

# **FINAL YEAR PROJECT**

---



University of Narowal

## **UTMS-UNIVERSITY TRANSPORT MANAGEMENT SYSTEM**

Ahmad Sultan

21-UON-0906

M. Adeel

21-UON-0923

Zeenat Arooj

21-UON-0948

**BS COMPUTER SCIENCE (SESSION 2021-2025)**

**SUPERVISED BY: MR. ZAEEM NAZIR**

**FACULTY OF COMPUTING AND INFORMATION  
TECHNOLOGY  
DEPARTMENT OF COMPUTER SCIENCE  
UNIVERSITY OF NAROWAL**

**Spring 2025**

**Shakargarh Road, Narowal, Punjab, Pakistan**

## STATEMENT OF SUBMISSION

This is certify that **Ahmad Sultan** Registration No. (21-UON-0906), **M. Adeel** Registration No. (21-UON-0923) and **Zeenat Arooj** Registration No. (201-UON-0948) successfully completed the final year project named as: **UTMS- University Transport Management System**, at the Department of Computer Science, University of Narowal, to fulfill the requirement of the degree of **BS in Computer Science**.

\_\_\_\_\_  
Project Supervisor

\_\_\_\_\_  
Officer In-Charge

\_\_\_\_\_  
External Examiner

## ACKNOWLEDGEMENT

With the blessings of Almighty Allah and the prayers of our parents, we have made this humble attempt to achieve the goal that was set for us at the beginning of this project. Although it was not an easy task, the research work embodied in this dissertation was accomplished under the able guidance and affectionate supervision of Mr. Zaeem Nazir, Associate Lecturer, University of Narowal, Narowal.

We are grateful for the support, guidance and cooperation of Mr. Zaeem Nazir without whom this project and research report would not have been possible. His continuous support and guidance helped us to work into and complete it. We are honored to acknowledge his support.

Date: July 15, 2025

## ABSTRACT

UTMS: A Campus University Transport Management System is a modern web-based platform aimed at increasing the efficiency, transparency, and accuracy of managing university transportation services. UTMS consists of two separate web interfaces—one for administrative staff and another for student users—creating a clear division of roles and responsibilities within the system. The transport administrator, often referred to as the super admin, manages the entire transport ecosystem. This includes the creation and supervision of treasurer admins and department-level admins in a hierarchical administrative structure. The super admin oversees crucial operations such as the generation and maintenance of online transport challans, transport card issuance, and the end-to-end management of buses, drivers, routes, schedules, and broadcast alerts. Treasurer admins are responsible for verifying submitted challans and managing fee-related tasks. Department admins, on the other hand, are tasked with validating and enrolling students into the transport system. On the student side, UTMS offers a dedicated user-friendly interface that allows students to register, upload challans, track their verification status, and receive their digital transport cards once approved. The entire process is optimized to reduce administrative workload, eliminate manual errors, and provide students with faster access to essential transport facilities.

## Table of Content

<b>STATEMENT OF SUBMISSION .....</b>	<b>1</b>
<b>ACKNOWLEDGEMENT .....</b>	<b>2</b>
<b>ABSTRACT .....</b>	<b>3</b>
<b>Chapter 1: Introduction to UTMS : University Transport Management System .....</b>	<b>7</b>
1.1 Introduction .....	8
1.2. Project Title .....	8
1.2.1. University Transport Management System (UTMS) .....	8
1.2.2. A Dual-Portal Web Application .....	8
1.2.3. Design, Development, and Deployment .....	8
1.3. Project Overview Statement .....	8
1.3.1. Admin Interface .....	9
1.3.2. Student/User Interface .....	9
1.4. Project Goals and Objectives: .....	11
1.5. High Level System Components .....	11
1.5.1. Administrative Interface: .....	11
1.5.2. Student User Interface: .....	11
1.5.3. Main features: .....	11
1.5.4. Connectivity and Integration: .....	12
1.6. Functional units list .....	12
1.6.1. Functional Requirements of UTMS .....	12
1.6.2. User Management: .....	12
1.6.3. Bus and Driver Management: .....	12
1.6.4. Schedule and Route Management: .....	12
1.6.5. Challan Management: .....	13
1.6.6. Communication and Notifications: .....	13
1.6.7. Reporting and Analytics: .....	13
1.6.8. Security and Access Control: .....	13
1.7. Website Architecture .....	13
1.7.1. System Architecture of UTMS (MERN Stack) .....	13
1.7.2. Key Functionalities: .....	14
1.8. Gantt Chart .....	15
1.9. Hardware and Software Specifications .....	16
1.10. Tools and Technologies used with Reasoning .....	16
<b>Chapter 2: Project Planning and Management.....</b>	<b>17</b>
<b>2.1 Introduction .....</b>	<b>18</b>
2.2. Project/Product Feasibility Report.....	18
2.2.1. Technical Feasibility .....	18
2.2.2. Operational Feasibility .....	19
2.2.3. Economic Feasibility .....	19

2.2.4. Legal and Ethical Feasibility.....	19
2.2.5. Schedule Feasibility.....	19
2.3. Project Plan .....	20
2.4. Introduction to Team Member and Their Skills.....	22
2.5. Risk List.....	23
<b>Chapter 3: UTMS System Architecture and Workflow .....</b>	<b>24</b>
3.1 Introduction .....	25
3.2 Website Features and User Experience .....	25
3.2.1. Signup Process .....	25
3.2.2. User Login and Access .....	26
3.2.3. Transport Fee Management .....	26
3.2.4. Route Schedule Viewing.....	26
3.2.5. Notification Management .....	26
3.2.6. Profile Management.....	26
3.2.7. Email Notifications .....	26
3.2.8. Administrative Functions.....	26
3.2.9. Security and Data Protection .....	27
3.2.10. Responsive Design.....	27
3.2.11. User-Friendly Interface.....	27
3.2.12. Feedback and support .....	27
3.3 Class Diagram.....	28
3.4 Entity Relation Diagram (ER) .....	29
3.5. User Flow chart .....	30
3.6. Verification flowchart.....	31
<b>Chapter 4: Software Requirement Analysis.....</b>	<b>32</b>
4.1 Introduction .....	33
4.1.1. User Interface.....	33
4.2 Specific or Functional Needs .....	34
4.3 Non-Operational Conditions: .....	35
4.3.1. Interface of Hardware .....	35
4.3.2. User Interface.....	36
4.3.3. Performance Criteria .....	36
4.3.4. Design Limitations.....	36
4.3.5. Security .....	36
4.3.6. Dependable .....	36
4.3.7. Reachable.....	36
4.3.8. Dependability.....	36
4.3.9. Mobility.....	37
<b>Chapter 5: Testing.....</b>	<b>38</b>
5.1 Introduction .....	39
5.2. Login Test.....	40
5.2.1. Login Successful .....	40
5.2.2. Wrong Credentials while Login .....	40

5.2.3.	Empty Fields while Login .....	41
5.2	Add User.....	42
5.2.4.	Add user with Valid Information.....	42
5.2.1	Add User with Invalid Information .....	42
5.3	Online Challan Genneration.....	43
5.3.1	Update Challan Data(Admin Site) .....	43
5.4	Notifications .....	44
5.4.	Update Schedule (Admin Site) .....	44
<b>Chapter 6:</b>	<b>Website Interface.....</b>	<b>46</b>
6.1.	Admin Login.....	47
6.2.	Manage User Request .....	48
6.3.	E-mail Notifications.....	49
<b>References</b>	<b>.....</b>	<b>53</b>

## **Chapter 1: Introduction to UTMS : University Transport Management System**

---



## 1.1 Introduction

The efficient operation of a university's transportation system plays a critical role in ensuring timely, safe, and reliable commute services for students and staff. Traditional methods of managing university transport often suffer from delays, miscommunication, and administrative inefficiencies — including manual challan processing, poor route coordination, and lack of centralized tracking for buses and drivers.

To address these challenges, the University Transport Management System (UTMS) has been developed as a modern, web-based solution tailored specifically for academic institutions. The system is divided into two main interfaces: one for students (users) and the other for administrators (transport office staff). This dual-portal design enhances functionality, security, and user experience.

The student portal enables users to register for transport services, upload challans, receive approval notifications, and view assigned bus schedules. Meanwhile, the admin panel empowers authorized personnel to manage key transport operations such as route creation, bus and driver assignment, transport challan generation and verification, and schedule updates — all through a centralized and secure dashboard.

By digitizing and streamlining core transport operations, this system reduces manual workload, minimizes errors, and improves service delivery. Furthermore, it promotes transparency, ensures timely communication, and enhances student satisfaction by providing a responsive and accessible online platform for all transportation needs.

Ultimately, this project modernizes the university's transport infrastructure and sets a foundation for scalable, efficient, and data-driven transport management across academic campuses.

## 1.2. Project Title

The following are the main topics this topic covers:

### 1.2.1. University Transport Management System (UTMS)

This is the title of the platform being developed, highlighting its purpose of efficiently managing university transportation through a centralized, web-based application

### 1.2.2. A Dual-Portal Web Application

The system features two distinct interfaces: one for students to manage their transport registrations and challans, and another for administrators to oversee operations such as route planning, bus and driver management, and challan verification.

### 1.2.3. Design, Development, and Deployment

This project involves the complete lifecycle of system creation: from requirement analysis and interface design to full-stack web development and implementation within a university environment for real-time use.

## 1.3. Project Overview Statement

The **University Transport Management System (UTMS)** is developed to enhance the accuracy, transparency, and operational efficiency of managing transportation services within a university

environment. This system provides a centralized, web-based solution aimed at automating manual processes, reducing administrative workload, and improving service delivery to students.

The system consists of two separate portals:

### **1.3.1. Admin Interface**

- Provides full administrative control to manage and monitor all transport-related operations.
- Allows the admin to add, update, or remove buses, drivers, and transport routes.
- Enables generation, verification, and tracking of transport challans.
- Supports real-time scheduling, driver assignments, and transport card issuance.
- Offers role-based access control to maintain system integrity.

### **1.3.2. Student/User Interface**

- Allows students to register for transport services and manage their personal profiles.
- Provides the ability to upload and verify transport fee challans online.
- Enables students to receive email and web-based notifications regarding approvals, schedule updates, and announcements.
- Offers access to current bus routes and time schedules through a user-friendly dashboard.
- UTMS streamlines transport operations by reducing the dependence on paper-based systems and manual verification. It minimizes errors, enhances decision-making, and improves the overall experience for both students and transport administrators. This system is scalable and adaptable, making it suitable for implementation across other campuses or educational institutions seeking efficient transport management.

