

# COP290: Institute Level Complaint Management System

Aditi Singla (2014CS50277)  
Shardul Tripathi (2014CS10253)  
Dhairya Sandhyana (2014CS10217)

October 4, 2018

## 1 Contents

S.No	Title	Page No.
1.	Introduction	2
2.	System Interface	
	2.1 Login Page	2
	2.2 Home Page (with Navigation Drawer)	2
	2.2.1 Write Complaint	3
	2.2.2 All Complaints	4
	2.2.3 Notifications	5
	2.2.4 View Single Complaint	5
	2.2.5 Change Password	5
	2.2.6 Add User	7
3.	Work Flow	8
4.	Entity Relationship Diagram	9
5.	Event Flow	12
	5.1 API's	12
	5.2 Flow Charts	13
6.	Features	15
7.	Backend	15
8.	Future Endeavours	14

## 2 Introduction

This app implements a complaint management system for an institute like IIT Delhi. It provides a platform for students, faculties and the institute employees to submit their complaints to the concerned authorities and get their problems rectified. The complaint can be at:

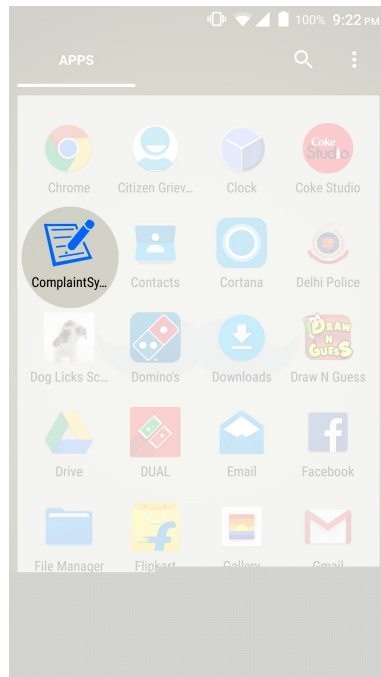
1. **Individual level:** It remains just between the complainant and the concerned authority. The complainant can mark it as resolved, once it has been addressed.
2. **Hostel level:** This is applicable only in case of students and the hostel wardens (who are although faculty members as well). It is visible to all the hostel residents, the warden and the concerned authority. The residents and the warden can upvote or downvote the complaint but only the warden can mark it as resolved after the required steps are taken.
3. **Institute level:** It is visible to all the users of the app (and the concerned authorities, obvious). It also has the option of downvote/upvote for all the students and the faculties. The complaint can be marked resolved only by the complainant.

The users for this app can be registered by the special users or the Admin. They involve Hostel Wardens, House Secretaries and Deans. Admin is the administrative head of this application who can only add users by populating the database. They (multiple admins are allowed) have no role in filing and viewing the complaints. The (other) Users can be categorised as follows:

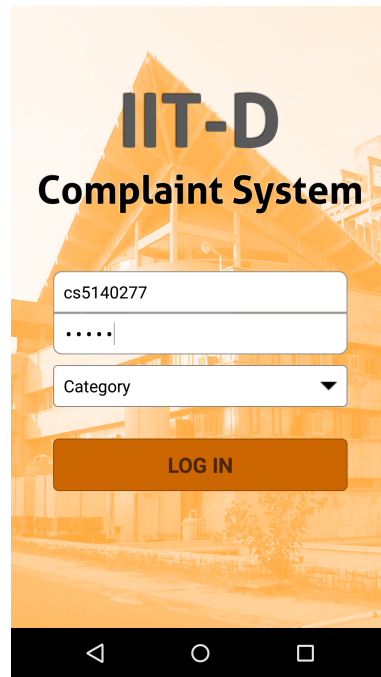
1. **Student:** Tag of hostel, and additional tags like Club General Secretaries, Hostel Level Secretaries, etc.
2. **Faculty:** Additional tags like Warden + Hostel, Deans, etc.
3. **Institute Employees:** Tags like Security, Electrician, Plumber, etc.

## 3 System Interface

1. **LoginPage :** This page is the main Login page. The users can login using a Login type (Admin or User) and a valid username and password (users already registered in the database, which are added by some special users like house secretaries, hostel warden, Admins, etc). The default password for a user is same as the username, which can be changed by the user. The credentials for login are:
  - Username
  - Password
  - Login Type:
    - Admin
    - User (Students, Faculty and Institute Employees are all Users)
2. **Home Page (with Navigation Drawer) :** It is the main home page of this app. It contains a Navigation Drawer, which can be drawn on swiping from left to right. When the drawer is closed, we find a container view, which contains a fragment (all other pages have been implemented as fragments). The default page opened is AllComplaints, in case of Users and



App Icon



Login Page

AddUser, in case of Admin. The navigation drawer contains the following options, which when selected, inflates the corresponding fragment in the Container View.

