Software Requirements Specification

for

Work Management System

Version 2.0 approved

Prepared by

Dewang Modi

B.Sashank Srivardhan

March 6 2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Changes in Section 2.2.2	18.3.19	Changed for better suitability	
Added Use Case, class and sequential diagrams	18.3.19	Required	

1. Introduction

1.1 Purpose

Work Management System(WMS) is a software intended to facilitate the process of dealing with urgent jobs by managing processes such as assigning workers to appropriate locations, managing notifications and recording previously completed job details.
This document is meant to explain the features of WMS in greater detail and hence serve as guidance to the developing team, describe the requirements in a complete, unambiguous and consistent manner and also as a validation document for the client.

1.2 Document Conventions

Important Abbreviations -

WMS - Work Management System
SRS - Software Requirements Specification
DB - Database

In a paragraph, section numbers are written in brackets.

Also, users of our software are organizations who perform jobs such as maintenance,cleanliness etc. We refer to them as Customer, user, or Concerned Organization, or organization interchangeably.

1.3 Intended Audience and Reading Suggestions

The document is intended primarily for users and customers, development team, test engineers, documentation writers, project managers,marketing staff and maintenance engineers.

The developers and project managers must be highly familiar with the SRS.

Users and customers are recommended to go through Overall Description(Section 2) to get a fundamental understanding of Work Management System.

Marketing staff should primarily focus on Overall Description(Section 2), particularly Product Functions(Section 2.2) and Operating Environment(Section 2.4) which shall be beneficial in advertising campaigns and strategies.

Testing staff need an understanding of the System features(Section 4) to properly test the system and prepare test cases covering all situations.

1.4 Product Scope

Work Management System is a software aimed to improve the present methods used in dealing with urgent work. Therefore, this is an extremely useful product for City Municipality, Colleges and Universities, NGOs etc to keep cleanliness and perform maintenance. It is aimed that the product is used throughout the country, in all cities,colleges and universities.

1.5 References

For more details, refer to www.wmsindia.org

2. Overall Description

2.1 Product Perspective

WMS is inspired by Swaccha Bharat Abhiyaan, and is aimed towards helping organisations and communities who handle cleanliness and other maintenance works by providing software assistance to their processes such as scheduling and notifications. Therefore, WMS should be easy to learn and handle,reliable,efficient. WMS is an independent software. It should run on both Ubuntu and Windows Operating System.

2.2 Product Functions

WMS should support the following features:

2.2.1.Maintain record of workers: This includes -

- Adding Worker Records
- Deleting Worker Records
- Updating Worker Records

The worker class contains the following attributes:

2.2.1.0. Worker Id
2.2.1.1. Worker Name (Full name of the worker)
2.2.1.2. Address (Residential Address)

2.2.1.3. Skill level (‘High’, ‘Medium’ or ‘Low’)
2.2.1.4. Availability
2.2.1.5. Email ID
2.2.1.6. Phone Number
2.2.1.7. Work Timings
2.2.1.8. Number of jobs done
2.2.1.9. Rating Points (Points given to workers on the basis of their work which also indicates their skill level, if Rating Points of a workers are greater than 350, he has ‘High’ skills, if they are between 200 and 350, he has ‘Medium’ skills, if Rating Points are less than 200, he has ‘Low’ skills)

2.2.2. Filling a new job: This includes adding a new job by choosing the appropriate worker/set of workers based on their availability, location, skill etc.

The job class contains following attributes:

2.2.2.1. Job Name
2.2.2.2. Job Description
2.2.2.3. Number of high skilled workers required
2.2.2.4. Number of medium skilled workers required
2.2.2.5. Number of low skilled workers required
2.2.2.6. Job Location
2.2.2.7. Priority of the job
2.2.2.8. Tools required
2.2.2.9. Start Time
2.2.2.10. Date
2.2.2.11. Worker List
2.2.2.12. Amount spent on the job
2.2.2.13. Rating of the job
2.2.2.14. Number of votes
2.2.2.15. JobID
2.2.2.16. Job Status

2.2.3. Scheduling and managing notifications: This includes notifying the set of nearest workers by sending them an email with the details of the location and work.

2.2.4. Taking public feedback: This includes taking online feedback from the concerned public. A separate Browser User Interface (BUI) is created to interact and take feedback from the concerned person.

2.2.5. Recording details of previously done jobs: This includes keeping a record of previously done jobs and the amount spent on the tools required and the wages given to the workers. The software will update this information in a file as soon as the job is marked done.

2.3 User Classes and Characteristics

2.3.1. Worker:

a. is expected to have basic knowledge of handling a smartphone or a computer
b. should be able to handle email notifications.

2.3.2. Administrator:

a. should have knowledge of handling databases
b. must have a deep knowledge of the software without any ambiguities.

2.3.3. Concerned Organization or User:

a. should have basic knowledge of computer
b. should be able to fill the basic details given in the form.

2.3.4. General Public:

a. should have enough knowledge to send feedback online in a browser.

2.4 Operating Environment

2.4.1. Operating System: Windows 7 or above, Ubuntu
2.4.2. Minimum RAM requirements: 512 MB
2.4.3. Disk Space: free space of 7 GB or above
2.4.4. Platform: Java

2.5 Design and Implementation Constraints

The data must be stored in a human-readable format in order to understand the data in times of technical difficulties.

2.6 User Documentation

2.6.1. Worker Manual
2.6.2. Administrator Manual
2.6.3. Customer Manual

All of these are available along with the software package.

2.7 Assumptions and Dependencies

We make the following assumptions:

2.7.1. The concerned organization is responsible for finding and adding new jobs. Our software is not responsible for finding new jobs in the city.
2.7.2. The software only records the wages to the workers and their calculation and distribution is to be handled by the municipality.

