CS19442 - SOFTWARE ENGINEERING CONCEPTS

BUS MANAGEMENT SYSTEM

Name	:	**Rethinaath S**
Year/Branch/Section	:	**II / CSE / D**
University Register No.	:	**2116220701222**
College Roll No.	:	**220701222**
Semester	:	**IV**
Academic Year	:	**2023-2024**

OVERVIEW OF THE PROJECT

PROBLEM STATEMENT:

Colleges and universities struggle to manage their bus fleets effectively, causing disruptions in student transportation. The main reasons for this inefficiency include:

- Lack of a centralized system: Important information is scattered, making it difficult to communicate effectively and manage resources efficiently.
- Ineffective communication: Delays and schedule changes are not communicated promptly to students, leading to frustration and missed classes.
- Potential safety concerns: Bus location and maintenance schedules might not be well-tracked, raising safety issues.
- Ineffective resource management: Without proper monitoring of bus use and location, resource allocation for routes and maintenance is difficult to optimize.

SOLUTION: COLLEGE BUS MANAGEMENT SYSTEM

A College Bus Management System can address these issues by providing a centralized platform for managing all aspects of the college bus fleet. Here's how it can improve efficiency and safety:

- Improved communication and transparency: Real-time bus tracking and schedule updates can be communicated to students through a mobile app or web interface.
- Enhanced safety: Features like GPS route tracking and automated maintenance scheduling can help ensure student safety and prevent breakdowns.
- Optimized resource allocation: By monitoring bus location and ridership, the system can identify underutilized or overcrowded routes, allowing for adjustments to optimize resource allocation.
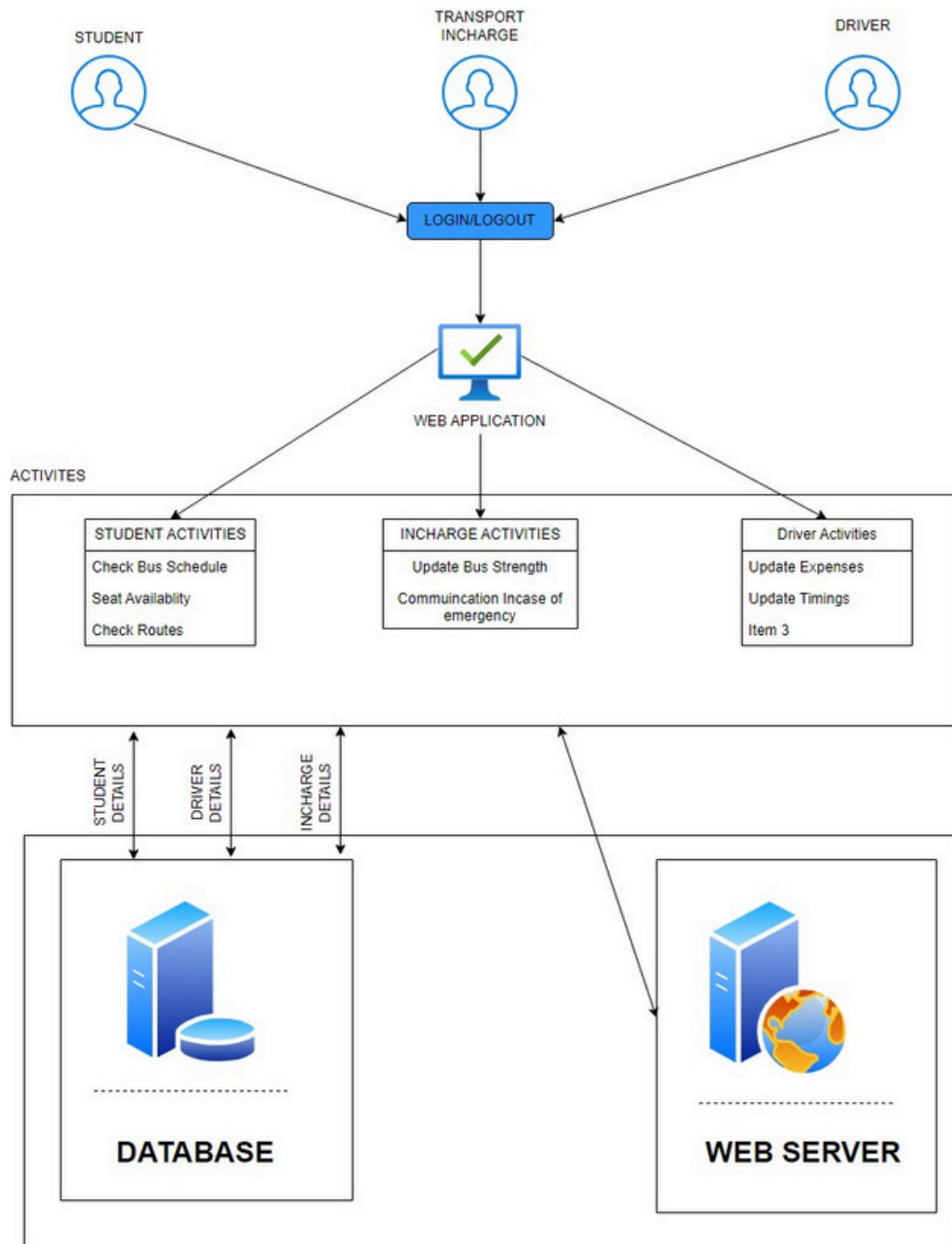
BENEFITS:

Implementing a College Bus Management System can result in several significant benefits:

- Improved efficiency in bus operations: Streamlined communication, route optimization, and maintenance scheduling can lead to a more efficient transportation system.
- Enhanced student safety and well-being: Real-time tracking and improved maintenance can promote a safer riding experience for students.
- Reduced transportation disruptions: Timely communication regarding delays or changes can minimize disruptions for students.
- Informed decision-making for resource allocation: Data collected by the system can be used to make data-driven decisions about fleet management and resource allocation.

BUSINESS ARCHITECTURE DIAGRAM

STUDENT

TRANSPORT INCHARGE

DRIVER

LOGIN/LOGOUT

WEB APPLICATION

ACTIVITES

STUDENT ACTIVITIES	INCHARGE ACTIVITIES	Driver Activities
Check Bus Schedule	Update Bus Strength	Update Expenses
Seat Availablity	Commuincation Incase of emergency	Update Timings
Check Routes		Item 3

STUDENT DETAILS

DRIVER DETAILS

INCHARGE DETAILS

DATABASE

WEB SERVER

Backend Overview

USER STORIES

- As a student, I want to view the bus routes, bus stops, bus schedules.
- As a student, I want to be notified about my bus schedule, delays, or cancellations so that I am informed of any changes.
- As a student, I want to provide feedback and report issues about my journey so that the college can improve its bus services.
- As a bus driver, I want to update fuel expenses and repair expenses.
- As a bus incharge, I want to allocate seat to the students, if not available, I want to communicate with other bus incharges to make everyone seated.
- As a Bus Incharge, I want to provide feedback and report issues about my journey so that the college can improve its bus services.
- As a Bus Incharge, I want to track driver performance and punctuality so that I can address any issues and improve service reliability.
- As an admin, I want to give access to other users and can overview the functioning of the system.
- As an admin, I should be able to create, delete, update, retrieve data on the website.
- As a Bus incharge, I want to communicate with the admin in case of any emergency situation.

NON-FUNCTIONAL REQUIREMENTS

PERFORMANCE:

- Response Time: The system should respond to user actions within 2 seconds for 95% of the requests.
- Scalability: The system must handle up to 1000 concurrent users without performance degradation.

RELIABILITY:

- Availability: The system should be available 99.9% of the time, ensuring minimal downtime.
- Data Integrity: The system must ensure data integrity, with no data loss or corruption during transactions.

SECURITY:

- Authentication: All users must authenticate using secure login mechanisms (e.g., multi-factor authentication).
- Authorization: Role-based access control must be implemented to restrict access to specific features based on user roles.
- Data Encryption: Sensitive data must be encrypted both in transit (using SSL/TLS) and at rest.

POKER PLANNING ESTIMATE

VIEW BUS ROUTES, STOPS, SCHEDULES:

- Estimate: 5
- Reason: This involves querying the database, presenting data in an organized manner, and possibly adding filters for better user experience.