## Project Overview Statement

Project Title: UTMS: A University Transport Management System			
Project Manager: Ahmad Sultan			
Project Members:			
Name	Registration #	Email Address	Signature
Ahmad Sultan	21-UON-0906		
M. Adeel	21-UON-0923		
Zeenat Arooj	21-UON-0948		
<b>Project Goal:</b> To enhance the efficiency and accessibility of university transport services by implementing a centralized and automated web-based platform that streamlines both student and administrative transport-related operations.			
Objectives:			
Sr.#	Description		
1	Implement adv. algorithms for optimal route planning & resource allocation.		
2	Automate scheduling and dispatch processes to reduce administrative workload.		
3	Ensure timely deployment of buses and minimize waiting times for students.		
4	Improve overall efficiency of university transport operations.		
<b>Project Success Criteria:</b> The success of the project will be measured by the complete development and functional deployment of a dual-interface web application — one for the transport administration and one for student users — within a six-month timeframe.			
<b>Assumptions:</b> Users have basic digital skills and access to internet-enabled devices. The university will support deployment and maintenance.			
<b>Risks &amp; Obstacles:</b> Competition from existing transport solutions Data security and privacy concerns Resistance to adopting the new platform Performance and scalability issues as user demand grows			
Organization Address: University of Narowal, Department of Computer Science			
Type of project: Website Development and Deployment			
Target End Users: Students of university of Narowal who use university transportation.			
Development Technology: React js , Node js , Express js , MongoDB			
Platform: Visual Studio CODE			

Approved By: Mr. Zaeem Nazir / FYP Committee
Date: June 04, 2025

## 1.4. Project Goals and Objectives:

- Enhance efficiency in transport operations by automating driver assignments, bus scheduling, and route planning.
- Reduce administrative workload through digital challan processing and transport card issuance.
- Improve punctuality and reliability of transport services for students and staff.
- Project Objectives
- Develop a user-friendly admin panel to manage buses, drivers, routes, schedules, challans, and alerts.
- Automate challan generation, upload, and verification processes to minimize errors and delays.
- Provide students with a self-service portal for registration, challan upload, and transport card management.
- Ensure timely communication through schedule updates and notifications.
- Implement strong data privacy and access control measures.
- Use efficient route planning to optimize resource allocation and reduce operational costs.
- Deploy the system within the university and gather feedback for future scalability.

## 1.5. High Level System Components

The high-level system components of Swift Campus include:

### 1.5.1. Administrative Interface:

Manages overall transport operations, including buses, drivers, routes, schedules, notifications, and challan generation. Verifies and manages transport challans uploaded by students. Verifies and signs up university students for transport services.

### 1.5.2. Student User Interface:

- **Registration Module:** Allows students to sign up for transport services.
- **Challan Upload and Verification:** Enables students to upload and verify transport challans.
- **Transport Card Issuance:** After receiving clearance, transport cards are issued to students.

### 1.5.3. Main features:

- **Bus management:** Oversees the routes, scheduling, and fleet of college buses.
- **Driver control:** continues tabs on and coordinates schedules and assignments for drivers.
- Transport challans are generated, verified, and accepted electronically by means of the Challan control department.

#### 1.5.4. Connectivity and Integration:

- **Web-based Platform:** each administrative and student up to date can also access it using internet browsers.
- **Database management:** effectively up to date and maintains records on timetables, routes, buses, drivers, and students.
- **Security and Compliance:** To guard sensitive facts, ensures records security and privacy compliance with relevant requirements.

Collectively those components form a complete shipping management machine designed specially for college operations, with the purpose of enhancing user revel in and streamlining techniques.

## 1.6. Functional units list

### 1.6.1. Functional Requirements of UTMS

Campus Wheel's functional requirements encompass a comprehensive set of features designed to effectively manage university transportation services. The key functional necessities are as follows:

#### 1.6.2. User Management:

- **Student Registration:** Enable students to enroll for transportation services offered by the university.
- **Admin Roles:** Define and assign roles with appropriate permissions, such as department admin, finance admin, and transport admin.
- **User Authentication:** Implement secure login and authentication mechanisms for both administrators and students.

#### 1.6.3. Bus and Driver Management:

- **Bus Operations:** Manage bus routes, seating capacities, and schedules to ensure smooth operation and maintenance.
- **Driver Assignments:** Coordinate driver schedules and assign buses accordingly.
- **Driver Profiles:** Maintain updated driver profiles including essential details like certifications and licenses.

#### 1.6.4. Schedule and Route Management:

- **Route Planning:** Design and optimize bus routes based on efficiency and demand.

- **Timetable Management:** Create and maintain bus timetables for various routes and schedules.
- **Real-time Updates:** Provide students with live information regarding bus locations and any changes to the schedules.

#### 1.6.5. Challan Management:

- **Challan Generation:** Generate transportation challans for students.
- **Challan Verification:** Allow the finance admin to verify submitted challans efficiently.

#### 1.6.6. Communication and Notifications:

- **Notification System:** Notify students about delays, schedule updates, and other relevant transportation information.
- **Communication Channels:** Facilitate communication between students and administration to address queries and provide support.

#### 1.6.7. Reporting and Analytics:

- **Data Reporting:** Produce detailed reports on bus usage, challan statuses, and other important metrics for administrators.
- **Analytics:** Analyze collected data to optimize routes, schedules, and resource distribution.

#### 1.6.8. Security and Access Control:

- **Data Security:** Ensure the protection and integrity of student and operational data.
- **Access Control:** Define user permissions and access rights according to roles to maintain confidentiality and system security.

### 1.7. Website Architecture

#### 1.7.1. System Architecture of UTMS (MERN Stack)

##### 1.7.1.1. Overview

UTMS is developed using the MERN stack, providing a comprehensive university transport management system (UTMS) that enhances the efficiency and accuracy of managing transportation services on campus.

##### 1.7.1.2.Components:

- **Frontend (Nodejs.):**
  - **Student Interface:** Enables students to register, log in, and manage their transportation-related activities. Key features include challan submission, verification status tracking, transport card issuance, as well as viewing bus schedules and notifications.
- **Administrative Interface:**

- **Super Admin (Transport Admin):** Oversees the entire system operations, manages roles such as Finance Admin and Department Admin, and controls buses, drivers, routes, schedules, notifications, challan generation, and transport card issuance.
- **Finance Admin:** Responsible for verifying submitted challans and handling all financial transactions related to transportation services.

- **Department Admin:**

Manages student registrations for transport services and verifies student information.

- **Backend (Node.js, Express.js):**
  - **API Layer:** Facilitates business logic and communication between the frontend and the database. Responsible for user authentication, authorization, and session handling.
  - **Database (MongoDB):** Stores all essential data including users, buses, drivers, routes, schedules, challans, and transport cards. Ensures data persistence and scalability.

### 1.7.2. Key Functionalities:

- **Transport Operations Management:** Manages bus fleets, driver assignments, routes, and timetables, while delivering real-time updates and notifications to users.
- **Challan Management:** Handles generation, submission, and verification of transportation challans through an online platform.
- **Student Management:** Supports student registration, verification, and approval workflows, and issues transport cards to eligible students.

#### 1.7.2.1. Technology Stack:

- **Frontend:** React.js, HTML, CSS, Bootstrap
- **Backend:** Node.js, Express.js
- **Database:** MongoDB
- **Integration:**  
Ensures smooth communication and data exchange between frontend and backend via RESTful APIs, maintaining data consistency and security across the system.
- **Scalability:**  
Designed with future growth in mind, allowing easy addition of new features and support for increasing numbers of users and transportation assets

## 1.8. Gantt Chart

The Gantt chart of the project is given below in figure 1.

Task	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Idea hunting / Proposal Writing								
Requirement gathering /risk Analysis								
Data Collection and Design of software architecture								
Front End Development								
Back End Development								
Integration								
Prepare for the Presentation								
Documentation								

Figure 1 : Gantt Chart



## 1.9. Hardware and Software Specifications

### Frontend:

- Framework: React.js
- UI Framework: Bootstrap
- Languages: JavaScript (ES6+)

### Backend:

- Framework: Node.js with Express.js o Database: MongoDB (NoSQL database)
- Authentication: JSON Web Tokens (JWT) for user authentication and authorization
- API Documentation: Postman for API documentation and testing

### Database:

- Database Management System: MongoDB
- Deployment: MongoDB Compass
- Data Access: Mongoose ODM for interacting with MongoDB

### Development Tools:

- Code Editor: Visual Studio Code
- Version Control: Git and GitHub for version control and collaboration
- Package Manager: npm (Node Package Manager)
- Debugging: Chrome DevTools for frontend debugging, Node.js Inspector for backend debugging

## 1.10. Tools and Technologies used with Reasoning

- **Microsoft Visual Studio Code:** Used as the primary code editor for the UTMS web applications.
- **Postman:** Used for testing APIs for the both websites application.
- **Microsoft Word Professional 2016:** Used for documentation purposes.
- **GitHub:** Used for version control and source code management.

## **Chapter 2: Project Planning and Management**

---

## 2.1 Introduction

The initial phase of the UTMS– University Transport Management System (UTMS) involves comprehensive project planning, scheduling, and resource allocation. This includes selecting appropriate technologies, defining roles, assessing risks, and mapping tasks to team members. The core deliverables of this stage include:

- Overview of the Project
- Defined Project Scope
- Future Scope and Expansion Potential
- Team Members and Their Roles
- Task Breakdown and Assignment Table
- System Architecture
- Risk Identification and Management

## 2.2. Project/Product Feasibility Report

A complete feasibility analysis was conducted to determine whether the UTMS can be successfully developed and deployed. The system is determined to be technically, operationally, economically, and legally feasible. Following are types of feasibilities:

- Technical
- Operational
- Economic
- Schedule
- Legal and Ethical

### 2.2.1. Technical Feasibility

- **Technology Stack:** The system uses the MERN stack (MongoDB, Express.js, React.js, and Node.js), which is robust, scalable, and ideal for full-stack JavaScript development. It supports efficient real-time updates and secure data handling, ideal for transport management needs.
- **Developer Expertise:** Resources skilled in JavaScript and the MERN stack are widely available. The development team can efficiently build and maintain the system using modern tools and best practices.
- **Deployment Infrastructure:** Cloud-based platforms like AWS, Render, or MongoDB Atlas provide scalable and cost-effective hosting and database solutions, ensuring uninterrupted service and easy scaling.
- **System Integration:** RESTful APIs will enable smooth communication between frontend, backend, and any existing campus systems like student portals or attendance platforms.
- **Tool Support:** Version control via Git/GitHub, testing frameworks like Jest, and CI/CD tools ensure fast and reliable development and deployment cycles.

### 2.2.2. Operational Feasibility

- **User Training:** Easy-to-follow guides and intuitive interfaces minimize training time for both administrators and students. Training sessions can be provided for initial onboarding.
- **System Maintenance:** A support plan will be established, including regular updates and bug fixes to ensure the system remains efficient and user-friendly.
- **Adoption and Transition:** To ensure smooth transition from manual processes, change management strategies will include phased rollouts, stakeholder feedback, and pilot testing.
- **User Acceptance:** The platform is specifically designed to address the pain points of current university transport systems, improving efficiency, reducing delays, and enhancing user satisfaction

### 2.2.3. Economic Feasibility

The financial evaluation confirms that UTMS is both cost-effective and sustainable over the long term.