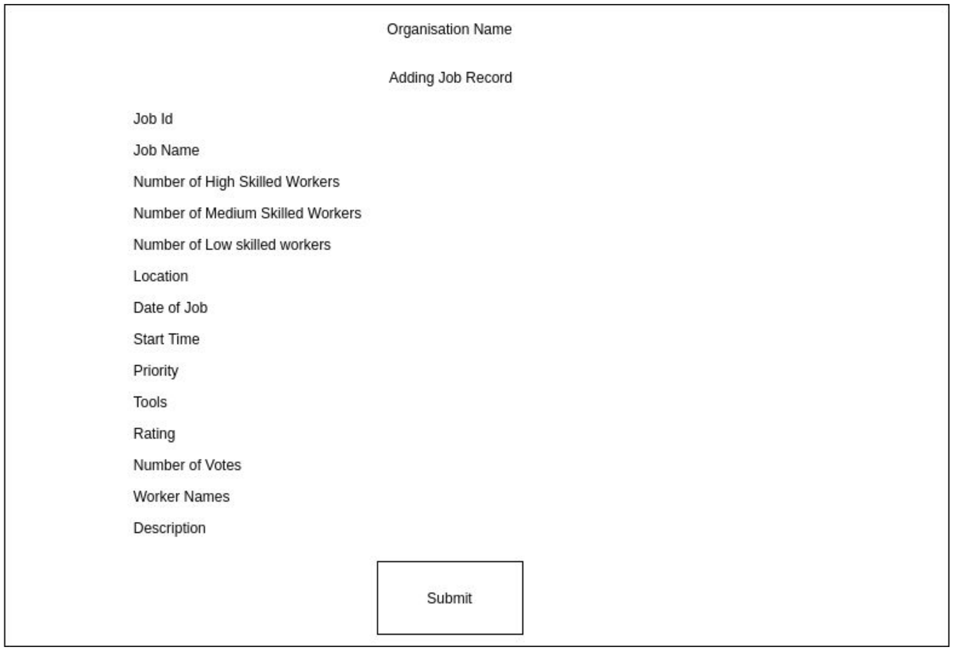
2.7.3. The workers are assumed to know simple english and are at least able to read the address sent in the notification.

3. External Interface Requirements

3.1 User Interfaces

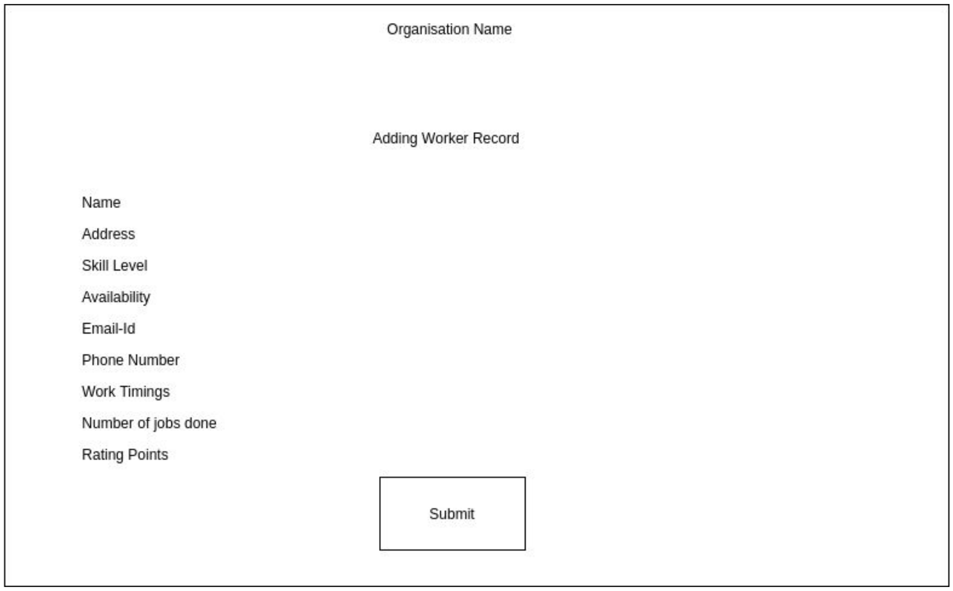
3.1.1.1. The users would view a GUI on their PC created using Java swing. The interface is user-friendly.
3.1.1.2. The concerned public can view a Browser User Interface (BUI) on their PC for giving feedback.

