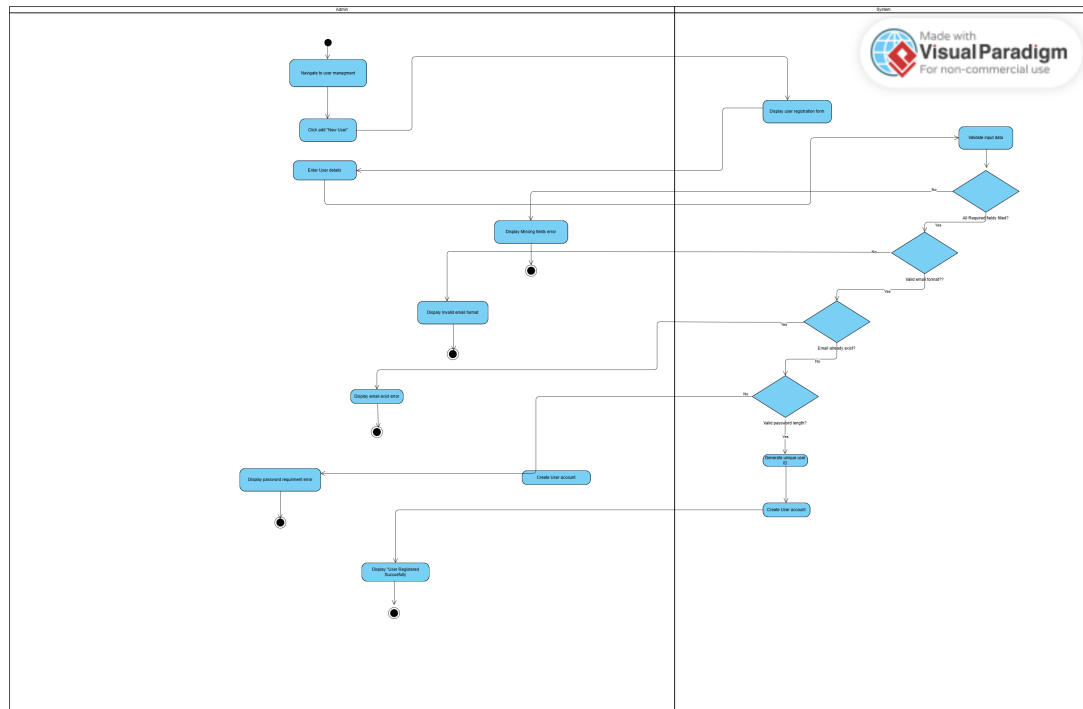


Figure 8: Activity diagram for user registration process

4.4.2 Login Activity

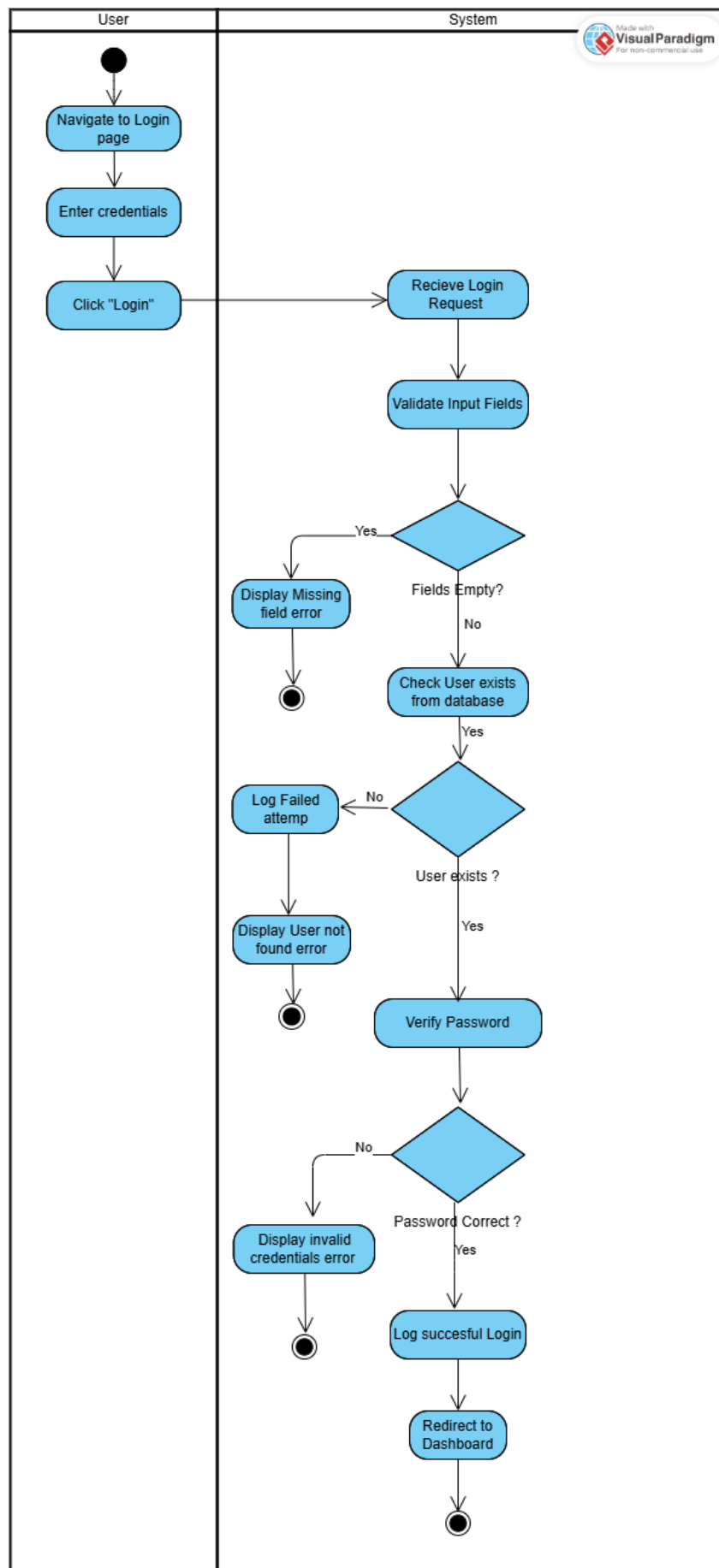


Figure 9: Activity diagram for user login process

