SOFTWARE ENGINEERING CONCEPTS (LAB MANUAL)

OVER VIEW OF THE PROJECT:

Traffic Violation Challan Generator

Project Overview:

The Traffic Violation Challan Generator is a comprehensive system designed to streamline the process of issuing traffic violation challans (fines). This system aims to assist law enforcement agencies in efficiently managing and recording traffic violations, calculating fines, and communicating with violators. The system will capture relevant details such as the violator's name, vehicle number, and a list of violations. It will also integrate with a database for record-keeping and utilize messaging services to notify violators upon fine payment.

Key Features

User Interface for Data Entry:

Input Fields: Capture violator's name, vehicle number, and other relevant details.

Violation Selection: Display a comprehensive list of traffic violations for the officer to select from.

Fine Calculation: Automatically calculate the total fine based on selected violations.

Database Management:

Storage: Securely store details of each violation, including violator information, violations selected, and fine amounts.

Retrieval: Provide functionalities to retrieve and update records as needed.

Notification System:

Message Generation: Generate notifications for violators once the fine is paid.

Delivery: Send notifications via SMS or email to inform violators of their payment status.

Technical Specifications

Frontend: The user interface will be built using modern web technologies such as HTML, CSS, and JavaScript for an intuitive and responsive experience.

Backend: A robust backend using a framework like Django or Flask (Python), or Node.js, to handle the business logic and database interactions.

Database: A relational database such as MySQL, PostgreSQL, or SQLite to store violation records and user information.

Messaging Service: Integration with an SMS gateway or email service provider (e.g., Twilio, SendGrid) to send notifications to violators.

Workflow

Data Entry:

The police officer inputs the violator's name and vehicle number into the system.

A list of possible violations is displayed for selection.

Violation Selection and Fine Calculation:

The officer selects the relevant violations from the list.

The system calculates the total fine based on predefined charges for each violation.

Record Storage:

The details are stored in the database for future reference and tracking.

Notification:

Once the fine is paid, the system sends a notification to the violator confirming the payment and providing any further instructions if necessary.

Benefits

Efficiency: Streamlines the process of issuing and managing traffic violation fines, reducing paperwork and manual errors.

