

COP290: Institute Level Complaint Management System

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

March 10, 2016

1 Contents

S.No	Title	Page No.
1.	Introduction	2
2.	System Interface 2.1 Login Page 2.2 All Complaints 2.3 Navigation Drawer 2.4 New Complaint 2.5 Notifications 2.6 ShowComplaint 2.7 My Account 2.8 Add User	2 2 3 3 4 4 4 4
3.	Work Flow	6
4.	Entity Relationship Diagram	8
5.	Event Flow	10
6.	Features	12
7.	Backend	13

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 and also 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. It also has the option of downvote/upvote for all the students and the faculties.

The users for this app can be registered by the special users. They involve House Wardens, House Secretaries and Admin. Admin is the administrative head of this application who can only add users by populating the database. They have no role in filing and viewing the complaints. The other users categorised into Students, Faculty and Institute Employees, can further have various tags like Secretaries, presidents etc.

3 System Interface

1. **LoginPage** : This page is the main Login page. The users can login using a Login type (Student, Faculty, Admin, Institute Employee) and a valid username and password (users already registered in the database, which are added by some special users like house secretaries, hostel warden, overall admin, etc).
 - Name
 - Category: (discussed later in details)
 - Student : Tag of hostel, and additional tags like Club secretaries+club, hostel level secretaries, etc.
 - Faculty : Additional tags like Warden+Hostel, Deans, Club presidents+Club (sub clubs for BRCA, for example), etc.
 - Institute Employees : Additional tags like Security Head, etc.
 - Password
2. **AllComplaints** : 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, with the latest at top, and unread complaints as highlighted.

There is an option to sort the complaints in order of the ascending or descending order of dates, number of upvotes, etc.

Also 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.

3. **NavigationDrawer** : A navigation drawer is linked to all the activities, which can be drawn on swiping from left to right. It contains all the basic options like:

- Write Complaint : On selecting this, the user is redirected to the activity- NewComplaint.
- All Complaints : The user is redirected to the HomePage which contains all the complaints which are either public or addressed to the user.
- Notifications : The user is redirected to the activity Notifications, where he/she can find all the unread complaints.
- My Account : The user is redirected to the activity- MyAccount, where he can find all the complaints, written by or addressed to the user.
- Add User (in case of special users) : This feature is applicable only to some special users, who are hostel wardens, deans, club presidents and the app admin. They have the right to add users to the app by registering them on the database. This redirects the user to the activity-Register.
- Logout : The user can logout of the app and come back to the login page.

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

- Complaint Level : Individual, Hostel or Institute
- Complaint type : The complaint types will be different for the students, faculties and institute employees like :
 - Student : At individual level, it can be maintenance, security, academics, etc., at hostel level, it can be mess related, maintenance, computer room, etc., and at institute level, it can be again for some board, common complaint for academics, security, etc.
 - Faculty : At individual level, it can be maintenance, cleanliness of his room, etc., at hostel level, only a warden can have a complaint like regarding discipline, etc., and at institute level, it can be security, discipline, etc.
 - Institute Employees : At individual level, it can be maintenance, security, academics, etc., and at institute level, it can be security, discipline, etc.
- Title : A subject/title must be given to the complaint, which will be visible to all in the notifications.
- 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 type 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 redirected to the AllComplaints activity.
- Help : There is a Help option at the bottom which can be clicked on to get an idea whom to complaint depending on the complaint type. This feature is useful since the users new to the institute (freshers), may not have any idea whom to complaint to.

5. **Notifications** : This contains all the complaints which are unread by the user and looks similar to the AllComplaints page, with 3 tabs, and the unread complaints classified under these sub-categories. The user can select any notification and can view the details as he is directed to the ComplaintDescription activity.
6. **ShowComplaint** : This contains all the information about the complaint like name of the complainant, title and description of the complaint, images attached to the complaint, date and time of the complaint filing, number of upvotes and downvotes, status(resolved or pending) and all the 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, and the warden can do this in case of hostel level complaints.
7. **MyAccount** : This contains all the basic information about the user like:
 - (a) User Profile : It contains the basic information like Name, Category(Student/Faculty/Institute employee) and further tags in the category.
 - (b) Inbox : On clicking over this option, a new activity Inbox opens, which contains all the complaints that have been addressed to the user by the other users of the app. On clicking on any complaint, the ComplaintDescription activity for that complaint shows up.
 - (c) Sent : On clicking over this option, a new activity Sent opens, which contains all the complaints that have been submitted by that user. Again, out of the list of all complaints, one can click on any and the ComplaintDescription activity for that complaint shows up.
 - (d) 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 activity ChangePassword shows up. This contains three fields: Current password, New password Confirm password. If the current password is correct, the new password follows the general rules 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 MyAccount activity.
8. **AddUser** : This activity contains a small registration form to be filled by the special user, with the following field:
 - Name
 - 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 :
 - (a) Student :
 - Hostel Name
 - Position Of Responsibilities :
 - * Normal (No tags)
 - * Hostel level secretaries :
 - Mess Secretary