4.4.3 Notification Activity

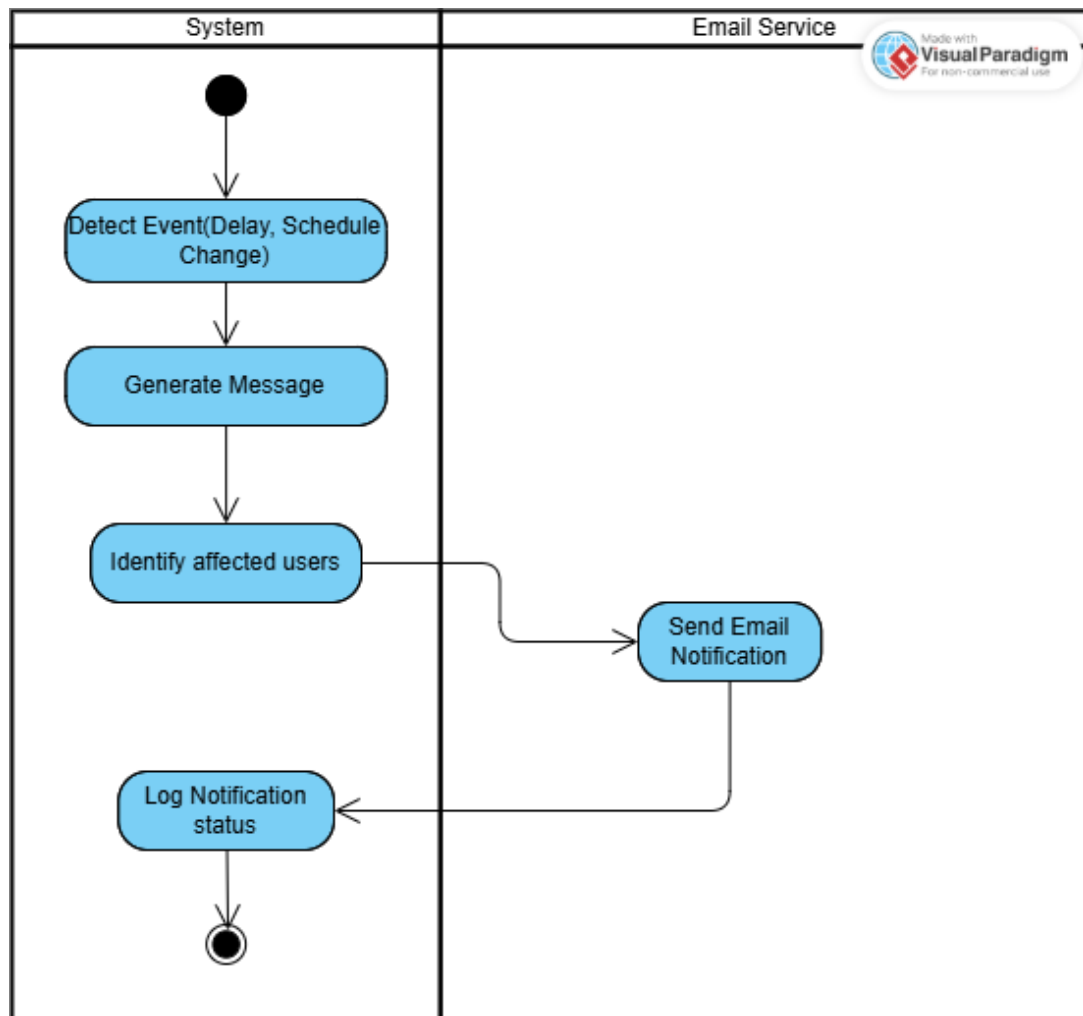


Figure 10: Activity diagram for notification management

4.4.4 Vehicle Booking Activity