- **Development Costs:** Leveraging open-source technologies significantly reduces initial development expenses. Most tools used in the MERN stack are free to use.
- **Operational Costs:** Cloud hosting platforms offer pay-as-you-go pricing, keeping running costs low and scalable. Maintenance costs are predictable and manageable.
- **Cost Savings:** Automating transport challan generation, route updates, and driver assignment reduces manual errors and improves staff productivity, translating to long-term savings.
- **Return on Investment (ROI):** The system improves operational workflows, reduces redundant efforts, and provides better visibility, leading to a strong ROI through efficiency and cost-cutting.

### 2.2.4. Legal and Ethical Feasibility

- **Data Privacy and Protection:** UTMS will comply with national data protection regulations. All sensitive data (e.g., student records, challan history) will be encrypted and stored securely.
- **University Compliance:** All modules will follow internal university policies regarding student data usage and transport services.
- **Ethical Use:** User consent will be obtained before data collection, and privacy policies will be clearly communicated. The system ensures transparency in how data is handled.
- **Accessibility:** The user interface will be designed to comply with accessibility standards, ensuring ease of use for individuals with visual, motor, or cognitive impairments.

### 2.2.5. Schedule Feasibility

Time is a crucial component. It is crucial to examine and evaluate whether a project can be finished on schedule and with the manpower and resources that are available. Three people make up our project team, and they are all engaged and constantly accessible for work. The amount of time needed to finish the job is sufficient. 18–20 weeks are allocated to this project. The job can be readily finished in this time frame. It is thus possible.

## 2.3. Project Plan

### 2.3.1. Initiation Phase

The **Initiation Phase** for UTMS focused on understanding the transportation-related issues faced by students and administrators in a university setting. The project aimed to develop an intelligent transport management system that enables students to register, upload challans, and book seats, while allowing transport administrators to manage buses, routes, and schedules effectively.

A feasibility study was conducted to assess:

- **Technical feasibility:** Using the MERN stack (MongoDB, Express.js, React.js, Node.js) for full-stack web development.
- **Operational feasibility:** Evaluating how easily the system could be integrated into the university's transport department.
- **Economic feasibility:** Ensuring the project could be developed with minimal cost using open-source tools.
- **Legal and ethical considerations:** Ensuring student data privacy and secure access.

A project charter was developed, outlining the core objectives, expected deliverables, major stakeholders, and potential risks. Approval was secured from the relevant academic authorities for project development.

### 2.3.2. Requirements Analysis Phase

In this phase, a detailed analysis was conducted to determine the exact needs of students, university administration, and transport staff. Several stakeholder meetings and surveys were carried out to gather requirements.

Identified requirements included:

#### 2.3.2.1. Functional Requirements:

- Student registration and login system
- Uploading of paid fee challans
- Admin approval for challans
- Seat booking and route selection
- Transport schedule management
- Bus and route data entry by admin

#### 2.3.2.2. Non-Functional Requirements:

- Secure authentication
- Responsive and user-friendly interface
- Scalability for future enhancements
- Performance under concurrent student usage

These were documented in the Software Requirements Specification (SRS), which included use case diagrams, system flow, and user interface sketches. The SRS served as a mutual reference for developers and stakeholders, reducing ambiguity and scope creep.

### 2.3.3. Design Phase

The Design Phase translated the approved requirements into an architectural and visual design for UTMS. The system was designed using the MERN stack, with the following architecture:

- Front-end: React.js for student and admin dashboards
- Back-end: Node.js and Express.js to handle API requests and business logic
- Database: MongoDB for storing student records, challans, bus details, and bookings

#### Key Design Components:

- UI/UX wireframes for student and admin portals were created with focus on usability and clarity.
- Database schema was designed to represent user roles, challan statuses, routes, buses, and bookings.
- API specifications were created to support interactions like challan approval, booking confirmation, and schedule updates.

Design standards were defined to ensure consistency in interface layout, performance optimization, and secure role-based access. The finalized design documents were used as the foundation for development.

### 2.3.4. Development Phase

During this phase, the system components were built according to the finalized design.

#### Front-End Development:

- Built in React.js to provide a smooth and interactive user interface.
- Separate dashboards for students (challan upload, booking status) and admins (challan verification, route management).

#### Back-End Development:

- Developed using Node.js with Express.js to handle routing, middleware, and API logic.
- Implemented features like challan validation, seat assignment logic, and bus-route management.

#### Database Development:

- Collections in MongoDB were created for users, challans, buses, routes, and bookings.
- CRUD operations ensured real-time data management for the transport system.
- Version control using Git was maintained for collaboration and code tracking. Development was broken down into sprints, with each sprint focusing on specific modules (e.g., challan system, admin portal, schedule planner).

### 2.3.5. Integration and Testing Phase

After development, all components were integrated and thoroughly tested.

#### Integration:

- Front-end and back-end modules were connected via APIs.
- Real-time interaction between user actions (like seat booking) and backend responses was tested.
- Testing Strategy:
  - Unit Testing: Each module (e.g., challan upload, login system) was tested in isolation.
  - Integration Testing: Ensured seamless flow between student and admin modules.
  - System Testing: Verified the complete CampusWheel system against the original requirements.
  - Security Testing: Verified secure login, access control, and data protection.

**Tools Used:**

- Postman for API testing
- Jest/Mocha for unit and integration testing
- Browser DevTools for UI/UX and performance checks
- Issues discovered during testing were logged and resolved before proceeding to deployment.

**2.3.6. Implementation Phase**

In this phase, CampusWheel was deployed for final use. The front-end and back-end systems were connected in a live environment, ensuring proper functionality.

**Activities:**

- Final testing was conducted to ensure production-readiness.
- Deployment scripts and environment variables were configured.
- Admin and student accounts were initialized for demonstration.
- Git repository was finalized and documented.

**Documentation was created for:**

- System usage
- Admin controls
- API endpoints
- Deployment and maintenance guidelines

After final validation from mentors and stakeholders, UTMS was officially marked as implemented and ready for use.

**2.4. Introduction to Team Member and Their Skills****Team Members:**

Zeenat Arooj  
Ahmad Sultan  
M. Adeel

**Skill:**

Front End and Documentation  
Back-End Developer and Quality Check  
Front-End and Attainment Manager

## 2.5.Risk List

- **Data & Authentication Security Risks:**  
Potential vulnerabilities in user registration, login, and challan generation modules may lead to unauthorized access or data breaches. Strong encryption and secure APIs are essential to mitigate this.
- **Challan Verification & Payment Integrity:**  
Risks of fake or unpaid challans being approved due to human error or insufficient admin validation checks. Mitigation includes backend validation and admin notification systems.
- **System Downtime or Server Failures:**  
Outages in hosting services or backend servers could disrupt student access and transport scheduling. Redundancy planning and periodic backups are needed to reduce this risk.
- **Manual Route Selection Errors:**  
Students selecting incorrect routes (due to lack of live GPS) could affect transport efficiency. In the future, route recommendation or live tracking can minimize this issue.
- **User Experience Issues:**  
If the web interface is not intuitive, students may find it difficult to complete tasks like registration or challan generation. Regular usability testing and feedback loops will help improve UX.



## **Chapter 3: UTMS System Architecture and Workflow**

---

## 3.1 Introduction

**UTMS** (University Transport Management System) is a comprehensive web-based solution developed to modernize and streamline the transport management system within universities. As student populations grow and transportation demands increase, traditional manual systems fall short in efficiency and reliability. UTMS addresses these challenges by providing a structured and user-friendly platform for students and administrators alike.

One of the core features of UTMS is its simplified registration and approval process. Previously, availing transport services required complex paperwork and in-person verification. Now, students can register online by submitting their personal and route details. This data is reviewed by the transport department, ensuring that only verified students gain access to the services. Email notifications keep students informed at every step, from registration confirmation to challan approval.

The system also enables secure and efficient challan generation. Students can manually generate a transport challan based on their selected route and submit payment proof by uploading a scanned image. The admin panel allows transport officials to verify and approve or reject these challans, streamlining the process and reducing delays. Approved students receive confirmation emails indicating their successful enrollment in the transport system.

Although UTMS currently lacks real-time GPS integration, it supports manual route selection and provides route listings to help students choose the most suitable option. Admins have full control to add, update, or remove routes, buses, and driver assignments through a dynamic dashboard, ensuring adaptability to scheduling needs.

A clean, interactive dashboard enhances the overall user experience. Students can easily navigate features such as registration status, challan tracking, and feedback submission. Administrators benefit from a robust control panel to manage transport logistics, view challan statuses, and respond to student queries.

The system is built using the **MERN stack** — MongoDB for the database, Express.js and Node.js for the backend, and React.js for the frontend. This modern tech stack ensures scalability, responsiveness, and data security, making UTMS a reliable and maintainable transport solution for universities.

By combining intuitive interfaces, smart approval workflows, and centralized management, UTMS delivers a smarter, faster, and more transparent transport management experience.

## 3.2 Website Features and User Experience

In order to provide a easy user revel in, the fast-Campus website offers a number of features that are useful for both administrative personnel and students. The system's person-friendly design and big functions guarantee that customers can results easily browse it and effectively manage their transportation necessities.

The following is a listing of the website's salient functions:

### 3.2.1. Signup Process

First, users must complete a thorough registration form with all the facts required for their signup request. users get an email confirmation that their request has been obtained after submitting it. After reviewing the request, the college's transportation branch decides whether or not to permit it or refuse it. electronic mail signals are then issued to the users to allow them to recognize what happened.

- Name
- Father Name
- Roll No
- Department
- semester
- Email
- Route Name
- Pickup Stop
- Profile image
- Password

### **3.2.2. User Login and Access**

Users who have been granted access may get entry to the website through entering their password and email address. Upon logging in, customers are granted get admission to to an intuitive dashboard that consists of navigation selections along with home, provider agenda, view profile, notifications, contact us, and about us. This enables customers to effectively manage their transportation-related tasks.

### **3.2.3. Transport Fee Management**

Through the internet, users may also create a transit rate challan and bodily pay it at the bank. They publish the paid challan photo to the system after making the charge. After reviewing the submitted image, the treasury admin decides whether to accept or reject it depending on its veracity. The fee control procedure is obvious for the reason that customers get e-mail notifications at the repute of their challan.

### **3.2.4. Route Schedule Viewing**

The website offers current facts on all routes that are on hand for transportation, along with bus information and timetables. This selection guarantees that clients get up-to-date and correct data on available transit options, which helps them plan their trips greater effectively.

### **3.2.5. Notification Management**

Users are guaranteed to get timely email information regarding important activities, such the progress of registration requests and challan approvals, thanks to the notification system. Notifications concerning new routes, timetable modifications, and different significant developments may also be posted by administrators and visible on the user dashboard.

