

TEST PLAN FOR ENHANCING URBAN MOBILITY

ChangeLog

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
001	30,October,2023	Khushi Vaish, Nikita sharma, Kashish Gupta	Initial Draft

1	INTRODUCTION	2
1.1	SCOPE	2
1.1.1	<i>In Scope</i>	2
1.1.2	<i>Out of Scope</i>	3
1.2	QUALITY OBJECTIVE	3
1.3	ROLES AND RESPONSIBILITIES	4
2	TEST METHODOLOGY.....	4
2.1	OVERVIEW	4
2.2	TEST LEVELS.....	4
2.3	BUG TRIAGE	6
2.4	SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS	ERROR! BOOKMARK NOT DEFINED.
2.5	TEST COMPLETENESS.....	7
3	TEST DELIVERABLES.....	7
4	RESOURCE & ENVIRONMENT NEEDS.....	7
4.1	TESTING TOOLS	7
4.2	TEST ENVIRONMENT.....	8
5	TERMS/ACRONYMS	8

1 Introduction

In today's world we have been facing a problem due to increasing traffic on roads. This causes a lot of waste of time and increases stress level. As many emergency vehicles are also stuck in traffic for hours, it leads to losing life of people. So, this requires development of a system to handle traffic in a smart way by automatically adjusting its timing based on traffic density to provide passage for ambulances using RCNN (Region-based Convolutional Neural Network), webcam, NodeMcu controller, LEDs. The core idea revolves around traffic management through the assessment of traffic volume on each side of the road, with the aim of implementing smart traffic signal control based on this density information. The webcam captures images of the vehicles on the road and sends images to the NodeMcu microcontroller and then we apply RCNN algorithm and OpenCV to detect and count number of vehicles on the lane and set traffic timing accordingly. If an ambulance is detected, traffic light turns green on that side and other side is turned red to provide passage for emergency vehicles.

1.1 Scope

1.1.1 In Scope

FUNCTIONAL REQUIREMENTS

1. Traffic Density Assessment:

- The system shall capture real-time images of the road using a webcam.
- It shall apply the RCNN algorithm to detect and count vehicles on each side of the road.
- The system shall update traffic density information based on the vehicle count.

2. Smart Traffic Signal Control:

- It shall implement intelligent traffic signal algorithms to optimize traffic flow.

3. Emergency Vehicle Detection:

- The system shall continuously monitor the road for the presence of an ambulance.
- It shall use the RCNN algorithm to detect an ambulance and its precise location.

4. Emergency Vehicle Priority:

- Upon detecting an ambulance, the system shall immediately turn the traffic light green on the ambulance's side.
- The system shall change the opposite side's traffic light to red, providing a clear passage for the ambulance.

NON-FUNCTIONAL REQUIREMENTS

Performance:

Response Time: The system should provide real-time responses to changing traffic conditions, ensuring minimal delay in adjusting traffic signal timings.

Throughput: The system should be able to process a large number of images from the webcam efficiently and analyze them in real-time.

Reliability:

Availability: The system should have a high level of availability, with minimal downtime to ensure continuous traffic management.

Usability:

User-Friendly Interface: The user interface should be intuitive and user-friendly, making it easy for administrators to monitor and configure the system.

Compliance:

Regulatory Compliance: Ensure compliance with relevant traffic and safety regulations in your region.

Maintainability:

Documentation: Provide comprehensive documentation for system maintenance and troubleshooting.

1.1.2 Out of Scope

The project is only intended to detect trained vehicles and within limited range.

1.2 Quality Objective

Here make a mention of the overall objective that you plan to achieve without your testing

Some objectives of your testing project could be

- Ensure that the RCNN algorithm achieves a high level of accuracy in detecting and counting vehicles, minimizing false positives and false negatives.
- The system should respond to changing traffic conditions in real-time, with minimal delays in adjusting traffic signal timings and detecting emergency vehicles.
- Maintain a high level of system reliability, with a target availability of at least 99%, to ensure continuous traffic management.

- Guarantee that the system provides immediate and priority passage for emergency vehicles, reducing response times and potentially saving lives.

1.3 Roles and Responsibilities

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

- QA Analyst – Khushi Vaish
- Test Manager – Abhishek Goyal
- Configuration Manager – Akanksha
- Developers – Khushi Vaish , Nikita Sharma, Kashish Gupta
- Installation Team - Khushi Vaish , Nikita Sharma, Kashish Gupta

Amongst others

2 Test Methodology

2.1 Overview

Mention the reason of adopting a particular test methodology for the project. The test methodology selected for the project could be

- **Agile**
Reasons to Adopt: Agile is highly flexible and iterative. It is a strong choice for projects with changing or evolving requirements, as well as those that require frequent testing and validation. For a dynamic project like smart traffic management, where quick responses to changes are crucial, Agile can be highly effective.