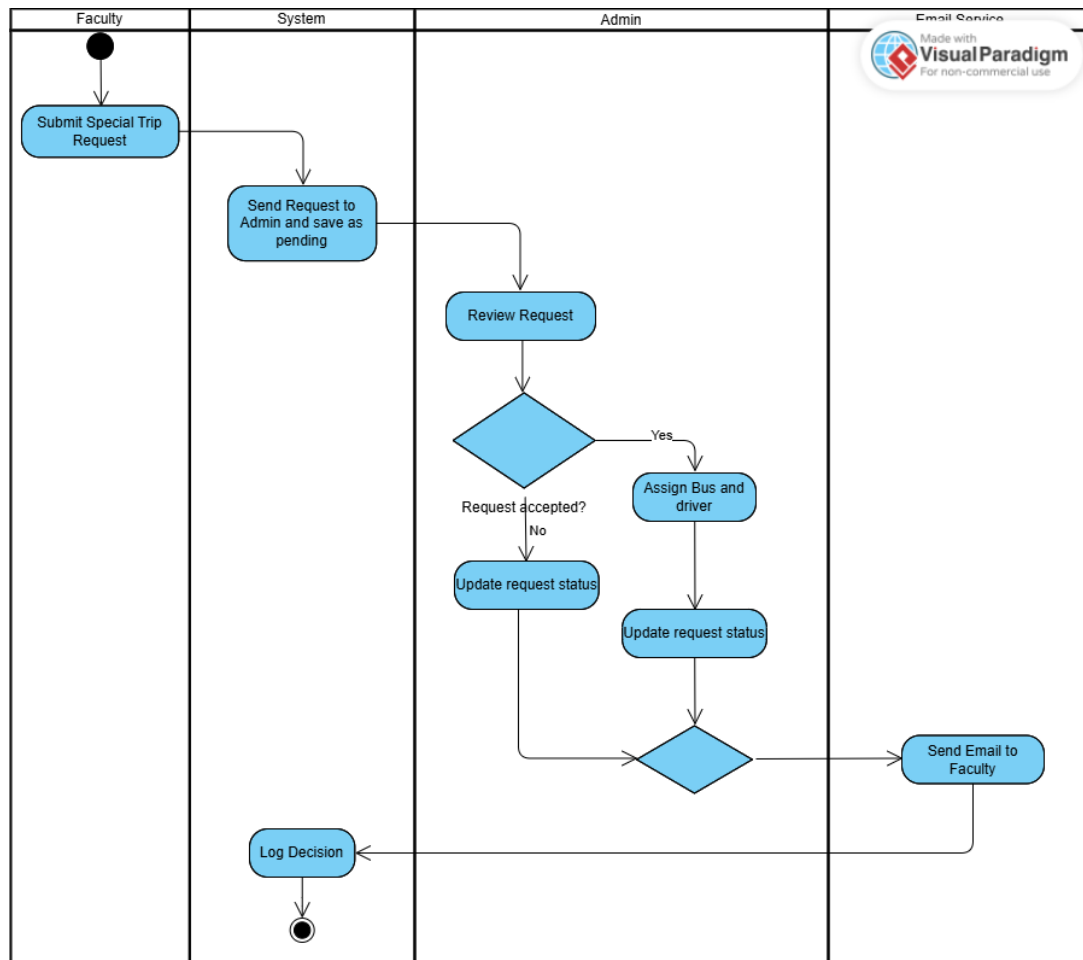


Figure 11: Activity diagram for vehicle booking process

4.4.5 Bus Tracking Activity

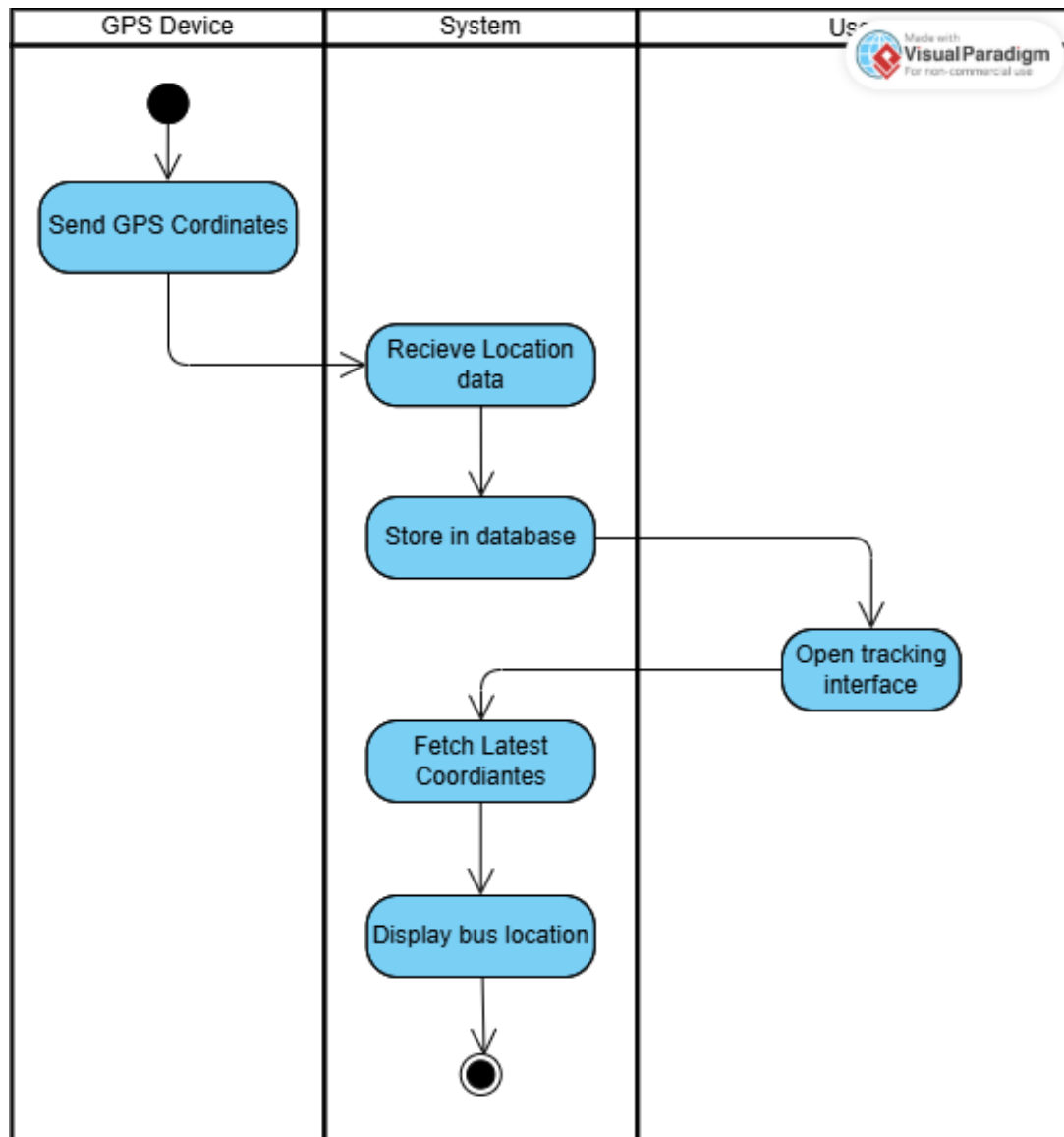


Figure 12: Activity diagram for real-time bus tracking process

