DIU Bus Management System

Course Title : Software Project VI

Course Code : CSE-336

Section : 60-A

Submitted By

Name	ID
Fatema Akter	213-15-4544
Kamruzzaman Hridoy	213-15-4565
Sanjeda Nasrin Tosiba	213-15-4344

Submitted To

Eng. Golam Rabbany

Department of Computer Science and Engineering

Daffodil International University

Submission Date: 2-12-2024

Course Teacher: Eng.Golam Rabbany
Course Title: Software Project VI
Course Code: **CSE336**

Dear Sir,

Re: **Enclosed Application Agreement for DIU Bus Management Web Application System.**

As students, we know that creating a client-oriented application requires a mixture of technical excellence and clear communication. Our firm hires only the very best to ensure you receive both. We know that every client is unique and we strive to deliver an individual, innovative, and affordable proposal every time and to follow it through with an outstanding delivery that is both on time and within budget.

According to your business requirements, we designed the DIU Bus Management Web Application System for you. In this agreement paper, we describe each part and all possible features of our Application. Please read carefully, and if it satisfies you, then sign and confirm the agreement.

Yours Truly,
Fatema Akter
ID:**213-15-4544**
Section:**60-A**
Daffodil International University

DIU Bus Management System

1. Project Overview

- **Purpose**: To develop an automated bus management system for DIU, offering seamless management of routes, schedules, and user interactions.
- **Scope**: The system will include modules for Admins, Drivers, and Passengers, as well as a Default View for non-logged-in users. Core functionalities include real-time tracking, ticketing, payments, and schedule optimization.
- **Definitions and Acronyms**: BMS (Bus Management System), HTML, CSS (User Interface), API, NodeJS(Application Programming Interface).