### **3.2.6. Profile Management**

Through the website's see Profile area, users may also see and trade their personal records. By enabling customers to maintain their facts up to date, this feature ensures accurate records across the system.

### **3.2.7. Email Notifications**

Key activities, such submitting a signal-up request, getting permission or rejection, and receiving updates at the progress of a project, cause automatic e-mail indicators. For vital announcements or changes, administrators may additionally send out bulk electronic mail indicators.

### **3.2.8. Administrative Functions**

Transport routes, timetables, and bus data are in the admin's authority. Admins can add, update, or delete driver and bus records. Administrative notifications and updates can be posted at once to the consumer dashboard and sent through email

### **3.2.9. Security and Data Protection**

User data is protected with secure login processes and data encryption. Regular backups and data protection measures are in place to guarantee the security and integrity of data.

### **3.2.10. Responsive Design**

The website is made to be easy to use and responsive across a range of platforms, such as PCs, tablets, and smartphones. Users may use any internet-connected device to access and manage their transportation requirements.

### **3.2.11. User-Friendly Interface**

The website features a user-friendly interface with intuitive navigation. Clear labels and instructions are provided to guide users through different processes. The design ensures ease of use for users with varying levels of technical proficiency.

### **3.2.12. Feedback and support**

The website includes a comprehensive feedback and support system. Users can submit feedback or support requests through the Contact Us section, ensuring their concerns and suggestions are heard. Administrators review and respond to these submissions, fostering a responsive and usercentric transport management system.

### 3.3 Class Diagram

The class diagram of the application is shown below in figure 4. It contains the 12 classes like , Driver, User, Admin user request contact us routes schedule upload challan update challan, buses, stop, Notifications, Challan . Every class contain different Items and Functions who perform different operations.

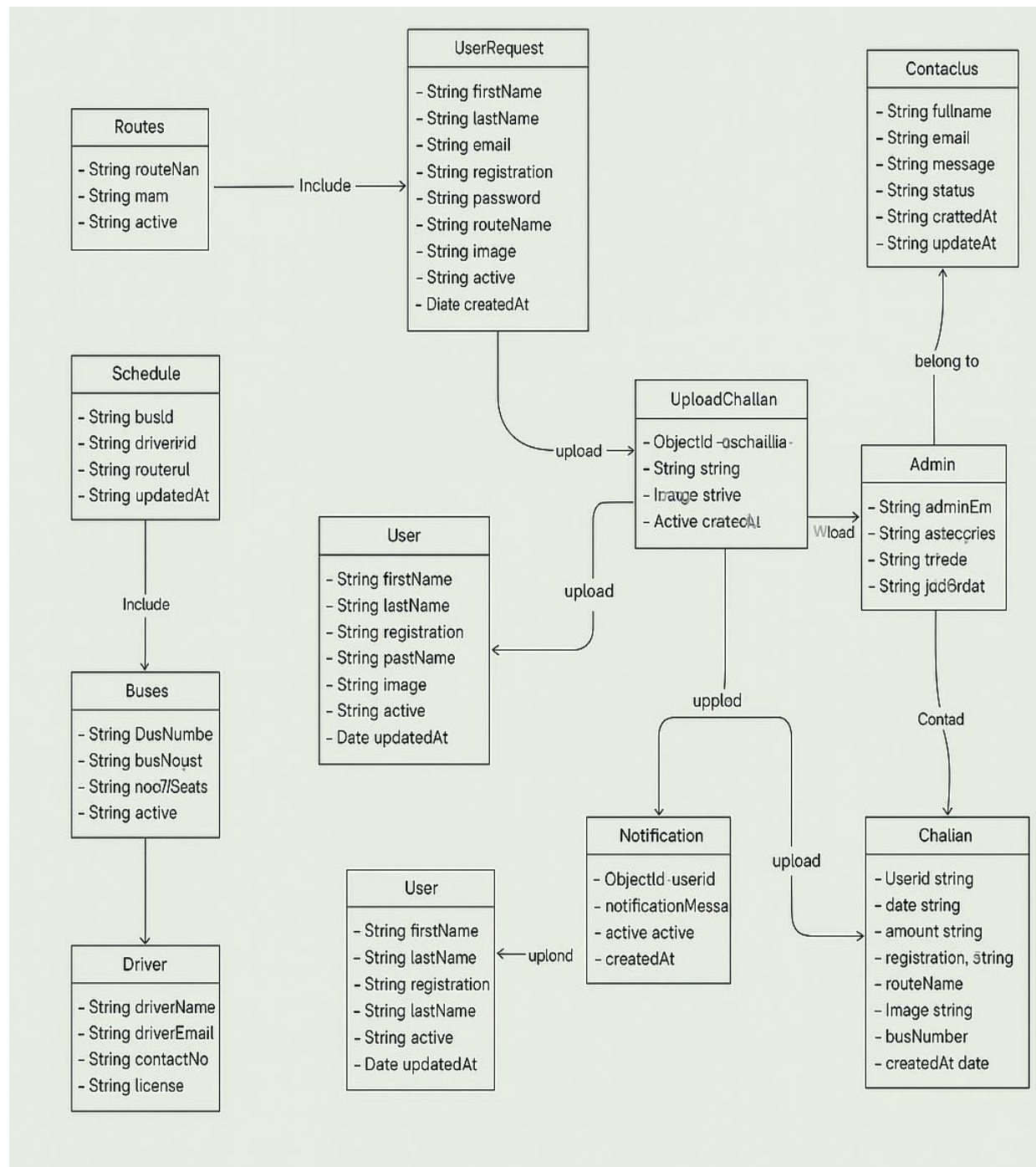


Figure 3 : Class Daigram

### 3.4 Entity Relation Diagram (ER)

The relationships between the processes and the characteristics are all represented in the entity relation (ER) diagram, as figure 5 below illustrates.

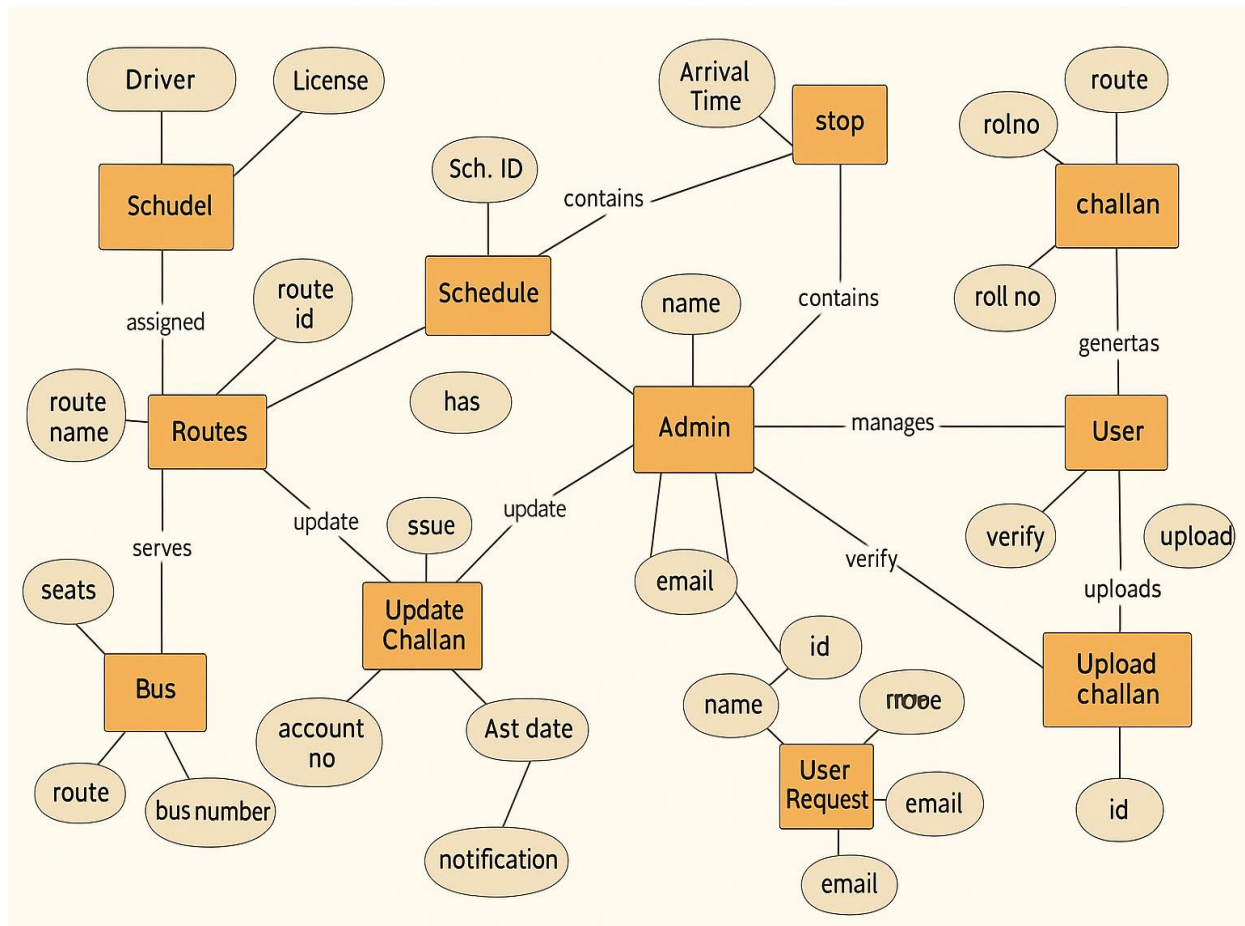


Figure 4 : ER Daigram

### 3.5. User Flow chart

It includes the website's flow, as shown in figure 6 below. If the credentials match those supplied by the administrator, the user may log in to the program and go to the homepage. If not, an error notice will be shown and the login page will be reloaded.