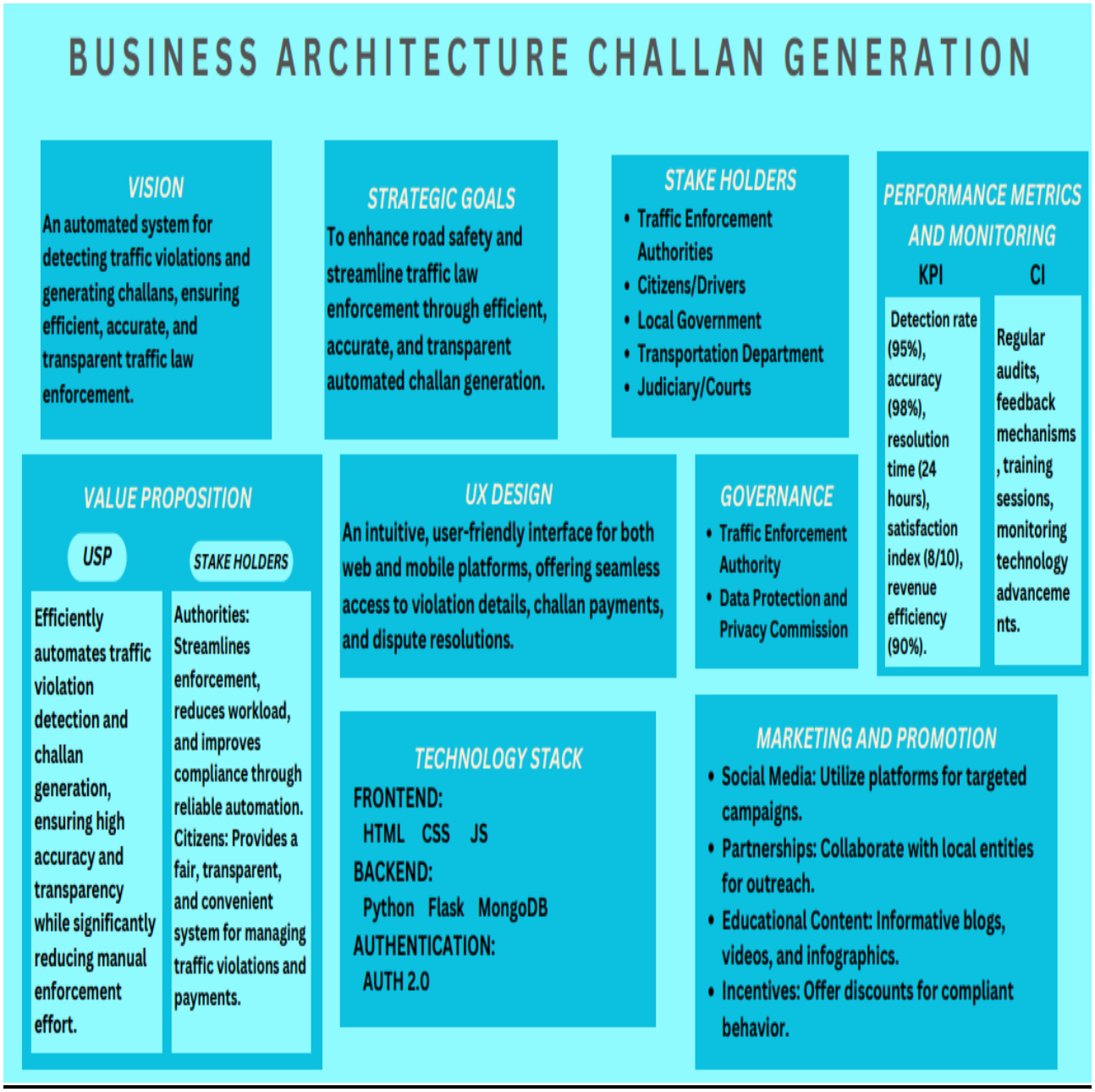
Transparency: Provides clear and transparent calculation of fines based on predefined rules.

Communication: Ensures violators are promptly informed of their payment status, enhancing compliance and reducing follow-up work for the police.

Conclusion:

The Traffic Violation Challan Generator is a crucial tool for modernizing traffic law enforcement. By automating the violation recording and fine management process, it enhances operational efficiency, ensures accurate record-keeping, and improves communication with violators. This project leverages technology to support law enforcement agencies in maintaining road safety and enforcing traffic laws effectively.

BUSINESS ARCHITECTURE DIAGRAM:



REQUIREMENTS AS USER STORIES:

1. As a traffic officer, I want the system to automatically detect traffic violations

So that I can ensure accurate and timely enforcement of traffic laws.

2. As a driver, I want to receive real-time notifications of my traffic violations

So that I can promptly address and pay the fines.

3. As a system administrator, I want to integrate AI-powered cameras and sensors

So that the system can efficiently detect and record traffic violations.

4. As a vehicle owner, I want to access a web portal and mobile app to view my violation history and challans

So that I can manage and resolve them conveniently.

5. As a citizen, I want a transparent process for disputing incorrect challans

So that I can contest them fairly with evidence.

6. As a data analyst, I want to generate reports on traffic violations and challan statistics

So that I can help improve traffic management and enforcement strategies.

7. As a local government official, I want to ensure the system complies with data privacy and security regulations

So that citizen information is protected.

8. As a traffic officer, I want to receive training on using the automated system

So that I can effectively operate and manage it.

