DATA DICTIONARY

For

Road Repair and Tracking System

Prepared by: **Archisman Chakraborty(224CS1004)**
Submitted to: **Professor Dr. Judhistir Mahapatro**

Date: **19th September, 2024**

National Institute of Technology, Rourkela

Table of Contents

1. Introduction

This data dictionary was created for a *Road Repair and Tracking System (RRTS)* for a major city corporation's Public Works Department. The document lists and details all the data items that will be used throughout this project. Going forward any doubts or conflicts regarding any data items in the project can be resolved with the help of this data dictionary.

2. Data Dictionary

2.1. User Authentication and Security

2.1.1. User Registration Data = valid-data = name+username+password+role+contactInfo

- name: string   *User's full name*

- username: string   *User's username/system name*

- password: string   *User's password*

- role: string = [Clerk, Supervisor, City Corporation Administrator, Mayor, Administrator]

- contactInfo: string   *User's contact information*

2.1.2. Login Credentials = valid-data = username+password

- username: string

- password: string

2.1.3. Access Control Data = eventTime+userId+event+details

- eventTime: datetime   *time stamp of the event that is recorded*

- userId: string   *User ID of the user associated with the access log*

- event: string   *Event shortcode*

- details: string   *Event details*

2.2. Complaint Management

2.2.1. Complaint Data = valid-data = residentName+residentContact+details+date+location+areaCode+status

- residentName: string   *resident's name*

- residentContact: string   *resident's contact information*

- details: string   *details of complaint*

- date: datetime   *Date the complaint was registered*

- location: string   *Exact location where repair is needed*

- areaCode: string   *Alphanumeric code for an area*

- status: string = [New, In Progress, Resolved, Waiting]

2.3. Priority Assessment

2.3.1. Priority Assessment Data = valid-data = repairID+analysis+priority

- repairID: string   *repair id*

- analysis: string   *Supervisor's analysis of the complaint*

- priority: string = [Critical, High, Medium, Low]   *Recommended priority*

2.3.2. Priority Data = repairID+priority

- repairID: string   *repair ID*

- priority: string = [Critical, High, Medium, Low]   *Set Priority*

2.4. Resource Management

2.4.1. Resource Data = valid-data = type+description+status+quantity+lastUpdated

- type: string = [Machinery, Personnel, Raw Material]   *resource id*

- description: string   *description of resource*

- status: string = [Allocated, Free, Unavailable]

- quantity: integer   *Quantity of resource*

2.4.1. Resource Data Update = valid-data = resourceID+description+status+quantity

- resourceID: string   *resource id*

- description: string   *description of resource*

- status: string = [Allocated, Free, Unavailable]

- quantity: integer   *Quantity of resource*

2.5. Scheduling and Repair Assignment

2.5.1. Generated Schedules = repairID+priority+status+scheduledDate+deadline+supervisorID

- repairID: string   *repair Id of repair work*

- priority: string = [Critical, High, Medium, Low]

- status: string = [New, In Progress, Waiting, Resolved]

- scheduledDate: datetime   *scheduled start date*

- deadline: datetime   *expected end date of repair*

- supervisorID: string   *supervisor's user id*

2.5.2. Repair Scheduled Notification = repairID+priority+scheduledDate+deadline

- repairID: string   *repair Id of repair work*

- priority: string = [Critical, High, Medium, Low]

- scheduledDate: datetime   *scheduled start date*

- deadline: datetime   *expected end date of repair*

3. Conclusion

In conclusion, if we require some clarifications regarding the data items in the system, we can use this data dictionary for all conflict resolutions. This provides a clear and concise view of the data items being handled by the system and their purpose, which would help developers maintain consistency throughout the project developement .