

COP290

# GRS - GRIEVANCE RESOLUTION SYSTEM

*Prakhar Agrawal (2014CS10207)*

*Mayank Rajoria (2014CS10233)*

*Prakhar Gupta (2014CS10290)*

Course Coordinator  
Prof. Vinay Joseph Ribeiro

## **Abstract**

The purpose of this assignment was to create a complaint system for our institute. The main platform decided for this platform is Android for front-end and a Web2Py server for back-end and API. The developed API would be Restful and would easily allow the system to be ported to any platform easily. There will be several levels of complaints allowed and each level shall have its corresponding sub-levels/domains. These shall together decide who has to solve the problem, who can mark it as resolved and who can see the problem. The problems shall have associated statuses with it which can be updated by allowed people and also comments which can be made on the overall complaint or on a particular status of the complaint.

## **1 Introduction**

This project aims to make a complaint system for our institute. Using the complaint system, the user shall be able to report complaints at various levels like individual, hostel and institute, direct them to different authorities for solving the problem. The users who can view the problem will also be able to see the latest status of work on the problem along with the history of its status changes. To voice their opinion, users shall be given the option of upvotes/downvotes and comments. This shall allow feedback from community and allow the problem solving to be a more transparent and community driven process.

We shall be developing our back-end in Web2Py and expose the API which can be called by the front-end on any platform. For the first phase of development, we will develop the complaint system on Android. This is keeping in mind that a majority of students on campus use an Android device and having an app on the mobile shall make the complaint registering process convenient. Our API shall adhere to the Restful practices and are designed in such a way that a front-end for it can be developed for any platform with minimal effort and knowledge of back-end.

We identified that the need of such a system may exist not only in our institute but also in other institute and even in other types of hierarchical settings like offices, societies etc. Keeping this in mind we have developed our database in such a way that it allows flexibility. To use the complaint system in any other place, only the entries in certain tables of the database

shall need to be changed. The tables, API, front-end can remain exactly the same.

## 2 Complaint Life-cycle

- The complaint is made by a user wherein he describes the level, domain and details of the complaint.
- The Users/Groups responsible for solving the complaint and the Users/Groups who can see the complaint are added based on the level and domain of the complaint.
- Additionally more people can be given the right to mark complaint as resolved by the user, for example I may want my room mate to be able to mark the complaint as resolved once the tube light of our room is fixed.
- The complaint status can be updated with custom text by the person responsible for solving the complaint or whoever has the permission to mark it as resolved.
- Predefined complaint status like "Work started", "Solved", "Resolved" and "Complaint forwarded" are also provided.
- Users who can view the complaint can comment on individual status and on overall complaint too.
- Each complaint shall have upvote and downvote buttons to allow a simple way for people to express their opinion.
- Complaint may also be bookmarked by users if they want to receive notifications about its status change and shall also be marked as "read" on server once the user has seen the complaint after latest status change.
- Complaint would be considered solved only after the person with rights to update its status as resolved has done so.

## 3 User Interface - Android Client Mockup

UI is planned out to be a complete swap/swipe based UI. The mockups presented below only represent the user/complaint information to be received/displayed. It is a rough representation of how application space can be used without crunching lots of information into too little space. Android

Client for the web application has been made fully sufficient with all functionality that the web interface provides. GRS will be divided into a large range of activities for easier user navigation and handling

- Login/Signup
- Complaint Lodge
- View Complaints Affecting You
- View Complaints you have received
- View Bookmarked Complaints
- View Notifications
- View User Profile
- Validation Requests from others
- Complaint Description
  - Complaint Timeline
  - Complaint Wall

### **3.1 GRS Main Screen**

Figure 1 is the opening screen of the application. If the user is not already logged in he is auto directed to login activity otherwise he is directly taken to the Main Application activity as shown in Figure ?? . Figure 2 is the login screen where the user enters his username and password to access the application interface. We plan to integrate the complaint System with the LDAP user profiles and authenticate the user via the LDAP-Kerberos used at IIT Delhi. This in the long run will enable us to integrate the user and their workgroup directly from the IIT Delhi database avoiding the redundancies of the registration or the signup process.

