COP290: Institute Level Complaint Management System

Aayan Kumar (2014CS10201)
Shreyan Gupta (2014CS10485)
Vaibhav Bhagee (2014CS50297)

10 March, 2016

Contents

1 Introduction

In this assignment we try to implement a basic Complaint Management System for IIT Delhi. The app will cater to people of the institute like faculties, students and institute employee.

Users with any grievances can submit their complaint and alert the concerned authorities. A complaint is visible to all people who are affected by it, and all such users can express their views on a particular complaint in the form of threads and comments.

The concerned authorities would take appropriate steps to resolve the complaint, and mark them as under resolution. If the end users are satisfied with the action of the authorities then the complaint would be marked as resolved, otherwise it would be forwarded to a higher authority.

2 Scope of Assignment

2.1 Current Aim

Currently we are focusing on making an Android App which does the basic functionality of lodging a new complaint (for students) and viewing unresolved complaints.

2.2 Possible future improvements

We can easily extend the app to cater to complaints by faculty and staff as well.

The assignment can be further extended to iOS, Windows as well as web apps in addition to android clients.

3 Implementation Details

3.1 Types of Complaints

There are 2 types of complaints we would be implementing

- The **Individual Complaints** are complaints which are posted by the user and the complaint directly goes to the concerned authority

- The **Community Complaints** are complaints which when posted by any user are visible to all concerned users defined by certain **Tags**. These include hostel level complaints and institute level complaints

3.2 Field of Complaints

The complaints can be broadly divided into the following

- **Maintenance**: Ex. Broken door in hostel or geyser not working.
- **Mess**: Ex. Mess food not upto the mark
- **Student Welfare Related Complaint**
- **Infrastructure**: Ex. Bad quality of seats in lecture halls.
- **Course related**: Complaints specific to courses.
- **NSO/ NSS/ NCC**
- **Security Complaint**: Ex: Stolen laptop, Ragging

4 Database Structure

The database we are implementing is in MongoDB. It is a NoSQL type database and all the objects would be stored as a collection of JSON Objects. The Collection of objects along with their description is described below.

Table 1: Database Collections

Database: Complaint Management System	
Collections	**Description**
Users	Collection of all the users registered in the system
Special_Users	Collection of all the special users in the system
Complaints	Collection of all the complaints (Both individual and community)
Notifications	Collection of all the notifications

- The User and Special User Collections are made up of User JSON Objects.
  - The User Object corresponds to each individual user of the Complaint System. This includes Students, Professors and Faculty Members. The structure of the User object is described in the table below.

Table 2: User JSON Object

Field_Names	**Type**	**Description**
Unique ID	String	PRIMARY KEY This is the unique identification number of i.e Entry Number for the students and registration id for faculty and staff
Name	String	Name of the User
Password	String	Hashed password of the User
Department	String	Associated Department
Contact Information	String	Contact information of the user
Tags	String Array	A list of the tags to which a user is associated (eg. Hostel, NCC/NSS/NSO etc)
Complaint List	String Array	An array of complaint IDs of the complaints to which a user is associated
Courses List	String Array	A list of courses in which the user in enrolled

- The Complaints Collections are made up of Individual Complaint Objects and Community Complaint Objects.
    - The Individual Complaint Object corresponds to a complaint which would only be visible to the lodger of the complaint and the concerned authority.

Table 3: Individual Complaint JSON Object

Individual Complaint JSON Object description		
Field_Names	**Type**	**Description**
Complaint ID	String	PRIMARY KEY This is the unique identification number of the complaint
Lodged_By	String	FOREIGN KEY Unique ID of the User who lodged the complaint
Title	String	Title of the complaint
Description	String	Description of the complaint
Timestamp	Timestamp JSON Object	A JSON object to store the a. Date of Lodging of complaint b. Date of last update
Is_Community	Boolean	A flag to store whether the complaint is individual type or not
Type	String	The broad area, from the Hierarchy Tree to which the complaint is related
Authority Hierarchy	JSON Object	A JSON object which stores the hierarchy of the complaint type
Current_Level	String	The current authority level with whom the complaint resides
Current_Status	String	Either one of below values Unresolved/Resolved/Under_Resolution
Associated Threads	Threads Object Array	The list of threads related to the current complaint

- The Community Complaint Object corresponds to a complaint which would be visible to the lodger of the complaint, the concerned authority, along with all the people who share the tag which corresponds to the complaint.