As a user,

I want to make online payments for my traffic fines

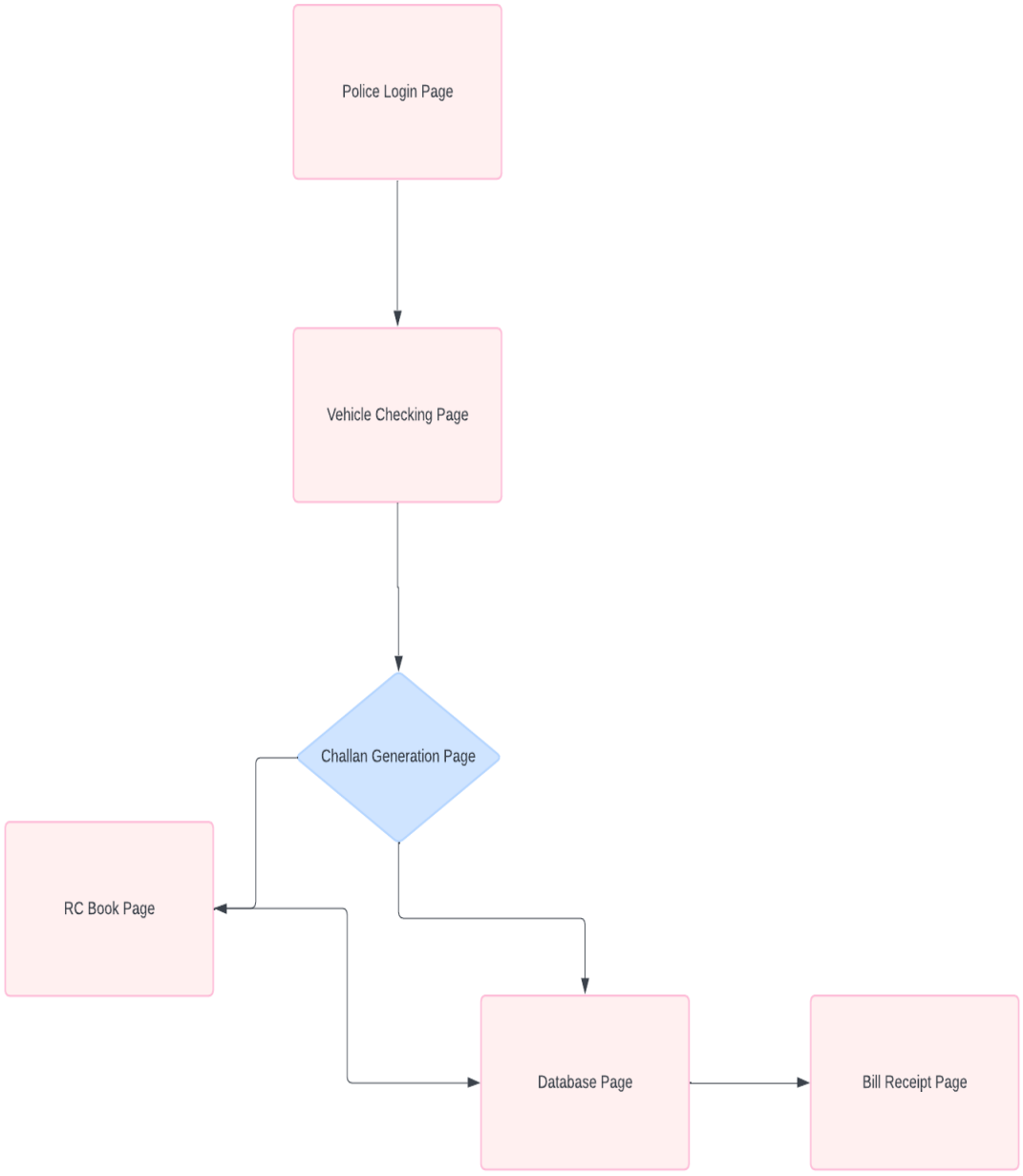
So that I can settle them conveniently without visiting an office.