Considerations: Agile methodologies, such as Scrum or Kanban, can ensure that the project team remains adaptable and can incorporate changes rapidly. Frequent testing and feedback loops align well with the real-time nature of traffic management.

2.2 Test Levels

Unit Testing:

Objective: At the lowest level, individual components or modules are tested in isolation to verify that they work as intended.

Scope for Your Project: Test individual components of your system, such as the NodeMcu controller, webcam image processing, and RCNN algorithm, to ensure they function correctly.

Integration Testing:

Objective: This level verifies the interactions and interfaces between different components when they are combined.

Scope for Your Project: Test the integration between hardware components (e.g., webcam, NodeMcu, LEDs) and software components (e.g., RCNN, traffic signal control).

System Testing:

Objective: Test the entire system as a whole to ensure that it meets the specified requirements and functions according to design.

Scope for Your Project: Evaluate the entire smart traffic management system, including its ability to assess traffic density, adjust traffic signal timings, and respond to emergency situations.

Performance Testing:

Objective: Assess the system's performance under various conditions, such as traffic volume and response time.

Scope for Your Project: Analyze the system's performance in terms of real-time responsiveness, scalability, and resource utilization.

Security Testing:

Objective: Evaluate the system's ability to protect against unauthorized access, data breaches, and cyber threats.

Scope for Your Project: Ensure that the system is secure and can safeguard sensitive data related to traffic management.

Usability Testing:

Objective: Assess the system's user interface for ease of use, accessibility, and user satisfaction.

Scope for Your Project: Evaluate the user interface for administrators, ensuring that it is user-friendly and accessible.

Regulatory Compliance Testing:

Objective: Ensure that the system adheres to relevant traffic and safety regulations and standards.

Scope for Your Project: Verify that the system complies with local and national traffic regulations and safety standards.

Acceptance Testing:

Objective: This level involves validating that the system meets the acceptance criteria defined by stakeholders.

Scope for Your Project: Work with stakeholders to define acceptance criteria and conduct testing to ensure that the system satisfies these criteria.

Emergency Response Testing:

Objective: Specifically, test the system's ability to detect and respond to emergency situations, such as ambulance arrivals.

Scope for Your Project: Ensure that the system reacts promptly and appropriately when an ambulance is detected, providing priority passage.

Load Testing:

Objective: Evaluate how the system performs under heavy traffic conditions to ensure it can handle peak loads.

Scope for Your Project: Test the system's performance when traffic is at its highest and assess how it manages congestion.

2.3 Bug Triage

Define Bug Resolutions:

The bug triage process should determine the type of resolution for each reported bug. This categorization helps prioritize and schedule fixes appropriately. Bug resolutions typically include categories like "fixed," "won't fix," "duplicate," "cannot reproduce," "postponed," and "by design."

Prioritize Bugs:

Triage is used to assess the severity and impact of each reported bug. This helps in prioritizing bugs based on their criticality, the potential to affect traffic management, and the urgency of the resolution.

Schedule Fixes:

Once bugs are categorized and prioritized, bug triage helps in determining a schedule for fixing these issues. Critical and high-priority bugs are scheduled for immediate attention, while lower-priority issues may be addressed in future releases or updates.

Improve System Quality:

By addressing and resolving reported bugs, the system's overall quality and reliability can be enhanced. Bug triage ensures that issues are systematically addressed to maintain or improve the system's performance and functionality.

Efficient Resource Allocation:

Bug triage aids in allocating resources efficiently. Critical bugs may require immediate attention from developers, while lower-priority issues can be scheduled when resources are available.

Effective Communication:

Bug triage serves as a communication mechanism between development, testing, and management teams. It helps in transparently conveying the status of reported issues, their severity, and the planned resolutions.

2.4 Test Completeness

Here you define the criterias that will deem your testing complete.

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

Here mention all the Test Artifacts that will be delivered during different phases of the testing lifecycle.

Here are the sample deliverables

- Test Plan
 - Test Cases
 - Requirement Traceability Matrix
 - Bug Reports
 - Test Strategy
 - Test Metrics
 - Customer Sign Off
-

4 Resource & Environment Needs

4.1 Testing Tools

Make a list of Tools like

- Requirements Tracking Tool
- Bug Tracking Tool
- Automation Tools

Required to test the project

4.2 Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application.

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

- Windows 8 and above
- Office 2013 and above
- MS Exchange, etc.

5 Terms/Acronyms

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

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