Rough GUI Examples given below -


Organisation Name

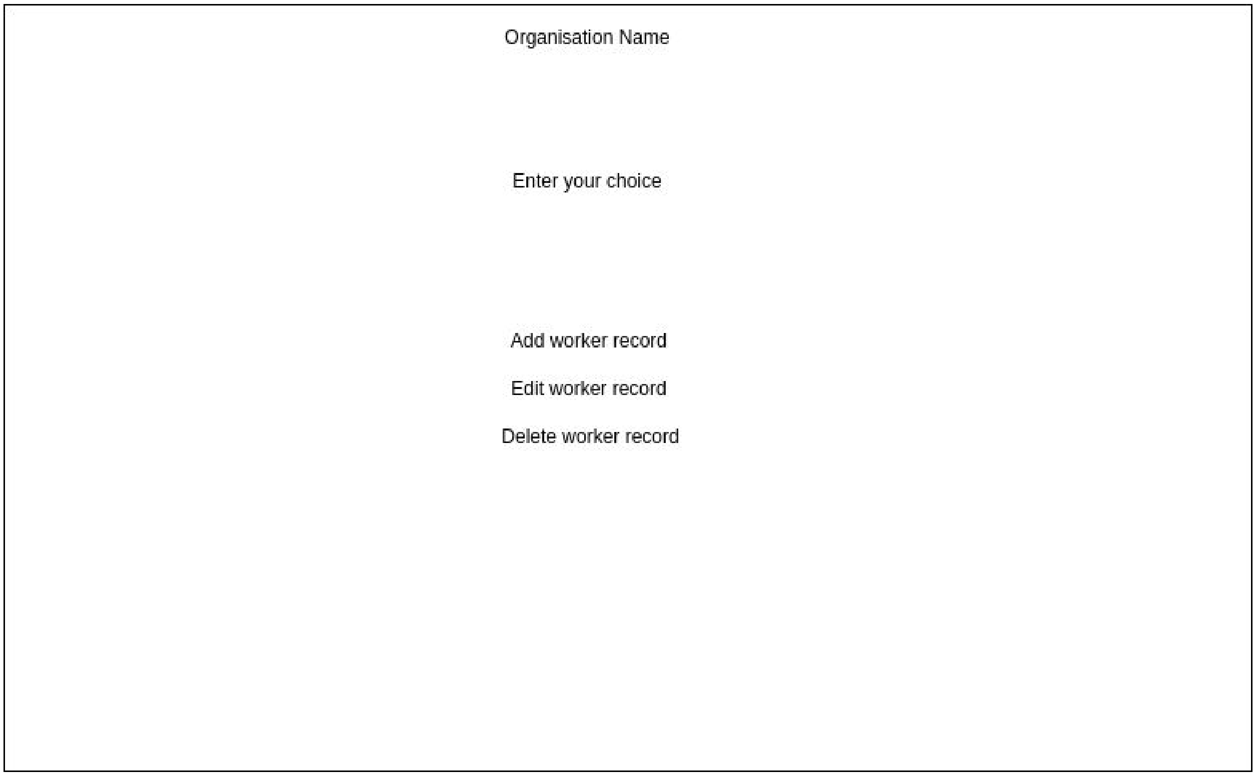
Adding Job Record

Job Id
Job Name
Number of High Skilled Workers
Number of Medium Skilled Workers
Number of Low skilled workers
Location
Date of Job
Start Time
Priority
Tools
Rating
Number of Votes
Worker Names
Description


Organisation Name

Adding Worker Record

Name
Address
Skill Level
Availability
Email-Id
Phone Number
Work Timings
Number of jobs done
Rating Points



3.2 Hardware Interfaces

3.2.1. Operating System: Windows 7 or above, Ubuntu 16.04 or above
3.2.2. Minimum RAM requirements: 512 MB
3.2.3. Disk Space: free space of 7 GB or above
3.2.4. Platform: Java

3.3 Software Interfaces

3.3.1. Our software uses Java Swing and other standard java libraries.
3.3.2. The worker's information is stored in text files.

3.4 Communications Interfaces

3.4.1. Information of the work to be done is sent through email to the workers.
3.4.2. Public feedback is sent through Browser based Interface and the server receives the rating and updates the rating of the job done and of the workers.

4. System Features

4.1 Maintaining record of workers:

4.1.1. Description and Priority

Priority: High

Description: We should be able to maintain the record of workers. The fields that are recorded have been specified in Section 2.2.1. We should be able to add, edit and delete these records.

4.1.2. Stimulus/Response Sequences

1. The system presents choice to user - to add, modify or delete a record, and user enters his choice.

1.a)The user chose ADD -

1.a.i) The system shows a form to the user, asking to fill all the details of a worker as specified in Section 2.2.1.

1.a.ii) The user enters details and the system verifies the format of these details. The system goes to 1.a.ii.a if if the details are not of appropriate data types, otherwise goes to 1.a.ii.b.

1.a.ii.a.) The system throws an error message “The details entered are inappropriate!” and goes back to 1).

1.a.ii.b) The system checks if a worker record exists in database with same email-id. The system displays an error message ”Email ID already exists!” if found, and returns to 1. Otherwise adds the entry to the database(with initial ratings points as 0 and skill ‘Low’) and displays “Successfully Added!” message.

1.b) The user chose MODIFY -

1.b.i) The system asks the user to enter name of the worker. The user enters the name, and the system searches the database for the worker record. If no entries are found, error message “No entry found” is displayed and system goes back to 1. Otherwise, if multiple entries are found, the system asks user to enter email-id. The system then searches for the worker.Once a unique worker is found, the system displays form of worker attributes(as specified in Section 2.2.1) and the user enters details.

1.b.ii) The user enters details and the system verifies the format of these details. The system goes to displays error “The details entered are inappropriate!” and goes to 1 if if the details are not of appropriate data types, otherwise 1.b.iia)

1.b.ii.a) The system checks if a worker record exists in database with same email-id. The system displays an error message”Email ID already exists!” if found, and returns to 1. Otherwise adds the entry to the database and displays “Successfully Updated!” message.

1.c) The user chose DELETE -

1.c.i) The system asks the user to enter name of the worker. The user enters the name, and the system searches the database for the worker record. If no entries are found, error message “No entry found” is displayed and system goes back to 1. Otherwise, if multiple entries are found, the system asks user to enter email-id. The system then searches for the worker.

Once a unique worker is found, the system displays the worker attributes(as specified in Section 2.2.1) and asks user confirmation with the message “Are you sure you want to delete this record?”. If user confirms, the record is deleted and system displays “Deleted successfully!” otherwise system displays “Did not delete the record!”. In both cases, the system goes back to 1.