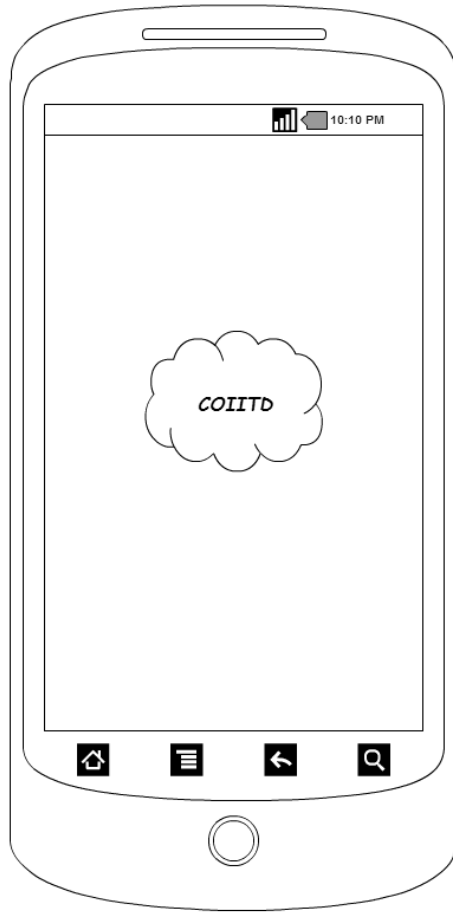


Figure 1: App Front Screen

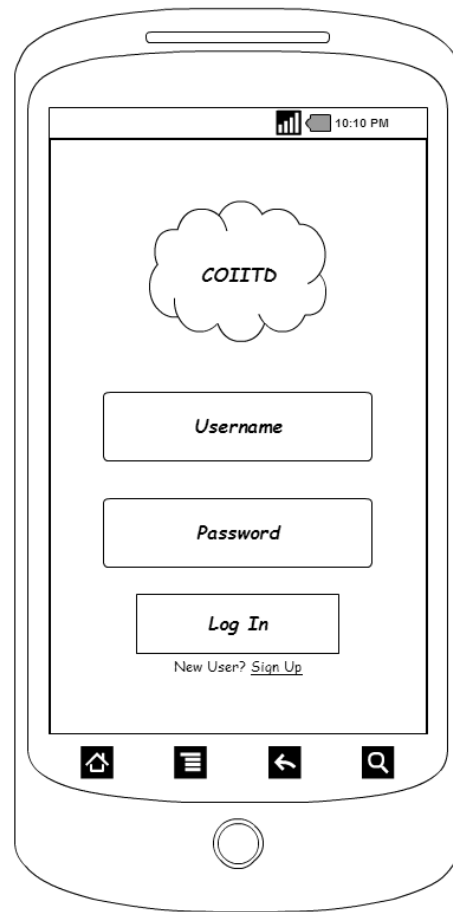
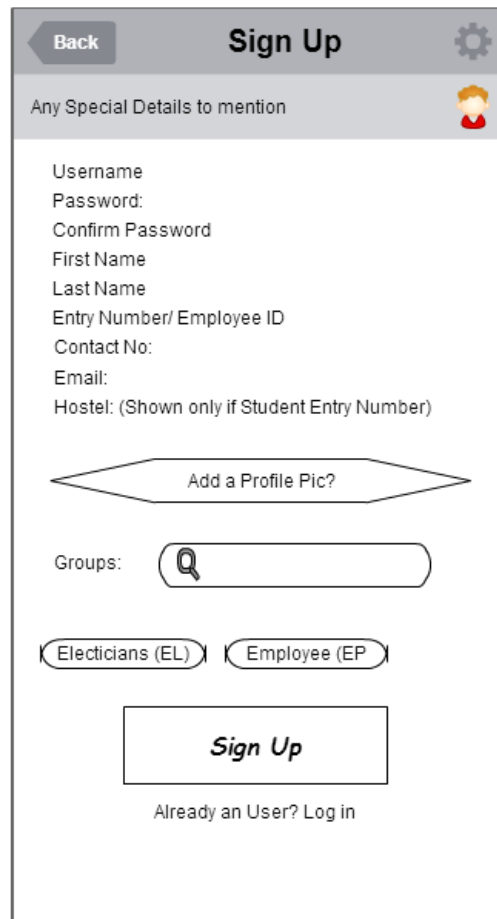


Figure 2: Login Screen

### 3.2 Sign Up Screen

- The Sign up activity has been added as a fallback for an institution that lacks user database or a common authentication APIS.
- User is allowed to enter all his workgroups for which he is responsible and answerable to the institution or the company.
- Care has been taken to not allow users to directly enter any group or subgroups in the backend by setting a validation flag to *false*.
- This temporary screen can be easily disabled by just removing the

singly provided link on the startup activity and *disabling the signup API*



The image shows a mobile application screen titled "Sign Up". At the top, there is a header bar with a "Back" button on the left, the title "Sign Up" in the center, and a gear icon on the right. Below the header, there is a section titled "Any Special Details to mention" with a small icon of a person with a speech bubble. The main content area contains a list of input fields: Username, Password, Confirm Password, First Name, Last Name, Entry Number/ Employee ID, Contact No:, Email:, and Hostel: (Shown only if Student Entry Number). Below these fields is a button labeled "Add a Profile Pic?". Underneath that is a "Groups:" label followed by a search bar with a magnifying glass icon. Below the search bar are two buttons: "Electricians (EL)" and "Employee (EP)". At the bottom of the form is a large "Sign Up" button. Below the "Sign Up" button is a link that says "Already an User? Log in".

Figure 3: Sign Up Screen

### 3.3 Main Activity

#### 3.3.1 Complaints Concerning User - Fragment 1

- This 4 screen shows the current user any *complaints directly or indirectly affecting the current user*.