- Maintenance Secretary
 - Computer Room Secretary
 - Library Secretary
 - Cultural Secretary
 - Sports Secretary
 - House Secretary
 - * Institute level Secretaries :
 - BRCA General Secretary + BRCA club Secretaries
 - BSA General Secretary
 - BSP General Secretary
 - BSW General Secretary
 - BHM General Secretary
 - SAC General Secretary
 - SIC General Secretary
- (b) Faculty :
- Normal (No tags)
 - Hostel Warden : Hostel
 - Dean : Department
 - Club Presidents :
 - * BRCA + BRCA clubs
 - * BSA
 - * BSP
 - * BSW
 - * BHM
 - * SAC
 - * SIC
- (c) Institute Employee :
- Normal
 - Tags like Senior Security Officer,etc.

4 Work Flow

[2]

The workflow can be as follows:

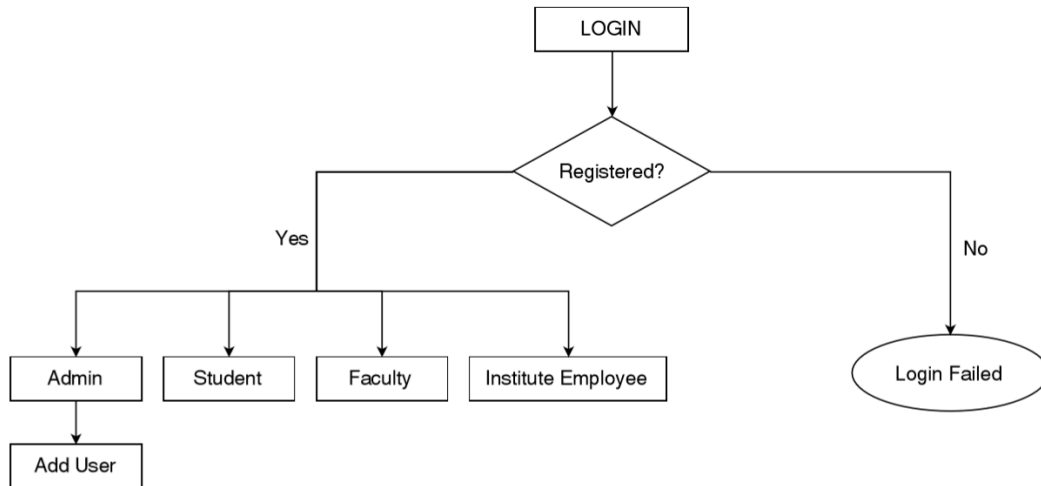


Figure 1: Work Flow

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