The options are:

- Write Complaint
- All Complaints
- Notifications
- View Single Complaint
- Change Password
- Add User\* (in case of special users)
- Logout

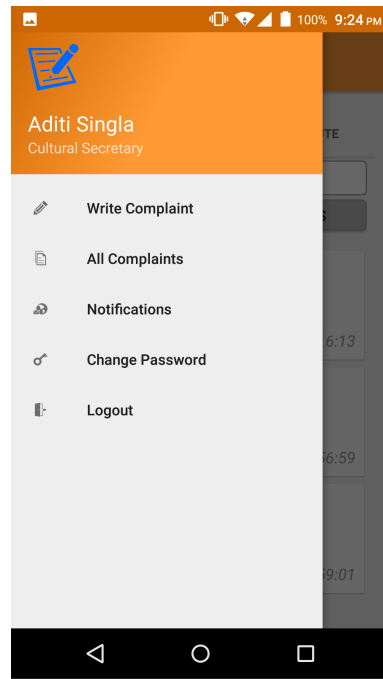
(a) **Write Complaint** : This will contain the following fields (all mandatory except add tags, add images, etc) to be filled in before submitting:

- Title : A subject/title must be given to the complaint, which will be visible to all in the notifications.
- Complaint Level : A drop down menu with options : Individual, Hostel(Visible only to students and Hostel Wardens) or Institute
- Complaint type : The complaint types will be different for various Complaint levels like :
  - **Individual** : It can be a complaint of Academics (to Dean Academics), Electrical (to Electrician), Plumbing (to Plumber) or Security (to Security Officer), where Electrician, Plumber and Security are Institute Employees.

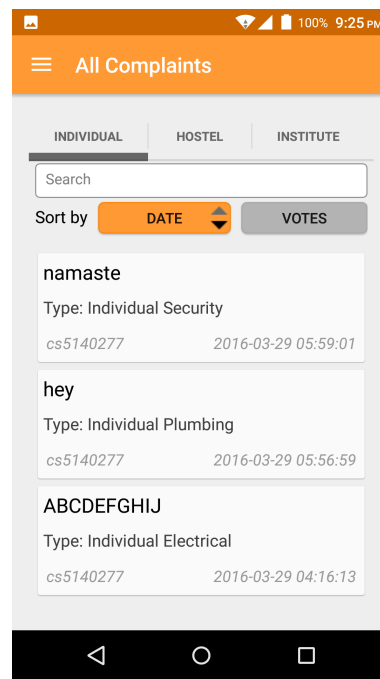
Write Complaint

Write Complaint : Dialog Box

- **Hostel** : Applicable only for Students and Hostel Wardens. The complaint can be of related to Mess (to Mess Secretary), Maintenance (to Maintenance Secretary), Cultural activities (to Cultural Secretary), Sports (to Sports Secretary) or others (to House Secretary).
  - **Institute** : It can be a complaint of Academics (to Dean Academics), Sports (to BSA General Secretary), Cultural (to BRCA General Secretary), Electrical (to Electrician), Plumbing (to Plumber) or Security (to Security Officer), where Electrician, Plumber and Security are Institute Employees.
  - **Description** : A short description about the complaint can be added here in the text box.
  - **Add Tags** : The complaint can be optionally put under a broad category of complaint types which can be sports, maintenance, academics, etc.
  - **Add images** : An optional feature to add to the description. Multiple images can be added.
  - **Submit button** : This can be clicked once all the data is entered. If any field is empty or invalid entry has been made, it doesn't allow to submit until that field is corrected. On submitting, the user is notified about the successful filing of complaint. The user can choose to return to the Home Page (All Complaints) or write another complaint.
- (b) **All Complaints** : This contains all the complaints that are either public in nature or addressed to that user. These are sorted into various tabs namely Individual, Hostel and Institute. In each tab,
- The complaints are sorted in descending order of time.
  - On selecting any complaint, the user is redirected to the View Single Complaint for that complaint.
  - There is an option to sort the complaints in order of the ascending or descending order of dates, votes (which is actually the difference between number of upvotes



Navigation Drawer



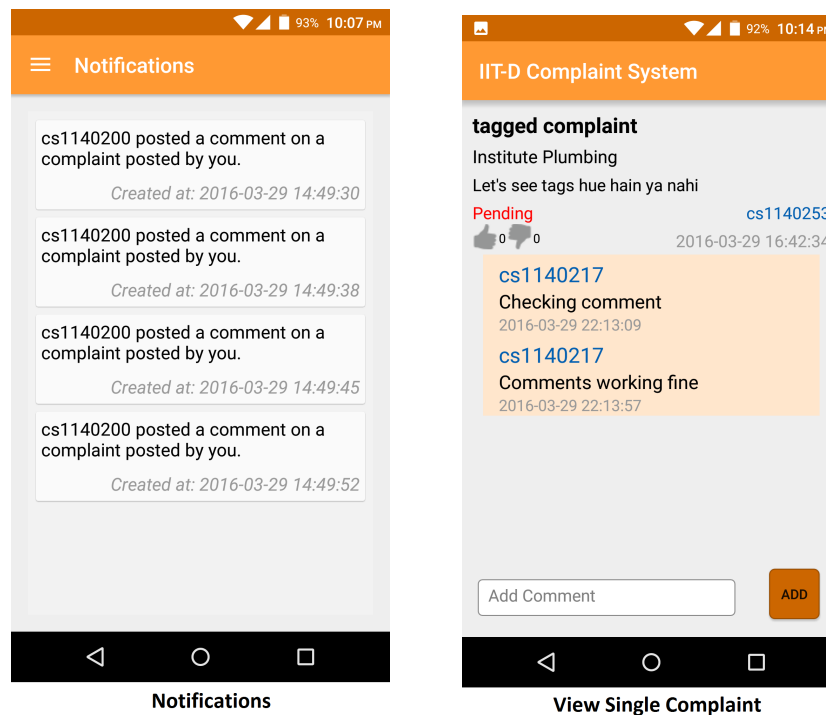
All Complaints : Individual

and downvotes), etc.