4.1.3. Functional Requirements

Note: Here information means all the attributes of a worker as specified in (Section 2.2.1).

1. The user should be able to add information of a new worker.
2. The user should be able to edit information of a worker.
3. The user should be able to delete information of an existing worker.

4.2 Filling a new job:

4.2.1. Description and Priority

Description: The user should be able to fill information of a new job, which includes description of the job, start time, duration, deadline, no. of workers required, location, priority, tools required and wages.

Priority: Medium

4.2.2. Stimulus/Response Sequences

1. The system first asks the user the name of the job that needed to be added.

2. After the user enters a valid name, it asks the user the number of high, medium and low skilled workers required for the job.
3. If the number of workers demanded by the job are greater than the number of available workers, the system shows an error message stating that the number of workers have exceeded the limit and also displays the number of available high, medium and low skilled workers on the screen.
4. Otherwise, the system asks the location of the job that need to be performed
5. Then it asks for the priority of the job.
6. Now it displays on the screen that the job has been successfully added and the workers will be notified.

4.2.3. Functional Requirements:

1. The user should be able to fill information about a new job.
2. The user should be able to mark completed tasks.

4.3. Scheduling and Managing Notifications

4.3.1. Description and Priority:

Description: For each unassigned job, the system should assign free workers automatically, and notify the workers through an email. The assignment is done based on various measures such as closeness, skill level, number of workers required.

Priority: High

4.3.2. Stimulus/Response Sequences

Here the system does not require inputs from user. This is automatically done.

4.3.3. Functional Requirements:

1. The system should schedule workers efficiently giving priority to closeness of workers. In case of similar closeness, the system should make assignments of workers to jobs such that jobs with higher priority is fulfilled.
2. The system should send mails to workers about the jobs assigned to them.

4.4. Taking Public Feedback

4.4.1. Description and Priority:

Description: After the completion of each job, feedback is taken from some chosen people in the vicinity of the job location, They are asked to rate it out of 5 based on the satisfaction of the individual on the work performed. They will enter the feedback (rating it out of 5) in a Browser Interface and will submit it. They can also add remarks about the job done (optional).

Priority: Medium.

4.4.2. Stimulus and Response Sequences:

1. The concerned person of that area will first rate the job out of 5 and add any remarks (optional) and submits it through a Browser based Interface.
2. Then the system receives the rating and remarks(if any) and updates the average rating of the job and the rating points of the workers involved in the job.
3. The system then checks if any worker has his skill level upgraded or downgraded and sends him notification regarding the change in his level.

4.4.3. Functional Requirements:

1. The system should receive ratings from the public and update the average job rating.
2. It then updates the ratings of the workers ie., it adds (average rating-3)*5 to the rating points of every worker of the job done .
3. It sends an email to the workers involved in the job and whose skill level has changed showing them their updated rating points and skill level.

4.5. Recording details of previously done jobs

4.5.1. Description and Priority:

Description: For each completed job, we store the amount spent on the tools required and the wages given to the workers. The software will update this information in a file as soon as the job is marked done.

Priority: Low

4.5.2. Stimulus/Response Sequences

1. The user selects a job and mark it done.
2. The system writes the information about the completed job in a file. The following information is written -

   i) Job Name
   ii) Number of high skilled workers required
   iii) Number of medium skilled workers required
   iv) Number of low skilled workers required
   v) Location
   vi) Priority of the job
   vii) Tools required and cost
   viii) Wages given to workers
   ix) Start time of the job
   x) End time of the job

4.5.3. Functional Requirements:

1. The system should ask the user to select the job he wants to mark as completed and display list of all assigned jobs.

2. The system should automatically prepare a report of the completed job in a text file.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. The scheduling and managing notifications(System feature 4.3) must be fast(with execution time ~ 5 secs) as there can be hundreds of urgent jobs(especially in case of disaster).
2. The system should be efficient in scheduling(System feature 4.3) so that maximum number of jobs must be assigned of high priority with workers as close as possible to the location of the job.

5.2 Safety Requirements

1. We should make sure that the system records public feedback for all jobs(System feature 4.4).
2. We should make sure that the reports for completed jobs are accurate(System feature 4.5).

5.3 Security Requirements

1. We should make sure that the data supplied for workers and jobs(System features 4.1 and 4.2) are appropriate without ambiguity or inconsistency or incompleteness. For example - if two workers have same email id in database, both of them would receive email and that would cause undesired outcomes.

5.4 Software Quality Attributes

1. The software should be user-friendly to all the users as general public is involved in it.
2. The software should be reliable as it is large scale and used in a whole city or state.
3. The software should have simple english so that even workers can understand.

5.5 Business Rules

In case the system is down due to unforeseen circumstances, the system administrator is responsible for making sure that any manual changes to the database are syntactically proper.

6. Other Requirements

Since the project is aimed to be used in different domains such as cities, colleges,campuses etc, the language used in the project should be general and not domain-specific.

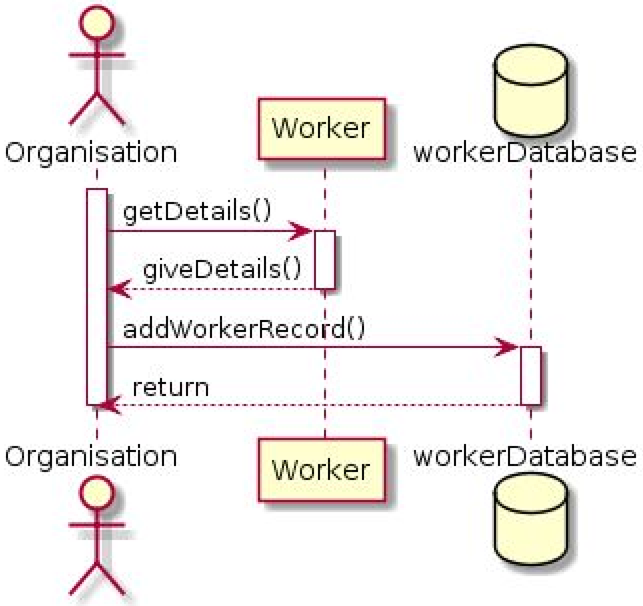
Similarly, the project is aimed to be useful country-wide, therefore terms should be general and not region-specific.

