National Institute of Technology, Rourkela
Computer Science Department
CS6471: Advanced Software Engineering Laboratory

Software Requirements Specification

Version (1.0)

Project Title: Road Repair and Tracking Software (RRTS) Development

Prepared by Utkarsh Chakrwarti

Roll Number: 223CS1098

NIT Rourkela

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the development of the Road Repair and Tracking Software (RRTS), which aims to streamline road maintenance, improve decision-making, and enhance resource allocation. It handles resident issues, evaluates road conditions, prioritizes repairs, schedules work, adapts to changes, and provides repair statistics to the mayor.

1.2 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, and users who need to understand the system's requirements and specifications. It provides comprehensive information on the RRTS project, including feasibility, recommendations, and financial analysis.

1.3 Product Scope

The RRTS system efficiently manages resident issue reporting, supervisor assessments, and repair prioritization based on severity and location. It optimizes resource allocation and considers climate conditions, optimal repair times, and traffic management. RRTS ensures adaptability to changing circumstances and provides detailed repair statistics for informed decision-making by the mayor.

1.4 Feasibility Study

The feasibility study for RRTS considers various aspects, including one-time (initial) costs, recurring costs, fixed and variable costs, benefits (tangible and intangible), and cost estimation. The study concludes with a 2.5-year payback period, subject to accurate cost and benefit estimates.

1.5 Management Summary and Recommendations

To establish a robust and adaptable RRTS system, the following recommendations are made:

1. Integrate weather data for efficient repair scheduling.
2. Create an algorithm prioritizing repairs based on severity and high-traffic areas.
3. Analyse historical data to find optimal repair times, reducing traffic disruption.
4. Implement real-time traffic monitoring to reroute traffic and ease congestion during repairs.
5. Build a dynamic resource allocation system adjusting personnel and machinery based on issue severity and location.
6. Continuously analyse repair costs to identify cost-saving opportunities.
7. Provide comprehensive reports and dashboards to inform the mayor and department heads.
8. Develop user-friendly training materials and offer support for all users, including residents.
9. Design scalable software to adapt to future growth and evolving needs.

2. Overall Description

2.1 Product Perspective

RRTS is a web-based system accessible through mobile devices, tablets, and laptops. It addresses the need for efficient road maintenance in large cities with expanding road networks. Residents can easily report road issues, enabling the Public Works Department to respond promptly.

2.2 Development Environment and Constraints

The system will be developed using Java and JavaScript for the front-end. It will be accessible over the internet for users and maintainers to interact with. Assumptions include regular updates of resource availability by administrators.

3. Functional Requirements

3.1 Inputs

1. User-submitted complaints, including road name, location, and issue description.
2. Data on road damage severity and resource requirements from supervisors.
3. Resource availability updates from administrators.

3.2 Processing

1. Processing user complaints and generating lists of damaged roads for inspection.
2. Analysing repair priorities and scheduling work based on resource availability.
3. Updating resource availability data.
4. Retrieving and generating repair statistics and reports.

3.3 Outputs

1. Area-wise lists of complaints for supervisors.
2. Detailed repair schedules.
3. Statistics on repair work and resource utilization.

4. Non-Functional Requirements

4.1 Performance Requirements

The system should perform efficiently even with a slow internet connection and low data consumption.

4.2 User-friendliness

The user interface should be intuitive and user-friendly for easy complaint submission.

4.3 Security

User information and complaint details must be securely maintained, ensuring user anonymity.

5. Goals of Implementation

5.1 Feedback

The system will incorporate user feedback to improve complaint processing and repair timeliness.

5.2 Cost Benefit Analysis

The cost benefit analysis demonstrates a 2.5-year payback period based on potential benefits and initial investment, contingent on accurate cost and benefit estimates.

5.3 Financial Analysis

The financial analysis highlights potential benefits, including cost savings, efficiency gains, and reduced maintenance costs, leading to a positive ROI.

5.4 Evaluation of Technical Risk

Technical risks associated with data integration, system performance, security, and user interface design are acknowledged and must be effectively managed.