- There is a search option, which the user can use to find a complaint by entering some keywords. All the complaints containing these words show up in order of date.
- (c) **Notifications** : This notifies the user about the comments added by the other users on the complaints posted by that user. So basically , it contains the list of these notifications about any complaint. On selecting any notification, the user is redirected to the View Single Complaint of that complaint, which contains the complete description and the comments.
- (d) **View Single Complaint** : This contains all the information about the complaint like:
- i. Date and time, when the complaint was registered
  - ii. Name of the complainant
  - iii. Title and description of the complaint
  - iv. Images attached to the complaint
  - v. Number of upvotes and downvotes
  - vi. Status(resolved or pending)
  - vii. Comments

The user can upvote or downvote the complaint and also add comments to it (along with the option of uploading images). The complainant can mark it as resolved/pending in case of individual and institute level complaints, while only the warden can do this in case of hostel level complaints.

- (e) **Change Password** : The user is initially assigned a password when he/she is registered on the database. This can be changed by the user by choosing this option. A new fragment ChangePassword shows up. This contains three fields: Current password, New password Confirm password. If the current password is correct, and the new password confirm password match, one can click on change password. The password is now changed and the user is redirected to the All Complaints Page.



(f) **AddUser** : This is available only in case of special users. This activity contains a small registration form to be filled by the special user, with the following field:

- Name : Name of the user
- Entry No. (optional) : Only for students
- Username : Every student, faculty and employee has a unique userId. This also becomes the username for that user account. The password by default, is same as the username on registration. The user can later change it to maintain the privacy.
- Category :
  - i. Admin
  - ii. User: Usertype can be:-
    - Student(Without any tag) : Hostel
    - Mess Secretary : Hostel
    - Maintenance Secretary : Hostel
    - Sports Secretary : Hostel
    - Cultural Secretary : Hostel
    - House Secretary : Hostel
    - BRCA General Secretary : Hostel
    - BSA General Secretary : Hostel
    - Faculty(Without any tag)
    - Hostel Warden : Hostel
    - Dean Academics
    - Electrician
    - Plumber
    - Security

Change Password

Old Password\* :

.....

New Password\* :

...

Confirm Password\* :

CHANGE PASSWORD

Change Password

Add User

Name\* : Shardul

Entry Number : 2014CS10253

Username\* : cs1140253

User

BRCA General Secretary

Udaigiri

\*mandatory fields

SUBMIT

Add User

## 4 Work Flow

[2]

The workflow can be as follows:

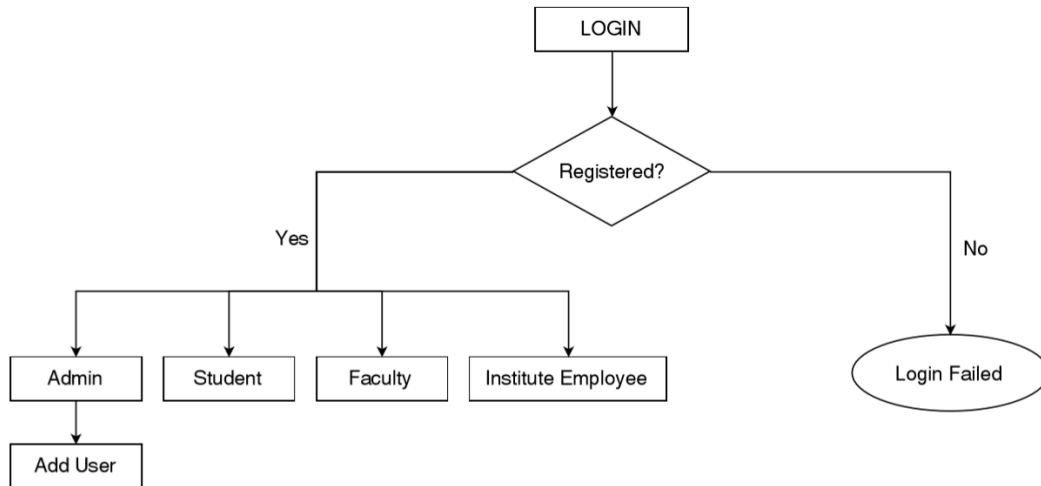


Figure 1: Work Flow

**Note:** The Admin can only add users (populate the database). He/she doesn't have any complaints forum.

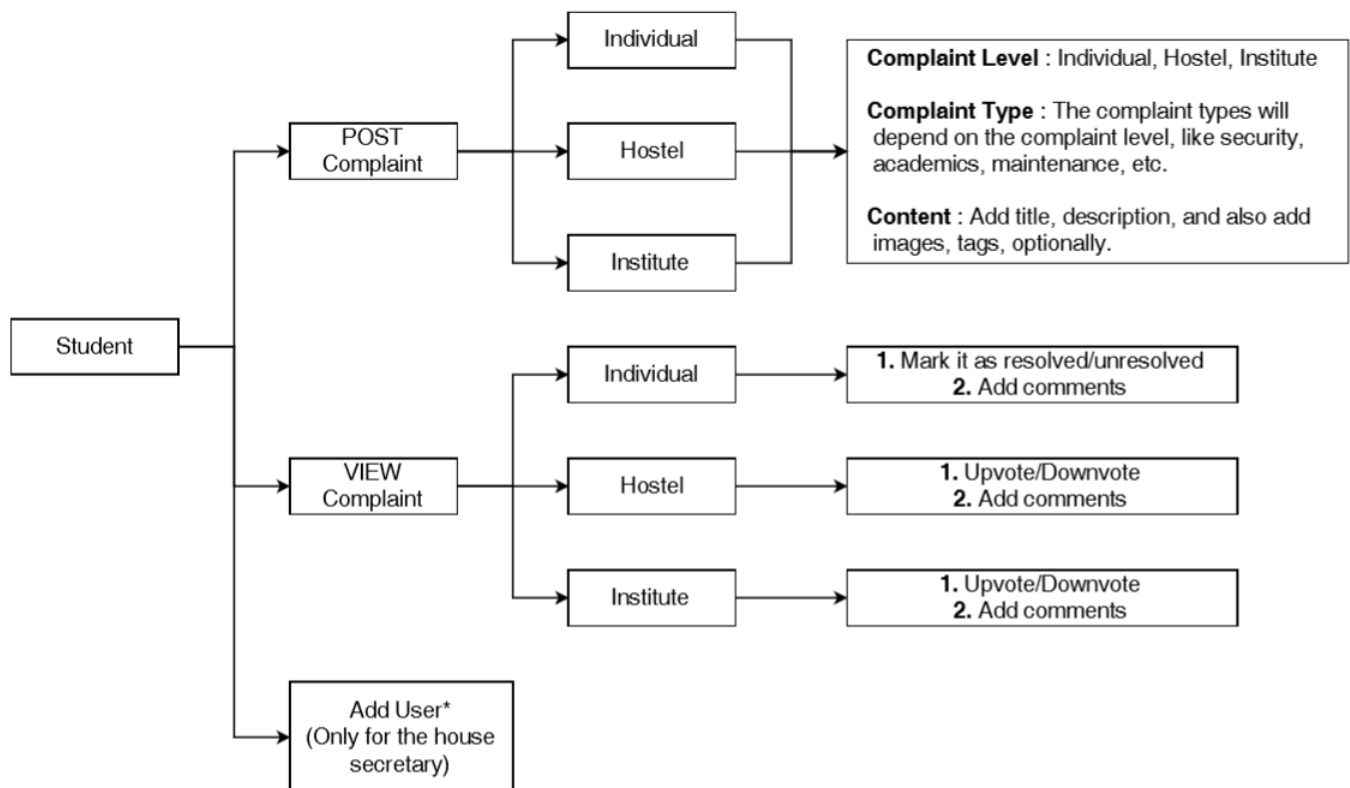


Figure 2: User : Student



**Note:** In the students, the house secretary has the right to add the users as well.

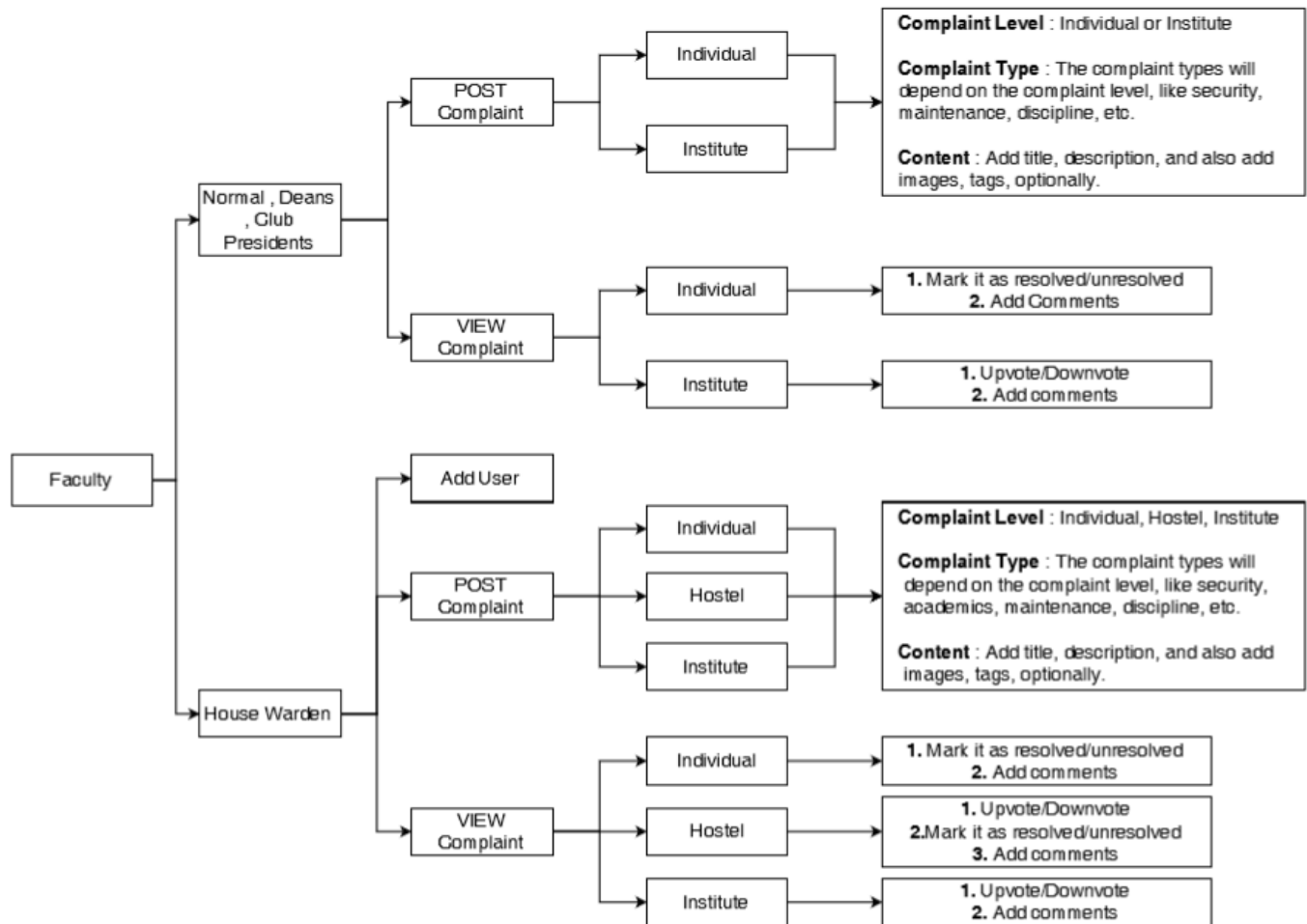


Figure 3: User : Faculty

**Note:** In the faculty, the warden has the right to add the users as well.

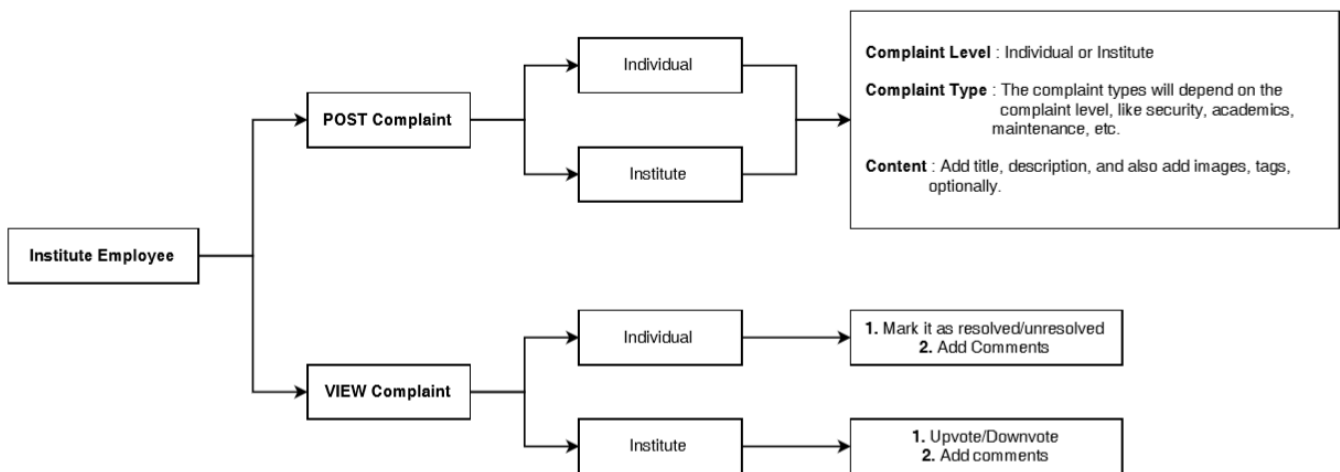


Figure 4: User : Institute Employee

## 5 Entity Relationship Diagram

[1]

The database in the server contains the following tables:

Figure 5: Users:

Column Name	Datatype	Constraint(s)
username	VARCHAR (9)	PK
password	VARCHAR (50)	NOT NULL
logintype	ENUM ('Admin', 'Categorised', 'User')	NOT NULL
usertype	ENUM ('Normal Faculty', 'Warden', 'Dean', 'BRCA President', 'BSA President', 'Normal Student',...)	NOT NULL
name	VARCHAR (100)	NOT NULL
entryno	VARCHAR (11)	UNIQUE
hostel	ENUM ('HIMADRI', 'KUMAON', 'KAILASH',...)	