- Complaints affecting the user and those that come to the current user have been wisely separated out in this fragment and the following fragment as shown in via the figures 4 and 5
- Each complaint listed in the figure has a **complaint title**, **Associated description in short**, **Complaint Pic** (if available), **Complaint Level Indicated by the diamond coloured according to a level**, **a bookmark status** and **a upvote/downvote count**
- Each complaint can be swiped left to view more options corresponding to the complaint such as ***Hide the complaint***, ***Bookmark the complaint***, ***etc.***

### 3.3.2 Complaints Addressed to User - fragment 2

- This Figure 5 screen shows the current user any ***complaints directly or indirectly addressed to the current user.***
- This part of the fragment can be easily swipe opened by swiping the frame 4 to the left to open up the new fragment.
- The ***Avatar*** on the top right in the action bar is a button to open the profile page which has been shown and described in the sections on profile. Reference Figure

## 3.4 Action Bar Functionality

- The action bar of the **Android Client** is designed to include a lot of the application functionality.
- On clicking more on the action bar the user is presented with a variety of buttons such as
  - Notifications
  - Bookmarks
  - Requests for validation
  - Submit A Complaint
  - Logout

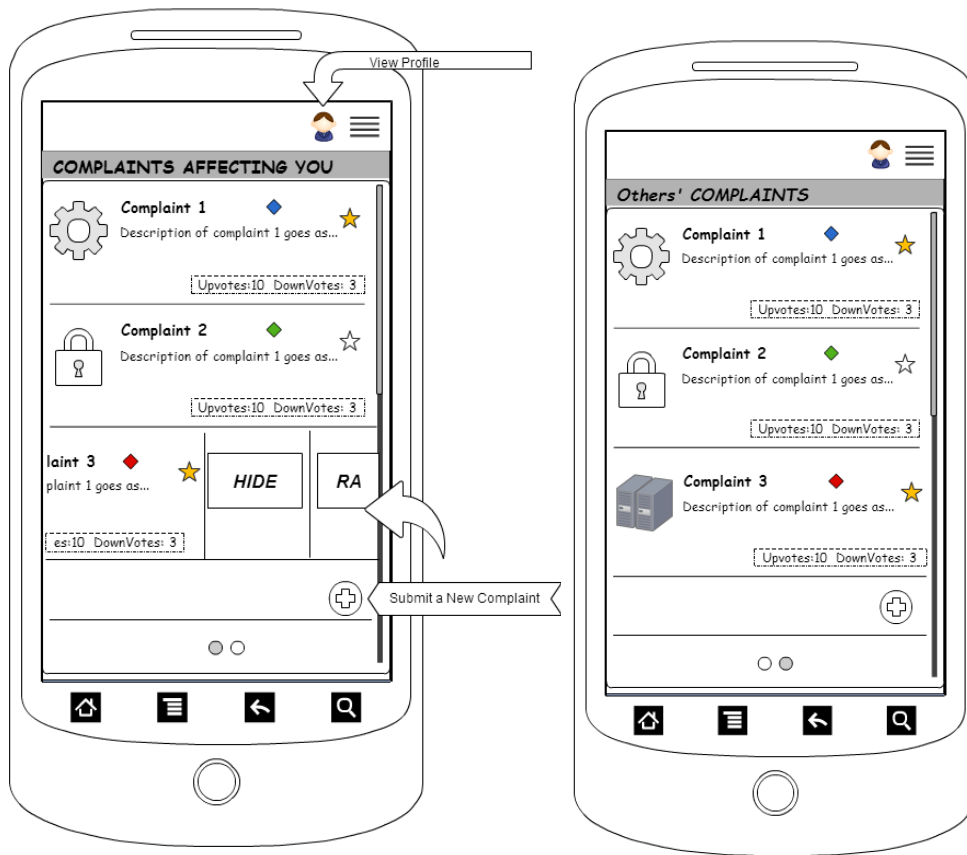


Figure 4: Complaints Concerning Current User

Figure 5: Complaints addressed to Current User

- All these buttons may have a slightly different view in the original application but the mockup gives a basic idea of how these components can be arranged.



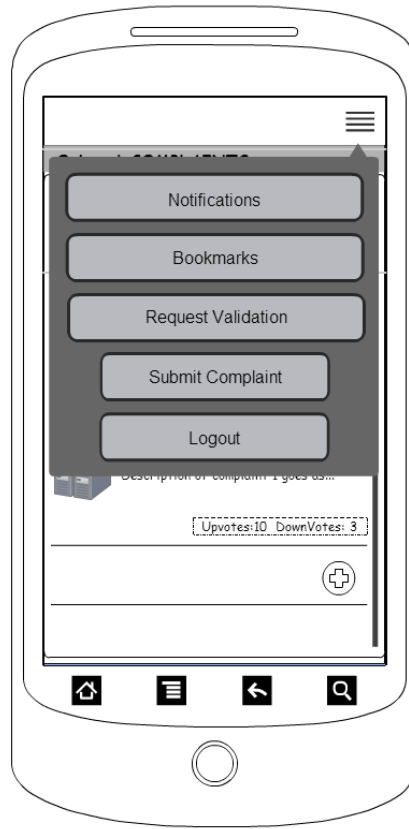


Figure 6: Action Bar functionality

### 3.5 Notifications and Bookmark

- The simple interface mockups for the notifications and the application Bookmarks (i.e. bookmarked complaints) has been shown in the Figure 7 and the Figure 8 respectively.
- These provide the end user a friendly environment to navigate through the active complaints and the complaints of importance i.e. the ones that are bookmarked by him.

