TEST PLAN FOR

ROAD ACCIDENT ANALYSIS & CLASSIFICATION

ChangeLog

Version	Change Date	By	Description
1.0.0	30.10.23	Ayush Pratap Singh	Live Severity

1 Introduction

Road accidents are a significant global concern, leading to immense human suffering and economic losses. To address this issue, our project focuses on developing a real-time road accident severity prediction system that leverages location data, vehicle information, and machine learning. This system aims to provide valuable insights to both drivers and traffic management authorities, helping them make informed decisions to enhance road safety.

In this testing report, we will delve into the details of our project, outlining its objectives, methodology, and the technology stack used. We will also present the results of our testing efforts, including the system's accuracy, performance, and user experience. This report serves as a comprehensive evaluation of the road accident prediction system, highlighting its potential to reduce accidents and their severity, ultimately making our roads safer for all.

1.1 Scope

1.1.1 In Scope

Functional Requirement: -

Real-time Location Data Collection: The system must collect real-time location data from various sources, such as GPS-enabled devices and traffic cameras.

Accident Data Analysis: The project must analyze data related to previous accidents, such as the accident location, weather conditions, and time of day, to predict accident severity.

Severity Prediction: The core functionality of the system is to predict the severity of road accidents in real-time, indicating whether an accident is likely to be minor or severe.

User Interface: The system should provide a user-friendly interface for both drivers and traffic authorities to access accident severity information.

Notifications and Alerts: Users should receive timely notifications and alerts about predicted accident severity to help them make informed decisions.

Non-Functional requirement: -

Scalability: The system should be capable of handling a high volume of real-time data and user requests.

Accuracy: The severity predictions must be accurate to ensure their reliability for decision-making.

Response Time: The system should provide real-time predictions with minimal latency to support quick decision-making by users.

Usability: The user interface should be intuitive and easy to use, catering to both drivers and traffic management authorities.

Performance: The system's performance, in terms of prediction speed and data processing, should meet or exceed predefined benchmarks.

1.1.2 Out of Scope

Equally important to defining what is within the project's scope is understanding what is explicitly excluded from the project. This section outlines the elements and functionalities that are out of scope for the Road Accident Severity Prediction System.

Traffic Control Systems: The project will not be responsible for controlling traffic signals, cameras, or any physical traffic management systems.

Vehicle Hardware Development: The development of hardware components for vehicles, such as sensors or cameras, is not within the scope.

Legal or Regulatory Changes: The system will not make or enforce changes to traffic laws or regulations. It will operate within the framework of existing traffic laws.

1.2 Quality Objective

Ensuring the Road Accident Severity Prediction System operates effectively and reliably is fundamental to its success. The following quality objectives define the expected performance and operational characteristics of the system:

- The system must provide accurate predictions of accident severity based on real-time and historical data, with a reliability rate exceeding 95%. Users should have confidence in the system's recommendations. Ensure the AUT meets the quality specifications defined by the client.
- The system should respond to user inputs and incoming data in real-time, with a latency of no more than 2 seconds. Quick responses are critical for timely accident management.
- The architecture of the system must be scalable to handle increased data and user loads as the user base grows. It should easily accommodate higher traffic volumes.
- The user interface should be intuitive, with clear and user-friendly features. Users of varying technical expertise should be able to interact with the system with minimal training.

1.3 Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

- **QA Analyst**- Anuj Jain
- **Test Manager**- Dr. Harsh Khatter
- **Configuration Manager**- Prof. Akankskha
- **Developers**- Ayush Pratap Singh, Arth Srivastava
- **Installation Team**- Ayush Pratap Singh, Arth Srivastava

2 Test Methodology

2.1 Overview

The primary purpose of the Road Accident Severity Prediction System is to predict the severity of road accidents in real-time. By analyzing a variety of data sources, including weather conditions, road type, vehicle count, and historical accident data, the system can provide valuable insights to both authorities and drivers.

2.2 Test Levels

Testing is a critical component of our road accident analysis project, ensuring that our system functions as expected and meets the defined requirements. We have organized our testing efforts into multiple levels to systematically evaluate different aspects of the project. These test levels help us identify and rectify issues at various stages of development. Below are the key test levels applied to our project:

Unit Testing:
Scope: Unit testing focuses on the smallest components of the project, such as individual functions, modules, or algorithms. In the context of the road accident analysis project, this level ensures that each component, like data collection scripts or analysis algorithms, performs as intended.