createdat	DATETIME	DEFAULT CURRENT_TIMESTAMP

Figure 6: Complaints:

Column Name	Datatype	Constraint(s)
complaint_id	SMALLINT	PK, AUTO_INCREMENT
title	VARCHAR (50)	NOT NULL
level	ENUM ('Individual', 'Hostel', 'Institute')	NOT NULL

Figure 7: Individual Complaints:

Column Name	Datatype	Constraint(s)
complaint_id	SMALLINT	PK, FK
title	VARCHAR (50)	NOT NULL
description	VARCHAR (1000)	NOT NULL
image	SMALLINT	FK
type	ENUM ('Maintenance', 'Security', 'Academics',...)	NOT NULL
username	VARCHAR (100)	NOT NULL
resolved	ENUM ('resolved', 'pending')	NOT NULL
createdat	DATETIME	DEFAULT CURRENT_TIMESTAMP

Figure 8: Hostel Complaints:

Column Name	Datatype	Constraint(s)
complaint_id	SMALLINT	PK, FK
title	VARCHAR (50)	NOT NULL
description	VARCHAR (1000)	NOT NULL
image	SMALLINT	FK
type	ENUM ('Maintenance', 'Security', 'Academics',...)	NOT NULL
username	VARCHAR (100)	NOT NULL
upvotes	SMALLINT	DEFAULT 0
downvotes	SMALLINT	DEFAULT 0
difference	SMALLINT	DEFAULT 0
resolved	ENUM ('resolved', 'pending')	NOT NULL
createdat	DATETIME	DEFAULT CURRENT_TIMESTAMP