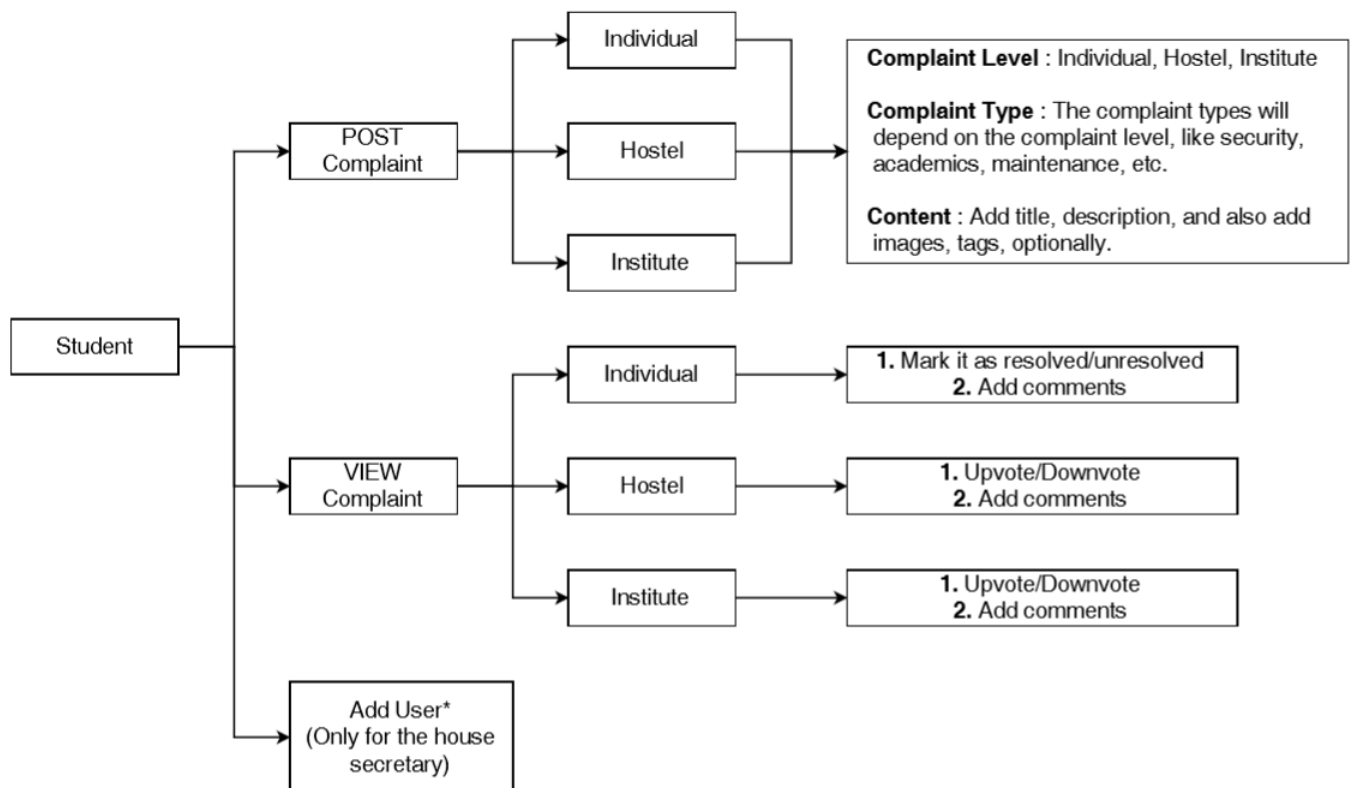


Figure 2: User : Student

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

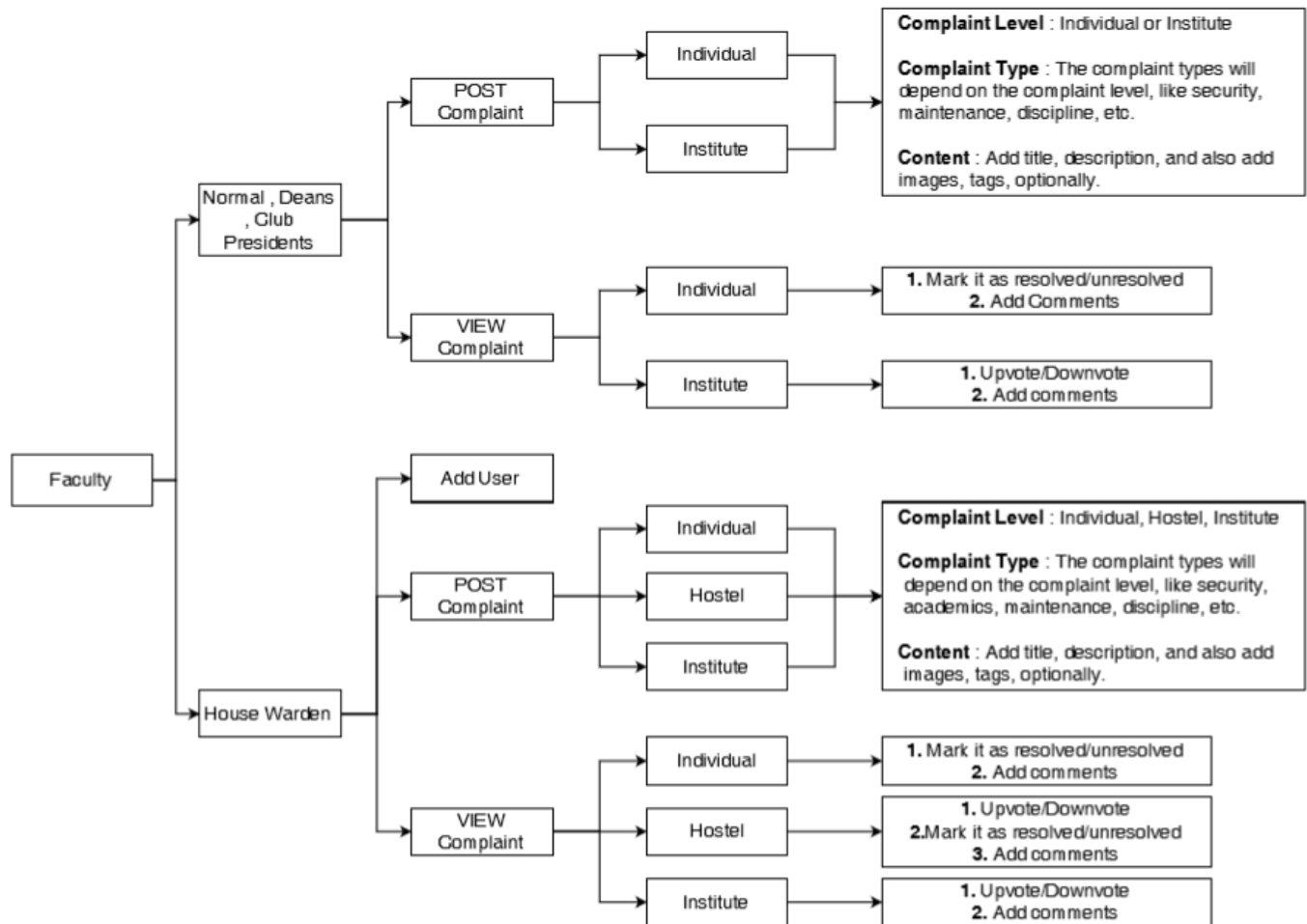


Figure 3: User : Faculty

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

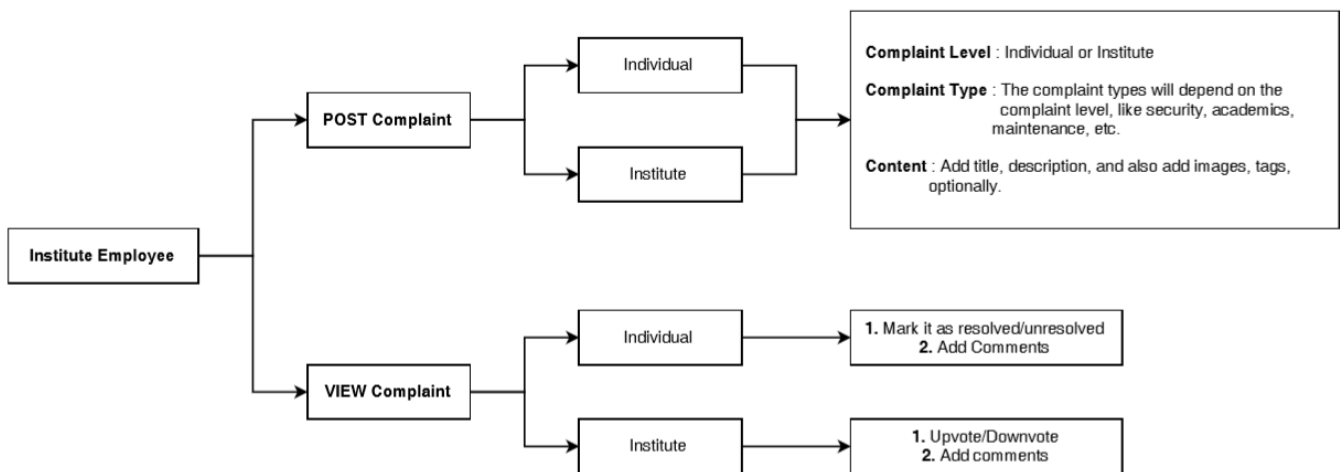


Figure 4: User : Institute Employee

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:

[3]