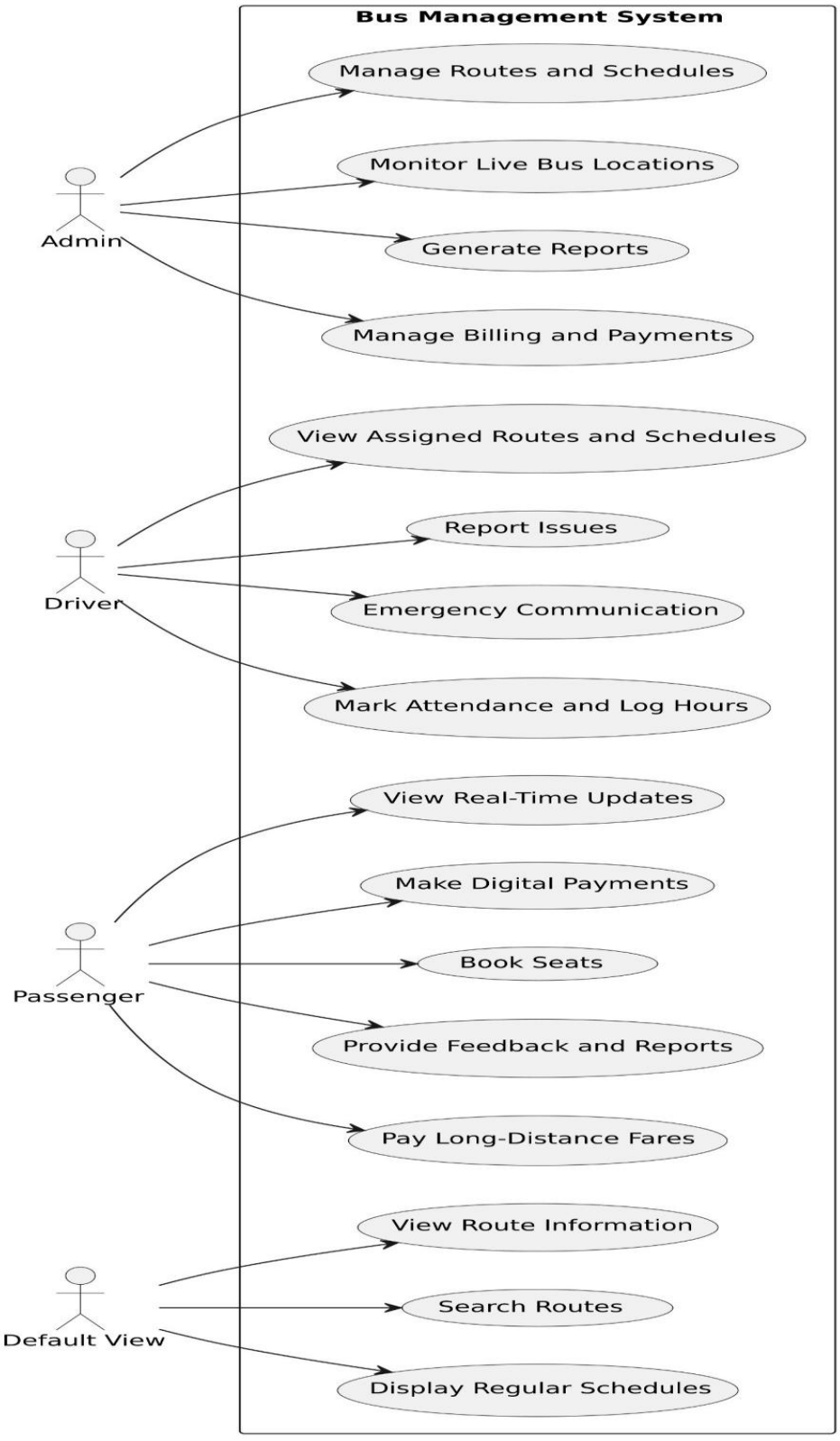
2. Overall Description

- **Product Perspective**: The BMS combines mobile and web-based solutions. It interacts with backend services to manage data such as SIGN-UP, LOG-IN
- **Product Functions**: Route and schedule management, real-time bus tracking, user ticketing, digital payment processing, and feedback collection.