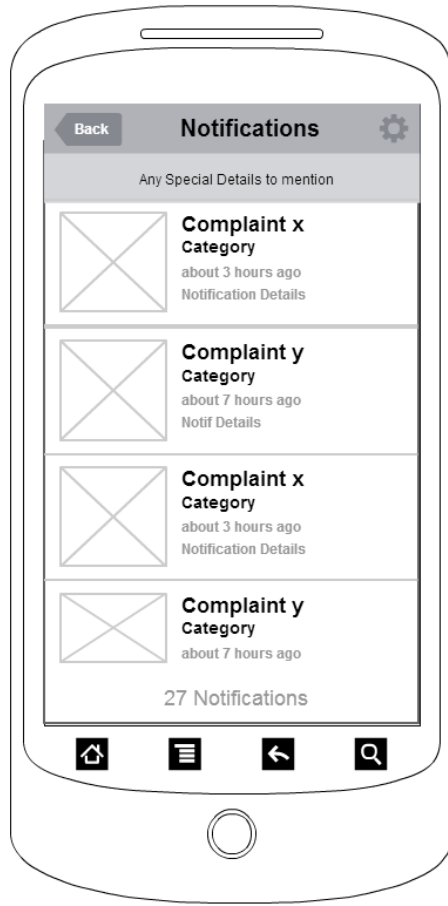


Figure 7: Notifications Screen

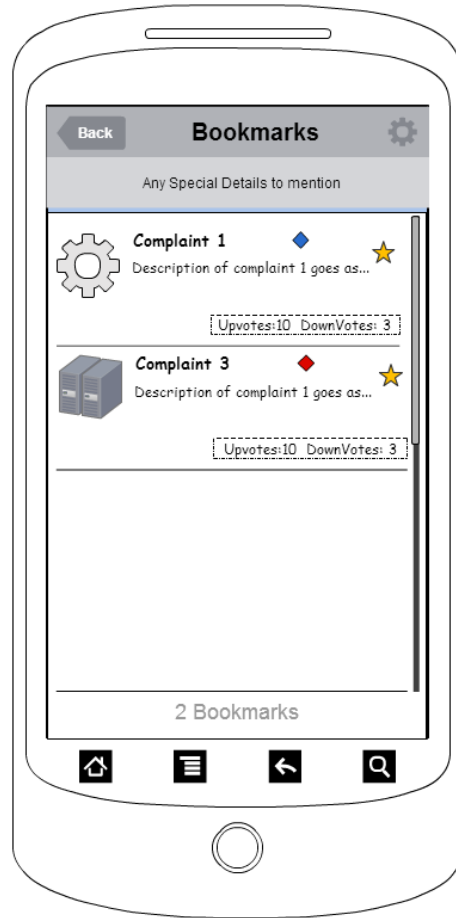


Figure 8: Bookmarks Screen

### 3.6 Profile, Lodge a Complaint, Validation Requests

- The application mockups for the above mentioned pages have been kept simple in the mockups but may exceedingly use swipe based User Interface that turns the Android application client attractive.
- These sections have been shown as in the order above in the Figure 9, 10 and 11

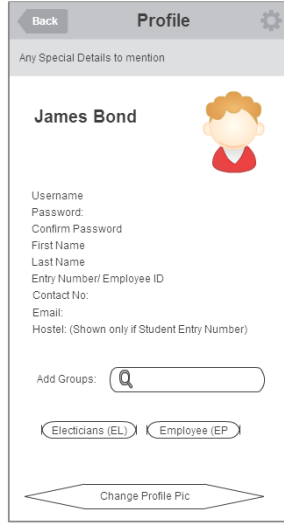


Figure 9: User Profile

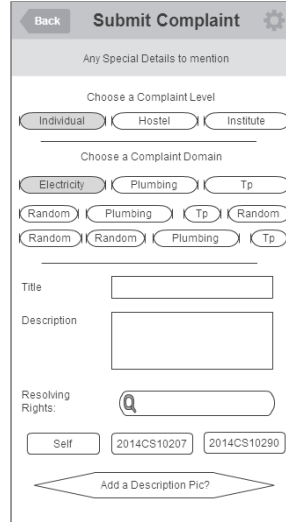


Figure 10: Complaint Lodge

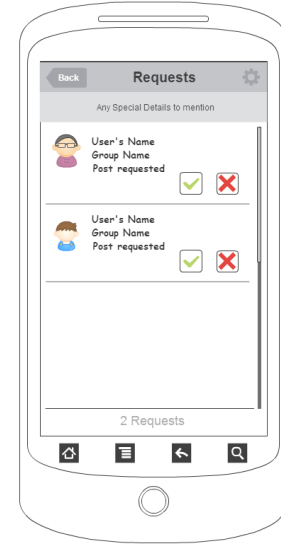


Figure 11: Validation Requests

### 3.7 Complaint Description

The most important part of the application i.e. **Complaint** itself has been detailed and described in a very user friendly manner. Each complaint has a corresponding description that can be accessed by clicking on the complaint or the complaint notifications as shown in the notifications Figure 7

On clicking any complaint the android client opens up a long complaint description Screen that includes complaint title, complaint description, total upvotes/downvotes, Bigger picture related to complaint, complaint current status, whether it has been undertaken by anyone, date undertaken on, posted by, posted on, Complaint location (if relevant) and numerous other details.