Objective: The primary goal of unit testing is to validate the correctness of individual components and to identify and rectify any defects at an early stage. This level aims to ensure that each piece of code works in isolation.

Testing Approach: Developers and testers collaborate to create test cases that assess the functionality of specific code units. Test data is designed to cover normal and boundary conditions. Automated testing tools can aid in the execution of a large number of tests quickly.

Integration Testing:
Scope: Integration testing assesses the interactions between various components or modules of the system. In the road accident analysis project, it verifies that the data collection process effectively feeds data into the analysis modules.

Objective: The objective of integration testing is to expose issues related to data flow, communication between modules, and any inconsistencies in the system's overall behavior when multiple components interact.

Testing Approach: Test cases are designed to evaluate how different components work together. This may involve testing data transfers between components and assessing the system's ability to handle and process the data as it flows through various stages.

System Testing:

Scope: System testing evaluates the entire road accident analysis system as a whole. It tests the system's compliance with the project requirements, ensuring that it meets the defined objectives and functions as expected.

Objective: The goal of system testing is to validate that the complete system fulfills the project's objectives and works seamlessly in a real-world context. It covers functionality, performance, security, and usability aspects.

Testing Approach: Test scenarios are developed to simulate real-world use cases. This testing phase assesses the system against predefined requirements and evaluates its performance under typical and stress conditions.

User Acceptance Testing (UAT):
Scope: UAT is the final testing phase, focusing on ensuring that the road accident analysis system meets the needs and expectations of its intended users. This testing level typically involves real end-users.

Objective: The primary objective of UAT is to validate that the system aligns with user requirements and that it is user-friendly. It confirms that the system is ready for production use.

Testing Approach: End-users and stakeholders participate in UAT. They execute test scenarios representing real-world tasks, and feedback is collected. Defects, if any, are addressed before system deployment.

2.3 Test Completeness

In software testing, "test completeness" refers to the extent to which a system or software application has been tested in terms of various aspects such as functionality, requirements coverage, and potential use cases. It is a critical aspect of the testing process that ensures that the system is rigorously evaluated to identify defects and ensure that it meets its intended objectives.

For instance, a few criteria to check Test Completeness would be

- 100% test coverage
- All Manual & Automated Test cases executed
- All open bugs are fixed or will be fixed in next release

3 Test Deliverables

Test cases: -

Test Case	Test Objective	Test Data	Expected Result	Actual Result	Pass/Fail
1	Verify Input for start and end location	Start & end location outside UK	Only UK Locations allowed	Invalid location	Pass

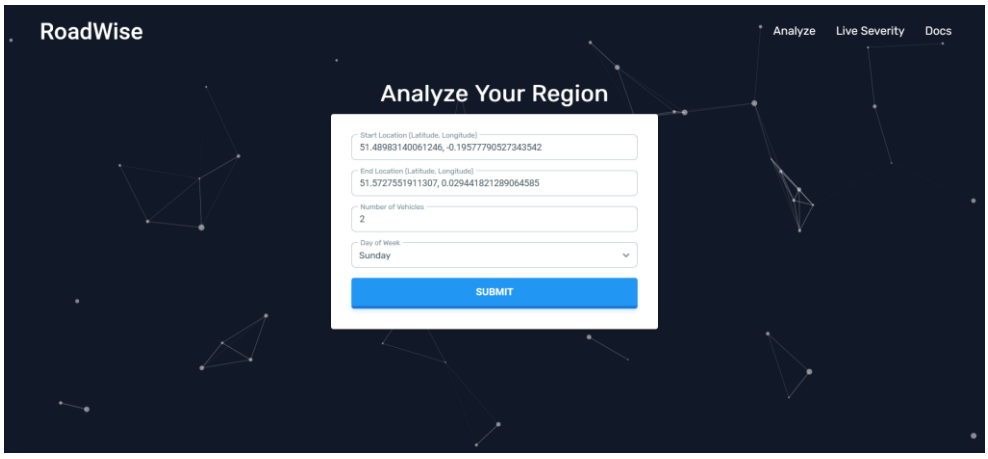
2	Verify Number of vehicles	Number of vehicles <= 3	Allowed	Valid Input	Pass
3	Verify route calculation	Start & end location inside UK	Should calculate route correctly & display on map	Map displaying highlighted route	Pass
4	Verify API Data Fetching	Start & end location inside UK	Fetch location-based data from API & feed it to model	Correctly fetching location-based data from API	Pass
5	Verify easting and northing	Start & end location inside UK	Correctly calculate location easting & northing	Model working correctly	Pass