Table 4: Community Complaint JSON Object

Community Complaint JSON Object description		
Field_Names	**Type**	**Description**
Complaint ID	String	PRIMARY KEY This is the unique identification number of the complaint
Lodged_By	String	FOREIGN KEY Unique ID of the User who lodged the complaint
Title	String	Title of the complaint
Description	String	Description of the complaint
Timestamp	Timestamp JSON Object	A JSON object to store the a. Date of Lodging of complaint b. Date of last update
Is_Community	Boolean	A flag to store whether the complaint is individual type or not
Type	String	The broad area, from the Hierarchy Tree to which the complaint is related
Authority Hierarchy	JSON Object	A JSON object which stores the hierarchy of the complaint type
Current_Level	String	The current authority level with whom the complaint resides
Current_Status	String	Either one of below values Unresolved/Resolved/Under_Resolution
Votes	Vote Object	A vote object which stores the information of number of up-votes, down-votes and list of people who have voted
Associated Threads	Threads Object Array	The list of threads related to the current complaint

- The Vote JSON Object corresponds to the vote associated with a single Community Complaint.

Table 5: Vote JSON Object

Vote JSON Object description		
Field_Names	**Type**	**Description**
No. of Upvotes	Integer	Number of upvotes
No. of Downvotes	Integer	Number of Downvotes
People Voted	String Array	Array of Unique ID of people who have voted

- The Thread Object corresponds to a single thread that would be associated with a single Community Complaint.

Table 6: Thread JSON Object

Thread JSON Object description		
Field_Names	**Type**	**Description**
Thread_ID	String	PRIMARY KEY This is the unique identification number of the thread
Complaint_ID	String	FOREIGN KEY Unique ID of the Complaint to which the thread is associated
Title	String	Title of the thread
Description	String	Description of the thread
Last Updated	Timestamp	Date of last update
Associated Comments	Comments Object Array	The list of comments related to the current thread

  - The Comment Object corresponds to a single comment what would be associated with a single Thread.

Table 7: Comment JSON Object

Comment JSON Object description		
Field_Names	**Type**	**Description**
Posted_By	String	This is the unique identification number of the person who posted the comment
Description	String	Description of the thread
Posted On	Timestamp	Date and time of Posting Comment

- The Notification Collection is made up of Notification JSON Objects
  - The Notification JSON Object has the information of a specific notification generated on updation of a complaint, along with the time stamp.

Table 8: Notification JSON Object

Notification JSON object description		
Field	**Type**	**Description**
Complaint ID	String	Unique ID of the complaint to which the notification is related
Timestamp	String	Time of the notification
Content	String	The content to be displayed in the notification

Table 9: Timestamp JSON Object

Timestamp JSON object description		
Field	**Type**	**Description**
Time of Lodging	String	Timestamp of lodging the complaint
Last Update	String	Timestamp of last update of the complaint

Table 10: Hierarchy JSON Object

Hierarchy JSON object description		
Field	**Type**	**Description**
Hostel Lists	JSON Array of Hostel Objects	Array of Hostel JSON Objects
Student Welfare Complaint	Student Welfare Complaint Object	Contains Hierarchy of Student Welfare Complaints
Infrastructure Complaint	Infrastructure Complaint Object	Contains Hierarchy of Infrastuctue Complaints
Course Complaint	Course Complaint Object	Contains Hierarchy of course complaints
NCC/NSO/NSS Complaint	N_Complaint_Object	Contains Hierarchy of NCC/NSS/NSO complaints
Security Complaint	Security Complaint Object	Contains Hierarchy of Security Complaints