#### 3.7.1 COMPLAINT TIMELINE

We plan to describe complaint as a series of progress or changes in the complaint status and we thereby invoke and introduce the concept of **"COMPLAINT TIMELINE"**. Complaint Timeline is chronological display of the progress or significant events in the life cycle of a particular com-

plaint. These events create notifications that are displayed to users that follow/bookmark the complaint to watch it closely.

We believe in taking righteous opinions of several people over each event in the lifecycle of the complaint and hence we have a comment box and a comment thread relating to each event in the Complaint life cycle.

The mockup for the above mentioned timeline has been kept simple and is shown in the figure alongside. The comment section for each event in lifecycle is as shown in the figure by a protruding arrow outside the figure in 12.

### 3.7.2 COMPLAINT WALL

The other tab beside the timeline tab in the complaint description view is of complaint wall, where user/users/resolvers/solvers all can post their opinion, view comments regarding any particular complaint.

This can be used for any number of comments by any user who can view the complaint. This helps in understanding the opinion of the people on widely affecting complaints such as the LAN BAN complaint and giving specific comments in case of the smaller complaints such as that of the tube light and plumbing for an individual hostelite or faculty member.

Figure 12: Complaint Timeline

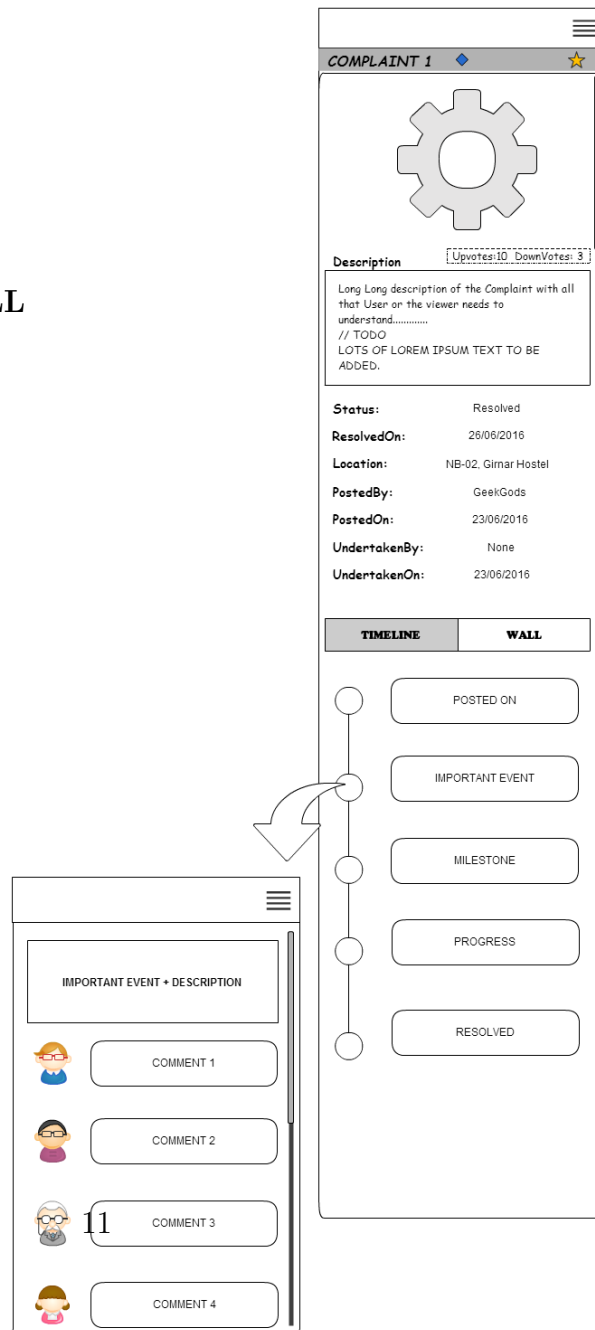
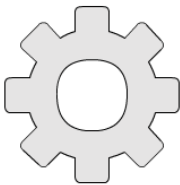


Figure 13: Complaint Wall

☰

COMPLAINT 1◆★



Description

Upvotes: 10 DownVotes: 3

Long Long description of the Complaint with all that User or the viewer needs to understand.....  
// TODO  
LOTS OF LOREM IPSUM TEXT TO BE ADDED.

Status: Resolved

ResolvedOn: 26/06/2016

Location: NB-02, Girnar Hostel

PostedBy: GeekGods

PostedOn: 23/06/2016


UndertakenBy: None

UndertakenOn: 23/06/2016


TIMELINE


WALL

Extra Details if any changes to API

 Comment

My Comment



 Comment

Send

### **3.8 Deploying complaint system in any institution**

- The database is made using MySQL and its design is highly modular
- There are tables for storing all kinds of data like possible complaint levels, complaint domains, groups of people existing, which groups can solve which kind of problems etc.
- From these, the required tables are exposed using the API for use with front-end.
- To be able to use the complaint system in another setting like an office, only the entries in the table of the database will need to be changed. The possible levels of complaint or various Groups can be put in its corresponding table. This can then be accessed by the front-end using the provided APIs.
- This modular design allows the complaint system to be used in various places without the need to change the front-end or the server side implementation.