Figure 9: Institute Complaints:

Column Name	Datatype	Constraint(s)
complaint_id	SMALLINT	PK, FK
title	VARCHAR (50)	NOT NULL
description	VARCHAR (1000)	NOT NULL
image	SMALLINT	FK
type	ENUM ('Maintenance', 'Security', 'Academics',...)	NOT NULL
username	VARCHAR (100)	NOT NULL
upvotes	SMALLINT	DEFAULT 0
downvotes	SMALLINT	DEFAULT 0
difference	SMALLINT	DEFAULT 0
resolved	ENUM ('resolved', 'pending')	NOT NULL
createdat	DATETIME	DEFAULT CURRENT_TIMESTAMP

Figure 10: Comments:

Column Name	Datatype	Constraint(s)
complaint_id	MEDIUMINT	PK, FK
comment	VARCHAR (1000)	NOT NULL
image	SMALLINT	FK
username	VARCHAR (100)	NOT NULL
createdat	DATETIME	DEFAULT CURRENT_TIMESTAMP

Figure 11: Tags:

Column Name	Datatype	Constraint(s)
complaint_id	MEDIUMINT	FK
tag	VARCHAR (100)	NOT NULL

Figure 12: Images:

Column Name	Datatype	Constraint(s)
image_id	SMALLINT	PK, AUTO_INCREMENT
complaint_id	SMALLINT	FK
url	VARCHAR (100)	NOT NULL

## 6 Event Flow

### 6.1 APIs:

The APIs are as follows: [3]

1. **Login:** [http://localhost/complaint\\_management/default/login.php](http://localhost/complaint_management/default/login.php)  
**Response:** JSONObject of Success Message
2. **Add User:** [http://localhost/complaint\\_management/default/adduser.php](http://localhost/complaint_management/default/adduser.php)  
**Response:** JSONObject of Success Message
3. **Change Password:** [http://localhost/complaint\\_management/default/changepassword.php](http://localhost/complaint_management/default/changepassword.php)  
**Response:** JSONObject of Success Message
4. **Logout:** [http://localhost/complaint\\_management/default/logout.php](http://localhost/complaint_management/default/logout.php)  
**Response:** JSONObject of Success
5. **All Complaints:** [http://localhost/complaint\\_management/complaint/allcomplaints.php](http://localhost/complaint_management/complaint/allcomplaints.php)  
**Response:** JSONArray of (JSONObject of Title, Description ComplaintId) called Complaint
6. **Sort:** [http://localhost/complaint\\_management/complaint/sort.php](http://localhost/complaint_management/complaint/sort.php)  
**Response:** JSONArray of sorted Complaints
7. **Search:** [http://localhost/complaint\\_management/complaint/search.php](http://localhost/complaint_management/complaint/search.php)  
**Response:** JSONArray of Complaints
8. **Get Complaint:** [http://localhost/complaint\\_management/complaint/getcomplaint.php](http://localhost/complaint_management/complaint/getcomplaint.php)  
**Response:** Complaint
9. **Submit Complaint:** [http://localhost/complaint\\_management/complaint/submitcomplaint.php](http://localhost/complaint_management/complaint/submitcomplaint.php)  
**Response:** JSONObject of Success Message
10. **Add resolve/pending:** [http://localhost/complaint\\_management/editcomplaint/addresolve.php](http://localhost/complaint_management/editcomplaint/addresolve.php)  
**Response:** JSONObject of Success Message
11. **Upvote/Downvote:** [http://localhost/complaint\\_management/editcomplaint/addvote.php](http://localhost/complaint_management/editcomplaint/addvote.php)  
**Response:** JSONObject of Success Message
12. **Notifications:** [http://localhost/complaint\\_management/notifications/getnotifications.php](http://localhost/complaint_management/notifications/getnotifications.php)  
**Response:** JSONArray of ComplaintId, Complaint title, type level
13. **Add Comments:** [http://localhost/complaint\\_management/editcomplaint/addcomment.php](http://localhost/complaint_management/editcomplaint/addcomment.php)  
**Response:** JSONArray of ComplaintId, Complaint title, type level

## 6.2 Flow Charts:

The flowcharts are as follows: [2]