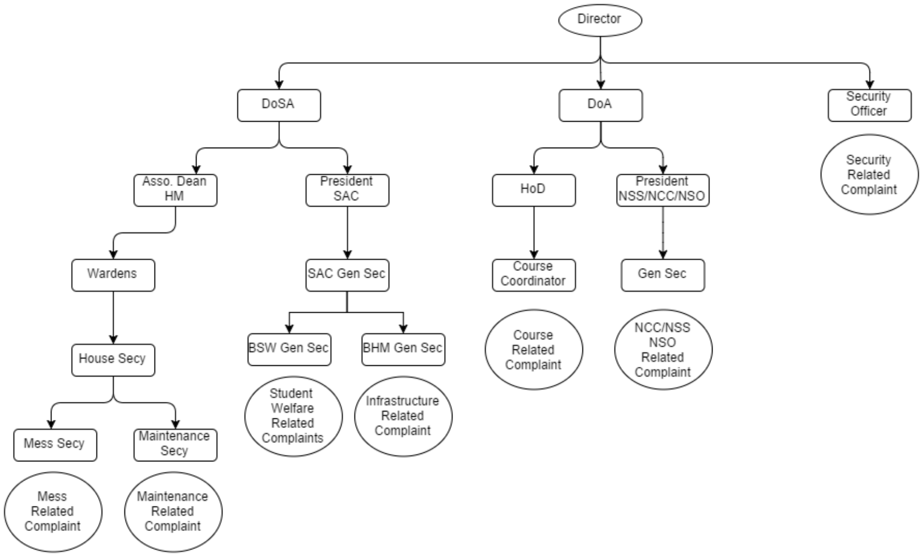

Figure 1: Complaint Hierarchy

5 APIs

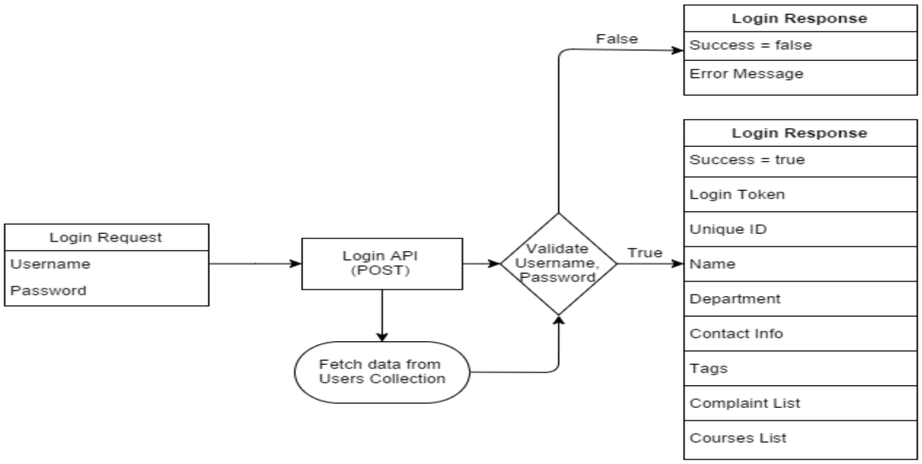

Figure 2: Login API

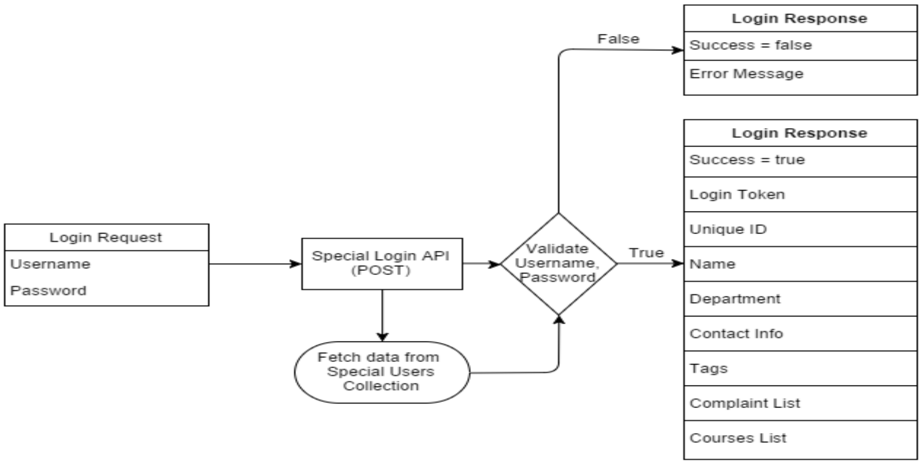

Figure 3: Special Login API

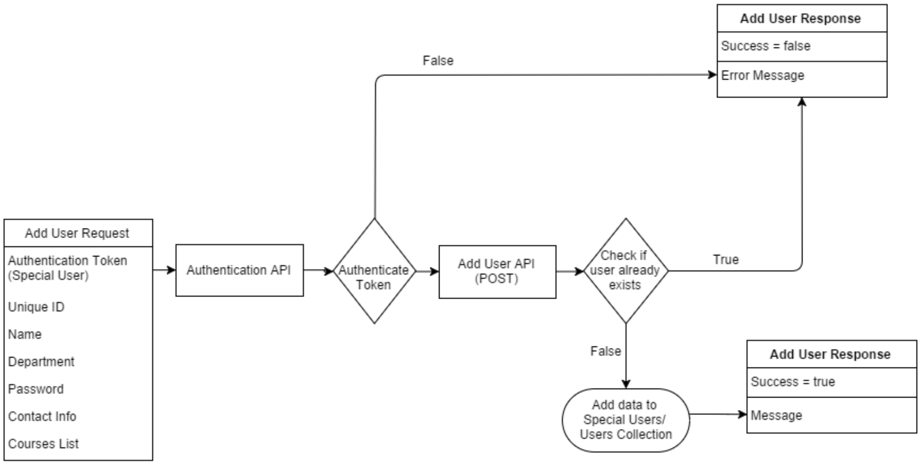

Figure 4: Add User API

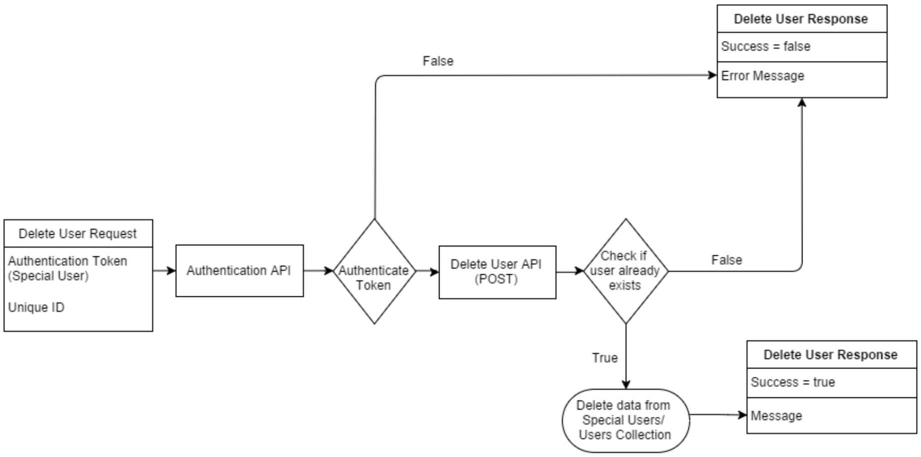

Figure 5: Delete User API

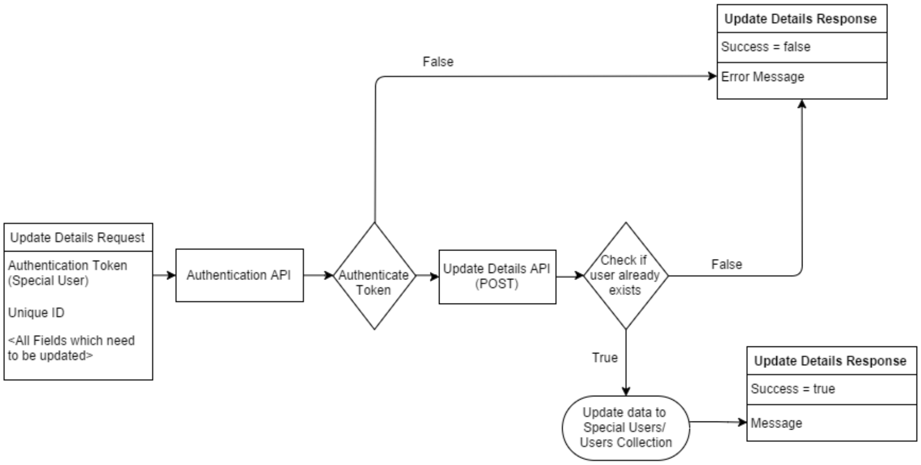