As a city planner,

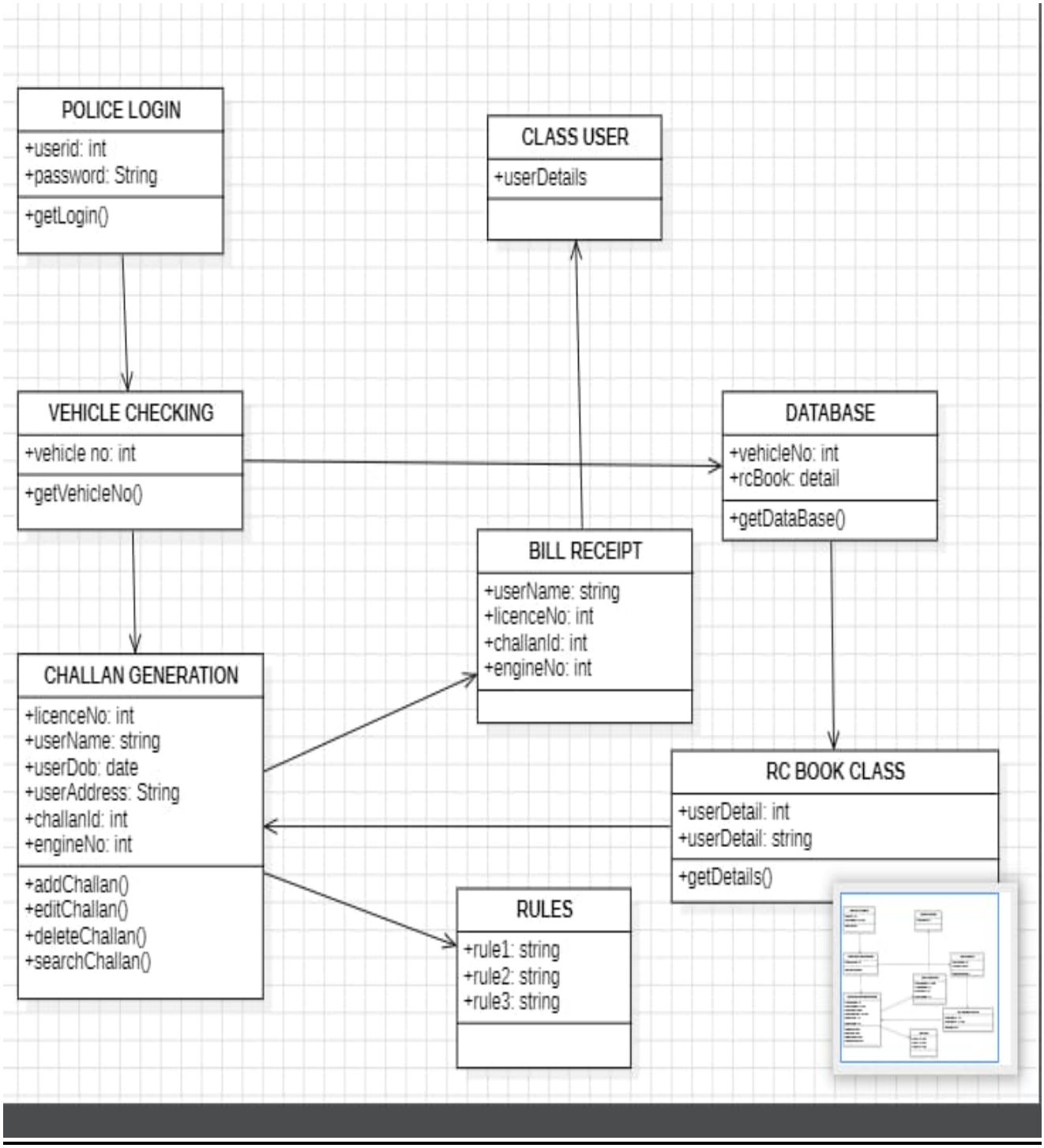
I want to analyze traffic violation data to identify high-risk areas

So that I can implement preventive measures to enhance road safety.