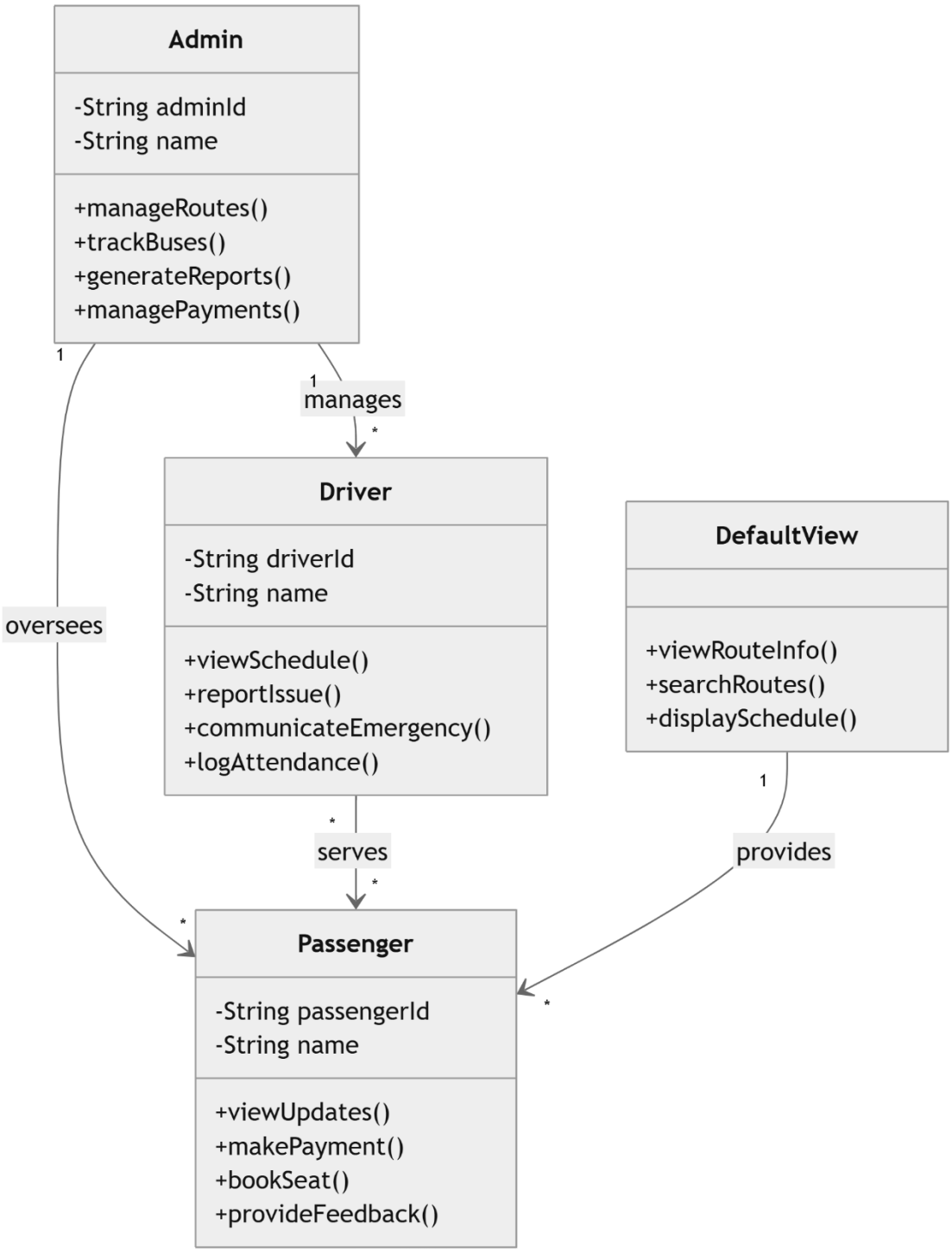
3. System Features and Characteristics

1. **Admin Module**:
    - Sign up on the website as an admin.
    - Real-time tracking of buses and data analytics.
    - Billing and payment management.(in future)
2. **Driver Module**:
    - Access to assigned routes and schedules.
    - Tools for issue reporting and emergency communication.
    - Automated attendance and payroll.
3. **Passenger Module**:
    - Live updates on bus location, schedule, and capacity.
    - Online payments and ticketing.
    - Seat booking and issue reporting.
4. **Default View**:
    - Basic access to schedules and route information for general users

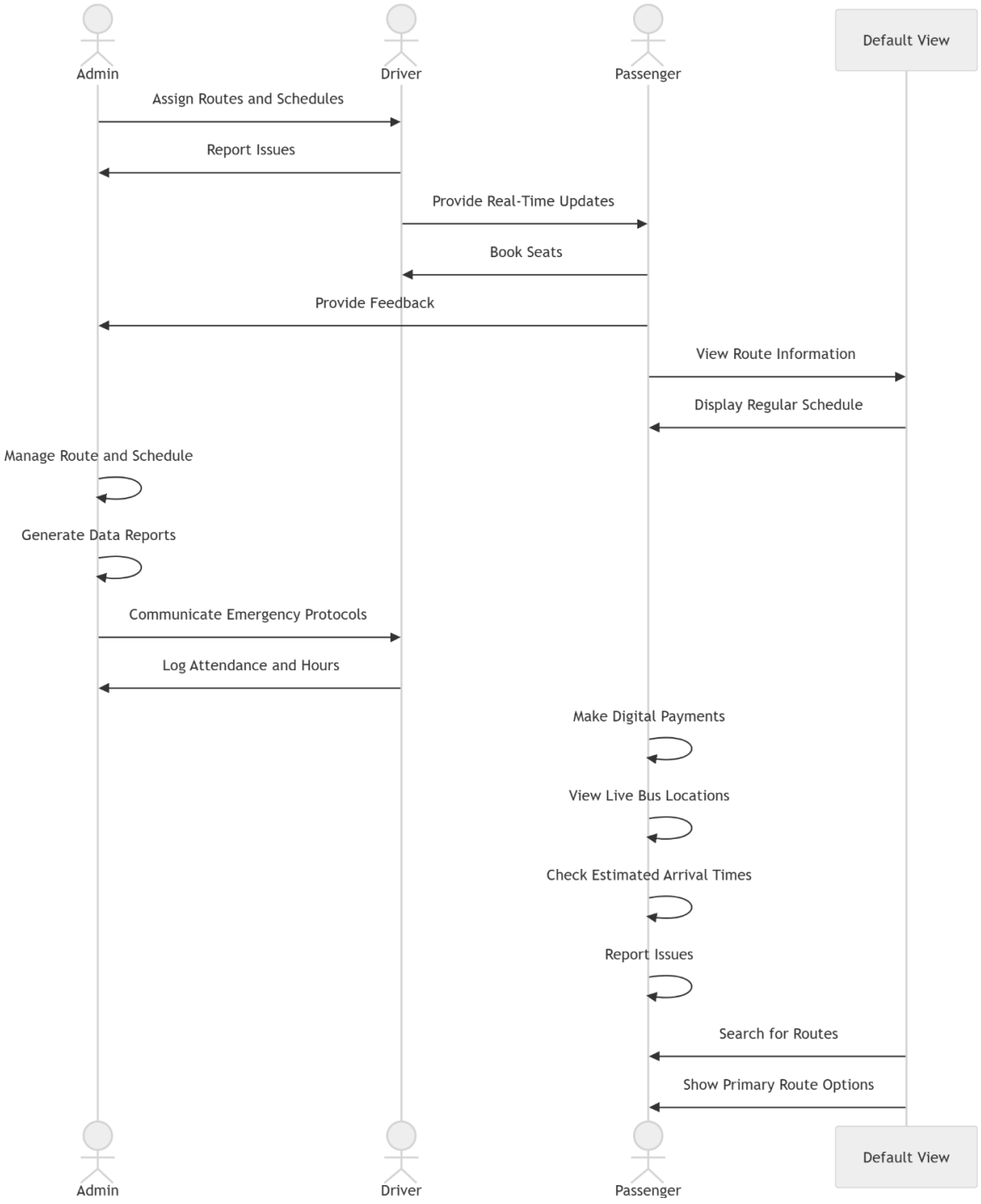
4.Use Case Diagram:



5.Unified Modeling Language:



6.Activity Diagram



7. Milestones and Reporting

Here's the **timing workflow** for the DIU Bus Management System (BMS) project, starting on **June 1, 2024**, and ending on **December 10, 2024**:

---

1. Requirement Gathering and Analysis

- **Duration**: 2 weeks (June 1, 2024 – June 14, 2024)
- **Tasks**:
  - Conduct stakeholder meetings to understand project requirements.
  - Document all functional and non-functional requirements.
  - Prepare initial wireframes for user interfaces.

---

2. Project Planning

- **Duration**: 1 week (June 15, 2024 – June 21, 2024)
- **Tasks**:
  - Create a detailed roadmap with milestones and deadlines.
  - Assign responsibilities to team members.
  - Plan resource allocation, including tools, budget, and personnel.

---

3. System Design

- **Duration**: 3 weeks (June 22, 2024 – July 12, 2024)
- **Tasks**:
  - Design the system architecture, including backend, frontend, and database interaction.
  - Finalize the technology stack (e.g., Flutter, Node.js).
  - Develop and finalize the UI/UX design for the system.

---

4. Development Phase

- **Duration**: 8 weeks (July 13, 2024 – September 6, 2024)

- **Tasks**:
    - **Frontend Development** (4 weeks, July 13, 2024 – August 9, 2024):
        - Build the interface by using HTML, CSS, and JavaScript.
        - Ensure responsiveness and implement state management.
    - **Backend Development** (4 weeks, August 10, 2024 – September 6, 2024):
        - Develop APIs for users sign-up and login.
        - Integrate the database for secure and scalable data storage.
        - Incorporate third-party APIs for payments and GPS tracking.

---

5. Integration and Testing

- **Duration**: 4 weeks (September 7, 2024 – October 4, 2024)
- **Tasks**:
    - Integrate frontend and backend components.
    - Perform unit testing for individual modules and features.
    - Conduct system testing to ensure the software meets all requirements.
    - Execute end-to-end testing to simulate real-world scenarios.

---

6. Deployment

- **Duration**: 2 weeks (October 5, 2024 – October 18, 2024)
- **Tasks**:
    - Set up production servers and deploy the application.
    - Implement CI/CD pipelines for smooth updates and maintenance.
    - Perform final system testing in the live environment.

---

7. User Training and Documentation

- **Duration**: 2 weeks (October 19, 2024 – November 1, 2024)
- **Tasks**:
    - Conduct training sessions for Admins, Drivers, and other stakeholders.
    - Prepare user manuals and API documentation.
    - Develop a knowledge base for troubleshooting.

---

8. Feedback and Refinement

- **Duration**: 2 weeks (November 2, 2024 – November 15, 2024)
- **Tasks**:
  - Gather feedback from a test group of users.
  - Address reported issues and optimize performance.
  - Perform bug fixes and implement minor enhancements.