4.4.6 Admin Activity

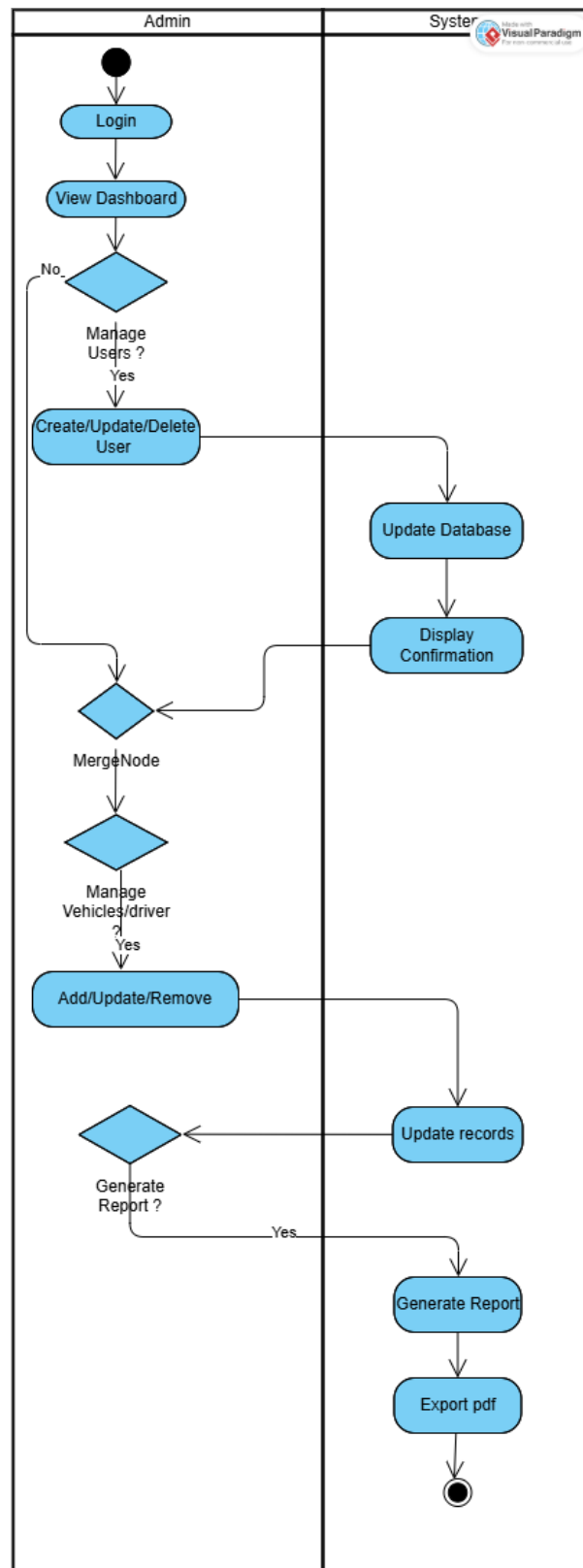


Figure 13: Activity diagram for administrator operations

4.5 Class Diagram

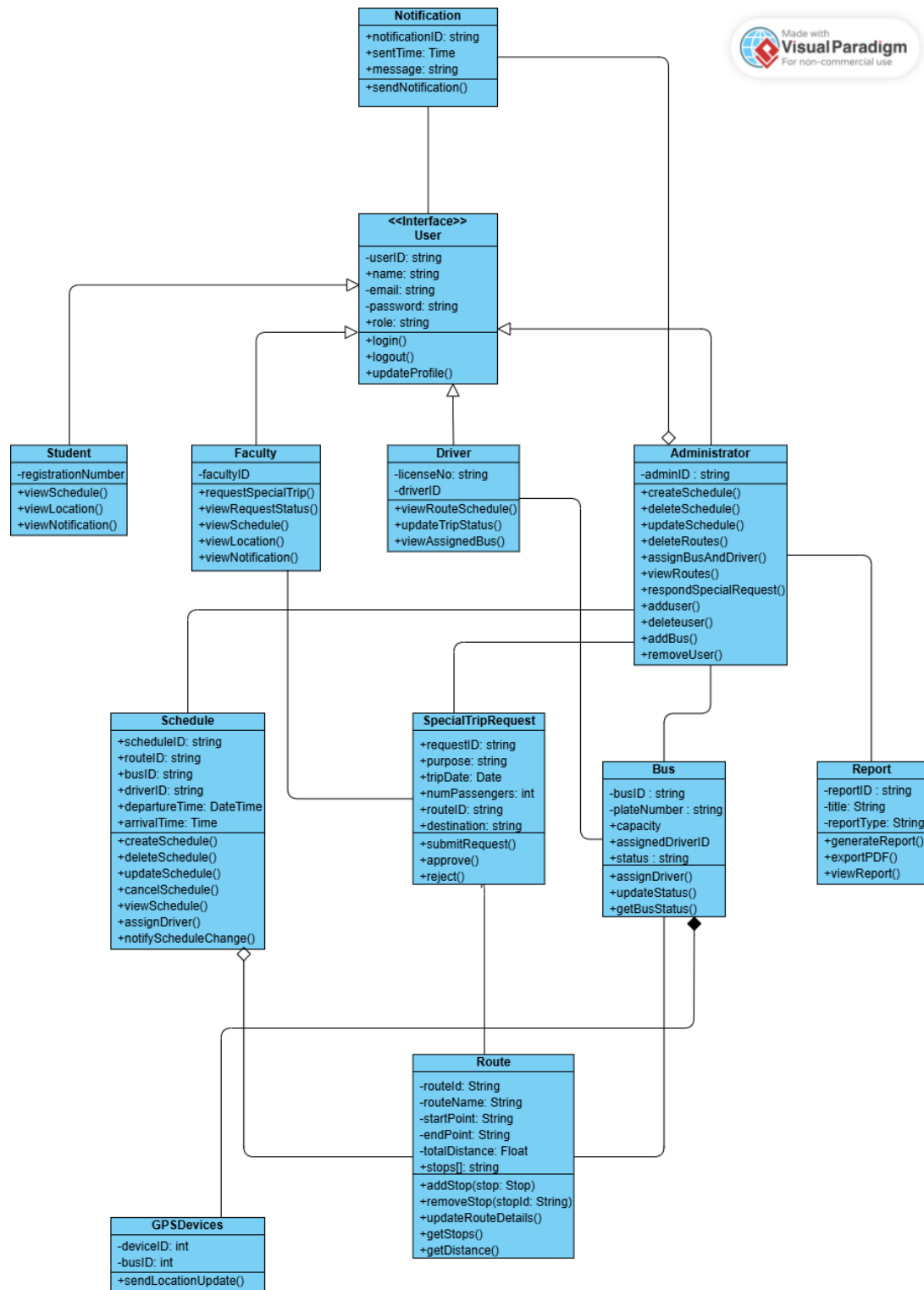


