Rochester Institute of Technology of Dubai

TigerBoard

Spring 2026

Prasanth Sathiyanarayanan(749002698),Manognya Kumar Dandala(412002881), Mohammed Baahir Yusuf(744005062), Hussain Aliasagar Dahodwala(418008681)

Executive Summary:

TigerBoard is a Commuter Management System designed to optimize RIT's commuting experience for both students and the drivers in RIT Dubai. The web application streamlines communication between commuter students and transportation services to a dedicated portal where both the students and drivers can manage their plans and commuting trips without any ambiguities or miscommunications that lead to unexpected delays or latencies for the trips.

Tigerboard helps students by providing real-time dashboards that display bus availability based on their registered commuter plan (morning-only, evening-only, or round-trip) and RITD-assigned schedules. Students can choose and book buses that align with their personal plans and academic schedule, reducing confusion and the risk of missing buses due to unawareness or last-minute changes. This approach ensures flexibility, transparency, and easy access to accurate commuting information.

For drivers, TigerBoard significantly reduces the redundant workload of tracking individual students before departure from RITD or at various checkpoints. Through the driver portal, drivers can review their daily trips and view a confirmed list of students for each trip, allowing them to depart on time without waiting for students who do not intend to use the bus.

Hence, this system minimizes delays and latency, improves time management, and reduces stress on both ends, resulting in a smoother, more reliable commuting experience for the entire commuting community in RITD.

Project Scope and Timeline:

Scope:

- Role Based Access Control: User Login and Registration Landing Page
- Booking Management System
- Dynamic Fleet Optimization
- Integration with Centralized External Database
- Responsive UI and Intuitive UX Dashboard for Students and Drivers

Timeline:

Week 5	Brainstorming UI and UX design using Figma or other software.
Week 6	Implementation of Basic Project Structure layout and Configuration of teammates local environments.
Week 7	Develop the Backend with REST APIs
Week 8	build the Student Dashboard and Driver Portal using HTML/CSS and Implement Javascript forms for user login and trip searching.
Week 10	Connect the frontend forms to the backend using JSON
Week 11-12	Design the MySQL database and implement the database.
Week 13	Implement Spring Security
Week 14	Perform end-to-end testing, record the demo video, and prepare the final presentation.

Timeline subjected to change according to course content, assessments and deadlines.

Key features:

1. Role Based Access – Student, Driver and Supervisor.
2. Driver and Student Dashboard
3. Responsive UI Design – Mobile, Desktop.
4. Cancel and Book buses (only Students)
5. Initiative Notifications for Students and Drivers
6. Automatic Attendance using QR codes (Verification)
7. Driver to Driver commuters Transferal

User / interaction scenario(s):

Student Scenario: After login, students access a personalized dashboard showing available buses, RIT-recommended bus, current allocation, commute history, and profile. Students receive a notification with bus number and timing, then confirm whether they will use the allocated bus. If yes, they scan a QR code while boarding to mark attendance as *present*. If no, they may select another bus and repeat the process. By default, students are marked as *going* to enhance user experience.

Driver Scenario: Drivers log in to a dashboard displaying current students, transfer options, supervisor contact, and profile. Drivers monitor student status and depart once all *going* students are marked *present*, or after a specified waiting time set by RIT. If three or fewer students board, drivers may request to transfer them to a similar-route bus. Drivers can also accept new students and adjust routes to reduce overcrowding.

Supervisor Scenario: Supervisors access bus schedules, active and past trips, and student statuses (*going* or *present*). They receive driver messages regarding trip cancellations or transfer requests and must approve or deny them according to RIT policies.

Implementation details:

- Implementing MVC architecture to build our Web Application for simplicity and efficient code management and maintenance.
- Use of HTML, CSS (Moustache/TailWind/Bootstrap) for structuring and styling our Web design and ensuring its responsiveness to different view ports.
- JavaScript for frontend processing and communication with the main java server.
- Java (Spring Boot) to Handle User Requests, implement Business data logic, and interact with the Database.
- MySQL (JPA) for User data retrieval and storage.
- REST APIs for Frontend-Backend communication.
- Intellij IDE and GitHub for Version Control

Anticipated challenges / limitations / mitigations

Challenges: Complex booking logic, late database implementation, version control, and lack of Time.
Limitations: No GPS tracking, unexpected commute incidents or traffic, and Internet dependent.
Mitigations: Creating a plan before implementing booking logic with the help of the instructor, early creation of an Data Model for the database and effective coordination between team members.