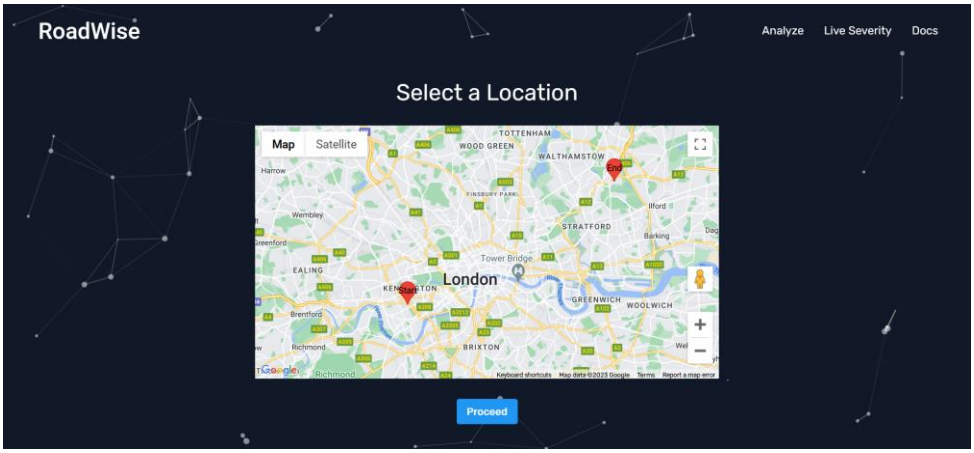
Decision Table

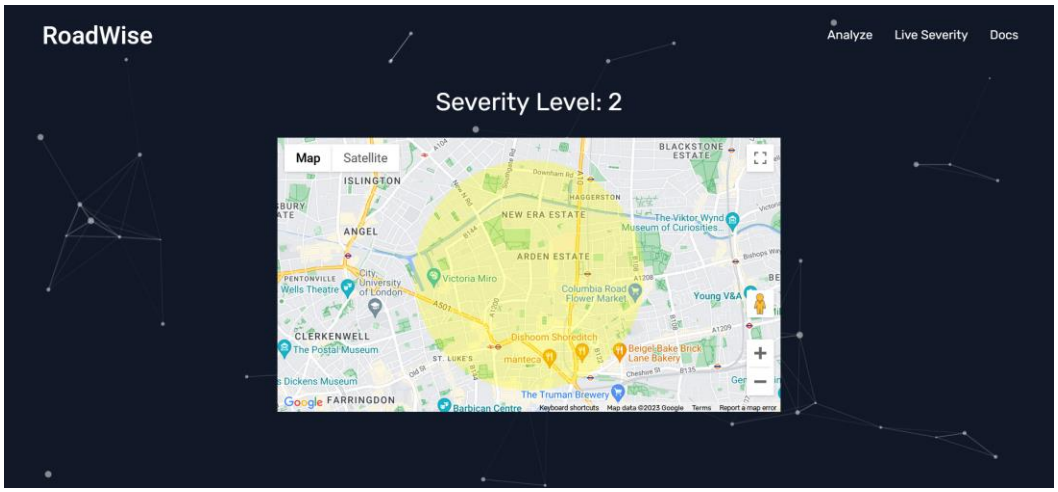
Conditions	**Rules**							
Traffic congestion is high	True				False			
Road condition is poor	True		False		True		False	
Number of vehicles > 3	True	False	True	False	True	False	True	False
Actions	**Entries**							
Alert The User	Yes	Yes	Yes	Yes	No	No	No	No
Record Data	Yes	Yes	Yes	No	Yes	Yes	No	No
Notify Authorities	Yes	No	No	No	Yes	No	Yes	No