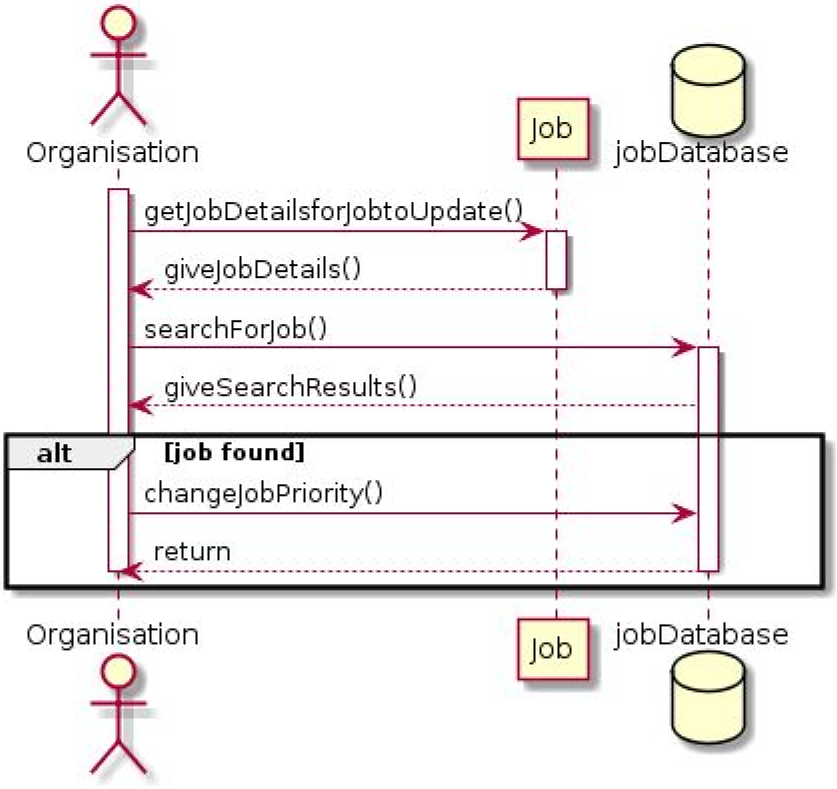
The database must be stored in a clean format such that the contents can be accessed and changed manually if required.