## 4 Deploying complaint system in any institution

- The database is made using MySQL and its design is highly modular
- There are tables for storing all kinds of data like possible complaint levels, complaint domains, groups of people existing, which groups can solve which kind of problems etc.
- From these, the required tables are exposed using the API for use with front-end.
- To be able to use the complaint system in another setting like an office, only the entries in the table of the database will need to be changed. The possible levels of complaint or various Groups can be put in its corresponding table. This can then be accessed by the front-end using the provided APIs.
- This modular design allows the complaint system to be used in various places without the need to change the front-end or the server side implementation.

## 5 Groups

- Various groups can exist in the institution and these are maintained in a dedicated table for groups.
- This table stores the groups, which users belong to the groups, their position and whether they are approved or not.
- Some example of groups are UGS, Girnar Hostel Admin, Aravali Mess Secy, Security Center, CSC etc.
- These groups will have to be initially defined in the database and a "Head" position for them must be given to a user.
- Whenever a new user sends request to be a part of a group via the app (possibly while registering), the request would go to the head of the group and he would have to accept it.

- Complaints made to a particular group will be visible to everyone in the group.

## 6 Complaint Levels

The complaint system allows complaints to be registered at various levels. The allowed levels are obtained using the provided API endpoint which fetches this data from a table in the database. Within each level, there are several complaint domains which are also loaded using the API. These domains decide to whom is the complaint directed. The following levels could be used in an education institute setting :-

### 6.1 Individual

- These are complaints like tube-light malfunction in room, theft of personal belonging, ragging etc.
- These complaints are directed to a particular Group/User to solve. For example any academic related complaint can be given to UGS.
- Only the user who makes the complaint has right to view the complaint or mark it as resolved. The user may also appoint additional people with this right.
- The Group/User to whom the complaint is made has right to see it and post any status updates, except that the issue is resolved and is closed. This can be done only by user who has posted complaint.

### 6.2 Hostel

- These are complaints like mess issues, sport equipment issues, water tap leakage etc.
- These complaints are directed to a Hostel admin and any other hostel secy as user may deem fit. They can make any status update on the complaint.
- Any user with same hostel as that of person making complaint has right to view the complaint and add comments to it.
- The hostel admin and warden has the right to mark it as resolved.



### **6.3 Institute**

- These are complaints like infrastructure, student demands etc.
- These complaints are directed to a group depending on the complaint domain selected by the user. They can make any status update on the complaint and have the right to mark it as resolved.
- Any user in the institute has right to view the complaint and add comments to it.

## **7 Notifications**

- Any complaint that is visible to the user can be bookmarked by him.
- Complaint made by a user and any complaints made to a group that the user is a part are automatically bookmarked for him.
- A user receives notification for his bookmarked complaints whenever there is a change in its status.