---

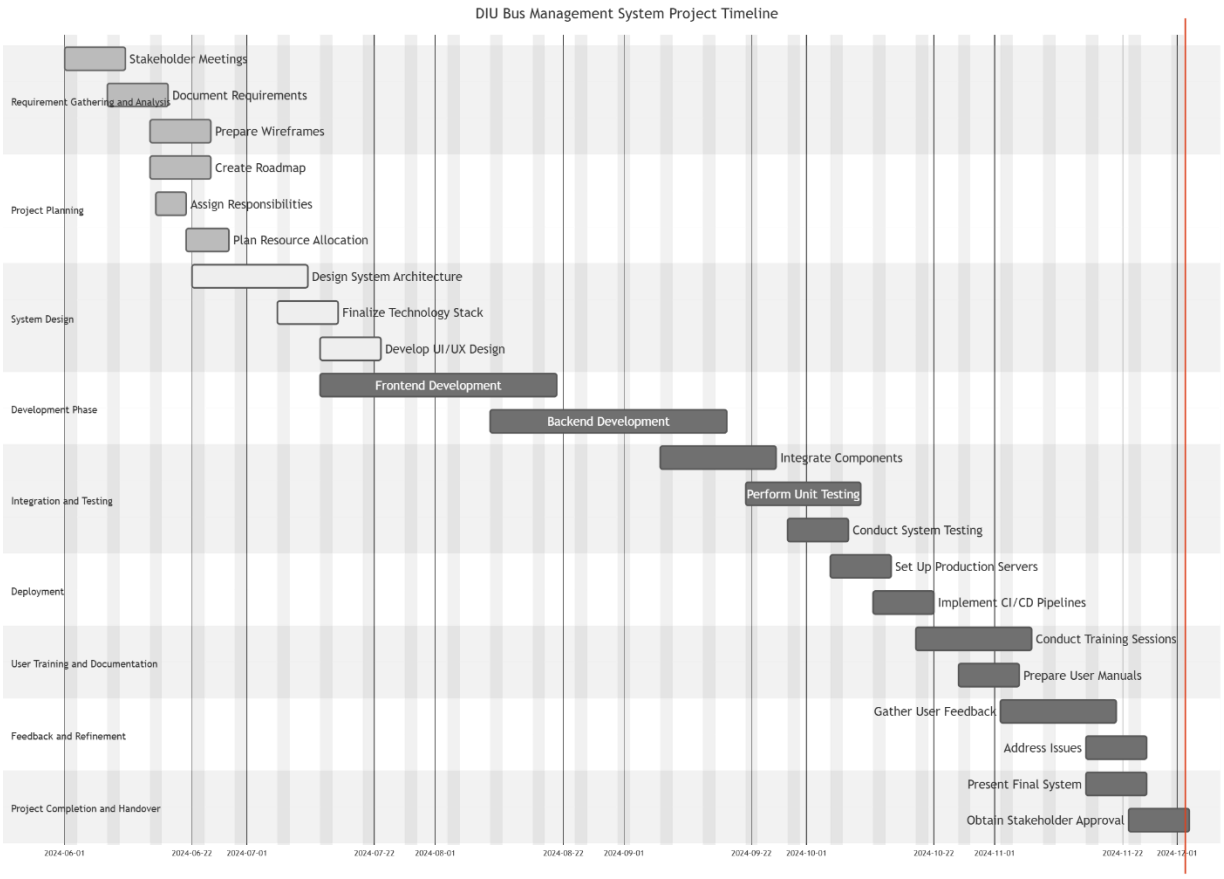
9. Project Completion and Handover

- **Duration**: 1 week (November 16, 2024 – December 10, 2024)
- **Tasks**:
  - Present the final system to DIU stakeholders.
  - Obtain stakeholder approval and sign-off.
  - Officially hand over the system along with documentation.

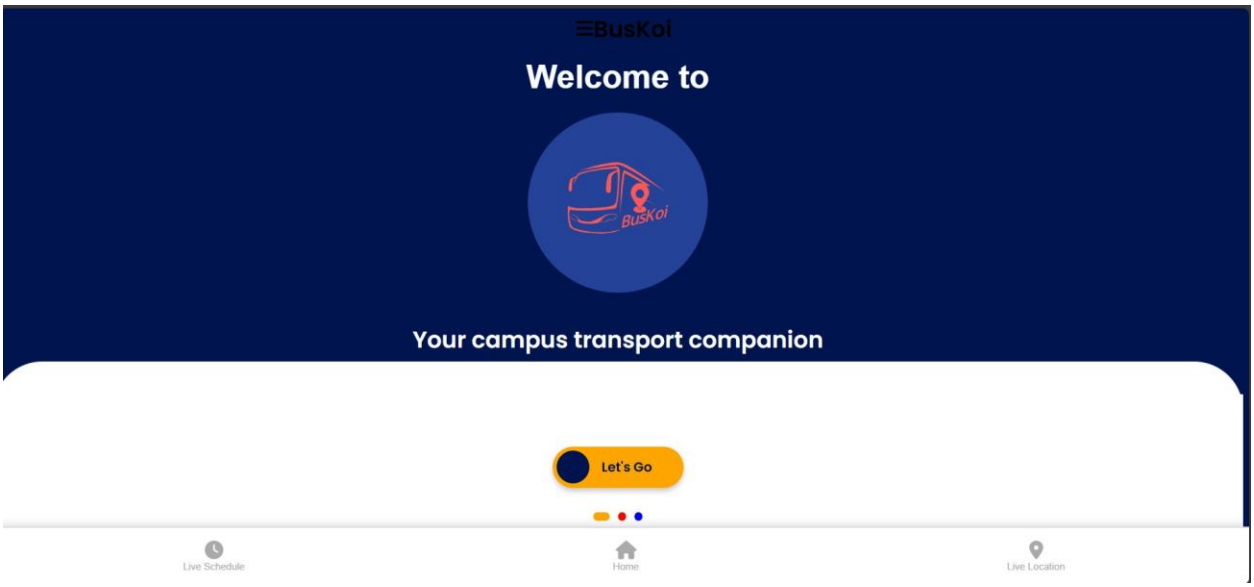
Summary

- **Start Date**: June 1, 2024.
- **End Date**: December 10, 2024.
- **Total Duration**: 27 weeks (approximately 6.5 months).

