Software Requirements Specification

for

Maintenance Automation

Version 2.0 approved

Prepared by

Aneesh Khunteta (20UCS020)

Mihir Mishra (20UCS118)

Kaustubh Nitin Narwade (20UCS095)

Vinay Sunil Bhagwat (20UCS227)

LNMIIT

24/11/2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
v2.0	22/11/2022	Updating of project functionalities and due to the limitations of the knowledge in some users.	2.0

1. Introduction

1.1 Purpose

The software package is developed from scratch exclusively for The LNM Institute of information Technology, to:

- Provide an online system through which users can easily register their complaints related to maintenance online.
- The *AMO* can assign tasks to the engineer staff.
- AMO would be able to create and delete new users and engineer staff accounts.
- This system will help in resolving maintenance related issues quickly without any hassle.

1.2 Document Conventions

The following documentation conventions are followed in preparing this SRS:

- All keywords related to the maintenance are formatted in italics.
- SRS-Software Requirement Specification.
- The Asset and Maintenance Officer is abbreviated to *AMO*.

1.3 Intended Audience and Reading Suggestions

This document is intended to: -

- The software development team for their use in analyzing the requirements.
- College administration staff to note down the complaints and take required actions regarding the issues.
- Asset and maintenance officer to track down all requests.
- Users can file the complaints.

1.4 Product Scope

The scope of the to-be-developed Maintenance automation software package is:

- To entertain all maintenance complaints regarding any area within campus premises.
- To cater a need for a suitable interface between the *complainant* and *maintenance staff*.
- To facilitate *AMO* to know the maintenance history.
- To bring transparency in the process of use of the budget incorporated during the time of maintenance.

1.5 References

The following references are used while preparing this SRS:

- The SRS Template and Sample sent by Prof. Vikas Bajpai and Dr. Shweta Saharan.
- Wikipedia
- IEEE guidelines for software requirements specifications writing

2. Overall Description

2.1 Product Perspective

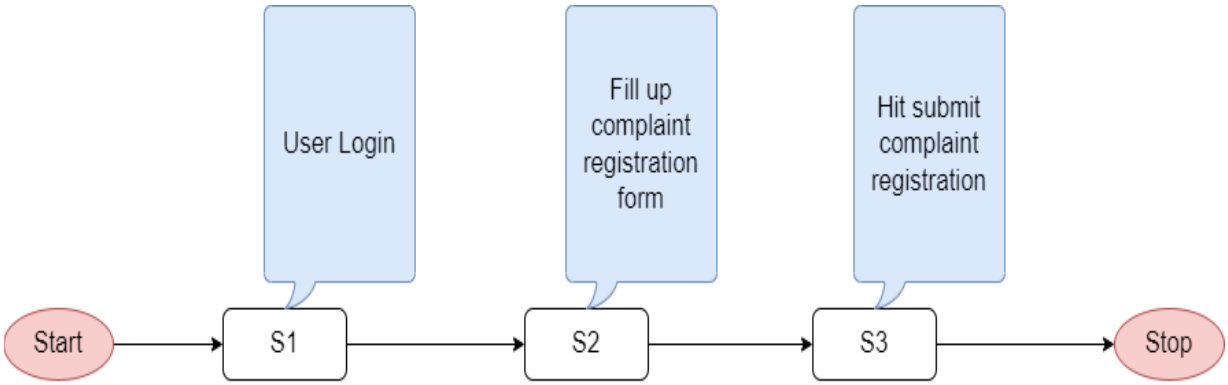
This is intended to replace the current complaint section of MIS portal as an independent complaint portal. it is not a follow-on member of any other product family

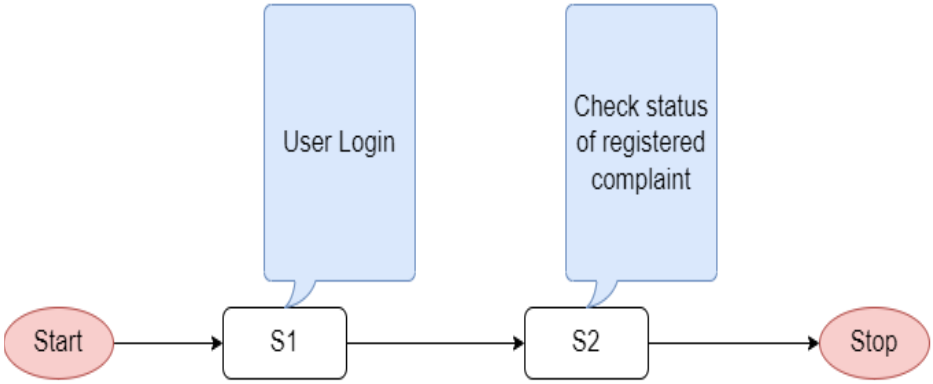
Our website will have a daily updated database of maintenance complaints. This includes:

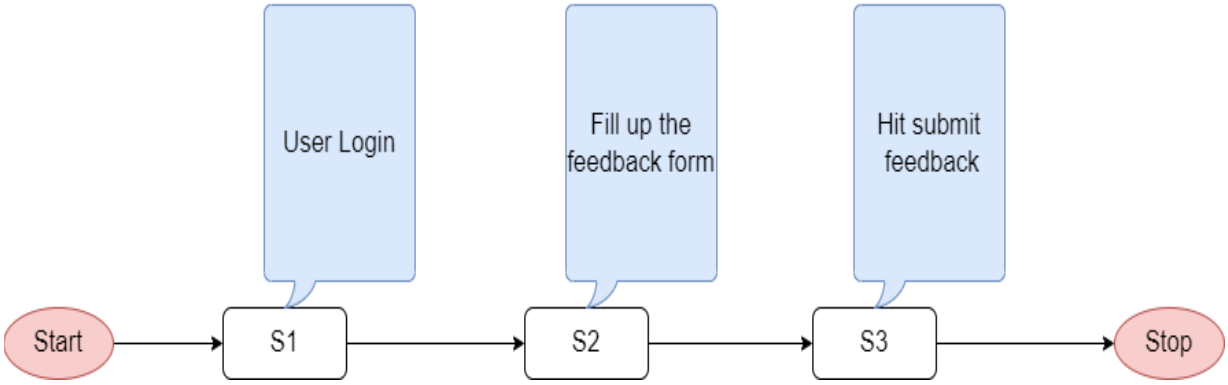
- Complaints from all the areas:
  - Hostel
  - Academic area
  - Canteen/Mess
  - Sports Facilities/GYM
  - Teacher’s Quarters
  - Miscellaneous (Medical Centre, College Gate)
- Separation of the accounts of college maintenance authorities from other accounts.

The following diagrams describes the high-level business process of the maintenance automation system:

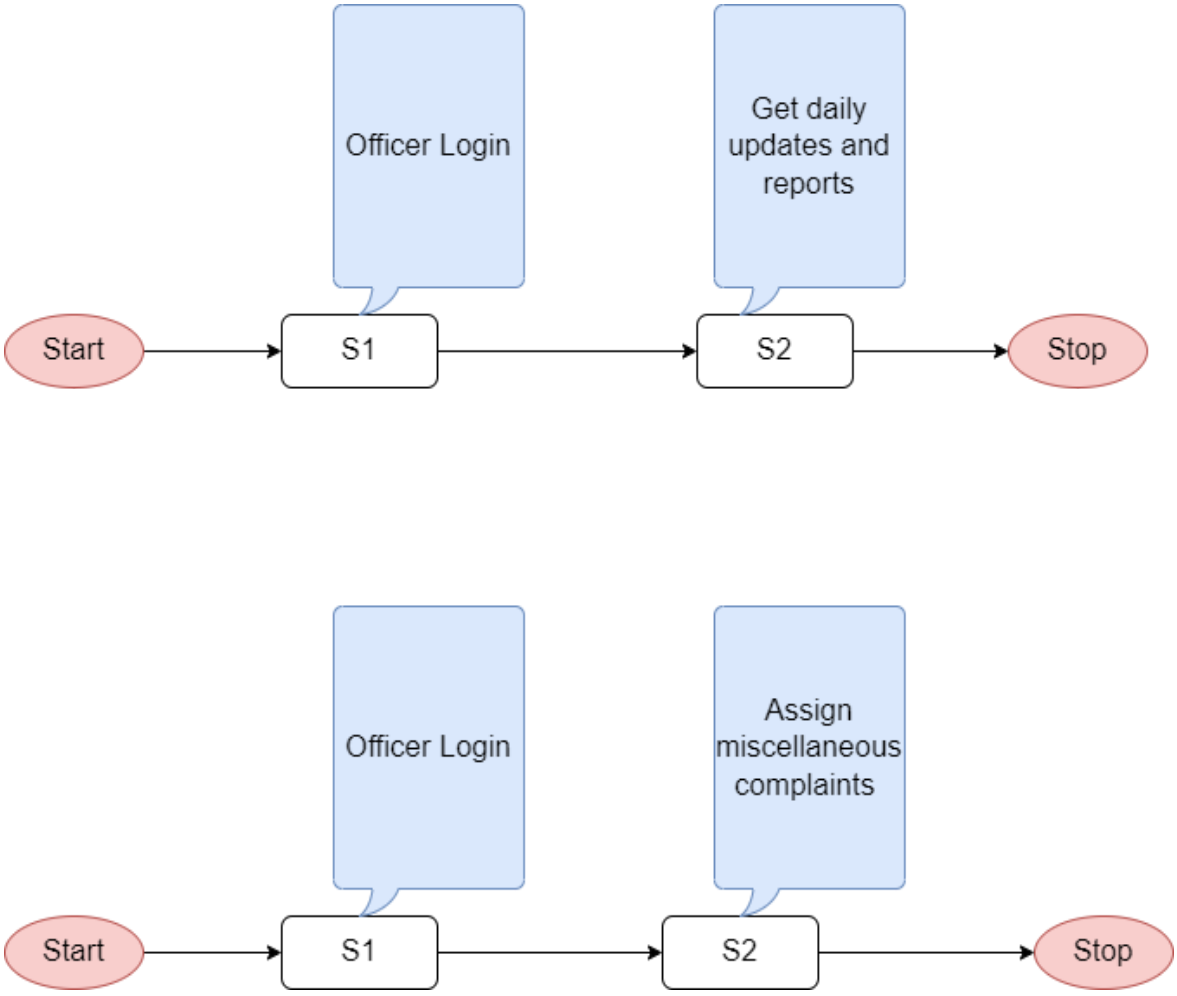
Maintenance Automation System Part 1: User



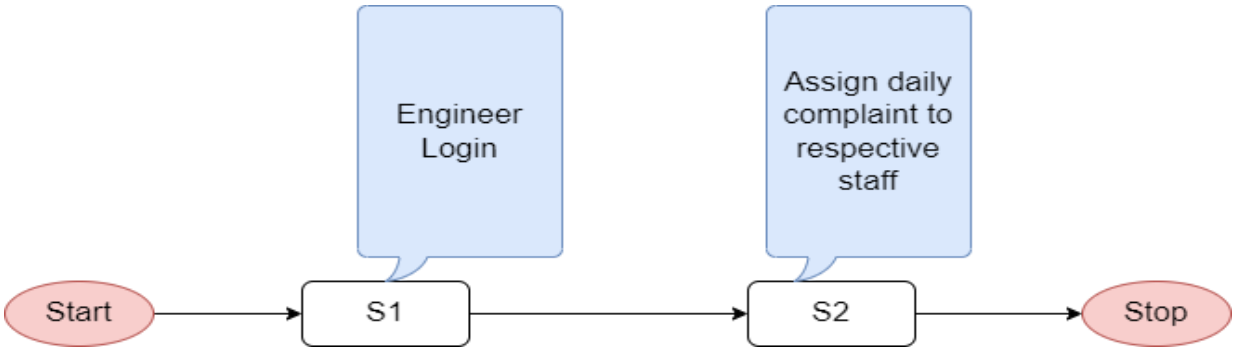




Maintenance Automation System Part 2: *AMO*



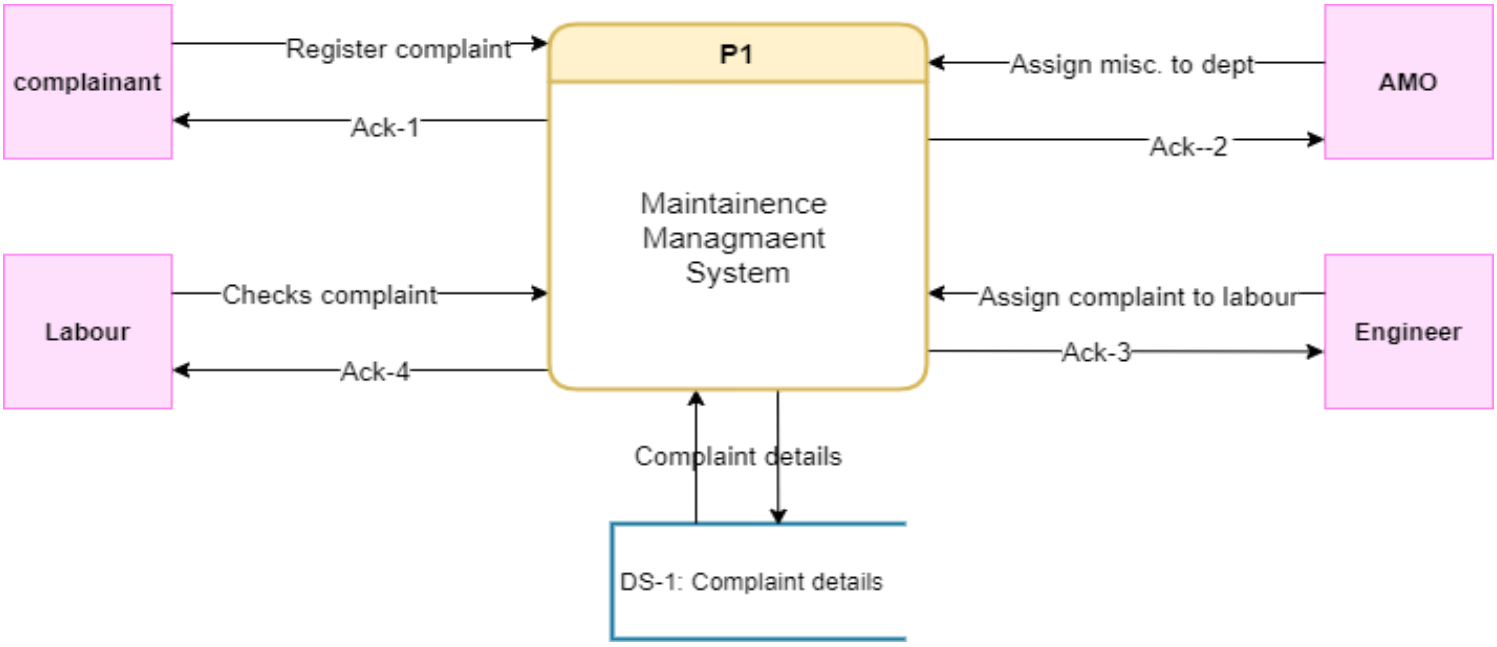
Maintenance Automation System Part 3: *Engineer*



2.2 Product Functions

The portal is going to provide following functionalities:

- Log in to the portal as a *complainant* or *maintenance staff*.
- It would let the *complainant* use the following functionalities:
  - Fill a fresh complaint
  - Check the previously filed complaint status
- The *AMO* will use the following functionalities:
  - Check the status of the problems.
  - Allot the complaint to the engineer.
- It would let the College *Engineer staff* use the following functionalities:
  - see their department specific complaints and assign them to the *labourers*.



2.3 User Classes and Characteristics

We have identified the following user classes -

- *Complainan*t: These users will file the complaint and choose the department to which it should be reported along with the details regarding the complaint.
- *AMO*: this user will be able to see all the complaints and sort them according to the requirements and can also view the feedback of the complaints. This user will allocate the miscellaneous complaints to the concerned department.
- College *Engineer*: These users will be able to see their department specific complaints and assign them to the *labourers*.

2.4 Operating Environment

The portal will be deployed on a server with high-speed Internet capability. The physical machine to be used will be determined by the college Administration. The software developed here assumes the use of a tool such as XAMPP for providing a PHP runtime environment and Apache and MySQL server. The speed of the user connection will depend on the hardware used rather than characteristics of this system. To open the webpage at the user's end, the user must have a compatible browser installed (such as Google Chrome v106, Mozilla Firefox v105).

2.5 Design and Implementation Constraints

The Design time constraints are:

- The information of all users and complaints must be stored in a database that is accessible by the website.
- MySQL will be used as the database.
- The Online Maintenance Automation system is running 24 hours a day.
- Users may access from any computer that has Internet browsing capabilities and an Internet connection.
- Users must have their correct usernames and passwords to enter their online accounts.
- The users must have basic knowledge of the English language to understand the portal.

2.6 User Documentation

- A tutorial video for every user's perspective will be provided along with the software package.
- The details of the Analysis, Design and Test Cases of this software package will be delivered along with this software.

2.7 Assumptions and Dependencies

The assumptions made for this to-be-made software are -

- The college will have sufficient funds, *maintenance staff* to resolve all the complaints.
- The users will be technically literate as well as have basic knowledge of the English language.
- The server will be available all the time except for maintenance / updating.

3.External Interface Requirements

3.1 User Interfaces

The set of User Interfaces are -

- To log into their respective accounts by users.
- To file the complaint by the *complainant*.
- To track down the status of the complaint.
- To see the department specific complaint for the *engineer* staff.
- To see all the complaints in the Database.
- Let AMO assign the complaints to the respective departments.

3.2 Hardware Interfaces

Since the application must run over the internet, all the hardware required to connect to the internet will be hardware interface for the system. As for e.g., Router, Wi-Fi, PC.

3.3 Software Interfaces

The Below table gives a description of each piece of software that will be used in making this website.

Software Used	Description
Database	We have chosen a relational database in MySQL. Its strong relational structure gives us a strong foundation to build our project.
HTML	The Hypertext Markup Language allows us to give structure to our webpage. It defines the skeleton of the webpage and
PHP (MyPHPAdmin)	PHP is directly embedded into HTML, so it gives us a solid work environment. It is a contemporary programming language with rich features.
CSS (Bootstrap)	CSS makes our HTML page easy to use and gives it beautiful workflow.

3.4 Communications Interfaces

The maintenance automation system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

4.SYSTEM FEATURES

The requirements of this software package are described per each category of user:

- All requirements of the maintenance automation portal for the LNMIIT *AMO*.
- All requirements of the maintenance automation portal for the LNMIIT *engineer* staff.
- All requirements of the maintenance automation portal for the LNMIIT *complainant*.

Business Use Case#1: All the Requirements of The LNMIIT *AMO*:

4.1 Checking the complaints and allocating miscellaneous complaints:

4.1.1 Description and Priority

The to-be-developed software should facilitate LNMIIT *AMO* to:

- View all the complaints and their feedback as per the requirements.
- Assign miscellaneous complaints to the concerned departments.

4.1.2 Stimulus/Response Sequences

Sr. No.	Stimulus from the User	Response from the software
1	AMO logs-in using user id and password	Software will validate the user-id and password. Software will display the AMO initial screen by recognising the user-id.
2	AMO chooses the view complaints option	Software will display the form to choose the filters and will show the complaints as per the filters selected (or will prompt a message in case the database is found empty).
3	AMO selects a complaint	Software will display all the details associated with that complaint (status, date of complaint, date of resolution, name of labour, feedback (if any)).
4	AMO chooses to view all the newly filed miscellaneous complaints	Software will display all the unallocated miscellaneous complaints.
5	AMO selects an unallocated miscellaneous complaint	Software would give all the details of the complaints along with the description given by the complainant. Software would open a form asking AMO to select the department to which the complaint must be assigned and assign the complaint to the chosen department.
6	AMO choose to logout	Software would logout AMO.

4.1.3 Functional Requirements

- *AMO* should be able to log in using the username and password and software must be able to verify it and must open the *AMO* module.
- The software must be able to fetch the data from the database and show the results as per the filters.
- The software must inform whenever the required data is absent.
- The software must directly assign the complaints as directed by *AMO*.

Business Use Case#2: All the Requirements of The LNMIIT *Complainants*:

4.2 Filing the complaints and giving the feedback for resolved complaints

4.2.1 Description and Priority

The to-be-developed software should facilitate LNMIIT complainants to:

- View their complaints and check the status.
- give the feedback for resolved complaints.

4.2.2 Stimulus/Response Sequences

Sr. No.	Stimulus from the User	Response from the software
1	User logs-in using user id and password	Software will validate the user-id and password. Software will display the complainant initial screen by recognising the user-id.
2	Users choose to file a complaint	Software will display a form in which User will be able to give all the details and submit it.
3	User chooses the view complaints option	Software will display the complaints filed by the user along with its status.
4	User selects a complaint	Software will display all the details associated with that complaint.
5	User chooses to give feedback on a resolved complaint.	Software will display a textbox to fill the feedback with a submit option.
6	Users choose to logout	Software would logout User.

4.2.3 Functional Requirements

- Users should be able to log in using the username and password and software must be able to verify it and must open the User module.
- Software must give the user selected options to choose for destination rather than taking a custom input.
- The software must be able to fetch the data from the database and show the User specific complaints.
- The software must inform whenever the required data is absent.

Business Use Case#3: All the Requirements of The LNMIIT *Engineer staff*:

4.3 Assign the complaints to the *labour*

4.3.1 Description and Priority

The to-be-developed software should facilitate LNMIIT *Engineer staff* to:

- View their department specific complaints and assign them to the *labourers*.

4.3.2 Stimulus/Response Sequences

Sr. No.	Stimulus from the User	Response from the software
1	Staff logs-in using user id and password	Software will validate the user-id and password. Software will display the Staff initial screen by recognising the user-id.
2	Staff chooses the view complaints option	Software will display the Department specific complaints.
3	Staff selects a complaint	Software will display a form with the list of labourers to select and submit buttons.
4	Users choose to logout	Software would logout User.

4.3.3 Functional Requirements

- Staff should be able to log in using the username and password and software must be able to verify it and must open the Staff module.
- Software must give the user selected options to choose for *labour* rather than taking a custom input.
- The software must be able to fetch the data from the database and show the department specific complaints.
- The software must inform whenever the required data is absent.

5. Other Nonfunctional Requirements

5.1. Performance Requirements

- At least 20 *maintenance staff* can log in on an average of four hours a day for five days a week.
- At least 50 students can log into their accounts for 3 hours for 5 days of the week.
- It should be able to handle the MySQL database of 20 *maintenance staff* and 2500 students.

5.2 Safety Requirements

- The database should be protected at the server's end.
- The software should be run in a system with minimum specifications otherwise it may harm the user experience.
- The server should be compatible to hold the software's needs.
- A proper scheduling software should be employed on the server so that the I/O request and output responses remain managed on both ends.

5.3 Security Requirements

- Firewall must be installed so that any attack on the system can be handled beforehand.
- All pages and resources require authentication except those specifically intended to be public.
- all password fields do not echo the user's password when it is entered, and that password fields (or the forms that contain them) have autocomplete disabled.
- sessions are invalidated when the user logs out.
- users can only access URLs for which they possess specific authorization.

5.4 Software Quality Attributes

- **availability:** website will be 24x7 available for all of its users
- **adaptability:** in order to get people acquainted with the to-be-developed website, we will list out the use cases and the robustness of the solutions which we are able to deliver by comparing it with the earlier products.
- **Scalability:** the system will always have space for accommodating multiple users at any given time and if the number of users increases in real time, allocation of additional resources can be done in easy steps.
- **Maintainability:** the system will always have the previous best performing snapshot. So, if at any given time the system fails/crashes due to unavoidable reasons it can always be rolled back to that previous snapshot or additional updates can always be applied in maintenance activities to avoid any such upcoming failures in near future.

- **Usability:** the software would let the users choose the destination of complaint only from the given options which would help in indicating it faster and help the *labour* to locate the site easily.

5.5 Business Rules

We have the following rules -

- Any account except the maintenance staff can register a complaint.
- The engineer staff will only see the complaints related to their department.
- Only AMO can see the complaints of all departments.

6. Other Requirements

The server should be able to keep up with MySQl's fast response time. This website will be available to access across the world wide web but will be registered in India and as such will have to comply with only local laws.

Appendix A: Glossary

The table below lists the acronyms and abbreviations used in this document.

Acronyms	**Meaning**
AMO	Asset and Maintenance Officer
SRS	Software Requirements Specification
PHP	Personal Home Page
MIS	Management Information Systems
HTML	Hyper Text Markup Language

Appendix B: Analysis Models

None

Appendix C: To Be Determined List

None