Figure 6: Update Details API

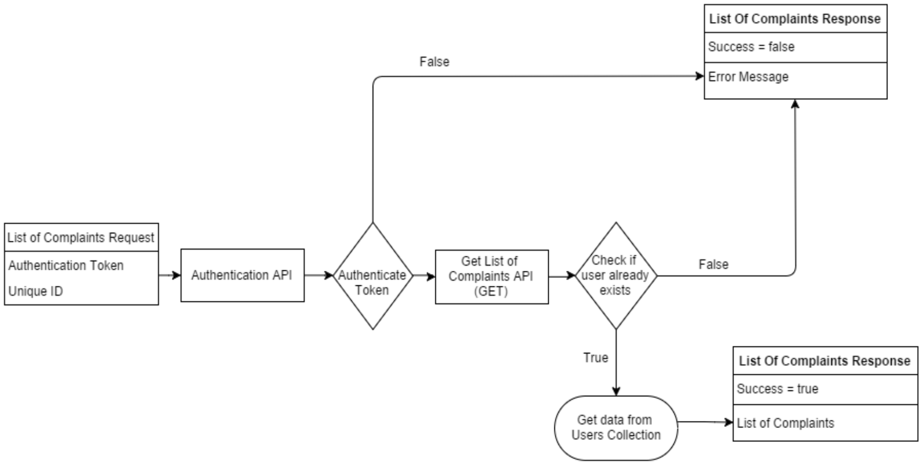
List Of Complaints Response
Success = false
Error Message
False
List of Complaints Request
Authentication Token
Unique ID
Authentication API
Authenticate Token
Get List of Complaints API (GET)
Check if user already exists
False
True
List Of Complaints Response
Success = true
List of Complaints
Get data from Users Collection

Figure 7: List of Complaints API

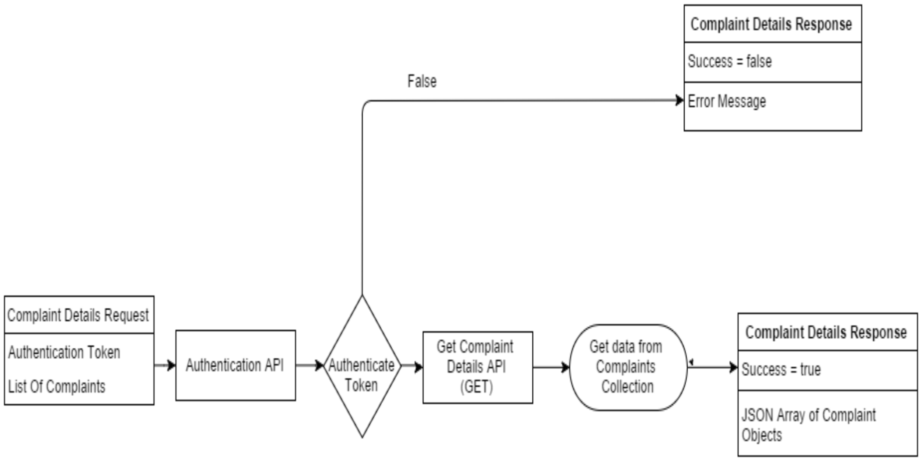
Complaint Details Response
Success = false
Error Message
False
Complaint Details Request
Authentication Token
List Of Complaints
Authentication API
Authenticate Token
Get Complaint Details API (GET)
Get data from Complaints Collection
Complaint Details Response
Success = true
JSON Array of Complaint Objects

Figure 8: Complaint Details API

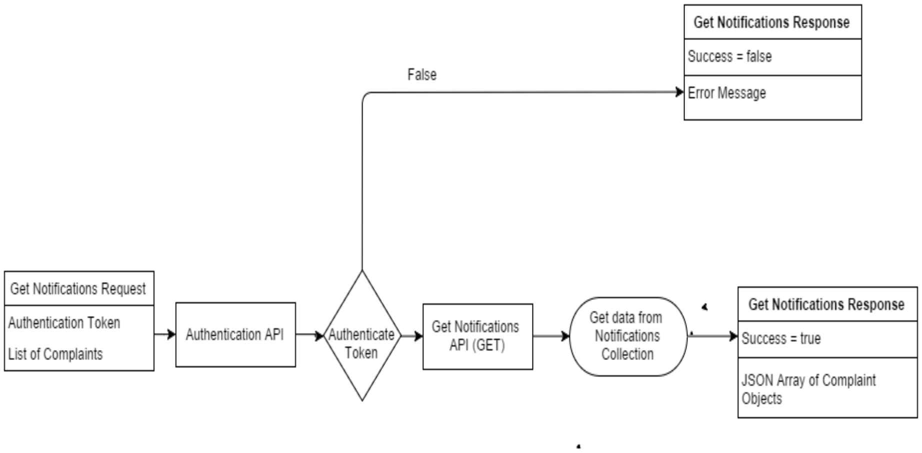
Get Notifications Response
Success = false
Error Message
False
Get Notifications Request
Authentication Token
List of Complaints
Authentication API
Authenticate Token
Get Notifications API (GET)
Get data from Notifications Collection
Get Notifications Response
Success = true
JSON Array of Complaint Objects

Figure 9: Get Notifications API

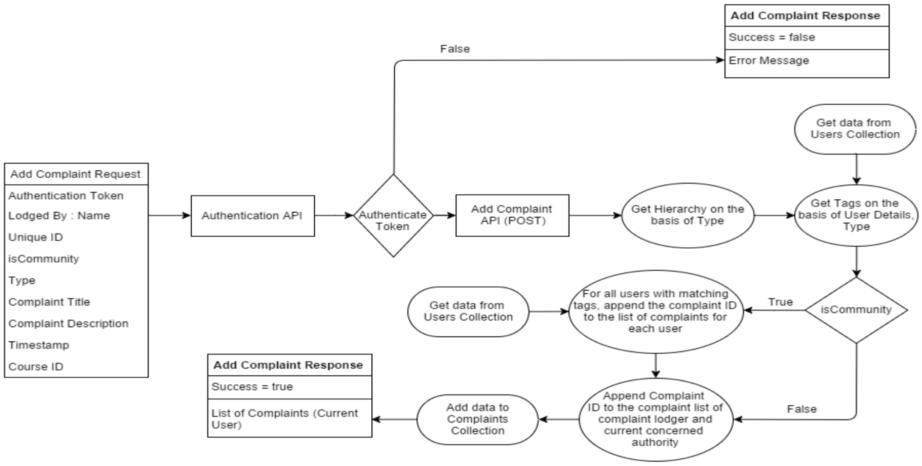
Add Complaint Response
Success = false
Error Message
False
Get data from Users Collection
Add Complaint Request
Authentication Token
Lodged By : Name
Unique ID
isCommunity
Type
Complaint Title
Complaint Description
Timestamp
Course ID
Authentication API
Authenticate Token
Add Complaint API (POST)
Get Hierarchy on the basis of Type
Get Tags on the basis of User Details, Type
Get data from Users Collection
For all users with matching tags, append the complaint ID to the list of complaints for each user
True
isCommunity
Add Complaint Response
Success = true
List of Complaints (Current User)
Add data to Complaints Collection
Append Complaint ID to the complaint list of complaint lodger and current concerned authority
False

Figure 10: Add Complaint API

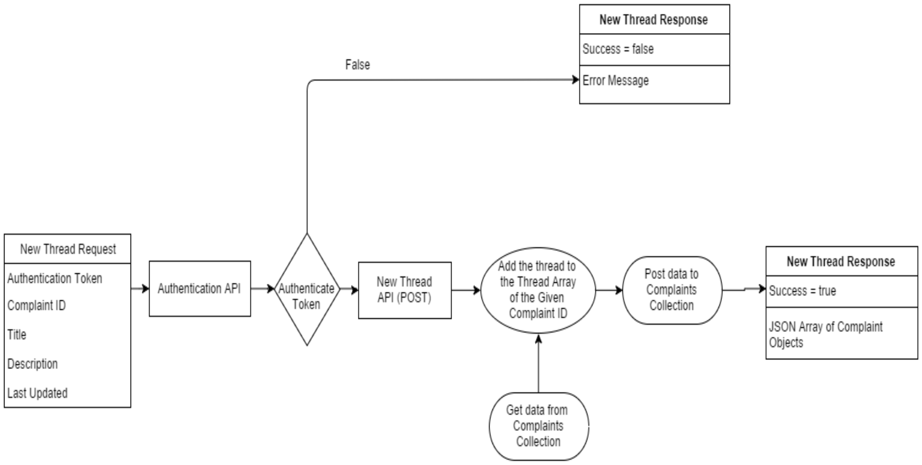

Figure 11: New thread API

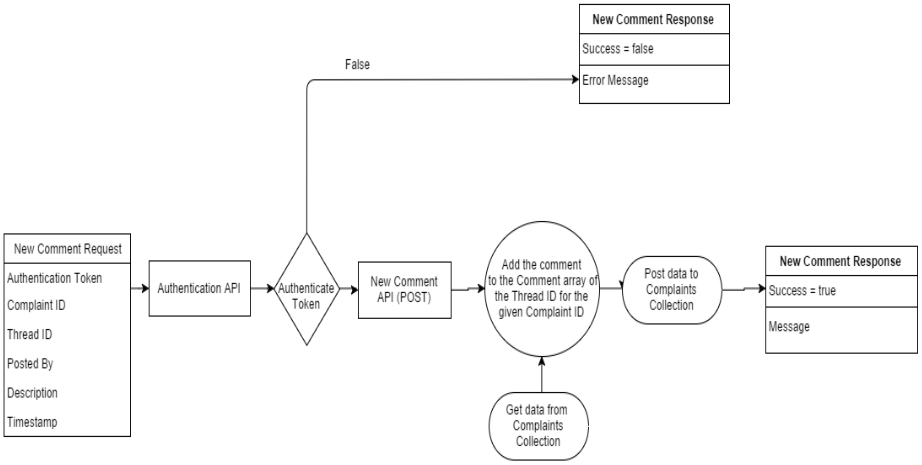

Figure 12: New Comment API

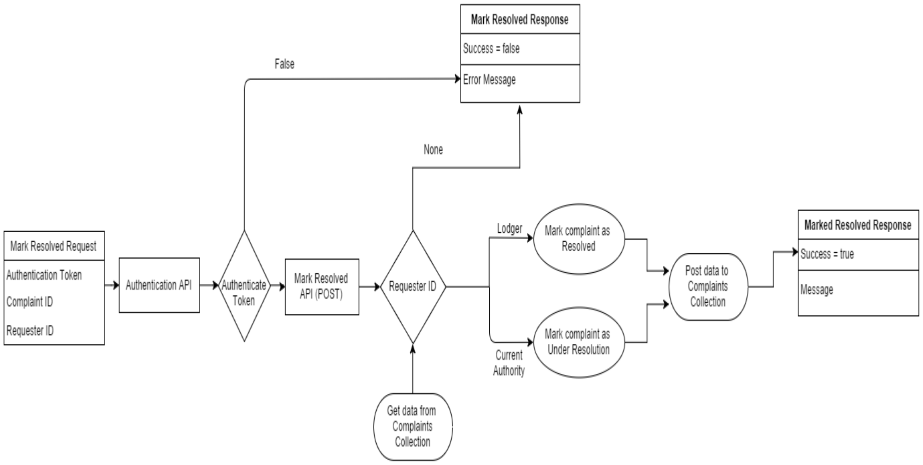

Figure 13: Mark Resolved API

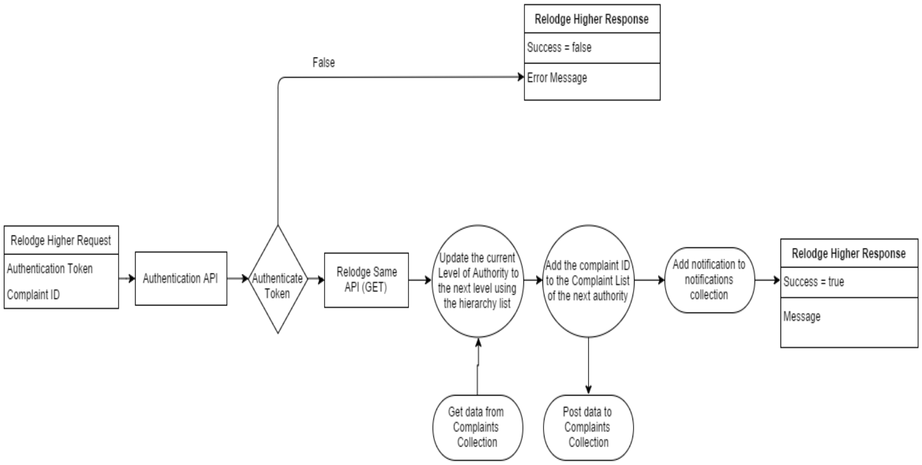
Relodge Higher Response
Success = false
Error Message
False
Relodge Higher Request
Authentication Token
Complaint ID
Authentication API
Authenticate Token
Relodge Same API (GET)
Update the current Level of Authority to the next level using the hierarchy list
Add the complaint ID to the Complaint List of the next authority
Add notification to notifications collection
Relodge Higher Response
Success = true
Message
Get data from Complaints Collection
Post data to Complaints Collection

Figure 14: Relodge higher API

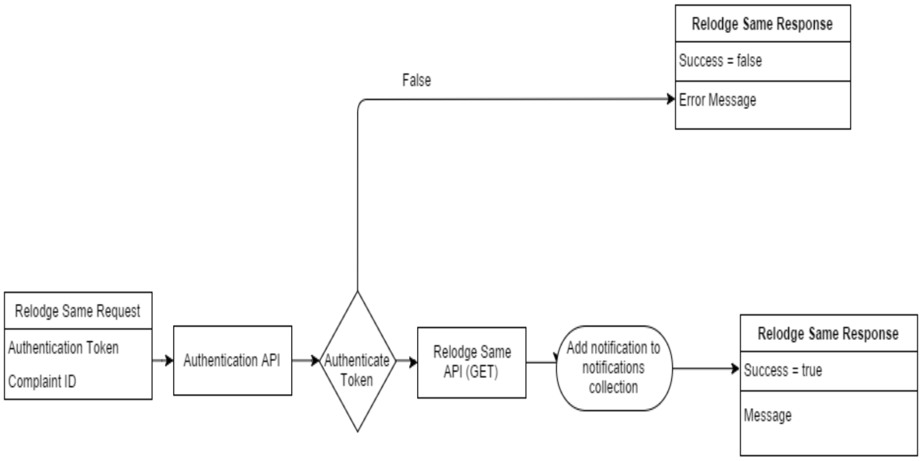

Figure 15: Relodge same API

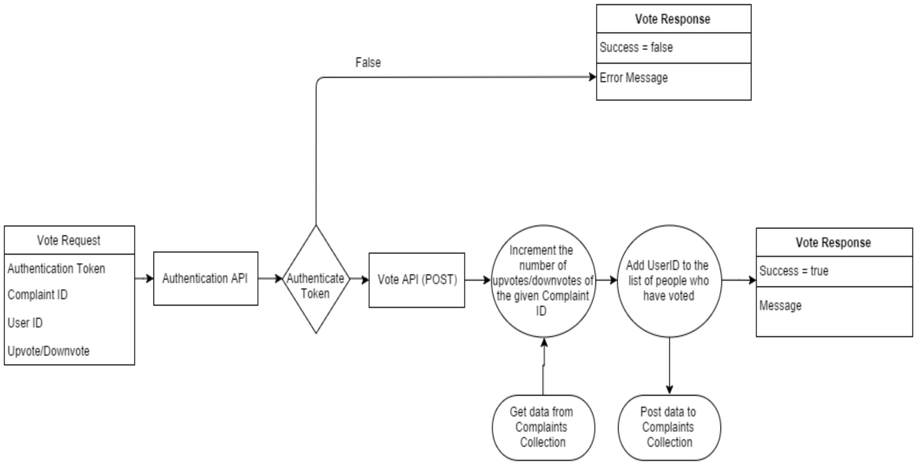

Figure 16: Vote API

6 Event Flow

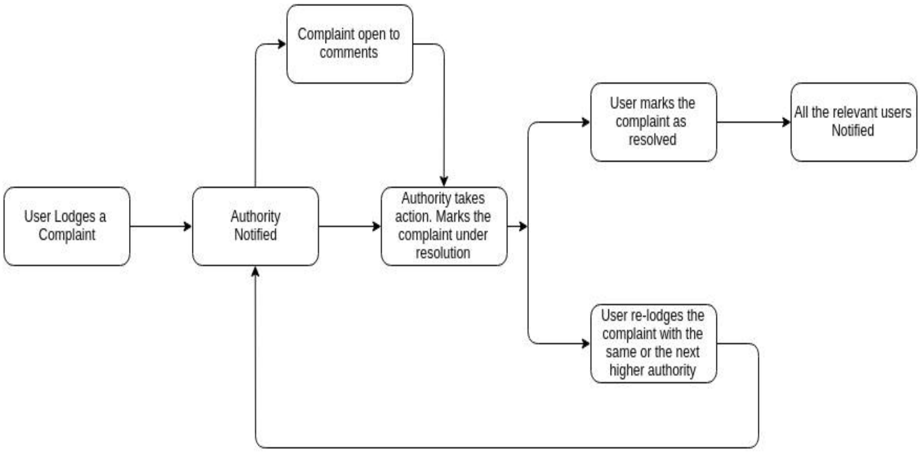
Complaint open to comments
User marks the complaint as resolved
All the relevant users Notified
User Lodges a Complaint
Authority Notified
Authority takes action. Marks the complaint under resolution
User re-lodges the complaint with the same or the next higher authority

Figure 17: Event flow for individual complaint

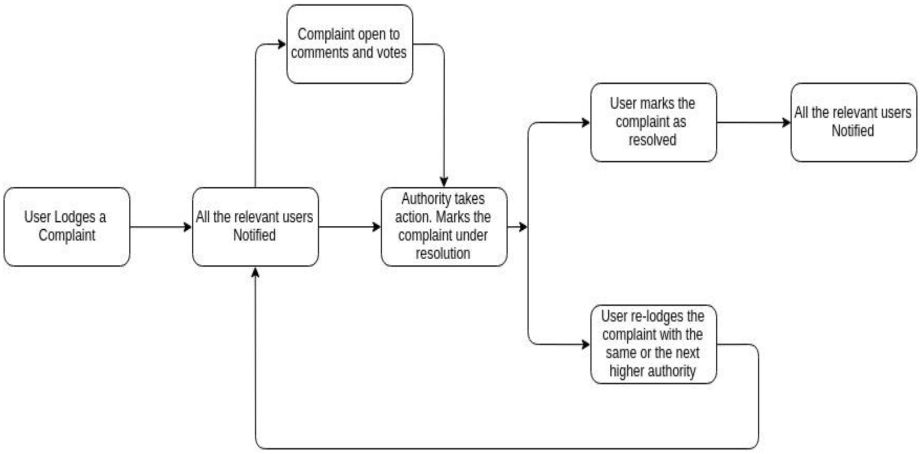
Complaint open to comments and votes
User marks the complaint as resolved
All the relevant users Notified
User Lodges a Complaint
All the relevant users Notified
Authority takes action. Marks the complaint under resolution
User re-lodges the complaint with the same or the next higher authority

Figure 18: Event flow for community complaint

7 User Interface

7.1 Login Page

The login page, as simple as it sounds, basically takes the user id and password input from the user and logs in the user

7.2 Notification Page

Notification page would display a list of all recent notifications specific to the user. A notification is generated when a **new community complaint is posted**, or when a **new thread is added** to a complaint concerning the user or when a **new comment** is added to a thread

7.3 Complaint Display Page

There are 3 types of complaint which would be displayed in 3 separate lists: Unresolved Complaints, Resolved Complaints and Complaints under Resolution.

7.4 Complaint Lodging Page

This page takes details for a new complaint, like Title, Description, Field of Complaint, Type of Complaint and lodges it.

8 References

- https://www.draw.io/
- http://www.tablesgenerator.com/
- http://www.sharelatex.com/