1. **Login:** <http://localhost/complaintsystem/default/login>
Response: JSONObject of Success Message
2. **Add User:** <http://localhost/complaintsystem/default/adduser>
Response: JSONObject of Success Message
3. **Change Password:** <http://localhost/complaintsystem/default/changepassword>
Response: JSONObject of Success Message
4. **Logout:** <http://localhost/complaintsystem/default/logout>
Response: JSONObject of Success
5. **All Complaints:** <http://localhost/complaintsystem/complaint/allcomplaints/level={level}&username={username}>
Response: JSONArray of (JSONObject of Title, Description ComplaintId) called Complaint
6. **Sort:** <http://localhost/complaintsystem/complaint/sort/level={level}&type={type}>
Response: JSONArray of sorted Complaints
7. **Search:** <http://localhost/complaintsystem/complaint/search/level={level}&str={str}>
Response: JSONArray of Complaints
8. **Get Complaint:** <http://localhost/complaintsystem/complaint/getcomplaint/id>
Response: Complaint
9. **Submit Complaint:** <http://localhost/complaintsystem/complaint/submitcomplaint>
Response: JSONObject of Success Message
10. **Add resolve/pending:** <http://localhost/complaintsystem/editcomplaint/addresolve>
Response: JSONObject of Success Message
11. **Upvote/Downvote:** <http://localhost/complaintsystem/editcomplaint/addvote/id={id}&vote={vote}>
Response: JSONObject of Success Message
12. **Get Image**(for a given complaint):<http://localhost/complaintsystem/images/getimage>
Response: JSONObject of ImageURL ComplaintId

13. **Upload Image:** <http://localhost/complaintsystem/images/uploadimage>
Response: JSONObject of Success Message
14. **Notifications:** <http://localhost/complaintsystem/notifications/allnotifications>
Response: JSONArray of ComplaintId, Complaint title, type level

6.2 Flow Charts:

[2]