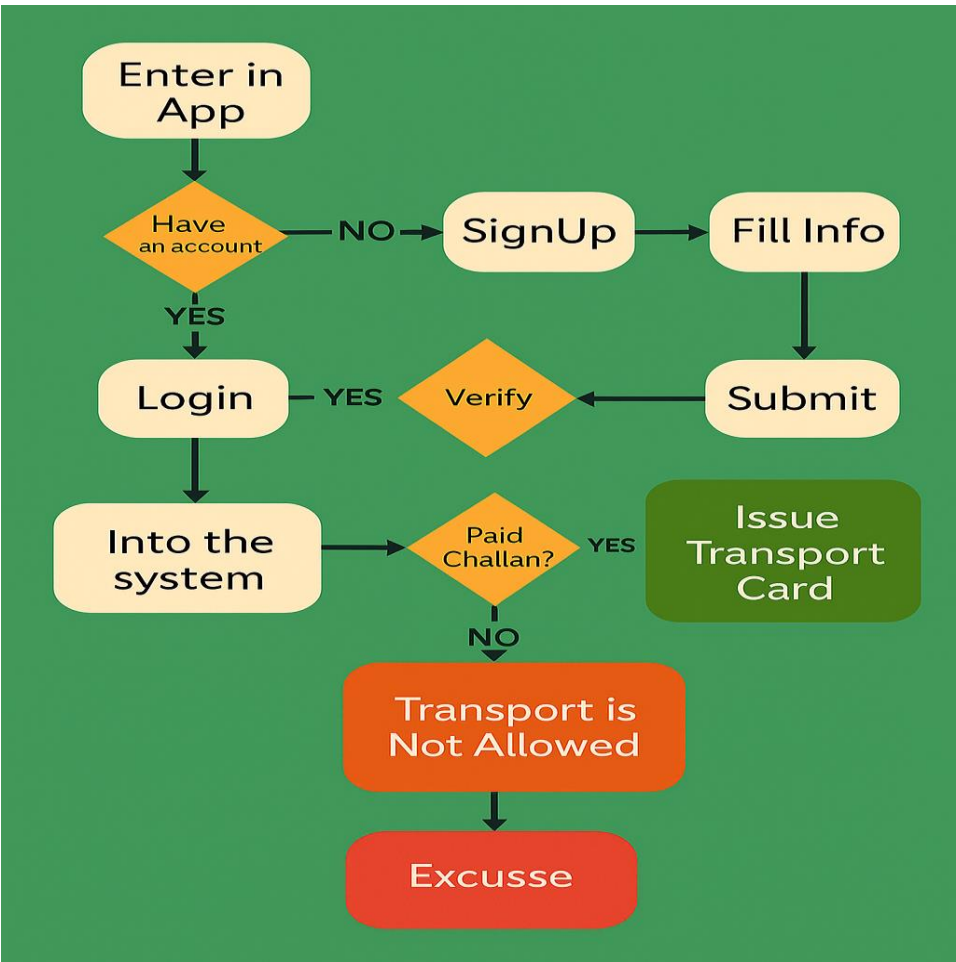


Figure 5 : User Flow Daigram

### 3.6. Verification flowchart

Figure 7 below illustrates the admin dashboard sequence that is included in it. If the credentials that the admin has supplied match, the admin may log in to the website and will be sent to their homepage based on their type. If not, an error notice will be shown and the login page will be reloaded.

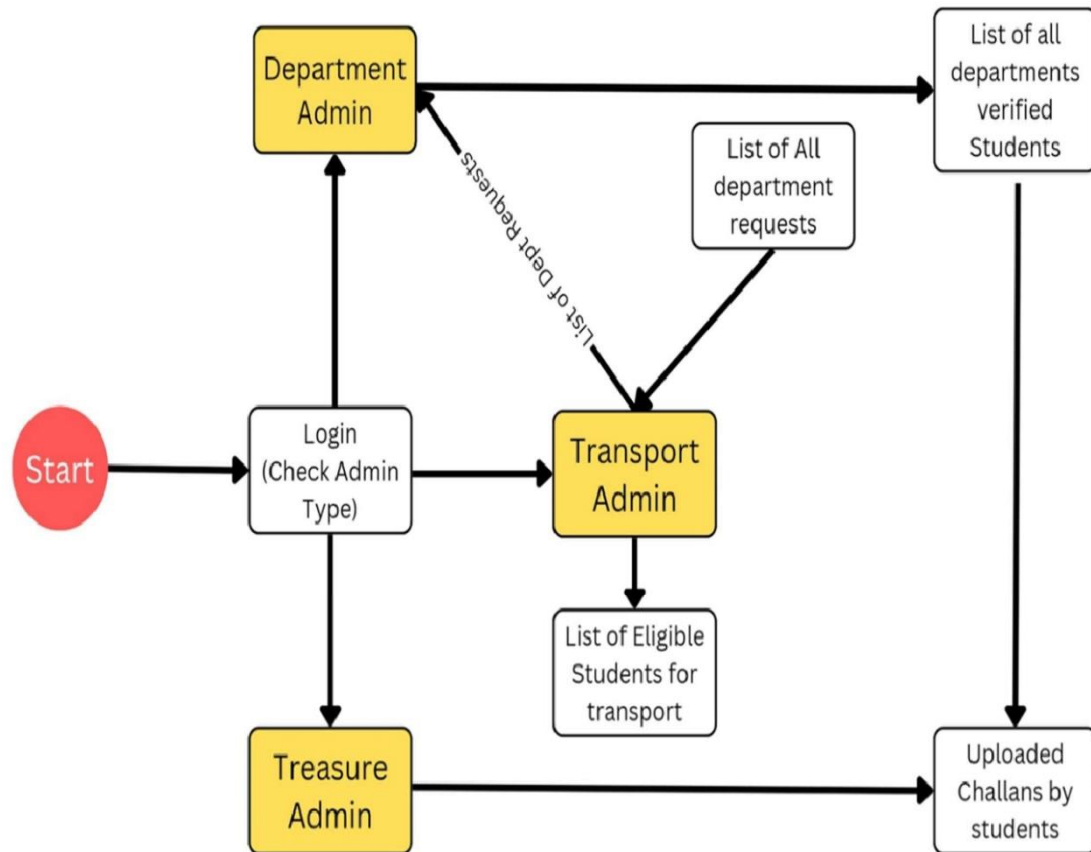


Figure 6 : Verification Flowchart



## **Chapter 4: Software Requirement Analysis**

---

## 4.1 Introduction

The multiple needs of the system from the perspectives of various users will be described in the next sections, together with a number of implementation-related choices that have been taken. These sections also make an effort to provide a brief overview of each user group's position inside the system by going over each one's specific responsibilities and capabilities.

### 4.1.1. User Interface

This system's user interface has to be straightforward and easy to use. The ages must, above all, be easily readable, comprehended, and accessible. There shouldn't be any problems with contrast, and the color palette should be adequate to provide a sense of familiarity with the institution.

#### 4.1.1.1. Transport Admin View Functionality:

- **Login:** Secure authentication process for the transport/super admin to access the system.
- **Manage Admins:**
  - Add, edit, and remove treasure admins and department admins.
  - Assign roles and permissions to different administrative users.
- **View Profile:**
  - Display the transport/super admin's profile information.
  - Provide options to update profile details and change passwords.
- **Manage Buses:**
  - Add, edit, and remove buses from the system.
  - Assign buses to specific routes and schedules.
- **Manage Drivers:**
  - Add, edit, and remove driver information.
  - Assign drivers to specific buses and routes.
- **Manage Routes:**
  - Create, edit, and delete transport routes.
  - Map routes with specific buses and schedules.
- **Manage Notifications:**
  - Create and send notifications to students and other admins.
  - View and manage sent notifications.
- **Manage Users:**
  - View and manage student user account
  - Approve or reject student registrations and challan uploads.
- **Manage Schedules:**
  - Create, edit, and delete transport schedules.
  - Assign schedules to specific buses and routes.
- **View Stats:**
  - Display statistics and analytics related to transportation operations.
  - View data on bus usage, driver performance, route efficiency, etc.
- **Update Challan:**
  - Update and manage transport challans.
  - Verify payment status and handle discrepancies.

- **Manage Transport Cards:**
  - Issue, update, and manage student transport cards.
  - Track card issuance and usage history.
- **View Feedbacks:**
  - View feedback and complaints from students and other users.
  - Address issues and improve service quality based on feedback.
- **Log Out:**
  - Securely log out from the admin interface.

#### 4.1.1.2. Treasure admin view Functionalities

- **Login:**
  - Secure authentication process for the treasure admin to access the system.
- **Verify Uploaded Fee Challans:**
  - Review and verify fee challans uploaded by students.
  - Approve or reject challans based on verification criteria.
- **Download Uploaded Challans:**
  - Download copies of uploaded challans for record-keeping or further verification.
- **View Profile:**
  - Display the treasure admin's profile information.
  - Provide options to update profile details and change passwords
- **View Stats:**
  - Access statistics and analytics related to challan verifications.
  - View data on the number of verified, pending, and rejected challans.
- **Log Out:**
  - Securely log out from the treasure admin interface

#### 4.1.1.3. Department Admin View functionalities

- **Login:**
  - Secure authentication process for the department admin to access the system.
- **Manage User Requests:**
  - View, approve, or reject student registration requests.
  - Verify student details and uploaded documents during the sign-up process.
  - Handle requests for updating student information or other related queries.
- **View Profile:**
  - Display the department admin's profile information.
  - Provide options to update profile details and change passwords.
- **View Stats:**
  - Access statistics and analytics related to user management.
  - View data on the number of student registrations, pending requests, and approved or rejected requests.
- **Log Out:**
  - Securely log out from the department admin interface.

## 4.2 Specific or Functional Needs

The system has to meet the following specifications:

### 4.2.1. Administrator Aspect

- Login and Authentication
- Manage Admins
- View and Update Profile
- Manage Buses
- Manage Drivers
- Manage Routes
- Manage Notifications

#### 4.2.2. Users Aspect

- Registration and Login
- Upload and Verify Challans
- View Profile
- Receive Notifications
- View Schedules
- Manage Transport Cards
- Submit Feedback
- Log Out

#### 4.2.3. Analysis

- Authenticating Users and Admins based on username and password.
- Checking whether challan is properly generated on student site.
- Checking whether challan is easily uploaded by student.
- Keeping history of generated challans. ☐ Keeping records of uploaded challans.
- Preserving a record of the customer input.

#### 4.2.4. Mailing

- While generating requests and when that request is approved by department admin, user gets all updates through email
- Email addresses of the clients should be used for transport.
- Upon confirmation of the registration, the user ought to get an email message.

#### 4.2.5. Performance Requirements

Here is a list of some of the performance criteria that were found:

- The database must have enough space to hold at least 10,000 user records.
- Several users may utilize the system simultaneously.
- Development won't be impacted by any additional particular performance requirements.

### 4.3 Non-Operational Conditions:

The client-server architecture of this system will function. An internet server that can execute web applications is necessary.

#### 4.3.1. Interface of Hardware

The system has to be operated via the internet, and any gear needed to connect to the internet will serve as the system's hardware interface. For example, WAN-LAN, Ethernet cross-cable, and modem.

#### Cab Management minimum system requirements:

- Processor: AMD comparable or Intel Core 2Duo, 2 GHz or above
- RAM: 2 GB; HDD: 160 GB; spin speed: 7200k

- Windows XP SP3 or Windows Vista Business SP1 operating system;
- Business broadband connection with static IP (8 MB download);

#### **4.3.2. User Interface**

Since the system is server-side, any scripting language—such as Node.js or React Native—is needed. The system needs a database to record any system transactions, such as MongoDB, etc. Additionally, DNS (domain name space) is needed for internet naming. In order to communicate with the system, both the admin and the user must have a web browser. An editor like Visual Studio Code is also necessary for developers.

#### **4.3.3. Performance Criteria**

There are no limitations on performance in this system since the server's requests and responses don't depend on the end user's connection to the web.