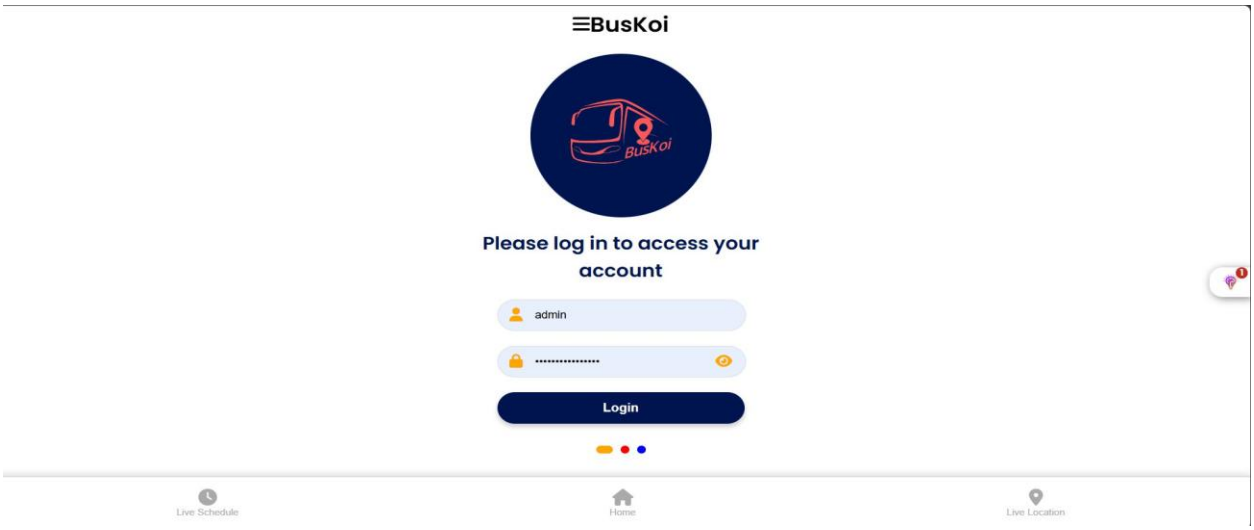
8. Timeline

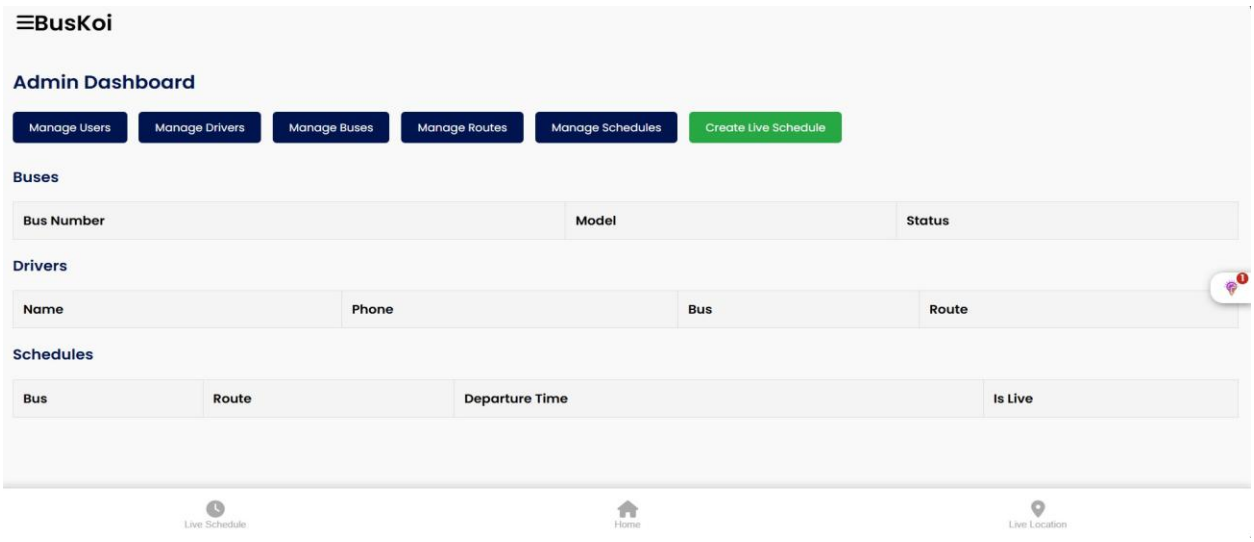


9. Image of our Website




BusKoi
Home to University
University to Home
Search your area
Live Schedule
Home
Live Location


BusKoi
BusKoi
Please log in to access your account
admin
Login
Live Schedule
Home
Live Location


BusKoi
Admin Dashboard
Manage Users
Manage Drivers
Manage Buses
Manage Routes
Manage Schedules
Create Live Schedule
Buses
Bus Number
Model
Status
Drivers
Name
Phone
Bus
Route
Schedules
Bus
Route
Departure Time
Is Live
Live Schedule
Home
Live Location

10. Our website full UI/UX design link:

https://www.figma.com/design/jSlwPl00Um9JOyUPl5hYBg/Bus?node-id=0-1&m=dev&t=9Fw5gRjxJ2jB3v9w-1

11. Workflow for DIU Bus Management System (BMS)

Admin Workflow

1. **Login/Registration**
    - Admins log in to the system using secure credentials.
    - Role-based authentication ensures access to only admin-specific functionalities.
2. **Route and Schedule Management**
    - Add, edit, or delete bus routes.
    - Assign drivers to specific routes and set schedules.
    - Make adjustments for special occasions (e.g., exams, events).
3. **Real-Time Tracking and Alerts**
    - View live bus locations on a map interface.
    - Receive alerts for deviations, breakdowns, or delays.
4. **Data Analysis and Reports**
    - Generate reports on metrics such as bus occupancy, punctuality, and route efficiency.
    - Use insights to optimize schedules and routes.

Driver Workflow

1. **Login and Route Access**
    - Drivers log in to the app to view their assigned routes and schedules.
2. **Route Navigation**
    - Access route details with turn-by-turn navigation.
    - View real-time traffic updates and suggested alternative routes.
3. **Issue Reporting**
    - Use the app to report breakdowns, delays, or capacity issues directly to Admins.
4. **Attendance and Payroll Management**
    - Mark attendance automatically through GPS-based tracking when starting a route.
    - Log working hours, which are integrated with the payroll system.
5. **Emergency Communication**

- Notify Admins instantly in case of emergencies, such as accidents or technical issues.

Passenger Workflow

1. **Registration/Login**
   - Passengers create an account or log in with their credentials.
   - Student or faculty ID verification is completed during registration.
2. **Viewing Schedules and Routes**
   - Access live schedules and route details via the mobile app.
   - Use filters to search for specific routes or buses.
3. **Real-Time Updates**
   - Receive live updates on bus location, estimated arrival time, and available seats.
   - Notifications are sent when the bus is approaching the passenger's location.
4. **Ticket Booking and Payment**
   - Book seats for specific routes in advance, especially during peak times.
   - Pay fares digitally via integrated platforms like Bkash, Rocket, or OneCard.
   - Tickets are displayed in the app for quick verification.
5. **Feedback and Issue Reporting**
   - Provide feedback on driver behavior, bus cleanliness, and schedule adherence.
   - Report issues such as overcrowding, unsafe driving, or broken amenities.