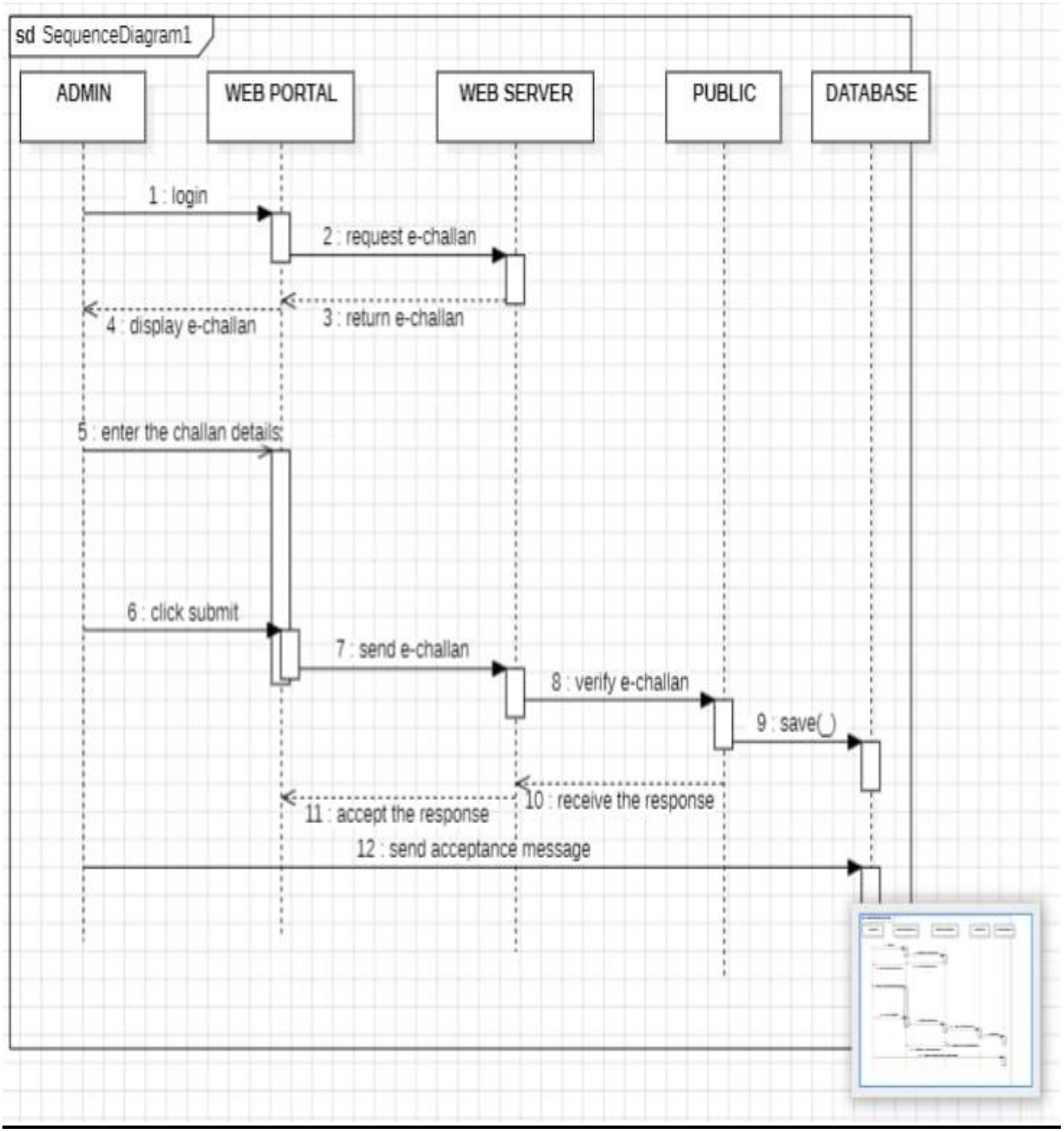
ARCHITECTURE DIAGRAM:



CLASS DIAGRAM:



SEQUENCE DIAGRAM:



DEPECTING THE TEST STRATERGY:

Test Strategy for E-Challan Generation System

The test strategy for the e-Challan generation system outlines the overall approach and methodologies used to ensure the system's functionality, reliability, and performance. It covers testing user stories, including happy path and error scenarios.

User Story 1: As a traffic officer, I want the system to automatically detect speeding and traffic violations so that accurate and timely enforcement of traffic laws can be ensured.

1.Happy Path:

Simulate a vehicle travelling at a speed above the set limit.

The system captures the vehicle's speed using the speed detection module.

The system verifies the speed against the permissible limit for the location.