#### **4.3.4. Design Limitations**

The system will be built using an accepted norm web page builder that adheres to Microsoft's Graphical standards, such as HTML, FIGMA, etc.

#### **4.3.5. Security**

The system uses Secure Socket Layer, or SSL, for every transaction involving confidential customer information. The system needs to automatically log all users out after a certain period of inactivity. There shouldn't be any cookies from the system containing the customer's password on their computer. The systems' back-end servers will only be accessible by authorized administrators. Sensitive information is encrypted before being sent across insecure networks, such as the internet.

#### **4.3.6. Dependable**

The network automatically switches between redundant computers to store all databases. The dependability of the software as a whole depends on how reliable each component is. The database backup is the key source of the system's reliability and is regularly updated and maintained to reflect the most recent changes. Consequently, the overall stability of the system is determined by the stability of the underlying operating system and the container.

#### **4.3.7. Reachable**

The only restrictions should be times when the system's hosting server is unavailable, other than the user's constant ability to access the system using a web browser. An alternative page will emerge in the event of a hardware breakdown or corruption in the database. In the event of a hardware failure or damage, the database administrator should also extract and preserve backups of the database from the server. After that, the service will be restarted. It indicates that you are available at all times.

#### **4.3.8. Dependability**

A commercial database is used to manage the database. If the program fails, it will be restarted from scratch. Furthermore, modularity is taken into account while designing the software in order to promote efficient maintainability.

#### **4.3.9. Mobility**

Programming languages and HTML are used in the program's construction. As a result, the end user piece is fully portable and should function on any computer running any web browser, including architectures for devices that have been or will be introduced in the future.

## **Chapter 5: Testing**

---

## 5.1 Introduction

Testing is a collection of methods to assess an application's correctness under a predetermined script, but it is not able to identify every application defect; rather, its primary goal is to identify application failures so that values can be found and fixed. Testing does not prove that a product works as intended under all circumstances; rather, it only shows that it malfunctions under a subset of conditions. The intended audience's adoption of the program, its simple graphical user interface, its robust functionality, testing, loading, etc., are all necessary for its success.

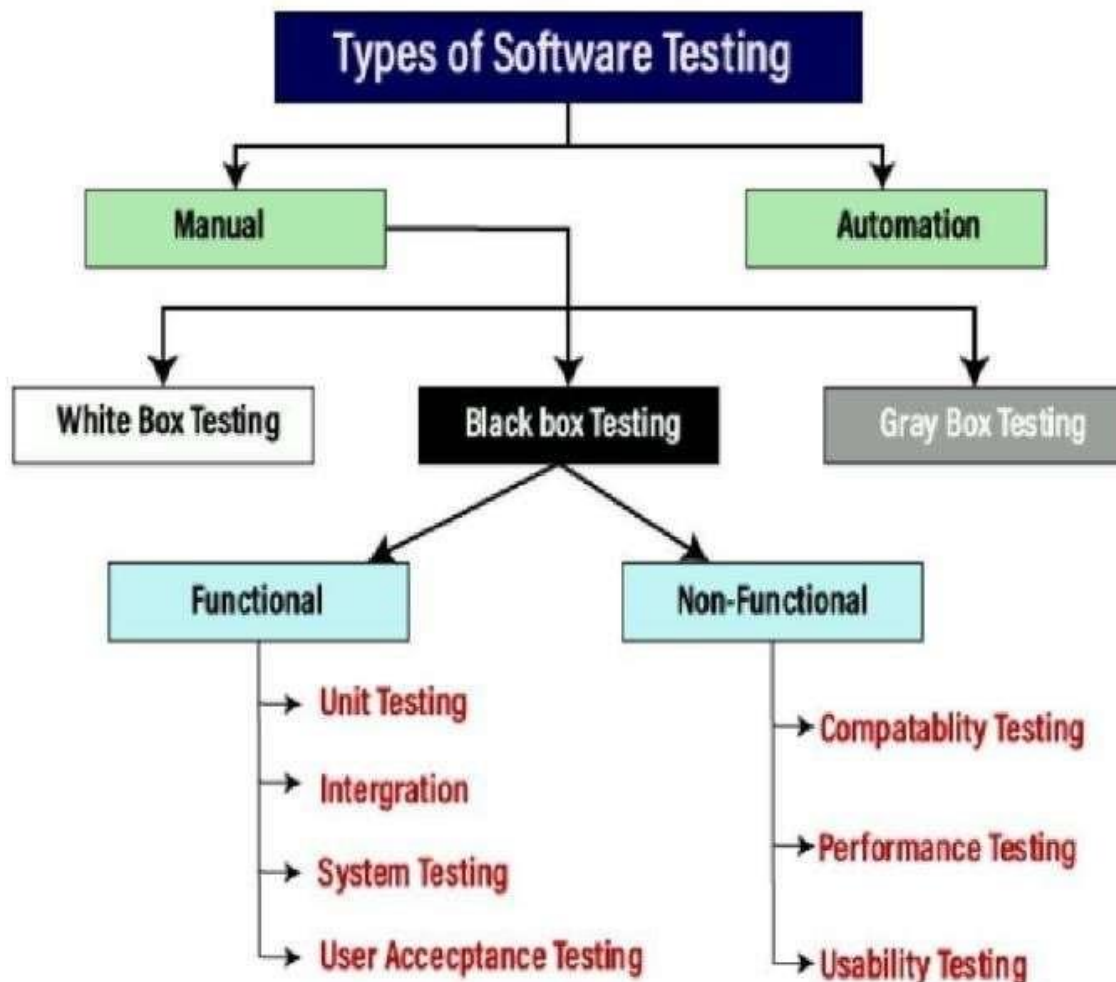


Figure 7: Testing



## 5.2. Login Test

We use a variety of procedures, such as Login with Valid, Invalid, and Empty Credentials, to verify the login.

### 5.2.1. Login Successful

Test By	Zeenat Arooj
Test Case ID	TC-1-01
Related UC/FR/NFR	UC-001
Date	02-06-2025
Purpose	To check whether the user successfully logs in after entering credentials.
Pre-Requisite	Website must be running and there should be internet access.
Test Data	User Name = “valid Username”, Password = “valid password”.
Steps	<ol style="list-style-type: none"> <li>1. Open the App.</li> <li>2. Tap the Login Button.</li> <li>3. Enter a valid Email or User Name and Password.</li> <li>4. Tap the Login Button.</li> </ol>
Expected Result	User must be “logged in successfully and redirected to the Homepage”.
Actual Result	User is “logged in successfully and redirected to the homepage”.
Status	Pass

Table 1 : Login Successful Test

### 5.2.2. Wrong Credentials while Login

Test By	Zeenat Arooj
Test Case ID	TC-1-01
Related UC/FR/NFR	UC-002
Date	02-06-2025
Purpose	To check whether the user successfully logs in after entering wrong credentials.

Pre-Requisite	Website must be running and there should be internet access.
Test Data	User Name = “invalid Username”, Password = “invalid password”.
Steps	<ol style="list-style-type: none"> <li>1. Open the App.</li> <li>2. Tap the Login Button.</li> <li>3. Enter an invalid Email or User Name and Password.</li> <li>4. Tap the Login Button.</li> </ol>
Expected Result	User should be shown an error “Invalid username or password. Please Try again.”
Actual Result	User must be shown an error “Invalid username or password. Please try Again.”
Status	Pass

Table 2 : Test Wrong Credentials while Login

### 5.2.3. Empty Fields while Login

Test By	Zeenat Arooj
Test Case ID	TC-1-03
Related UC/FR/NFR	UC-003
Date	02-06-2025
Purpose	To check the response of the system while trying to Login without providing an Email or User Name and Password.
Pre-Requisite	Website must be running and there should be an internet access.
Test Data	User Name = “blank field”, Password = “blank field”.
Steps	<ol style="list-style-type: none"> <li>1. Open the App.</li> <li>2. Tap the Login Button.</li> <li>3. Left the email and password field empty.</li> <li>4. Tap the Login Button.</li> </ol>
Expected Result	User should be shown an error “Please provide both Email/User Name And Password.”
Actual Result	User must be shown an error “Please provide both Email/User Name and Password.”
Status	Pass

Table 3 :Test Login with Empty Fields

## 5.2 Add User

### 5.2.4. Add user with Valid Information

Test By	M. Adeel
Test Case ID	TC-1-04
Related UC/FR/NFR	UC-001
Date	02-06-2025
Purpose	To check the response of website after adding the user.
Pre-Requisite	Website must be running and there should be an internet access.
Test Data	The personal information of new user Member.
Steps	<ol style="list-style-type: none"> <li>1. Open the App.</li> <li>2. Tap Signup Button.</li> <li>3. Add user Details.</li> <li>4. Tap the Submit Button.</li> </ol>
Expected Result	User member must be added successfully
Actual Result	User member added successfully
Status	Pass

Table 4 : Test of Adding User with valid information

### 5.2.1 Add User with Invalid Information

Test By	M. Adeel
Test Case ID	TC-1-05
Related UC/FR/NFR	UC-002
Date	02-06-2025
Purpose	To check the response of website after adding the user.
Pre-Requisite	Website must be running and there should be an internet access.
Test Data	Invalid information of New User Member.
Steps	<ol style="list-style-type: none"> <li>1. Open the App.</li> <li>2. Tap Signup Button.</li> <li>3. Add User.</li> <li>4. Tap the Submit Button.</li> </ol>
Expected Result	Request is then send to Department admin who will reject it.
Actual Result	Email notification of request registration failed send to that user

Status	Pass
--------	------

Table 5 :Test of adding User with Invalid Information

### 5.3 Online Challan Genneration

Test By	Ahmad Sultan
Test Case ID	TC-1-05
Related UC/FR/NFR	UC-001
Date	03-06-2025
Purpose	To check the response of website after user registration.
Pre-Requisite	Website must be running and there should be an internet access.
Test Data	Valid information of the user i.e. Route and Stop
Steps	<ol style="list-style-type: none"> <li>1. Open the Website.</li> <li>2. Go to Generate Challan.</li> <li>3. Enter Information.</li> <li>4. Tap the Submit Button.</li> </ol>
Expected Result	Challan must be downloaded in PDF Successfully.
Actual Result	Challan is downloaded Successfully.
Status	Pass

Table 6 : Test Online Challan Generation

#### 5.3.1 Update Challan Data(Admin Site)

Test By	Ahmadd Sultan
Test Case ID	TC-1-06
Related UC/FR/NFR	UC-002
Date	03-06-2025
Purpose	To check the response of Challan after updating challan.
Pre-Requisite	System must be running and there should be an internet access.
Test Data	Provide Data to Update challan credentials
Steps	<ol style="list-style-type: none"> <li>1. Open the Dashboard.</li> <li>2. Go to Update Challan.</li> <li>3. Provide new data.</li> <li>4. Tap the Submit button.</li> </ol>

Expected Result	Challan data must be update i.e. Account number, Due date etc.
Actual Result	Data is updated while generating new challan at user site
Status	Pass

Table 7: Test to Update Challan data.

## 5.4 Notifications

Test By	Ahmad Sultan
Test Case ID	TC-1-07
Related UC/FR/NFR	UC-001
Date	03-06-2025
Purpose	To check whether the notifications are delivered to user from admin
Pre-Requisite	Website must be running and there should be an internet access.
Test Data	Information that will be sent to user on user site and Email
Steps	<ol style="list-style-type: none"> <li>1. Open the Admin Dashboard.</li> <li>2. Go to Notifications.</li> <li>3. Enter message to send.</li> <li>4. Click on Send</li> </ol>
Expected Result	Notification should be received at user site and Email.
Actual Result	Notification is received Successfully.
Status	Pass

Table 8 : Notification test

### 5.4. Update Schedule (Admin Site)

Test By	M. Adeel
Test Case ID	TC-1-08
Related UC/FR/NFR	UC-001
Date	03-06-2025
Purpose	To check whether schedules are view by student and updates in schedules by admin are received at user site.
Pre-Requisite	Websites must be running and there should be an internet access.
Test Data	Valid information of the Schedule.

Steps	<ol style="list-style-type: none"> <li>1. Open the Admin Dashboard.</li> <li>2. Go to Schedules.</li> <li>3. Click on Add new.</li> <li>4. Provide the route data and also pickup and drop-off time.</li> <li>5. Then press submit.</li> </ol>
Expected Result	Schedules must be updated at user site and user can easily view.
Actual Result	Schedules are updated and viewed successfully.
Status	Pass

Table 9 : Update Data

## **Chapter 6: Website Interface**

---

## 6.1. Admin Login

An important component of the quick-Campus challenge is the admin login characteristic, which gives distinctive administrative roles—including department admin, transport admin, and treasure admin—secure access to the system. By means of limiting access to the administrative functions to only authorized employees, this machine protects safety and integrity.

Maintaining security and integrity, the admin login gadget makes certain that handiest legal people may additionally use the executive capabilities of the rapid-Campus device.



Figure 8 : Admin Login

Admins should input a valid username and password combination in order to access the system securely. The database encrypts passwords to enhance protection and prevent undesirable access.

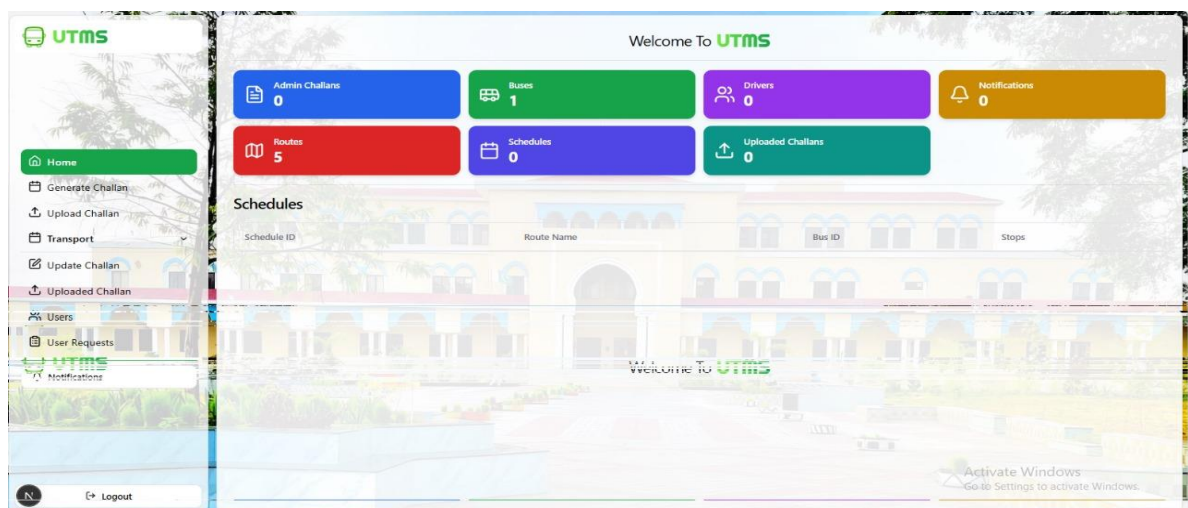


Figure 9 : Admin Home page



## 6.2. Manage User Request

A student uploads a profile picture and provides the required personal information when they register for the swift-Campus system. After then, this sign-up request is sent in and is pending department administrator verification.

After logging onto the executive portal, the department administrator may be able to access a section meant for coping with patron inquiries. They may go over the student records, uploaded profile photo, and any pending registration requests on this part.

Every request is placed via checking out by using the branch administrator to make sure the information is correct and the profile picture meets all requirements.

Figure 10 : User/Student SignUp

The student will be granted access to the tool and all of its functions if the department administrator reviews the request and decides it's suitable. The student receives a phrase and has access to the quick-Campus gadget after politely registering. The department administrator may also reject a request if it is incomplete or includes mistakes; in this case, they will provide an explanation for their choice. The student has the option to either make the necessary corrections or submit any missing information for reevaluation after receiving the rejection. By granting only the most deserving certified students access to the university's transport options, this tactic preserves the correctness and integrity of the device.

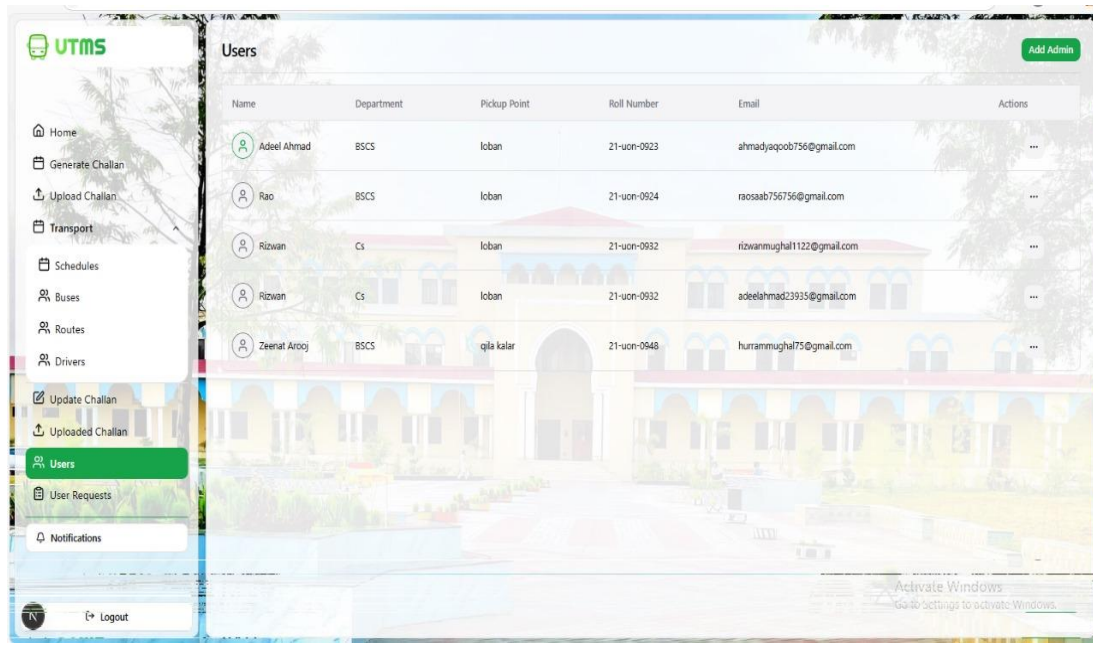


Figure 11 : Department Admin manages SignUp Requests

### 6.3. E-mail Notifications

Students who use the quick-Campus device to submit their registration requests get an e-mail confirming that their requests had been received. This email informs them that their sign-up request is being reviewed via the department administrator and consists of information about them, such as their software identity. The student gets an extra email note detailing the result after the department administrator's assessment of the request. The email will notify the student that their request has been accepted and emphasize the benefits they may now enjoy, such as the ability to create on-line challans, get signals, and test transportation schedules. Need to the request be denied, the student gets an e mail outlining the reasons for the denial along with commands on how to address any issues.

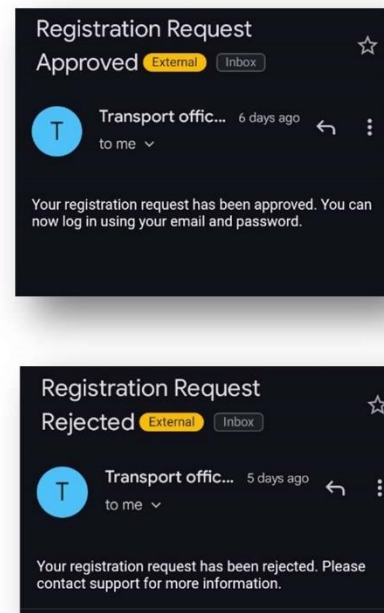


Figure 12 : Email Notifications

## 6.4. challan Generation

Allows the users to generate the challan.

Bank Copy	Student Copy	Treasure Copy	Office Copy																																				
Sr. No.	Sr. No.	Sr. No.	Sr. No.																																				
<b>UNIVERSITY OF NAROWAL</b> <small>University of Narowal</small>																																							
The Bank of Punjab A/C- 1234567890																																							
Name: zeenat arooj	Name: zeenat arooj	Name: zeenat arooj	Name: zeenat arooj																																				
Father Name: mirza asghar	Father Name: mirza asghar	Father Name: mirza asghar	Father Name: mirza asghar																																				
Roll No: 21-uon-0948	Roll No: 21-uon-0948	Roll No: 21-uon-0948	Roll No: 21-uon-0948																																				
Contact No:	Contact No:	Contact No:	Contact No:																																				
Semester: 8      2023-2024	Semester: 8      2023-2024	Semester: 8      2023-2024	Semester: 8      2023-2024																																				
Program: Morning	Program: Morning	Program: Morning	Program: Morning																																				
Department: BSCS	Department: BSCS	Department: BSCS	Department: BSCS																																				
Route: Narang      Bus Stop: qela	Route: Narang      Bus Stop: qela	Route: Narang      Bus Stop: qela	Route: Narang      Bus Stop: qela																																				
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Sr.</th> <th>Particulars</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Transport Fee</td> <td>15000/-</td> </tr> <tr> <td></td> <td>Grand Total</td> <td>15000/-</td> </tr> </tbody> </table>	Sr.	Particulars	Amount	1	Transport Fee	15000/-		Grand Total	15000/-	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Sr.</th> <th>Particulars</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Transport Fee</td> <td>15000/-</td> </tr> <tr> <td></td> <td>Grand Total</td> <td>15000/-</td> </tr> </tbody> </table>	Sr.	Particulars	Amount	1	Transport Fee	15000/-		Grand Total	15000/-	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Sr.</th> <th>Particulars</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Transport Fee</td> <td>15000/-</td> </tr> <tr> <td></td> <td>Grand Total</td> <td>15000/-</td> </tr> </tbody> </table>	Sr.	Particulars	Amount	1	Transport Fee	15000/-		Grand Total	15000/-	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Sr.</th> <th>Particulars</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Transport Fee</td> <td>15000/-</td> </tr> <tr> <td></td> <td>Grand Total</td> <td>15000/-</td> </tr> </tbody> </table>	Sr.	Particulars	Amount	1	Transport Fee	15000/-		Grand Total	15000/-
Sr.	Particulars	Amount																																					
1	Transport Fee	15000/-																																					
	Grand Total	15000/-																																					
Sr.	Particulars	Amount																																					
1	Transport Fee	15000/-																																					
	Grand Total	15000/-																																					
Sr.	Particulars	Amount																																					
1	Transport Fee	15000/-																																					
	Grand Total	15000/-																																					
Sr.	Particulars	Amount																																					
1	Transport Fee	15000/-																																					
	Grand Total	15000/-																																					
Amount Fifteen thousand Only/-																																							
<b>Fee Submission Date:</b> 2025-05-08 to 2025-06-07																																							
<b>Note:</b> Challan after due date will not be accepted. Student has the responsibility to retain challan copy till completion of degree.																																							