Figure 14: Class Diagram showing all system classes and relationships

4.6 Component Diagram

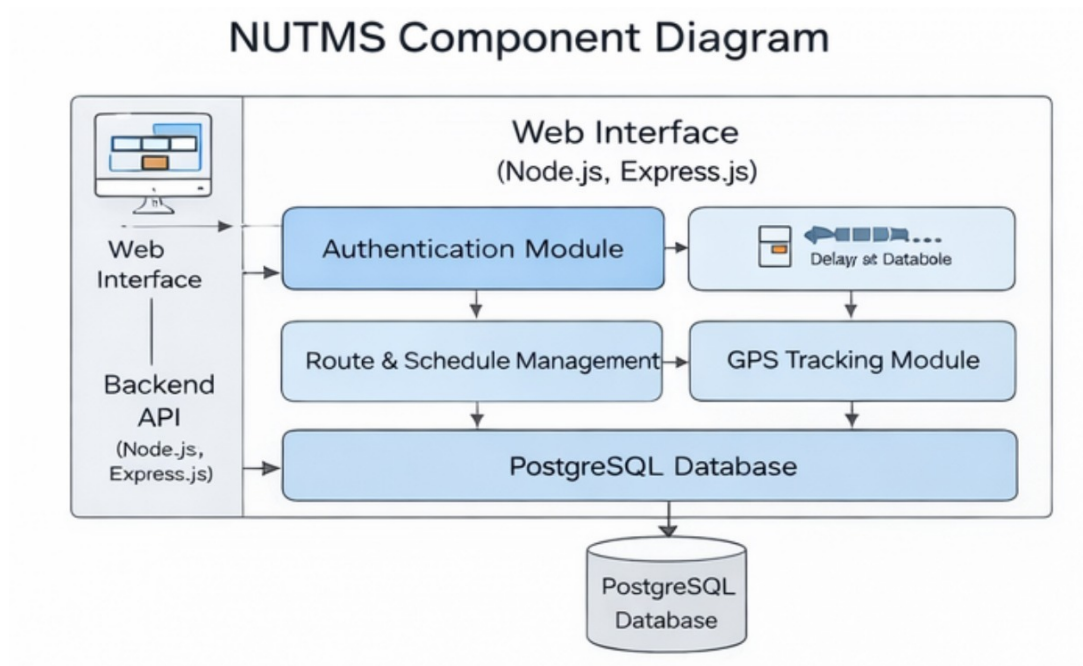


Figure 15: Component diagram showing modular decomposition of NUTMS

5 Requirements–Design Traceability Table

This traceability table demonstrates how each functional requirement from the SRS is addressed in the system design through various design artifacts.

Req ID	Requirement	Design Artifacts
FR-1.1	User Registration	<ul style="list-style-type: none"> - Use Case: User Registration - Context Diagram (Level 0) - DFD Level 1: Authentication Module - Sequence Diagram: User Registration Flow - Activity Diagram: Registration Activity - Classes: User, Student, Faculty, Driver, Administrator - Components: Authentication Module - Prototype: Registration Form, User Profile Setup
FR-1.2	User Authentication	<ul style="list-style-type: none"> - Use Case: User Login - Context Diagram (Level 0) - DFD Level 1: Authentication Module - Sequence Diagram: Login Sequence - Activity Diagram: Login Activity - Classes: User (login() method) - Components: Authentication Module - Prototype: Login Screen, Dashboard Access
FR-1.3	Profile Management	<ul style="list-style-type: none"> - Use Case: Manage Profile - DFD Level 1: Authentication Module - Sequence Diagram: Profile Update Flow - Activity Diagram: User Profile Management - Classes: User (updateProfile() method) - Components: Authentication Module - Prototype: User Profile Screen, Edit Profile Interface
FR-2.1	Route Definition	<ul style="list-style-type: none"> - Use Case: Manage Routes - Context Diagram (Level 0) - DFD Level 2: Maintain Routes & Stops - Sequence Diagram: Route Creation Sequence - Activity Diagram: Route Management Activity - Classes: Route (addStop(), removeStop(), updateRouteDetails()) - Components: Route & Schedule Management - Prototype: Admin Dashboard, Route Management Interface

Req ID	Requirement	Design Artifacts
FR-2.2	Schedule Management	<ul style="list-style-type: none"> - Use Case: Manage Schedules - Context Diagram (Level 0) - DFD Level 2: Maintain Schedules - Sequence Diagram: Schedule Creation/Update Flow - Activity Diagram: Schedule Management Activity - Classes: Schedule (createSchedule(), updateSchedule(), deleteSchedule()) - Components: Route & Schedule Management - Prototype: Schedule Configuration Screen, Calendar View
FR-2.3	Route Assignment	<ul style="list-style-type: none"> - Use Case: Assign Resources - DFD Level 2: Save Schedule - Sequence Diagram: Driver/Bus Assignment Sequence - Activity Diagram: Assignment Workflow - Classes: Schedule (assignDriver()), Bus (assignDriver()) - Components: Route & Schedule Management - Prototype: Driver Assignment Interface, Bus Allocation Screen
FR-3.1	GPS Integration	<ul style="list-style-type: none"> - Use Case: GPS Tracking - Context Diagram (Level 0) - DFD Level 1: GPS Tracking Module - Sequence Diagram: Real-Time Bus Tracking Sequence - Activity Diagram: GPS Data Reception Activity, Bus Tracking Activity - Classes: GPSDevices (sendLocationUpdate()) - Components: GPS Tracking Module - Prototype: Live Tracking Map

Req ID	Requirement	Design Artifacts
FR-3.2	Live Location Display	<ul style="list-style-type: none"> - Use Case: Track Buses - DFD Level 1: Tracking Module - Sequence Diagram: Real-Time Bus Tracking Sequence - Activity Diagram: Bus Tracking Activity, Bus Location Display - Classes: Route, GPSDevices - Components: GPS Tracking Module - Prototype: Interactive Map Screen, Bus Position Viewer
FR-3.3	Route Progress Tracking	<ul style="list-style-type: none"> - Use Case: Track Buses - DFD Level 1: Tracking Module - Sequence Diagram: Real-Time Bus Tracking Sequence - Activity Diagram: Bus Tracking Activity - Classes: Route (getStops(), getDistance()) - Components: GPS Tracking Module - Prototype: Progress Indicator, ETA Display
FR-4.1	Booking Confirmation	<ul style="list-style-type: none"> - Use Case: Notification System - Context Diagram (Level 0) - DFD Level 1: Notification Management Module - Sequence Diagram: Booking Confirmation Flow - Activity Diagram: Notification Activity - Classes: Notification (sendNotification()) - Components: Notification Module - Prototype: Booking Confirmation Email Template
FR-4.2	Delay Notifications	<ul style="list-style-type: none"> - Use Case: Notification System - DFD Level 1: Notification Management Module - Sequence Diagram: Delay/Emergency Notification Sequence - Activity Diagram: Notification Activity, Delay Alert Activity - Classes: Notification (sendNotification()) - Components: Notification Module - Prototype: Delay Alert Email, Push Notification