## 8 Database Structure

The Entity Relationship Diagram for the database is shown in figure 14

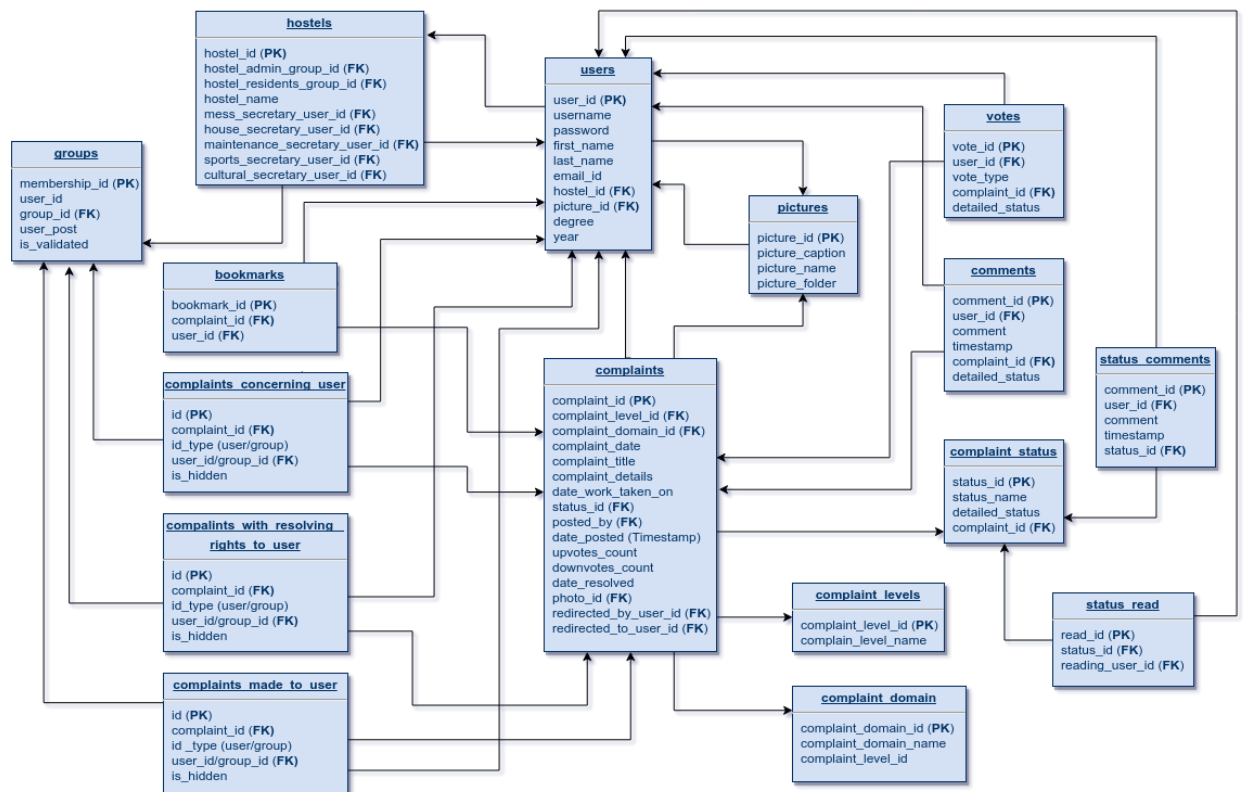


Figure 14: Database Schema

## 9 Application Workflow

The flowcharts in figures Figure 15, Figure 16 and Figure 17 represent the work flow of the application:

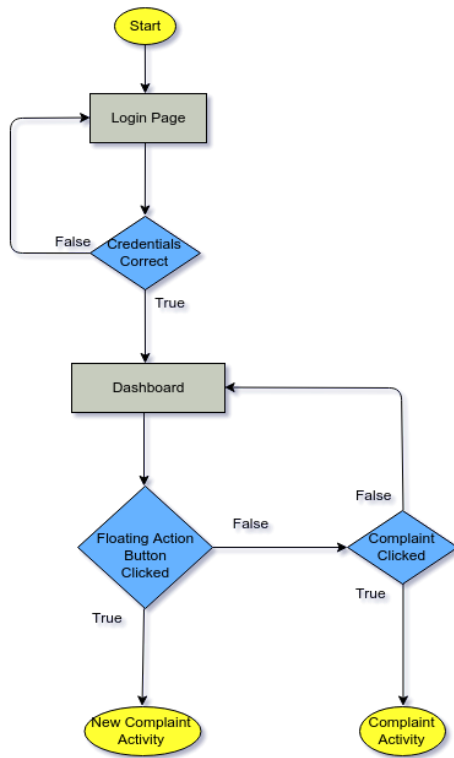


Figure 15: WorkFlow 1

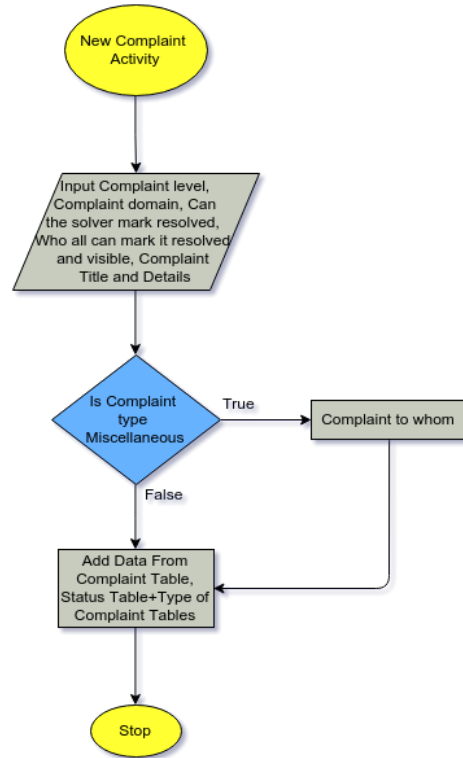


Figure 16: WorkFlow 2

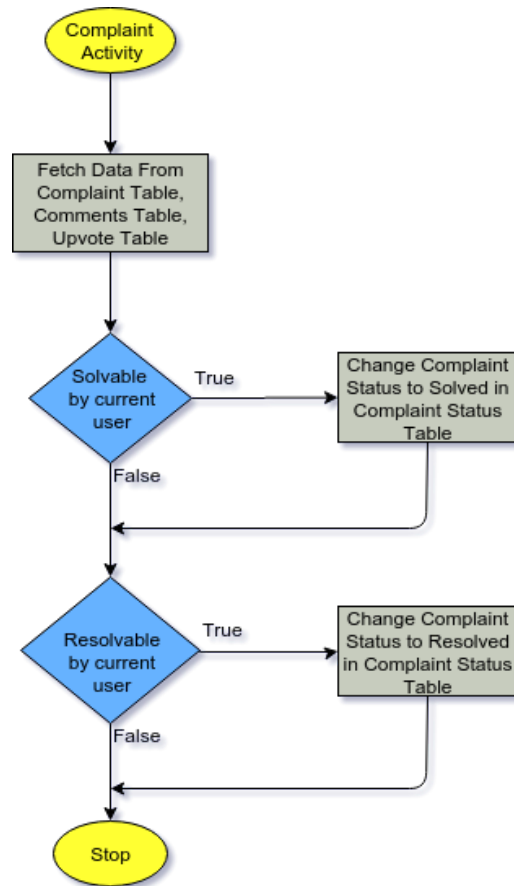


Figure 17: WorkFlow 3

## 10 Application Programming Interface (API)

- We have designed an API that adheres to the principles of REST.
- This allows modularity and easier development of the complaint system on multiple platforms using the same back-end.
- The full API description with detailed request and response parameters has been well discussed and developed at [www.google.com/PU8T1m](http://www.google.com/PU8T1m)
- The following end-points are provided :-