Upon detecting a speeding violation, the system captures the vehicle's image and licence plate number.

Error Scenario:

Simulate a vehicle travelling within the speed limit.

The speed detection module incorrectly records the vehicle as speeding due to a sensor error.

The system captures the vehicle's speed and license plate number.

The system compares the recorded speed with the permissible limit and detects an erroneous speeding violation.

The system logs the error and does not generate a challan.

User Story 2:As a driver, I want to receive real-time notifications of my traffic violations so that I can promptly address and pay the fines.

2.Happy Path:

Simulate a traffic violation (e.g., speeding) by a registered vehicle.

The system detects the violation and captures relevant details (vehicle number, violation type, time, location).

The system generates a challan with the captured details.

The system sends a notification to the driver via SMS and email, including challan details and payment instructions.

Verify the driver receives the notification promptly.

Error Scenario:

Simulate a traffic violation by a vehicle with missing or incorrect contact details.

The system detects the violation and captures relevant details (vehicle number, violation type, time, location).

The system attempts to generate a challan and send a notification.

The system detects the incorrect or missing contact details.

The system logs the error and retries sending the notification.

If the retry fails, the system flags the issue for manual review and resolution.

User Story 3:

As a vehicle owner, I want to access a web portal and mobile app to view my violation history and challans so that I can manage and resolve them.

3.Happy Path:

Open the web portal or mobile app.

Log in using valid credentials (username and password).

View the list of past violations and challans with details .

Complete the payment process and verify the status update.

Error Scenario

Open the web portal or mobile app.

Attempt to log in using incorrect credentials.

Attempt to log in using valid credentials but with an inactive account.

User Story 4:

As a citizen, I want a transparent process for disputing an incorrect challan so that I can contest it fairly with evidence.

4.Happy Path:

Log in to the web portal or mobile app using valid credentials.

Select the incorrect challan from the list.

Providing necessary details.Upload supporting evidence (photos, videos, documents)..

Verify that the system confirms the submission and provides a reference number for tracking.

Error Scenario

Log in to the web portal or mobile app using valid credentials.

Select the incorrect challan from the list.

Fill in the dispute form with incomplete information.

Observe the system's response.

User Story 5: As a traffic officer, I want to receive training on using the automated system so that I can effectively operate it.

5.Happy Path:

Log in to the training portal using valid credentials.

Access the training dashboard and view available training modules.

Select and start a training module.

Verify completion status and score on the training dashboard.

Error Scenario

Log in to the training portal using valid credentials.

Midway through the training module, encounter a technical issue.

Attempt to refresh or navigate back to the training module.

GITHUB REPOSITORY STRUCTURE:

.github/: Contains GitHub-specific configurations such as issue templates and CI/CD workflows.

ISSUE_TEMPLATE/: Templates for bug reports and feature requests.

workflows/: CI/CD pipeline configurations.

docs/: Documentation for the project.

api/: API specifications.

user_stories/: Detailed user stories.

README.md: General documentation.

backend/: Backend code and configurations.

src/: Source code organized by controllers, models, routes, services.

tests/: Unit and integration tests.

config/: Configuration files.

frontend/: Frontend code for web and mobile applications.

web/: Web application source code and tests.

mobile/: Mobile application source code and tests for both Android and iOS.

training/: Training modules for traffic officers.

modules/: Organized by module, each containing videos, documents, and quizzes.

README.md: Instructions and descriptions of training modules.

scripts/:

maintenance/: Maintenance scripts.

README.md: Documentation for scripts.

Root Files:

README.md: Overview of the project.

LICENSE: Licensing information for the project.

Azure DevOps Architecture

1.Source Control Management

Tool: Azure Repos

Continuous Integration and Continuous Deployment (CI/CD)

Tool: Azure Pipelines

Project Management

Tool: Azure Boards

Infrastructure as Code (IaC)

Tool: Azure Resource Manager (ARM)

Package Management

Tool: Azure Artifacts

Security and Compliance

Tool: Azure Security Center

DEPLOYMENT ARCHITECTURE OF THE APPLICATION: