Software Requirement Specification (SRS) for Complaint Management System (CMS)

1. Introduction

1.1 Purpose:

Complaint Management System (CMS) is intended to enable easy accessibility to a shared platform for the users to raise complaints and also effectively monitor and track the lodged complaints thereby prompting necessary action by the management.

This document is meant to delineate the features of CMS, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

1.2 Scope:

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

In-scope :

- Announcements about newly resolved issues
- Provision of shared platform for smooth functioning
- Streamlining complaints - Maintenance of dashboard and complaint history
- Ensuring anonymity of complainer
- User/admin authentication and profile storage.

Out-scope :

- Filtering complaints as spam
- Reporting users as spam

1.3 Definitions, Acronyms, and Abbreviations:

Acronyms and Abbreviations :

- CMS - Complaint Management System
- SRS - Software Requirements Specification.

1.5 Overview:

The rest of this SRS is organized as follows: Section 2 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software and some assumptions and dependencies that are assumed. Section 3 gives specific requirements which the software is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given. Section 4 gives some possible future extensions of the system.

2. Overall Description:

2.1 Product Perspective:

The goal of CMS:CMS should be user-friendly and reliable software for the above purpose. If managed effectively, complaints can help businesses grow and improve its operations and helps to gain a better understanding of clients.

It should run on both UNIX and Windows based platforms.

2.2 Product Functions:

CMS should support the following use cases:

Use cases for Complainant		
Class of Use cases	**Use cases**	**Description of use cases**
Use case related to Installation	Installation	Creates and initializes working files.
Use cases related to authorization	Login	Login to CMS with user-id and password
	Register	Create an account
	Forgot Password	Change password in case of forgotten password
Use cases related to Account Settings	Profile Info	Enter/Update personal information
	Change Password	Enables password change
Use cases related to Lodging Complaint	Lodging a complaint	Write a category-based complaint

Use cases related to complaint history	Complaints history	List of complaints lodged by the user with the status of complaints
Use cases related to Announcements	Resolved major complaints	List of all resolved complaints written by many users
Use case related to sign-out	Signout	End session of user

Use cases for Admin		
Class of Use cases	**Use cases**	**Description of use cases**
Use case related to Installation	Installation	Creates and initializes working files.
Use cases related to authorization	Login	Login to CMS with user-id and password
	Register	Create an account
	Forgot Password	Change password in case of forgotten password
Use cases related to Account Settings	Profile Info	Enter/Update personal information
	Change Password	Enables password change
Use cases related to Resolving Complaint	Resolving a complaint	Write a response for complaints received
Use cases related to Announcements	Resolved major complaints	List of all resolved complaints written by many users
Use cases related to complaint history	Complaints history	List of complaints received by the admin with the status of complaints

2.3 User Characteristics:

a. Users should be familiar with the category the complaint belongs to.

b. Users should be part of the system/organization.

2.4 Principal Actors:

The three principal actors in CMS are “user”, “admin” and “system”.

2.5 General Constraints:

a. For full working CMS requires Internet connection.

b. CMS is multi-user software.

2.6 Assumptions and Dependencies:

a. Assuming that there are only valid complaints and users.

3. Specific Requirements:

3.1 Functional Requirements:

We describe the functional requirements by giving various use cases.

Use case related to installation:

Use Case 1: Installation.

Use cases related to authorization:

Use Case 2: Login

Primary Actor: User/Admin

Pre Condition: Nil

Main Scenario:

1. Load the website. User/Admin prompted for login and password.
2. User/Admin gives the login and password.
3. System does authentication.
4. Main screen is displayed.

Alternate Scenario:

4(a). Authorization fails

4(a)1. Prompt the user/admin that he typed the wrong password

4(a)2. Allow him to re-enter the password.

Use Case 3: Register

Primary Actor: User/Admin

Pre Condition: Nil

Main Scenario:

1. Load the website. User/Admin prompted to register.
2. User/Admin gives his/her details including login,password.
3. System sends verification code to the entered mail-id.
4. User/Admin enter verification code.
5. System does authentication.
6. Main screen is displayed.

Alternate Scenario:

4(a). Authorization fails

4(a)1. Prompt the user/Admin that he typed the wrong verification code
4(a)2. Allow him to re-enter the verification code.

4(b). Login id already exists

4(b)1. Prompt the user/Admin that the login already exists.

Use Case 4: Forgot Password

Primary Actor: User/Admin

Pre Condition: Nil

Main Scenario:

1. User/Admin initiates the forgot password command.
2. User/Admin gives his/her login id.
3. System sends verification code to the entered mail-id.
4. User/Admin enters verification code.
5. System does authentication.
6. User/Admin gives the new password and confirms the new password.
7. System does authentication and updates password.
6. Main screen is displayed.

Alternate Scenario:

4(a). Authorization fails

4(a)1. Prompt the user/Admin that he typed the wrong verification code
4(a)2. Allow him to re-enter the verification code.

4(b). New password and confirm new password do not match.

4(b)1. Allow him to re-enter the attributes.

Use cases related to Account Settings :

Use case 5: Update profile info

Primary Actor: User/Admin

Pre Condition: User/Admin is logged in.

Main scenario :

1. User/Admin updates his personal information.

Use Case 6: Change Password

Primary Actor: User/Admin

Pre Condition: User/Admin logged in

Main Scenario:

1. User/Admin initiates the password change command.
2. User/Admin is prompted for old password, new password and confirm new password.
3. User/Admin gives the old password, new password and confirms the new password
4. System does authentication.
5. New password is registered with the system.

Alternate Scenario:

4(a). Authorization fails

4(a)1. Prompt the user/Admin that he typed the wrong password

4(a)2. Allow him to re-enter the password. .

4(b). New password and confirm new password do not match.

4(b)1. Allow him to re-enter the attributes.

Use cases related to Lodging Complaint :

_ **Use Case 7: Lodging Complaint**

Primary Actor: User

Pre Condition: User logged in

Main scenario :

1. User chooses the category.
2. User chooses the sub-category.
3. User enters the complaint.
4. User submits the complaint.

Use cases related to complaint history:

Use Case 8: Complaint History

Primary Actor: User

Pre Condition: User logged in

Main scenario :

1. User initiates complaint history command.
2. System shows the list of previously lodged complaints along with the status - Resolved, Unresolved, Yet to be confirmed.

Elaborating Yet to be confirmed :

1. If a user receives a response from admin regarding a lodged complaint, the user confirms the status as resolved or unresolved.

Use cases related to Announcements :

Use Case 9 : Resolved major complaints

Primary Actor: User/Admin

Pre Condition: User/Admin logged in

Main scenario :

1. User/Admin initiates Announcements command.
2. System shows a list of major complaints.

Use cases related to Resolving a Complaint :

Use Case 10: Resolving Complaint

Primary Actor: Admin

Pre Condition: Admin logged in

Main scenario :

1. Admin opens the category.
2. Admin opens the sub-category.
3. Admin sees the complaint and tries to resolve it.
4. Admin responds to the user.

Use cases related to Complaint History:

Use Case 10: Complaint History

Primary Actor: Admin

Pre Condition: Admin logged in

Main scenario :

1. Admin opens the complaint history.

2. Admin can see the sub-category wise sorted complaints
3. Admin sees all the complaints and their corresponding status(elaborated below).

Elaboration :

The status of each complaint will have 3 categories

- Every new complaint lodged will be under the category of "Unresolved".
- Once the issue reported by the user is looked and resolved by the admin, he marks the issue as "Awaiting Confirmation".
- The user then gets this notification and gives acceptance/rejection.If user gives acceptance ,
    - The issue will be marked as "Resolved" on the Admin's side.
    - Else if the user rejects the response by the admin , it will be marked as "Rejected" and admin will further look into it to satisfy the user.

3.2 Performance Requirements:

(a) Should run on 500 MHz, 64 MB machine. (b) 90% of the responses should be within 2 sec.

3.3 Design Constraints:

1. *Security:* The files should be secured against malicious deformations.
2. *Fault Tolerance*: Data should not become corrupted in case of system crash or power failure.

3.4 External Interface Requirements:

On entering the webpage, based on the role(student/admin) page will be redirected for Sign in. Dashboard customised for different roles will be displayed, containing Lodge/Resolve complaints dropdown which expands and contracts as per user action, Complaint History, Settings which include Profile Info and Change Password.

Ref Appendix(a) for sample images.

4. Future Extensions:

a. CMS is a two-level hierarchy of user and admin. It can be extended to a three-level hierarchy of user, admin and super admin.

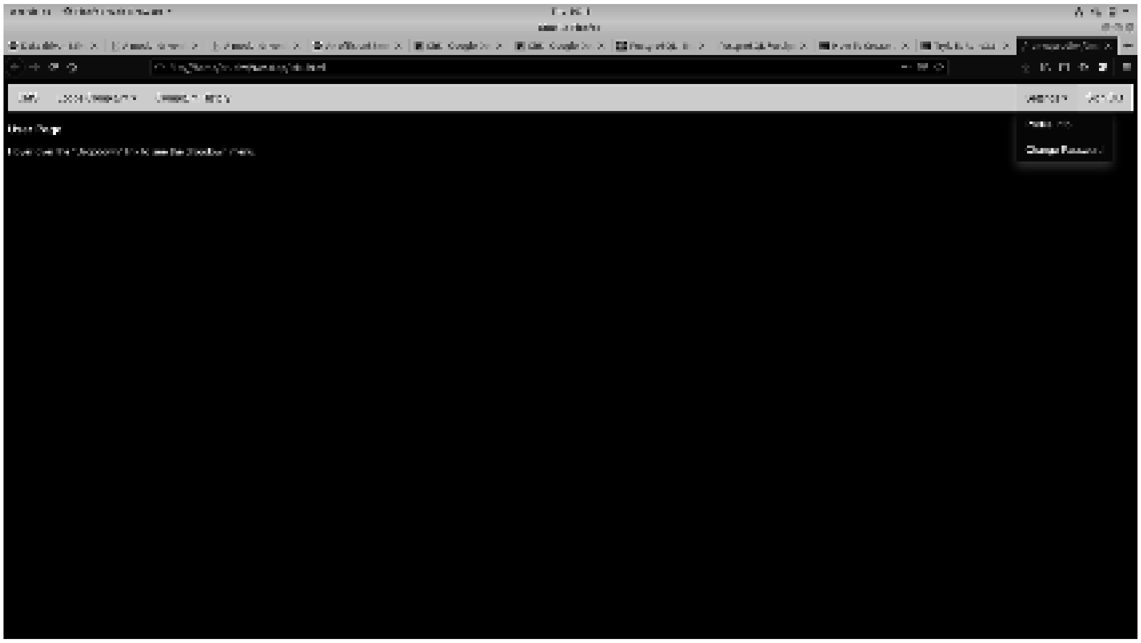
b. Super admin can be given rights to add/delete complaint categories.

c. Admins can complain to super admins.

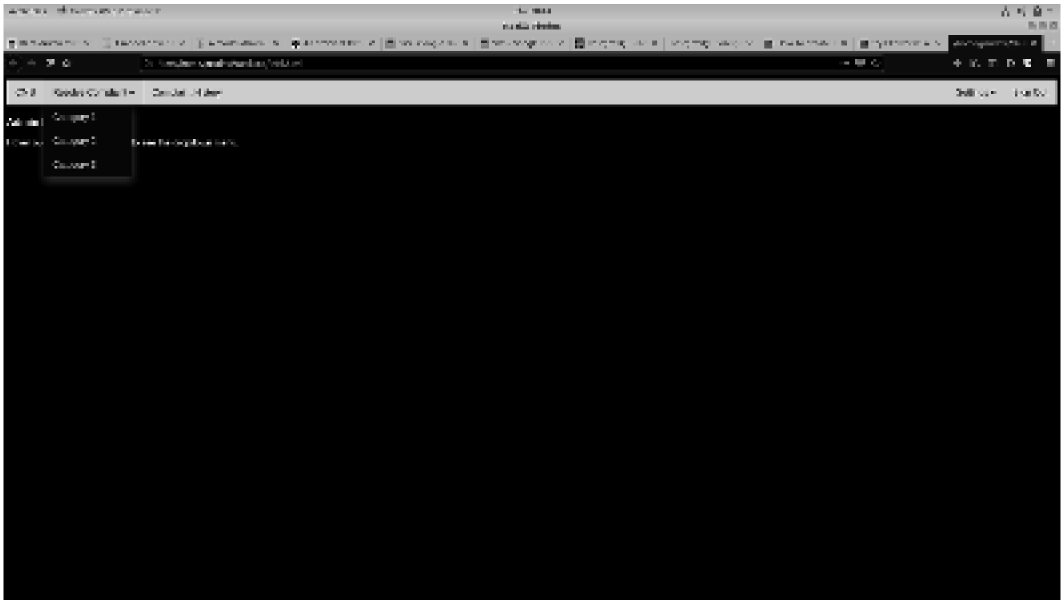
5. Appendix(a):

Main Page:

User Page :



Admin:



Team Members :

- Gorikapudi Prudhvi - CS16BTECH11016
- Mangu Venkata Sai Subhadra Lakshmi - CS16BTECH11021
- Gonela Deepika - CS16BTECH11015
- Sanivarapu Uma - CS16BTECH11033.