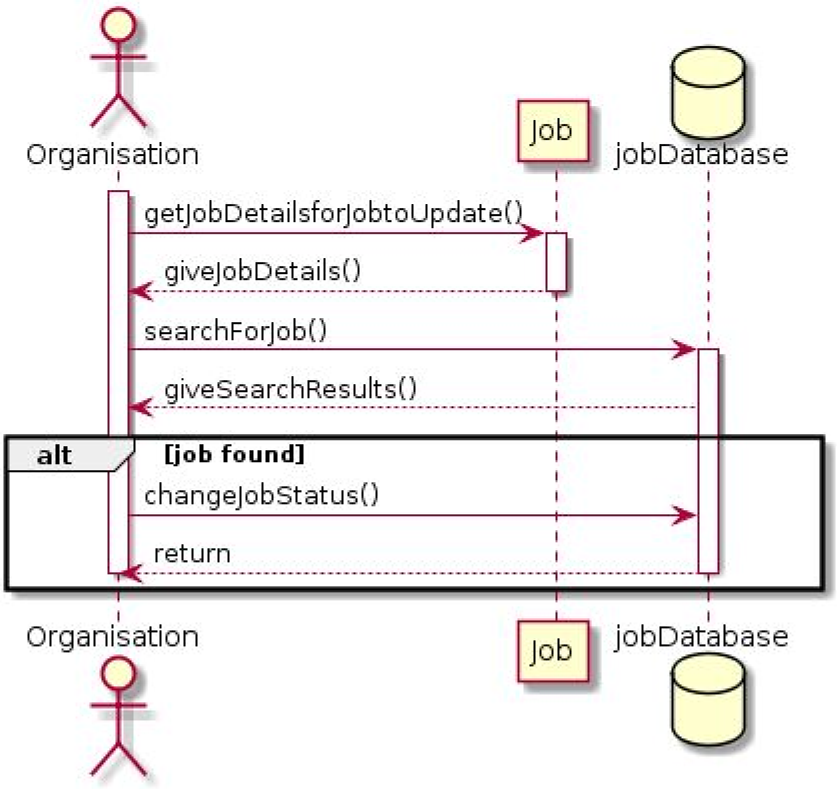
Appendix A: Glossary

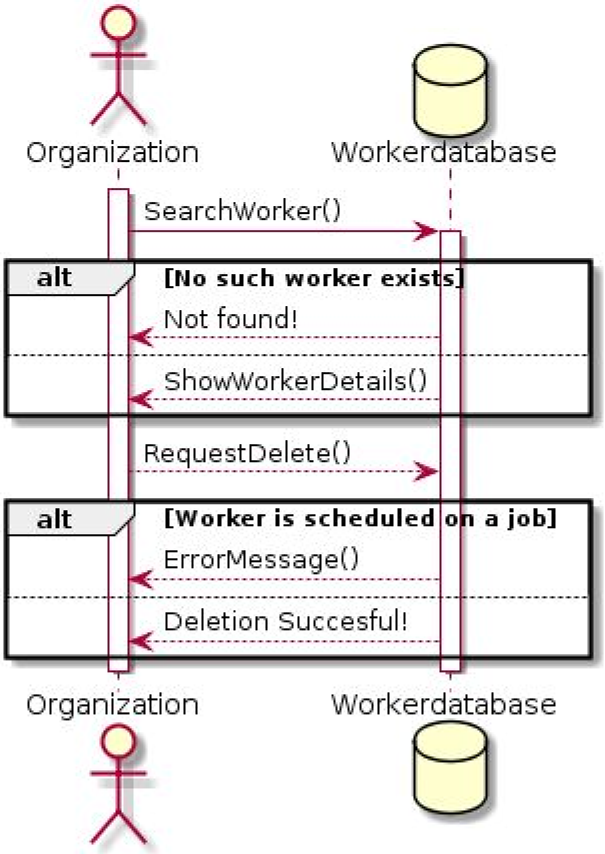
WMS - Work Management System
SRS - Software Requirements Specification
DB - Database
GB - Gigabyte
MB - Megabyte
Ambiguous - not clear or decided
Consistent - not containing any logical contradictions
Swaccha Bharat Abhiyaan - Swachh Bharat Abhiyan or Swachh Bharat Mission is a nation-wide campaign in India that aims to clean up the streets, roads and infrastructure of India's cities, towns, and rural areas.
City Municipality - A municipal corporation, city corporation, Mahanagar Palika, Mahanagar Nigam or Nagar Nigam or Nagara Sabha is a local government in India that administers urban areas with a population of more than one million. The growing population and urbanization in various cities of India were in need of a local governing body that can work for providing necessary community services like health care, educational institution, housing, transport etc
NGO - Non-Governmental Organisation
Windows - An operating system developed, marketed and sold by Microsoft
Ubuntu - A free and open-source Linux distribution
Java - Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.
GUI - Graphical User Interface
RAM - Random Access Memory