## 10.1 Complaints

- Adding a new Complaints [/complaint] [POST]
  - This adds the complaint details to the "complaint" table.
  -
- Upload Images [/image/complaint] [POST]
  - Uploaded image is saved into a folder and is given an ID, which is stored in database against the image name.
  - The image ID is returned back.
- Download Image [/image/complaint/{image\_id}] [GET]
  - Download image corresponding to the ID from the folder.
- Get complaints list [/complaints] [POST]
  - The type of table to be requested is posted at the given end point. Its possible values are :-
    - \* bookmarked: For list of complaints that are bookmarked.
    - \* concern: For list of complaints that concerns the user.
    - \* resolve: For list of complaints that the user has right to mark resolved.
    - \* solve: For list of complaints that the user is responsible for solving.
  - The response is a list of complaints with their complaint id, title, description and other details required to show the complaint in a concise form.
- Complaint details [/complaint/{complaint\_id}] [GET]
  - Get required complaint details from the "Complaints" table.
  - Check the upvote and downvote table to see if current user has performed the said action.
  - Table will also be updated to mark the current status of this complaint as "read".

## 10.2 Complaints Levels

- Get possible Complaint Levels [/levels] [GET]
  - Get data from "levels" table.

- Get possible Complaint Domains [/domain/{level\_id}] [GET]
  - Get possible domains of complaint in given level.

### 10.3 Complaints Attributes

- Vote complaint [/vote] [POST]
  - The vote status is updated in upvote/downvote tables as required.
- Bookmark Complaint [/follow] [POST]
  - User ID of current user is added against the complaint ID in the "bookmark" table.
- Change Complaint Status [/status] [POST]
  - New status entry is made in "status" table and the status id is updated in "complaints" table.
  - All users who have bookmarked this complaint are sent a push notification and the notification entry is made in the notification table.
- Complaint Status List [/status/Complaint ID] [GET]
  - Get the list of all status ID and their titles.
- Get Bookmark status of user Bookmark [/bookmark/{complaint\_id}] [GET]
  - Returns the bookmark status of user from the "bookmark" table.
- Update Bookmark Status [/bookmark] [POST]
  - Complaint id and new bookmark status is posted.
  - Update bookmark status in the "bookmark" table.
- Mark complaint as read [/read] [POST]
  - Complaint id as posted.
  - Complaint is marked as read by current user.
- Redirect complaint to another user/group [/redirect] [POST]
  - Complaint id and new user/group id is posted.

- A new status is added to the complaint and thus all users who have bookmarked it receive a notification. The people responsible for solving the complaint and who can resolve it are updated in respective tables.

## 10.4 Comments

- Post a comment [/comment] [POST]
  - Complaint ID, User ID and comment is provided. This is updated in the "comments" table.
  - Comment ID is returned back as indication of success.
- Get wall comment [/comment/{complaint\_id}] [GET]
  - All comments on wall of the given complaint are returned.
- Get comments on a complaint status [/comment/status/{status\_id}] [GET]
  - All comments on given status of the complaint are returned.

## 10.5 Users

- Login a User [/user/login] [POST]
  - Username and Password is sent in post request.
  - If credentials are valid, the user ID and other details are returned back
- Signup a User [/user/signup] [POST]
  - User details are sent in POST request.
  - Entries for requested groups of user are entered in "groups" table.
  - User ID is sent back.
- Upload User image [/image/user] [POST]
  - Image is uploaded which is saved in a folder and its entry is made in a table that stores image ID against image name in the folder.
  - Image ID is sent back.
- Download Image [/image/user/{image\_id}] [GET]

- Get the user’s image.
- View validation requests received by user [/validation] [GET]
  - Validation requests are received by head of groups when someone wants to join their group.
  - This endpoint returns all such received requests with the user names, claimed posts, group name and primary key of the entry in groups table.
- Accept a validation request [/validation/{GroupUsersID}] [POST]
  - A post request at this endpoint with the primary key of the validation request in the groups table would accept that validation request.

## 11 Assignment Foresight.

This project has been made highly modular and in such a way that it could be deployed in any institute, office, society etc with minimal effort. The API are also adhering to **REST guidelines**. This allows a complaint system on any platform to access them easily and be made with almost no knowledge of server side implementation.

This work could really be considered as basis of a **B2B startup** wherein we could provide this complaint system to various institutions and organizations. After collecting minimal information about their hierarchy, the database and server could be hosted by us or on one of their servers. The members of the organization could then be provided with credentials to login and use the complaint system. All this would require no logical changes to back-end or front-end, only the database, making our task much simpler.

This application has been developed in a way that it can be easily integrated with the ***LDAP-Kerberos used at our institution***. This gives this application the ability to self populated database using the **DIT (Directory information tree)** in the LDAP and authenticate users using the newly planned kerberos authentication api that we as updaters at CSE are planning,



## 12 Citations

- [www.draw.io](http://www.draw.io)
- [https://www.sharelatex.com/learn/Inserting\\_Images](https://www.sharelatex.com/learn/Inserting_Images)
- <https://en.wikipedia.org/wiki/Flowchart>
- <https://apibluprint.org/>
- [www.cacoo.com](http://www.cacoo.com)