Default View Workflow

1. **Route Information**
   - Users without login credentials can view general route information.
2. **Search Functionality**
   - Use a search bar to filter routes by name or destination.
3. **Regular Schedule Display**
   - View regular schedules and expected bus timings for specific routes.

Technology Stack

- **Frontend**: Flutter for mobile apps, React (or similar framework) for the web dashboard.
- **Backend**: Node.js with Express for APIs.

- **Database**: XAMPP for scalable and efficient data storage.
- **Payment Integration**: Bkash, Rocket, or OneCard API for digital payments.
- **Real-time Updates**: WebSockets for live tracking and notifications.

12.Development Process

1. **Initial Planning**:
    - Requirement gathering and stakeholder interviews.
    - System architecture and UI/UX design in Figma.
2. **Frontend Development**:
    - Flutter mobile app for passengers and drivers.
    - Admin dashboard with React components.
3. **Backend API Development**:
    - APIs for schedules, tracking, payments, and notifications.
4. **Testing**:
    - Unit testing, integration testing, and user acceptance testing.
5. **Deployment**:
    - Pilot launch followed by full-scale implementation.
6. **Support and Maintenance**:
    - 24/7 technical support and regular updates.

Technology Stack

- **Frontend**: Flutter for mobile apps, React (or similar framework) for the web dashboard.
- **Backend**: Node.js with Express for APIs.
- **Database**: MongoDB for scalable and efficient data storage.
- **Payment Integration**: Bkash, Rocket, or OneCard API for digital payments.
- **Real-time Updates**: WebSockets for live tracking and notifications.

13. Testing Process

To ensure the DIU Bus Management System operates smoothly and efficiently, the following testing processes are employed:

1. **Unit Testing**
    - Test individual modules (e.g., route management, payment processing, live tracking) to ensure they function as expected.
    - Tools: Jest for mobile and web applications, Mocha for backend APIs.

2. **Integration Testing**
    - Test interactions between the Admin, Driver, and Passenger modules.
    - Validate data flow between the frontend and backend through APIs.
3. **System Testing**
    - Test the entire system to verify compliance with requirements.
    - Ensure seamless operation under real-world conditions.
4. **Performance Testing**
    - Simulate high-traffic scenarios to test the system's responsiveness and reliability.
5. **User Acceptance Testing (UAT)**
    - Engage students, faculty, staff, and drivers in real-world use to identify usability issues and gather feedback.
6. **Regression Testing**
    - Test updates and bug fixes to ensure existing functionality remains unaffected.

14. Support and Maintenance

To ensure the system remains reliable and user-friendly:

1. **Support Services**
    - **Technical Support**: 24/7 helpdesk for resolving user queries and issues.
    - **User Training**: Tutorials and guides for Admins, Drivers, and Passengers.
2. **Maintenance Plan**
    - **Bug Fixes**: Regular updates to address software bugs and issues.
    - **Feature Enhancements**: Continuous addition of new features based on user feedback.
    - **Performance Optimization**: Regular monitoring and optimization for speed and efficiency.
3. **Monitoring**
    - Real-time system monitoring for downtime or errors.
    - Automated alerts for immediate resolution of critical issues.

15. Pricing and Payment Terms

- **Total Project Fee**: BDT 15,00,000
- **Payment Schedule**:
    1. **20%** upon completion of requirement analysis.
    2. **40%** upon delivery of the initial development phase and UI/UX design.
    3. **30%** after testing and UAT completion.
    4. **10%** upon final deployment.

16. Roles and Responsibilities

1. **Client (DIU Administration)**
    - Provide requirements and feedback during the project lifecycle.
    - Facilitate testing by providing real-world scenarios and user data.
2. **Project Manager**
    - Oversee the entire development process and ensure timely delivery.
3. **Frontend Developer**
    - Build and maintain mobile and web interfaces for Admins, Drivers, and Passengers.
4. **Backend Developer**
    - Develop APIs for real-time tracking, payment integration, and data storage.
5. **QA Engineer**
    - Conduct thorough testing to identify and resolve bugs.
6. **System Admin**
    - Monitor server infrastructure and ensure system uptime post-deployment.

17. Contact Us

For queries or support, please contact:

- **Email**: fatema@diu.edu.bd
- **Phone**: +8801626410045
- **Location**: Daffodil International University, Main Campus, Dhaka, Bangladesh