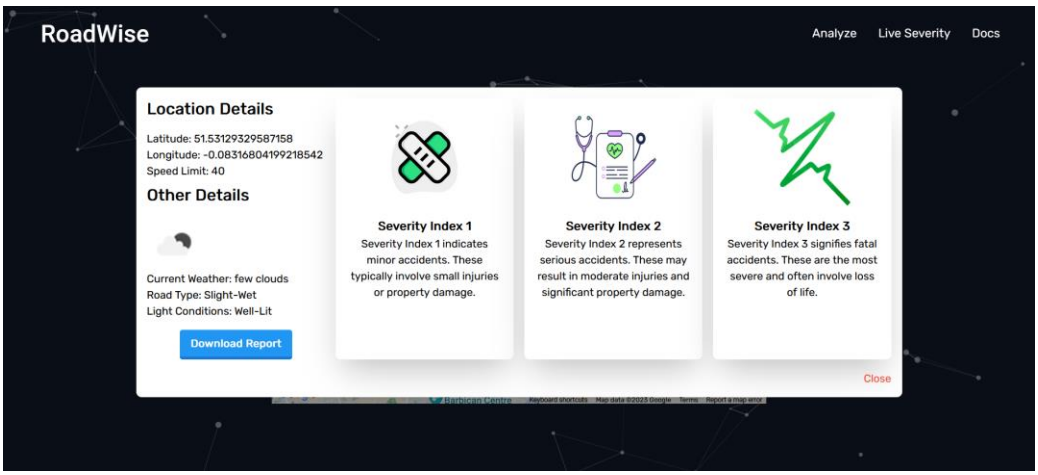
Test case Output Images

1. **Manual Testing:**

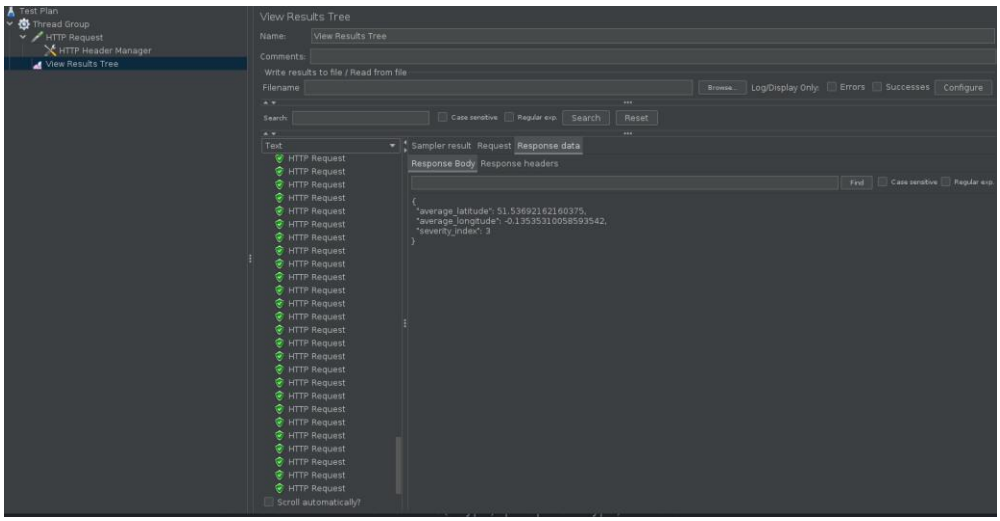


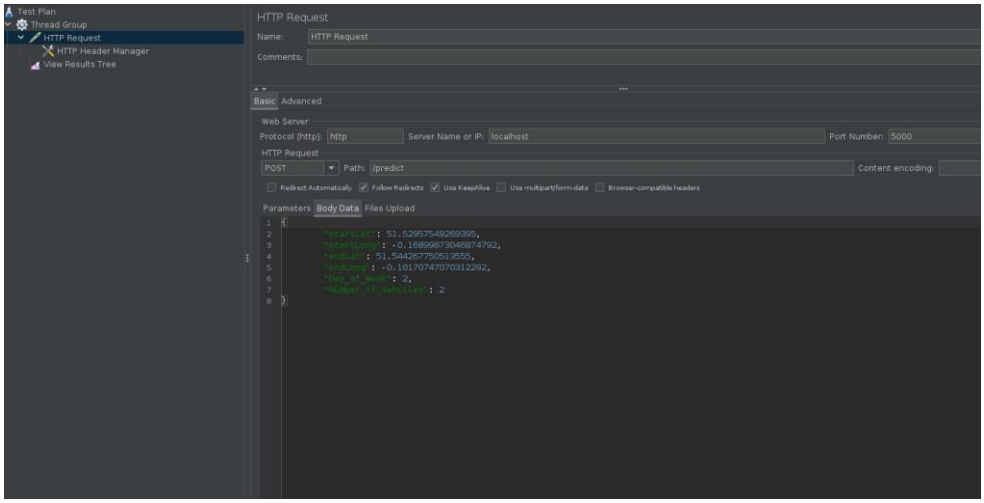

RoadWise
Analyze
Live Severity
Docs
Select a Location
Map
Satellite
London
Proceed