NOTIFICATIONS ABOUT SCHEDULE CHANGES:

- Estimate: 8
- Reason: This requires real-time notifications, which might involve integrating with external services like SMS or email gateways, and ensuring reliable and timely delivery of notifications.

PROVIDE FEEDBACK AND REPORT ISSUES:

- Estimate: 5
- Reason: This requires creating a feedback form, storing the feedback in the database, and possibly some analytics to track and manage feedback.

UPDATE FUEL AND REPAIR EXPENSES:

- Estimate: 3
- Reason: This involves creating forms for input, validation, and storing the data in the database.

ALLOCATE SEATS AND COMMUNICATE WITH OTHER INCHARGES:

- Estimate: 13
- Reason: This involves seat allocation logic, handling edge cases where seats are not available, and a communication system between incharges.

INCHARGE FEEDBACK AND ISSUE REPORTING:

- Estimate: 5
- Reason: Similar to the student feedback, but might have additional details specific to incharge requirements.

TRACK DRIVER PERFORMANCE AND PUNCTUALITY:

- Estimate: 8
- Reason: This requires collecting data on driver performance, creating reports, and possibly integrating with other systems for tracking punctuality.

ADMIN USER ACCESS AND SYSTEM OVERVIEW:

- Estimate: 8
- Reason: This involves user role management, permissions, and an overview dashboard to monitor system functions.

ADMIN CRUD OPERATIONS:

- Estimate: 8
- Reason: This requires a comprehensive CRUD interface, validation, and ensuring data integrity and security.

INCHARGE EMERGENCY COMMUNICATION WITH ADMIN:

- Estimate: 5
- Reason: This involves creating a communication channel, which could be as simple as messaging or as complex as integrating with an external communication tool.

MVC ARCHITECTURE DIAGRAM

MODEL:

- Bus (id, route, capacity, availability)
- Route (id, origin, destination, stops)
- Schedule (id, bus_id, route_id, departure_time, arrival_time)
- Passenger (id, name, contact_info)

VIEW:

- Bus List View (display all buses with their routes and availability)
- Route View (display route details with stops)
- Schedule View (display schedule for a specific bus or route)
- Passenger View (display passenger details)

CONTROLLER:

- BusController (handle bus-related operations like add, edit, delete)
- RouteController (handle route-related operations like add, edit, delete)
- ScheduleController (handle schedule-related operations like add, edit, delete)
- PassengerController (handle passenger-related operations like add, edit, delete)

DESIGN PRINCIPLES

SEPARATION OF CONCERNS:

The system is divided into distinct modules such as user management, bus scheduling, route management, and notifications. This modular approach facilitates efficient management and allows individual components to be updated or replaced without affecting the entire system.

HORIZONTAL AND VERTICAL SCALING:

The design accommodates an increasing number of users and buses without significant performance degradation. Scalable databases and cloud services have been incorporated to ensure the system can expand to meet the college's growing needs.
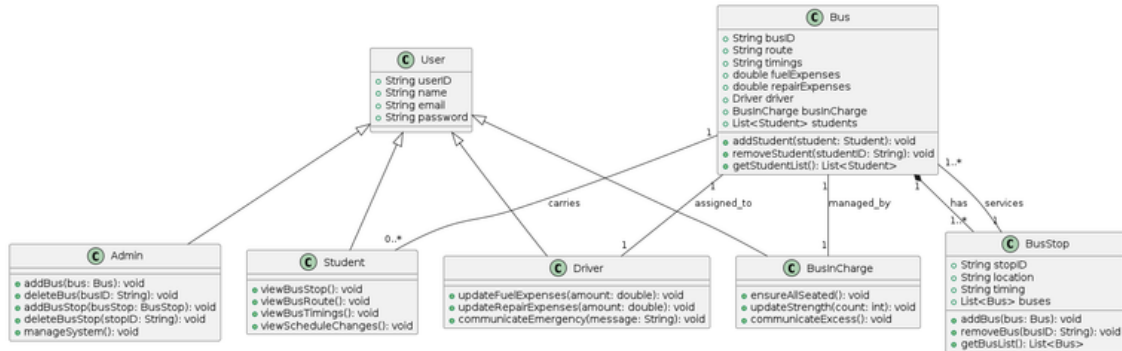
ROLE-BASED ACCESS CONTROL (RBAC):