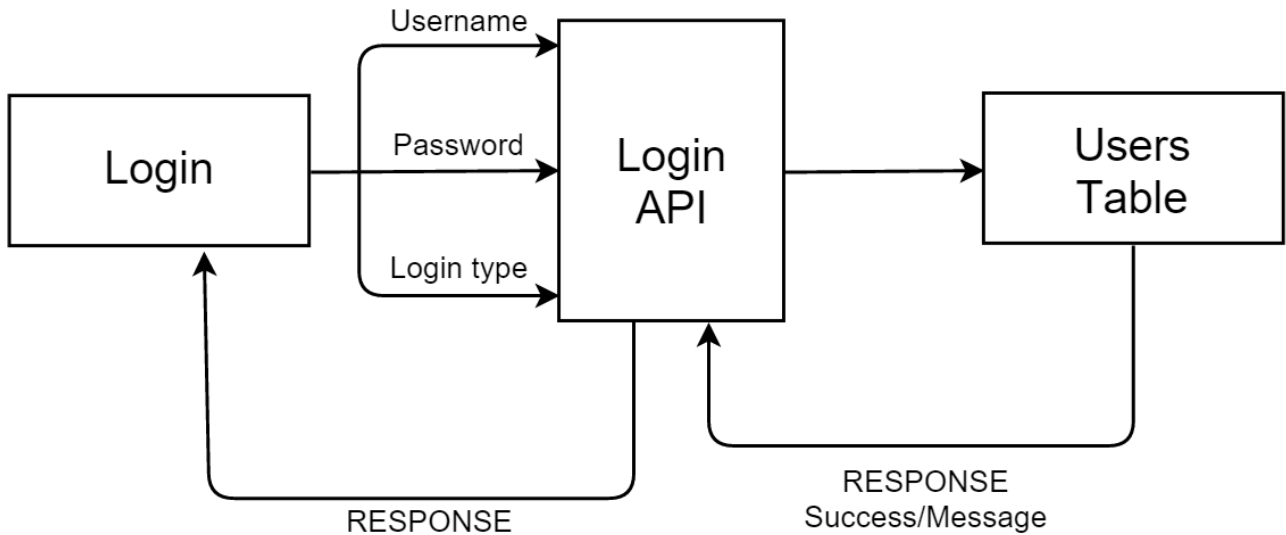


Figure 13: Login: Flow of events when a person tries to login

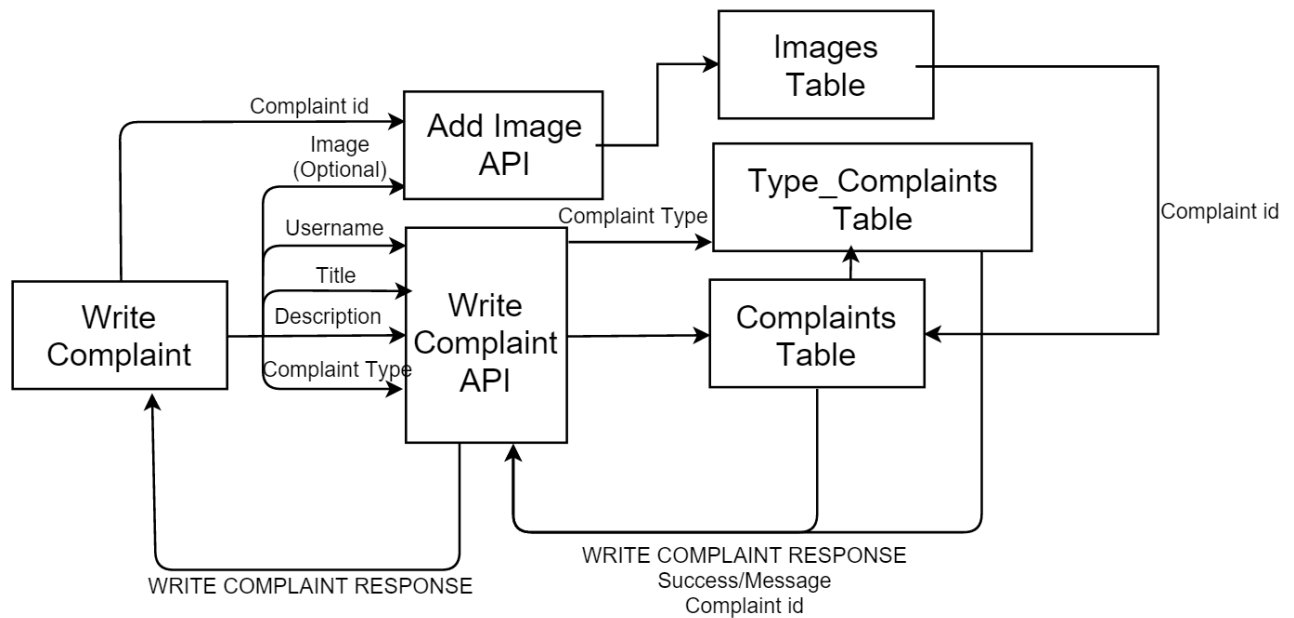


Figure 14: Write Complaint: When a user adds a new complaint with a provision of adding image(s)