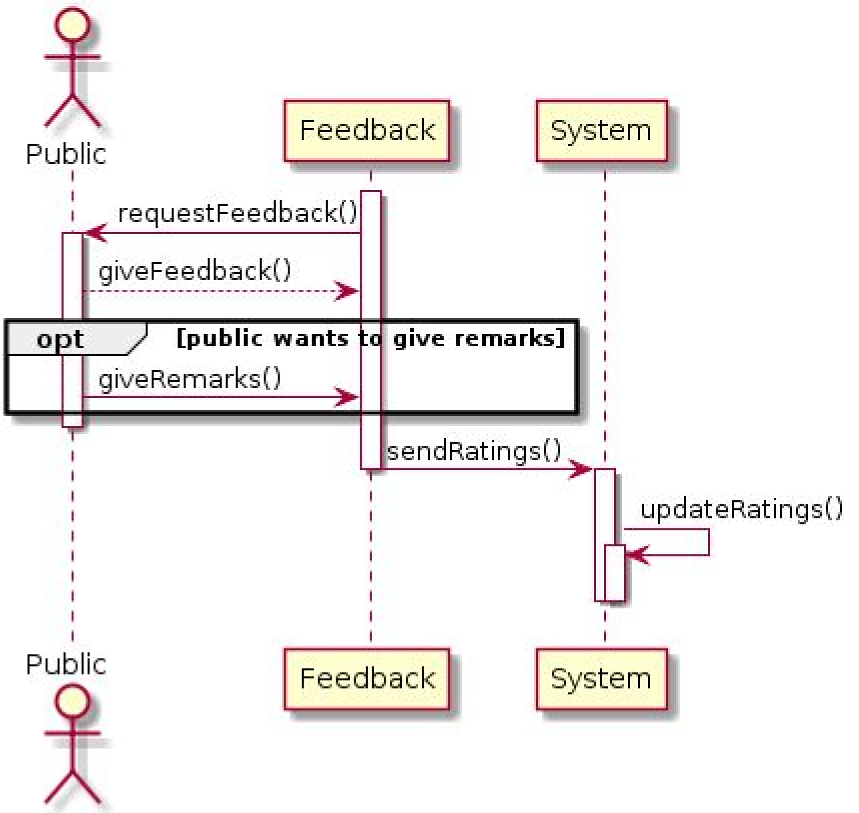
Appendix B: Analysis Models

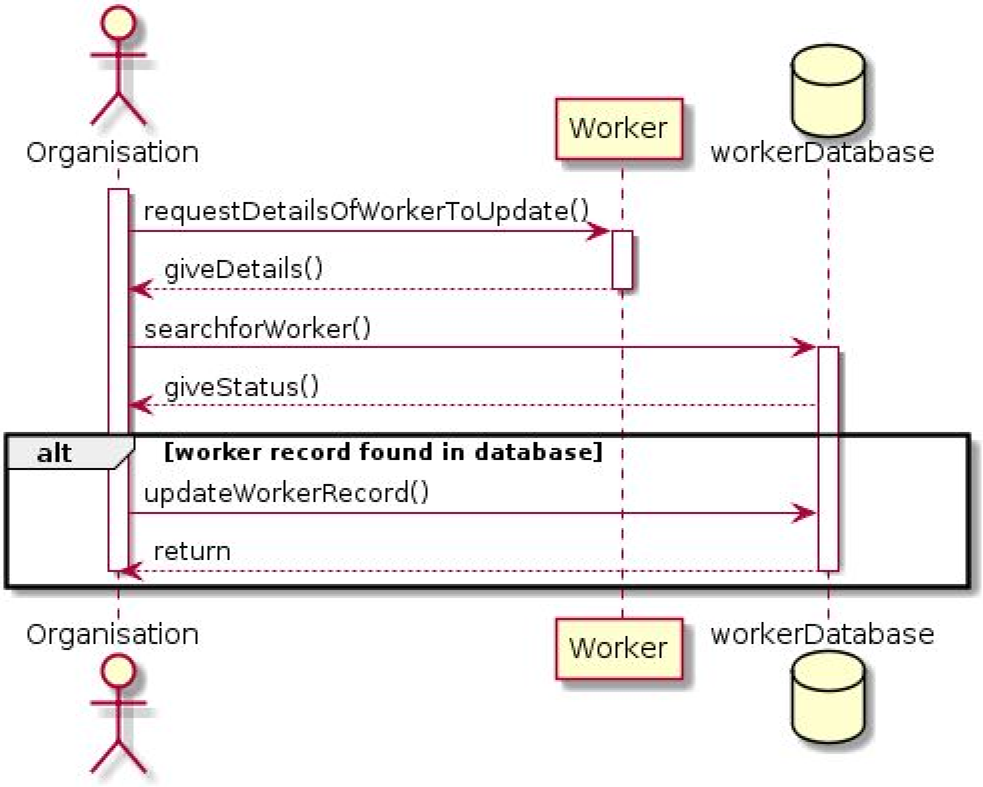


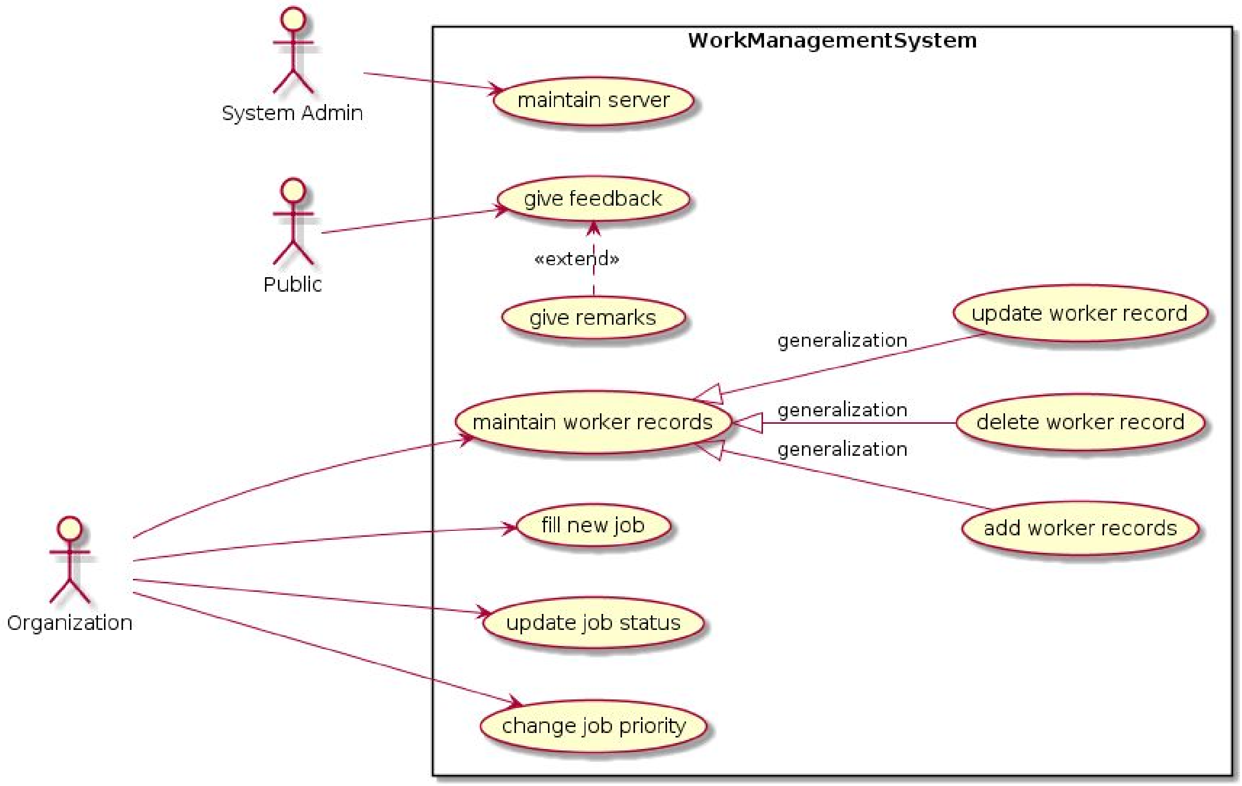


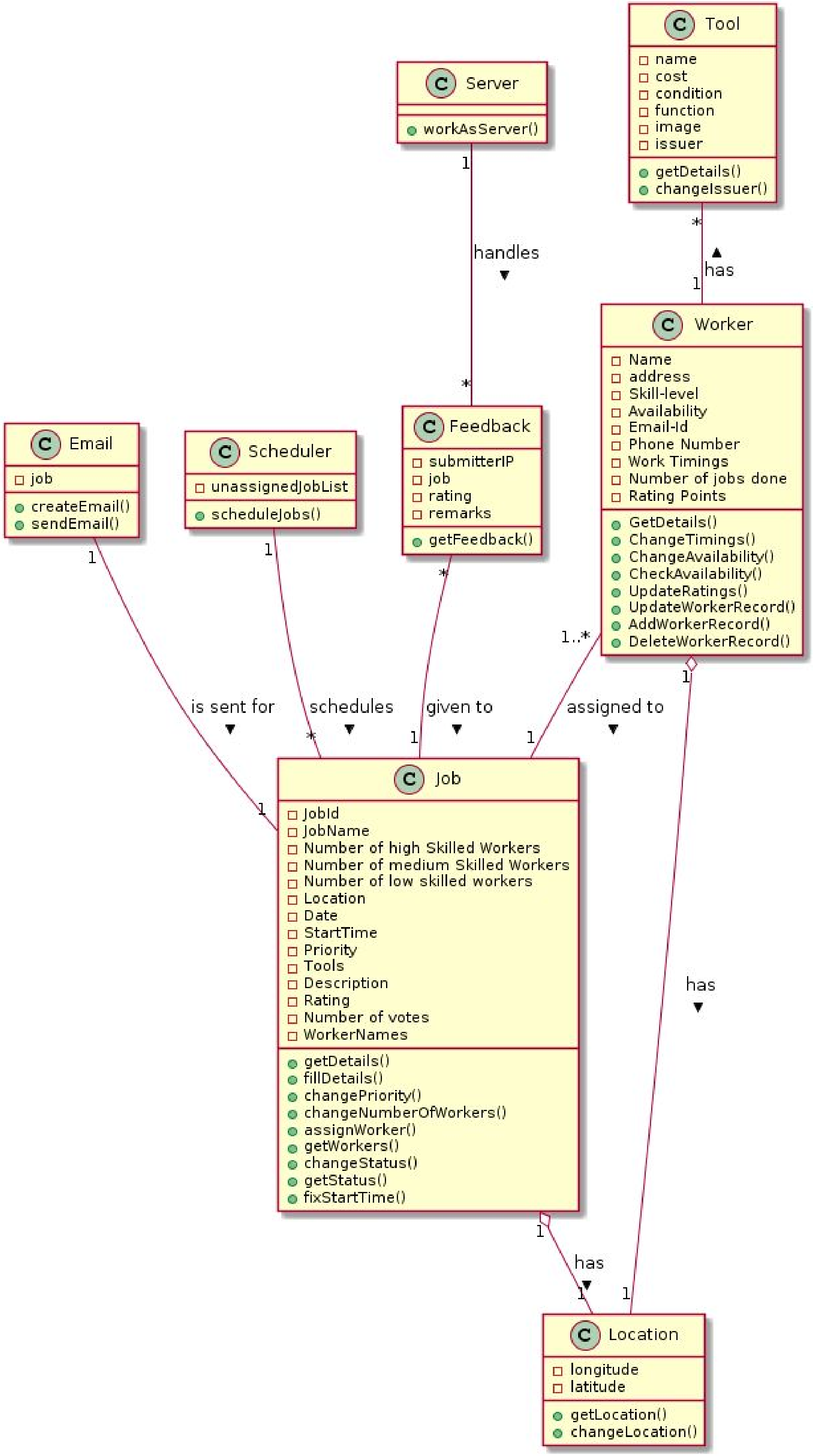

Organisation
Job
jobDatabase
getJobDetailsforJobtoUpdate()
giveJobDetails()
searchForJob()
giveSearchResults()
alt
[job found]
changeJobStatus()
return
Organisation
Job
jobDatabase


Organization
Workerdatabase
SearchWorker()
alt
[No such worker exists]
Not found!
ShowWorkerDetails()
RequestDelete()
alt
[Worker is scheduled on a job]
ErrorMessage()
Deletion Succesful!
Organization
Workerdatabase


Public
Feedback
System
requestFeedback()
giveFeedback()
opt
[public wants to give remarks]
giveRemarks()
sendRatings()
updateRatings()
Public
Feedback
System


Organisation
Worker
workerDatabase
requestDetailsOfWorkerToUpdate()
giveDetails()
searchforWorker()
giveStatus()
alt
[worker record found in database]
updateWorkerRecord()
return
Organisation
Worker
workerDatabase


WorkManagementSystem
System Admin
Public
Organization
maintain server
give feedback
«extend»
give remarks
update worker record
generalization
maintain worker records
generalization