Specific roles (students, bus in-charges, administrators, and bus drivers) with appropriate permissions have been defined. This ensures that each user has access only to the functionalities necessary for their role, enhancing security and usability.
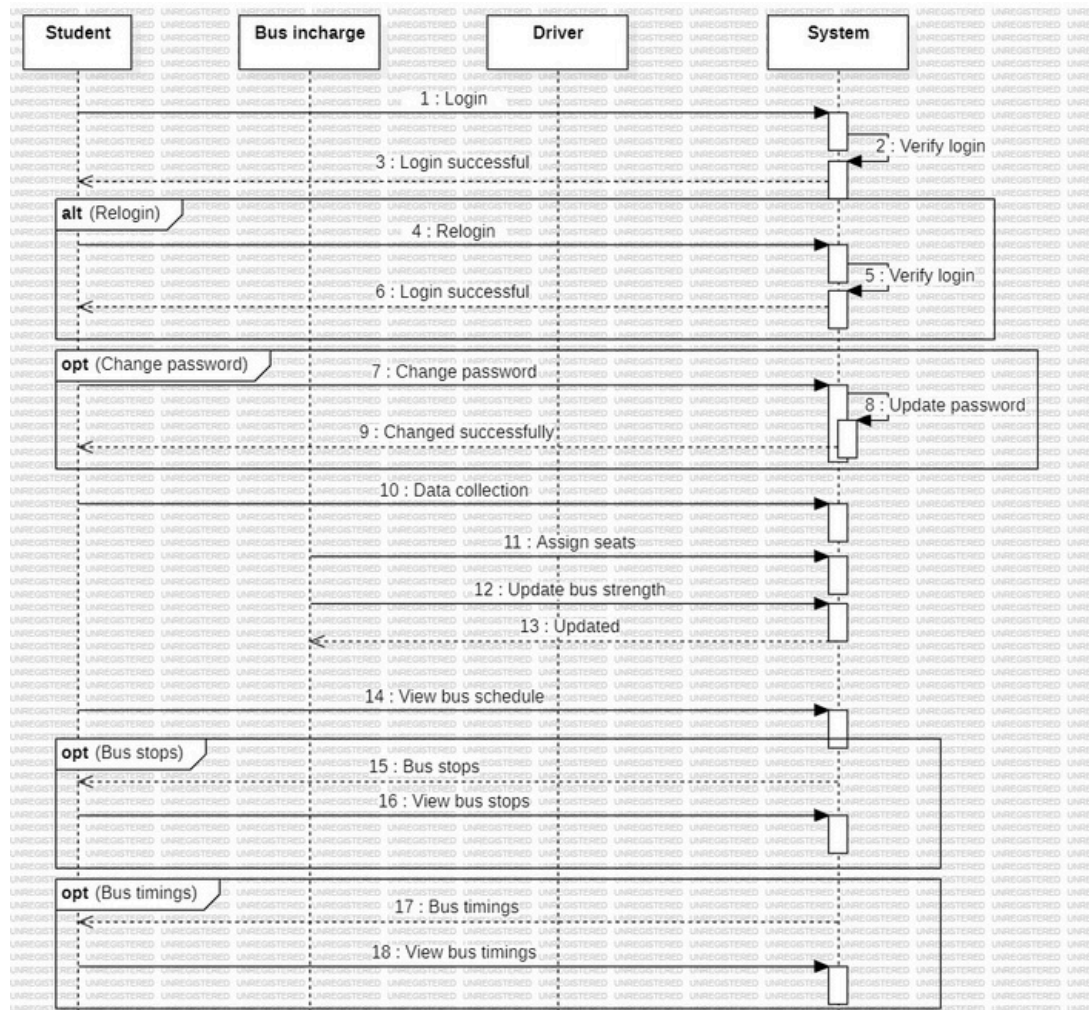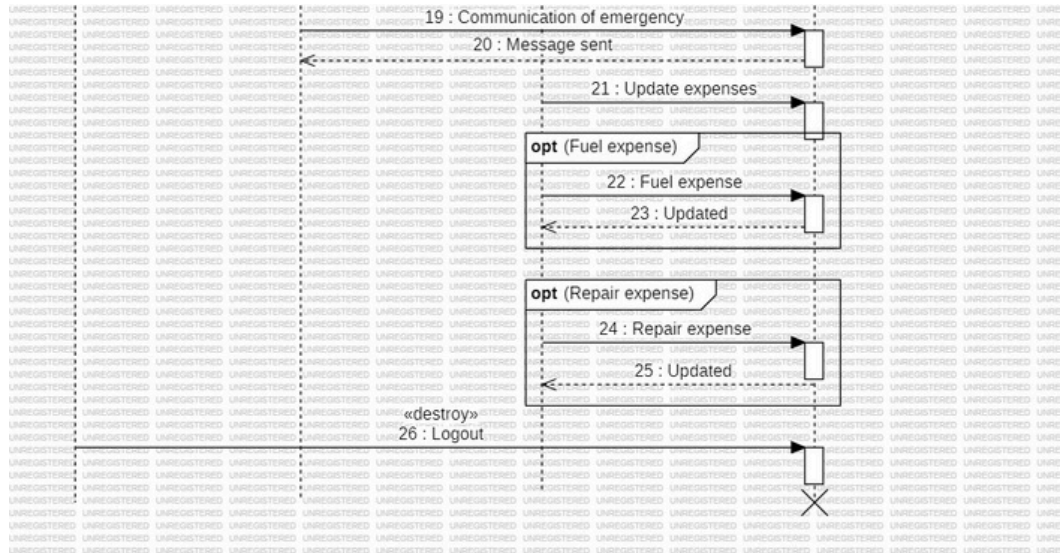
USER-FRIENDLY INTERFACE:

The interface is designed to be intuitive and easy to navigate. Responsive design principles have been employed to support various devices, including desktops, tablets, and smartphones.

CLASS DIAGRAM

SEQUENCE DIAGRAM

19 : Communication of emergency

20 : Message sent

21 : Update expenses

opt (Fuel expense)

22 : Fuel expense

23 : Updated

opt (Repair expense)

24 : Repair expense

25 : Updated

«destroy»
26 : Logout

TEST STRATERGY

UNIT TESTING

Scope: Individual components/modules
Responsibility: Developers
Tools: JUnit, pytest
Automation: High

INTEGRATION TESTING

Scope: Interaction between integrated components
Responsibility: QA team
Tools: Selenium, Postman
Automation: Medium

SYSTEM TESTING

Scope: Complete integrated system
Responsibility: QA team
Tools: JMeter, OWASP ZAP
Automation: Medium to High
Types: Functional, Performance, Security.

ACCEPTANCE TESTING

Scope: Validation against business requirements
Responsibility: QA team and stakeholders
Tools: Manual
Automation: Low

<u>TEST CASES</u>

1. User Story: As a student, I want to view the bus routes, bus stops, bus schedules.

Happy Path Scenario: Student views bus routes, bus stops, and schedules.

Test Case 1.1: View Bus Routes
- **Input:** Student navigates to the "Bus Routes" page.
- **Expected Output:** The list of bus routes is displayed.

Test Case 1.2: View Bus Stops
- **Input:** Student navigates to the "Bus Stops" page.
- **Expected Output:** The list of bus stops is displayed.

Test Case 1.3: View Bus Schedules
- **Input:** Student navigates to the "Bus Schedules" page.
- **Expected Output:** The list of bus schedules is displayed.

2. User Story: As a student, I want to be notified about my bus schedule, delays, or cancellations so that I am informed of any changes.

Happy Path Scenario: Student receives notifications about bus schedule changes.

Test Case 2.1: Receive Schedule Notification
- **Input:** Bus schedule is updated.
- **Expected Output:** Student receives a notification about the updated schedule.

Test Case 2.2: Receive Delay Notification
- **Input:** Bus delay is reported.
- **Expected Output:** Student receives a notification about the bus delay.

DEPLOYMENT ARCHITECTURE