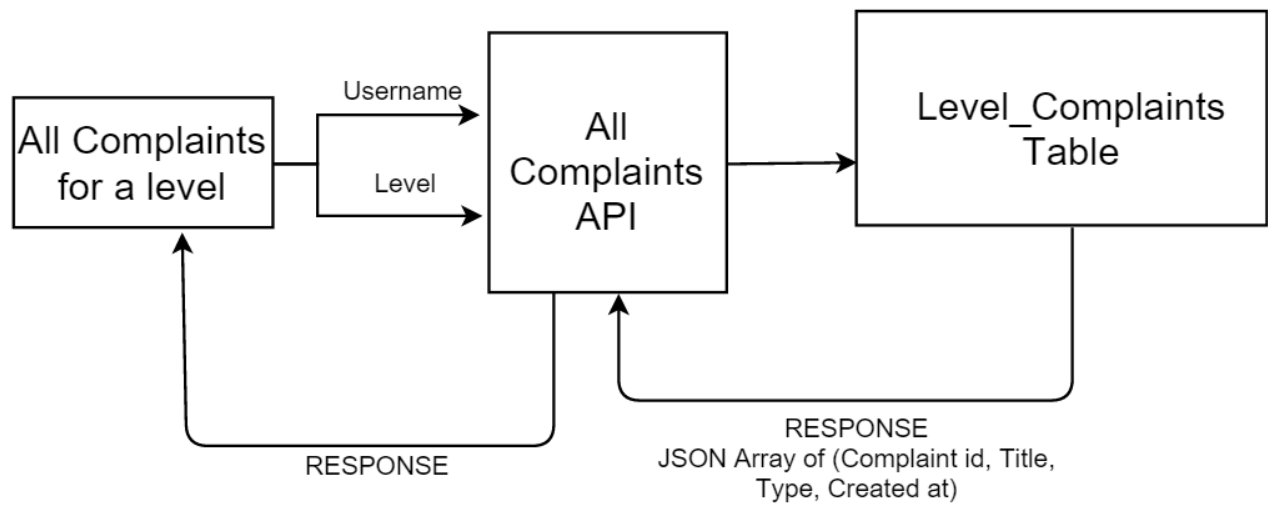


Figure 15: All Complaints: Displaying all complaints related to the user for a particular level

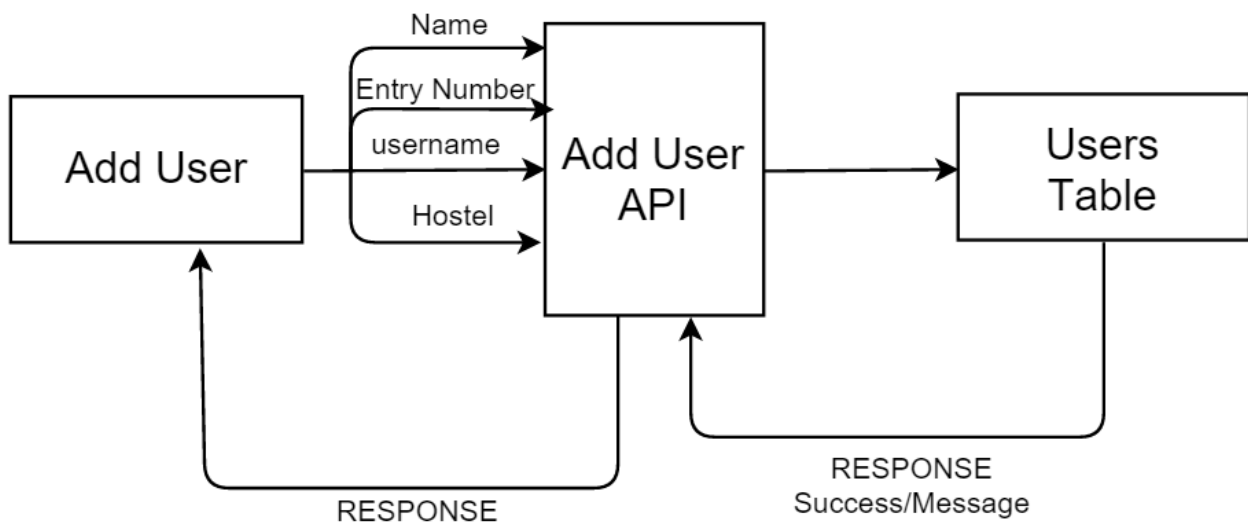


Figure 16: Add User: Flow of events when a specialised user adds a new user to the database

## 7 Features

1. Navigation Drawer on main HomePage activity which contains options to write a complaint, view all complaints, view all notifications, change password, add user(in case of special users) and logout, where these options inflate as fragments.
2. Recycler view to dynamically display notifications and complaints.
3. Write Complaint:
  - Upvote/Downvote a complaint
  - Add comment to a complaint (Can include images)
  - Add tags to a complaint
  - Ability to mark the status of a complaint as Resolved/Pending by the complainant or House Warden.
4. All Complaints:
  - Tabs using Pager Sliding Tab Strip: Complaints are classified according to levels and are visible in different tabs
  - Sort the complaints according to the date or votes.
  - Search a complaint by some keywords.
5. Notifications
6. Alert Dialog Box to give a confirmation to the user about the successful registration of complaint.

## 8 Backend

1. XAMPP server is used to install mainly the Apache HTTP Server, MySQL database, and interpreters for scripts written in PHP.
2. **Request:** Volley library used to send request to the server and handle response.
3. A folder "complaintsystem" is created in the www directory of XAMPP, which further contains the following folders:
  - **include:** Contains the config and connect PHP files.
  - **libs:** Contains third-party libraries
  - **default:** Contains the PHP file to manage the login and logout activities.
  - **complaint:** Contains the PHP file to return all complaints related to a user, sort them and search a complaint in the list of complaints.
  - **editcomplaint:** Contains the PHP file to change the status of the complaint as resolved or pending and adding votes (upvote/downvote).
  - **images:** Handles uploading and downloading of image.

- **notifications:** Returns the notifications for a particular user.
  - **uploads:** Images sent to the server are saved in this folder.
4. **Scope:** The APIs have been designed such that the application can be extended to the web clients as well.

## 9 Future Endeavours

1. Persistent Login
2. Push Notifications
3. Image Uploading and downloading in complaints



## References

- [1] Android network tutorial. <http://dev.mysql.com/>.
- [2] Flowcharts made using:. <https://www.draw.io/>.
- [3] Rest api. <http://www.androidhive.info/2014/01/how-to-create-rest-api-for-android-app-using-php-slim-and-mysql-day-23/>.