Software Requirements Specification

For

Road Repair and Tracking System (RRTS)

Version 1.0

Prepared By: Naveen Kumar Khuntey (121CS0176)

Submitted to : DR. Judhistir Mohapatro, Professor

Software Engineering Lab - CS3074

Date: 16 Jan 2024

INDEX

1. Introduction

1.1 Purpose

The purpose of this document is to define is to define the requirements that Road Repair and Tracking System (RRTS). RRTS is to automate various book keeping activities associated with road repair tasks carried out by the Public Works Department of a large city. RRTS aims to efficiently manage repair requests, prioritize them based on severity and locality, and schedule repair work while considering the availability of resources.

This document will be used by Developers: Who will implement the RRTS system. Supervisors: Responsible for overseeing road repair work. Administrators: Managing resource availability data. Mayor: Seeking road repair statistics.

1.2 Scope

People can raise repair requests via phone or written complaints. A clerk enters these complaints into the system. Supervisors receive an area-wise list of fresh complaints each morning. They visit assigned areas, assess road conditions, and determine repair priorities based on severity and locality type. Supervisors estimate raw material requirements, machine types, and personnel needed for repair work. The system schedules repairs based on priority, resource availability, and severity. Administrators update data on available manpower, machines, and raw materials. The mayor can request road repair statistics.

1.3 Environmental characteristics

The software will interact with public works department's database system for complaint storage and retrieval. The system will be accessible through web browsers, ensuring flexibility and ease of use for clerks, mayor, supervisors, and administrators.

1.4 Definition

Acronyms and Abbreviations:

- **RRTS**: Road Repair and Tracking System
- **SRS**: Software Requirements Specification

1.5 References

This SRS is developed using Microsoft word. The Structure and conventions of the document follow the guidelines outlined in the pdf given by Dr. Judhistir Mohapatro sir, and IEEE standards for software Requirements Specifications.

2. General Description

2.1 Product perspective

The **Road Repair and Tracking System (RRTS)** is designed to automate various bookkeeping activities associated with road repair tasks handled by the Public Works Department of the Corporation of a large city. The system will serve as a comprehensive solution to efficiently manage the entire road repair process, from complaint registration to scheduling and tracking repairs.

2.2 Product functions overview

The primary functions of RRTS include:

1. **Complaint Registration**: Residents can submit repair requests either via phone or written complaints. Clerks enter the complaints into the system for further processing.
2. **Supervisor Assessmen**t: Supervisors examine the complaints and determine severity and priority. They estimate raw material requirements, machine types, and personnel needed for each repair.
3. **Scheduling**: The system schedules repair work based on priority, considering available resources. Daily reports are generated for supervisors with area-wise lists of fresh complaints.

4. **Administrator Management**: The city corporation administrator manages and updates data on available manpower and machines. Changes trigger a reschedule of ongoing projects.
5. **Statistics Reporting**: The mayor can request various road repair statistics, including the number and types of repairs over time, outstanding repairs, and utilization statistics of repair manpower and machines.

2.3 User characteristics

The key users of RRTS include:

Residence: Submit road repair request through phone or written complaints.

Clerks: Responsible for entering repair requests into the system.

Supervisors: Assess complaints, determine priorities, and estimate resources needed.

Administrator: Manages and updates data on available manpower and machines.

Mayor: Requests and reviews road repair statistics.

2.4 Operating environment

The Road Repair and Tracking System (RRTS) consists of:

• Web App: Accessed via web browsers for complaint submission.

• Backend: Operates in the database for data storage and application logic.

2.5 Design and Implementation constraints

The system will use NodeJS for backend implementation and React (JavaScript Framework) for frontend.

3. Functional Requirements

3.1 Complaint Registration

Introduction: The Complaint Registration module allows residents to submit repair requests, which are then entered into the system by clerks. This ensures a systematic approach to recording and managing repair complaints.

Inputs: Resident-provided repair requests via phone or written complaints. Relevant details, such as location, type of issue, and any additional comments.

Processing: Clerks input resident-provided data into the system, ensuring accuracy and completeness. The system generates a unique complaint ID for each entry.

Outputs: A confirmation message to the resident with the assigned complaint ID. The stored complaint data is made available for further processing by supervisors.

3.2 Supervisor Assessment

Introduction: The Supervisor Assessment module empowers supervisors to evaluate and prioritize repair requests. They estimate the required resources for each repair based on the severity of the road condition and the locality type.

Inputs: Complaint data entered by clerks. Supervisor's examination of the road conditions and severity. Locality type (e.g., commercial, busy, deserted).

Processing: Supervisors assess the complaints, assigning priority based on severity and locality type. Estimate the raw material, machine types, and personnel required for each repair.

Outputs: Prioritized repair requests with detailed resource estimates. Daily reports for supervisors with area-wise lists of fresh complaints.

3.3 Scheduling

Introduction: The Scheduling module generates an optimal repair schedule based on prioritized repair requests and the availability of resources such as raw materials, machines, and personnel.

Inputs: Prioritized repair requests from the Supervisor Assessment module. Availability status of raw materials, machines, and personnel. Daily reports with area-wise lists of fresh complaints.

Processing: The system schedules repair work, considering priority and resource availability. Daily reports are updated to reflect scheduled repairs.

Outputs: Detailed repair schedule, including assigned personnel, machines, and estimated completion times. Updated daily reports for supervisors.

3.4 Administrator Management

Introduction: The Administrator Management module allows the city corporation administrator to maintain and update data on available manpower and machines, facilitating efficient resource allocation.

Inputs: Current data on available manpower and machines. Changes made by the administrator, such as additions or removals.

Processing: The system updates the database with the administrator's changes. If there are ongoing projects, the system triggers a reschedule based on the updated data.

Outputs: Updated records of available manpower and machines. Notifications to supervisors if ongoing projects are affected by changes.

3.5 Statistics Reporting

Introduction: The Statistics Reporting module enables the mayor to request and review various road repair statistics, providing insights into the city's repair activities.

Inputs: Mayor's request for specific statistics, such as the number and types of repairs over time. Data stored in the system regarding completed and outstanding repairs.

Processing: The system compiles and processes relevant data based on the mayor's request. Generates statistical reports as per the requested parameters.

Outputs: Detailed statistical reports, including the number and types of repairs over time. Utilization statistics of repair manpower and machines.

4. Non-Functional Requirements

4.1 Performance Requirements

The Road Repair and Tracking System (RRTS) demands efficient performance to meet user expectations. The system responds promptly to user interactions, ensuring a seamless experience. The React frontend and Node.js backend maintain a good response time. Additionally, the system also supports more concurrent users to accommodate varying usage patterns during peak periods.

4.2 Reliability

Reliability is crucial for the successful operation of RRTS. The system aims for a higher availability rate, minimizing downtime for both the React frontend and Node.js backend. Data integrity is important to maintain accuracy and consistency. The reliability of both frontend and backend components will be measured to ensure a dependable system.

4.3 Security

Security is a top priority to safeguard sensitive data and user interactions. The Node.js backend will implement data encryption for transmitted and stored data in database. Access control like authenticate and authorize users effectively. Regular reviews of access logs will ensure the detection and prevention of unauthorized access attempts by supervisors.

4.4 Maintainability

Maintainability is essential for the long-term viability of RRTS. The React frontend and Node.js backend codebases prioritize maintainability. Regular code reviews and documentation assessments will be conducted to facilitate easy maintenance and updates by using version control like git. System updates, including those to React and Node.js components, will be deployed.

5 Other Requirements

5.1 Accessibility

RRTS is committed to accessibility, in the React frontend. The user interface design will adhere to accessibility standards, providing an intuitive experience for all users.

5.2 Documentation

Comprehensive documentation, including user manuals system architecture, and code documentation, should be provided to aid system understanding and maintenance.

5.3 User Training

The React frontend's user interface is designed to be intuitive, minimizing the need for extensive user training. However, supplementary training materials will be provided to enhance user understanding and proficiency. Training sessions, if necessary, will be conducted to familiarize users with the system's functionalities.

5.4 Error Handling

RRTS prioritizes effective error handling to ensure a resilient user experience. The software implement error handling mechanisms, providing clear error messages to users.