Generate Challan

Figure 13 : Online Challan Generation

## 6.5. User Site

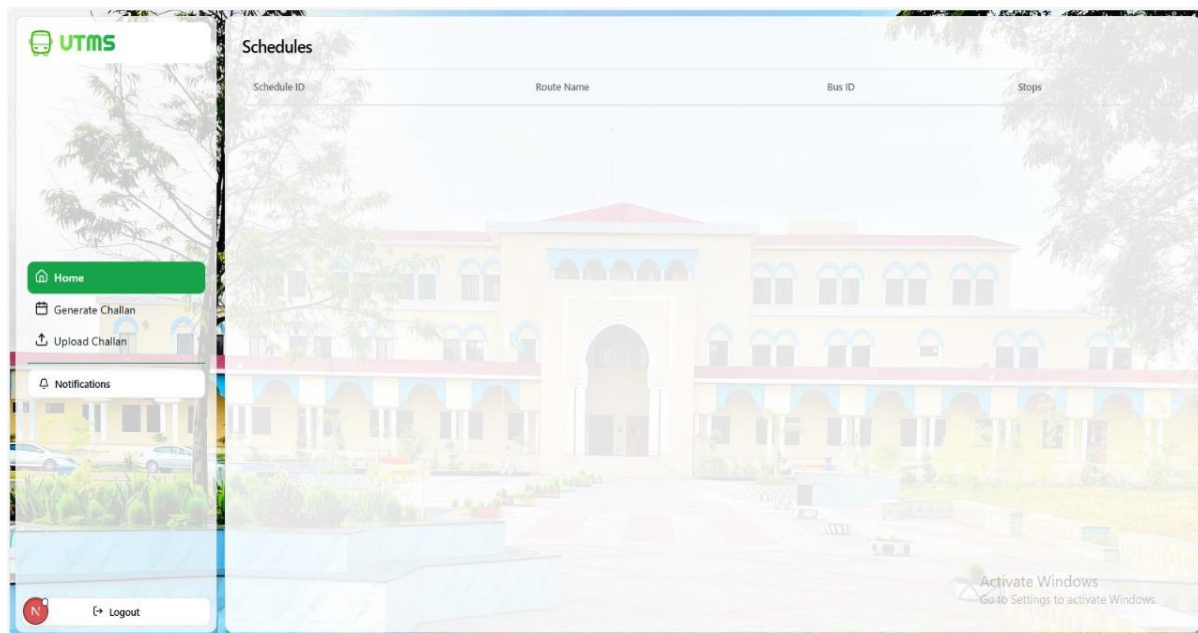


Figure 14 : User Site

## 6.6. Paid Challan Verification

The screenshot shows the 'Update Challan' form in the UTMS dashboard. The form is located on the right side of the page, with a sidebar on the left containing navigation links. The form fields are as follows:

Field	Value
Account No	1234567890
Session	2023-2024
Amount	15000
Issue Date	05/08/2025
Last Date	06/07/2025
Max Challan	10

A green 'Submit' button is located at the bottom left of the form. The sidebar on the left includes links for Home, Generate Challan, Upload Challan, Transport, Schedules, Buses, Routes, Drivers, Update Challan (highlighted), Uploaded Challan, Users, User Requests, and Notifications. The background of the dashboard features a blurred image of a university building.

Figure 16 : Challan Process

## 6.7 Stats

The screenshot shows the 'Buses' section in the UTMS dashboard. It displays a table of bus routes with the following columns: Bus ID, Route, Number, Seats, Active, and Actions. The table contains 8 rows of data. The sidebar on the left is the same as in Figure 16, with 'Buses' highlighted. The background features a blurred image of a university building.

















Bus ID	Route	Number	Seats	Active	Actions
SAL-863	Qasim pura	1	40	Yes	 
GTJ-26	Narang	2	40	Yes	 
GTJ-29	Pasrur	3	40	Yes	 
GTJ-28	Dairavala	4	40	Yes	 
GTJ-30	Darman Chock	5	40	Yes	 
SAX-142	Baddomalhi	6	40	Yes	 
SAL-119	Narang	7	40	Yes	 
SAL-865	Molan Mor	8	40	Yes	 

Figure 17 : Buses Routes



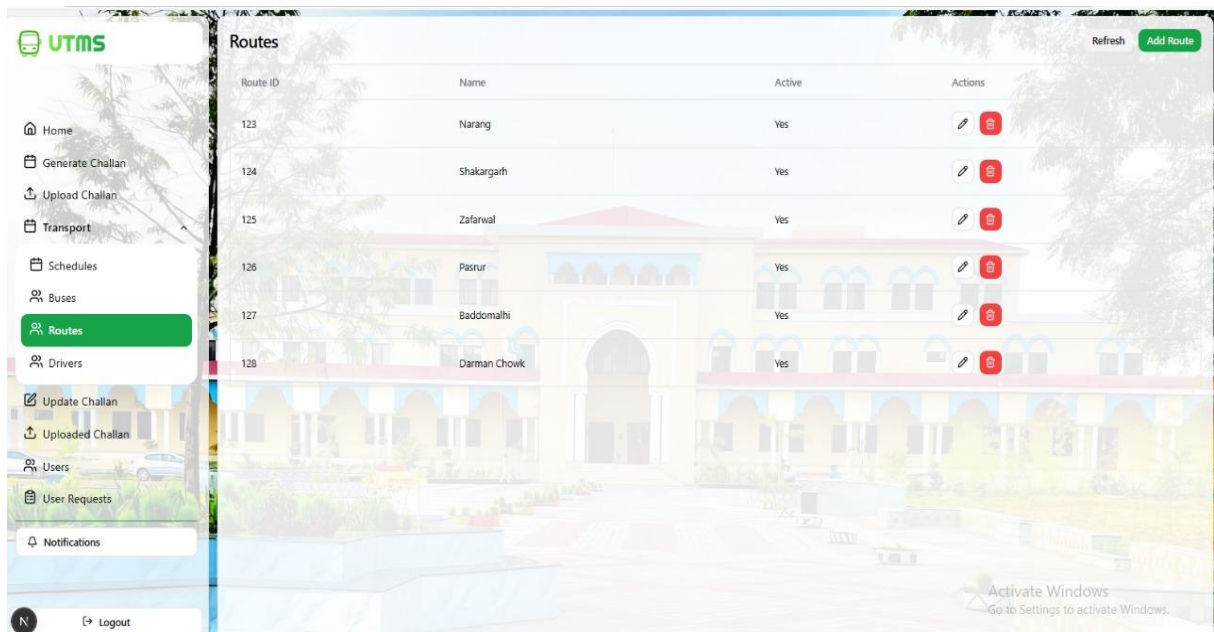


Figure 18 : Routes

## References

- [1] J. Obetta, L. Nwakwusi, I. C. Dialoke, et al., “Design and Implementation of Web-Based Smart Campus Bus Transport Management System with Real-Time Display of Arrival Time,” *Int. Multilingual J. Sci. Technol.*, vol. 8, no. 2, Feb. 2023.
- [2] G. Sivakumar, B. Abinayasri, M. Jayaprakash, and M. Kannan, “Automated Transport Management System: Route Optimization and Attendance Tracking for Institutions,” *Int. Res. J. Adv. Eng. Hub*, vol. 2, no. 12, pp. 2707–2714, Dec. 2024.
- [3] A. Lushi, D. Daas, and M. Nadeem, “IoT-Based Public Transport Management System,” in *Proc. 2022 IEEE Global Conf. Artif. Intell. Internet Things (GCAIoT)*, Dec. 2022.
- [4] N. Khamis and K. K. Li, “User Experience Evaluation for a Bus Tracking App in Smart Campus Initiative,” *Bull. Electr. Eng. Informatics*, vol. 10, no. 4, pp. 2254–2262, Aug. 2021.
- [5] Y. Chen et al., “Enhancing Campus Mobility: Achievements and Challenges of Autonomous Shuttle ‘Snow Lion’,” *arXiv preprint*, Jan. 2024.
- [6] I. C. Haron, N. S. Saleem, M. H. Hasan, et al., “A RFID-Based Campus Context-Aware Notification System,” *arXiv preprint*, Mar. 2010.
- [7] B. P. Ashwini, R. Sumathi, and H. S. Sudhira, “A Dynamic Model for Bus Arrival Time Estimation Based on Spatial Patterns Using Machine Learning,” *arXiv preprint*, Oct. 2022.
- [8] N. C. Petersen, F. Rodrigues, and F. C. Pereira, “Multi-Output Bus Travel Time Prediction with Convolutional LSTM Neural Network,” *arXiv preprint*, Mar. 2019.
- [9] “Campus Area Bus Service,” Wikipedia, accessed Jul. 2025. [Online]. Available: [https://en.wikipedia.org/wiki/Campus\\_Area\\_Bus\\_Service](https://en.wikipedia.org/wiki/Campus_Area_Bus_Service)
- [10] “Intelligent Transportation System,” Wikipedia, accessed Jul. 2025. [Online]. Available: [https://en.wikipedia.org/wiki/Intelligent\\_transportation\\_system](https://en.wikipedia.org/wiki/Intelligent_transportation_system)
- [11] H. X. Liu and S. Feng, “Traffic Light Optimization with Low Penetration Rate Vehicle Trajectory Data,” *Nature Communications*, vol. 15, art. 1306, 2024.
- [12] B. Abdulhai, H. Abdelgawad, and others, “Adaptive Traffic Signal Control with Deep Reinforcement Learning and High Dimensional Sensory Inputs: Case Study and Comprehensive Sensitivity Analyses,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20021–20035, Nov. 2022.