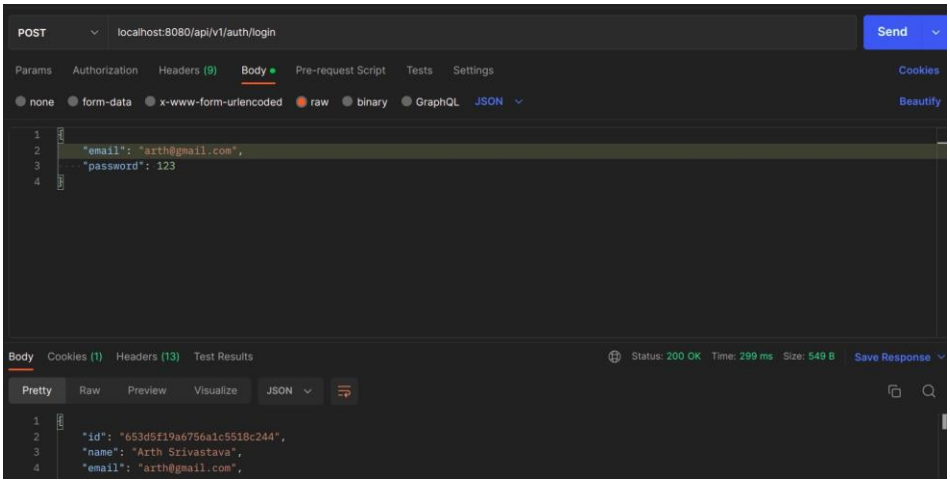

RoadWise
Analyze
Live Severity
Docs
Severity Level: 2
Map
Satellite


RoadWise
Analyze
Live Severity
Docs
Location Details
Latitude: 51.53129329587158
Longitude: -0.08316804199218542
Speed Limit: 40
Other Details
Current Weather: few clouds
Road Type: Slight-Wet
Light Conditions: Well-Lit
Download Report
Severity Index 1
Severity Index 1 indicates minor accidents. These typically involve small injuries or property damage.
Severity Index 2
Severity Index 2 represents serious accidents. These may result in moderate injuries and significant property damage.
Severity Index 3
Severity Index 3 signifies fatal accidents. These are the most severe and often involve loss of life.
Close

2. Automation Testing:







4 Resource & Environment Needs

4.1 Testing Tools

In your road accident analysis project, various testing tools can be employed to ensure the efficiency, accuracy, and reliability of your application. These tools assist in automating tests, managing test cases, tracking defects, and more. Here are some of the testing tools that can be beneficial for your project:

1. **Postman:**
    - Scope: Postman is a popular API testing tool. It can be used to test the RESTful APIs integrated into your project, such as those fetching accident data.
    - Objective: To validate API endpoints, check responses, and ensure that the API requests and data retrieval function correctly.
    - Testing Approach: Develop API test collections and scripts to send requests, assess responses, and verify the accuracy of data retrieved from external sources.
2. **Load Testing Tools (e.g., Apache JMeter):**
    - Scope: Load testing tools help assess how your application performs under various levels of concurrent user activity.
    - Objective: To identify performance bottlenecks, server capacity, and response times.
    - Testing Approach: Design test scenarios that simulate real-world traffic and evaluate the application's stability under load.

4.2 Test Environment

Following **software's** are required in addition to client-specific software.

- Web browsers (e.g., Chrome, Firefox) for testing the web interface.
- Internet access to retrieve real-time data.
- Testing tools (e.g., Selenium, Postman, load testing tools) for test automation and API testing.
- Database management systems (e.g., MySQL) for data storage and retrieval testing.
- Development environments (e.g., Visual Studio Code) for scripting automated tests.
- Continuous Integration/Continuous Deployment (CI/CD) tools (e.g., Jenkins) for automating build and deployment.
- Test data generation tools for creating realistic test scenarios.
- Source code repositories (e.g., GitHub) for version control.
- Collaboration and project management tools (e.g., JIRA) for issue tracking and task management.

5 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
AUT	Application Under Test
API	Application Program Interface