delete worker record
generalization
add worker records
fill new job
update job status
change job priority


C Tool
name
cost
condition
function
image
issuer
getDetails()
changeIssuer()
C Server
workAsServer()
1
handles ▼
*
*
▲ has
1
C Worker
Name
address
Skill-level
Availability
Email-Id
Phone Number
Work Timings
Number of jobs done
Rating Points
GetDetails()
ChangeTimings()
ChangeAvailability()
CheckAvailability()
UpdateRatings()
UpdateWorkerRecord()
AddWorkerRecord()
DeleteWorkerRecord()
C Email
job
createEmail()
sendEmail()
C Scheduler
unassignedJobList
scheduleJobs()
C Feedback
submitterIP
job
rating
remarks
getFeedback()
1
1
*
1..*
1
is sent for ▼
schedules ▼
given to ▼
assigned to ▼
*
1
1
1
C Job
JobId
JobName
Number of high Skilled Workers
Number of medium Skilled Workers
Number of low skilled workers
Location
Date
StartTime
Priority
Tools
Description
Rating
Number of votes
WorkerNames
getDetails()
fillDetails()
changePriority()
changeNumberOfWorkers()
assignWorker()
getWorkers()
changeStatus()
getStatus()
fixStartTime()
has ▼
1
has ▼
1
1
C Location
longitude
latitude
getLocation()
changeLocation()

Appendix C: To Be Determined List

None