Req ID	Requirement	Design Artifacts
FR-4.3	Schedule Change Alerts	<ul style="list-style-type: none"> - Use Case: Notification System - DFD Level 1: Notification Management Module - Sequence Diagram: Schedule Change Notification Flow - Activity Diagram: Notification Activity - Classes: Schedule (notifyScheduleChange()), Notification - Components: Notification Module - Prototype: Schedule Update Email Template
FR-4.4	Emergency Notifications	<ul style="list-style-type: none"> - Use Case: Notification System - DFD Level 1: Notification Management Module - Sequence Diagram: Delay/Emergency Notification Sequence - Activity Diagram: Notification Activity, Emergency Broadcast - Classes: Notification, Administrator (sendEmergencyAlert()) - Components: Notification Module - Prototype: Emergency Alert Interface, Broadcast Message Screen
FR-5.1	Dashboard Overview	<ul style="list-style-type: none"> - Use Case: Administrative Dashboard - Sequence Diagram: Admin Dashboard Access - Activity Diagram: Admin Activity, Dashboard Loading Activity - Classes: Administrator (viewDashboard()) - Components: Administrative Module - Prototype: Admin Dashboard Homepage
FR-5.2	User Management	<ul style="list-style-type: none"> - Use Case: User Management - Context Diagram (Level 0) - DFD Level 1: Authentication Module (Create/Update/Delete User) - Sequence Diagram: User Management Flow - Activity Diagram: Admin Activity, User CRUD Activity - Classes: Administrator (addUser(), deleteUser(), updateUser()) - Components: Administrative Module - Prototype: User Management Panel, Role Assignment Screen

Req ID	Requirement	Design Artifacts
FR-5.3	Driver and Vehicle Management	<ul style="list-style-type: none"> - Use Case: Driver/Vehicle Management - DFD Level 1: Add/Update/Remove (Vehicles/Drivers) - Sequence Diagram: Driver Assignment Sequence - Activity Diagram: Admin Activity, Driver/Vehicle CRUD Activity - Classes: Bus (assignDriver(), updateStatus()), Driver (viewSchedule()) - Components: Administrative Module - Prototype: Driver Management Screen, Vehicle Registry
FR-5.4	Report Generation	<ul style="list-style-type: none"> - Use Case: Reporting - DFD Level 1: Generate Report Process - Sequence Diagram: Report Generation Flow - Activity Diagram: Admin Activity, Report Creation Activity - Classes: Report (generateReport(), exportPDF(), viewReport()) - Components: Administrative Module - Prototype: Reports Dashboard, Export Options
FR-6.1	Special Trip Request	<ul style="list-style-type: none"> - Use Case: Special Vehicle Booking - Context Diagram (Level 0) - DFD Level 1: Request Submission Flow - Sequence Diagram: Faculty Special Vehicle Request Sequence - Activity Diagram: Vehicle Booking Activity, Special Trip Request - Classes: SpecialTripRequest (submitRequest()), Faculty (requestSpecialTrip()) - Components: Booking Management Module - Prototype: Special Trip Request Form

Req ID	Requirement	Design Artifacts
FR-6.2	Request Approval Workflow	<ul style="list-style-type: none"> - Use Case: Approve Special Requests - DFD Level 1: Approval Workflow - Sequence Diagram: Faculty Special Vehicle Request Sequence - Activity Diagram: Vehicle Booking Activity, Request Approval - Classes: SpecialTripRequest (approve(), reject()), Administrator (respondSpecialRequest()) - Components: Booking Management Module - Prototype: Request Review Screen, Approval Interface

6 Appendix

6.1 GitHub Repository

The complete project repository is available at following link :

<https://github.com/Altaf-Hussain-11/SE-Project-Transport-Management-System-.git>

6.2 Meeting Minutes Reference

You can access the meeting minutes document from the following link:

<https://docs.google.com/spreadsheets/d/180hqUDHGpHp8CMqTwAZxo9n5CBqNpvm2eWgDv0x2>

6.3 Figma Link

<https://www.figma.com/make/BSx3HRTdplYAIGmDAfMgEJ/Namal-University-Transport-Management-System?t=CSUEbjsCcbeJFhXb-1>