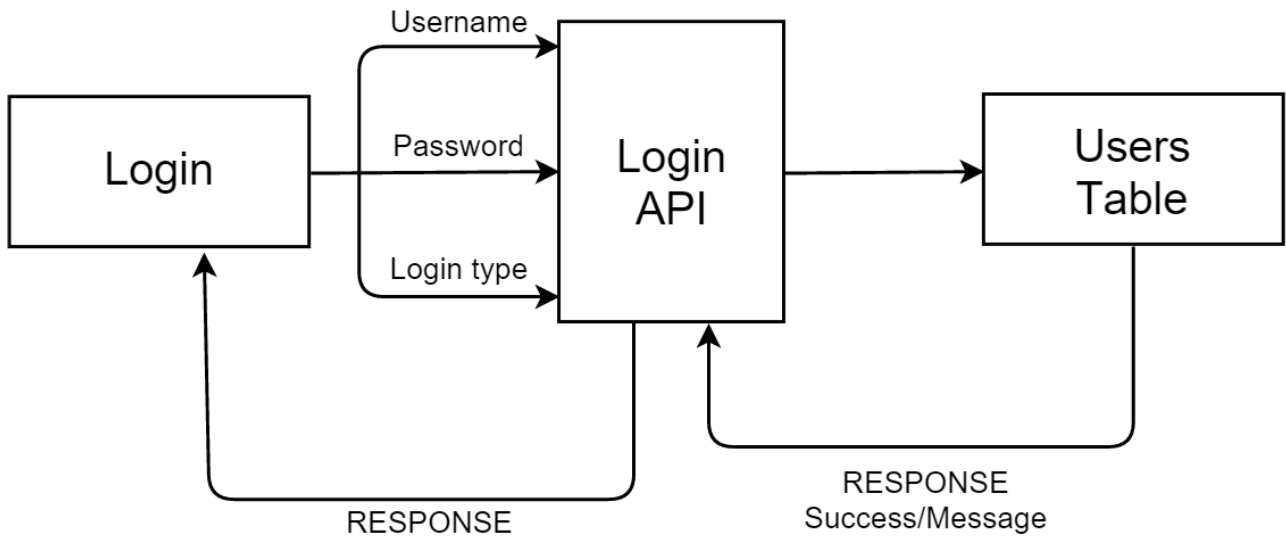


Figure 13: Login: Flow of events when a person tries to login

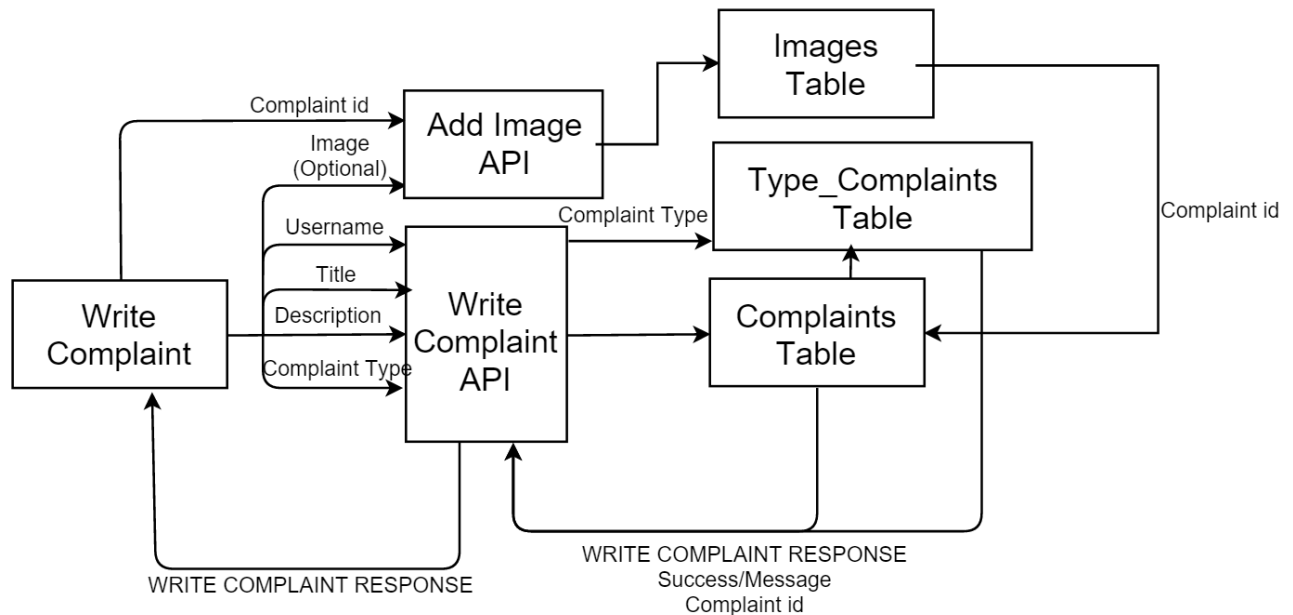


Figure 14: Write Complaint: When a user adds a new complaint with a provision of adding image(s)

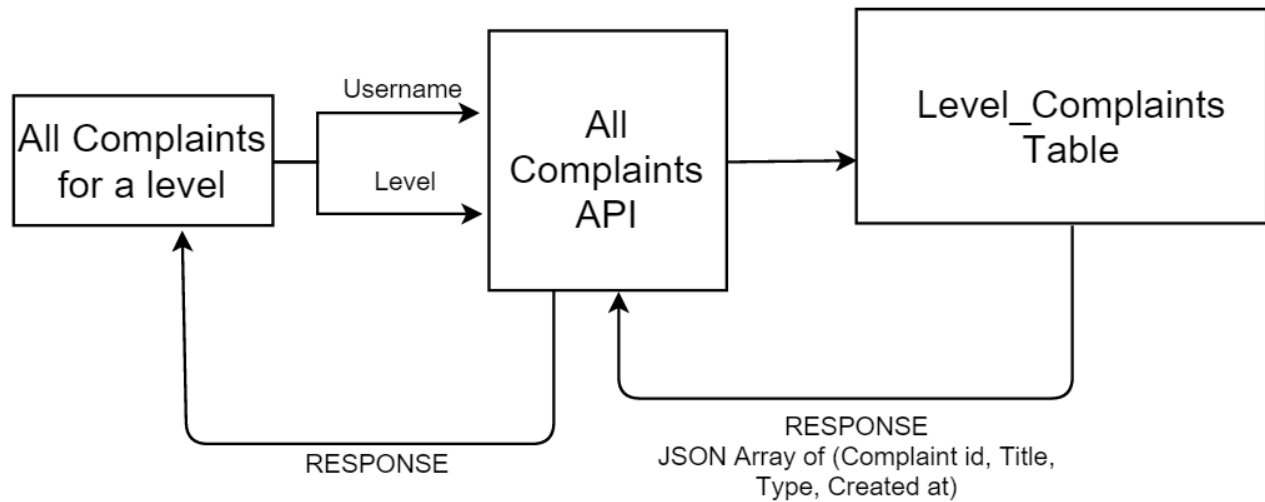


Figure 15: All Complaints: Displaying all complaints related to the user for a particular level

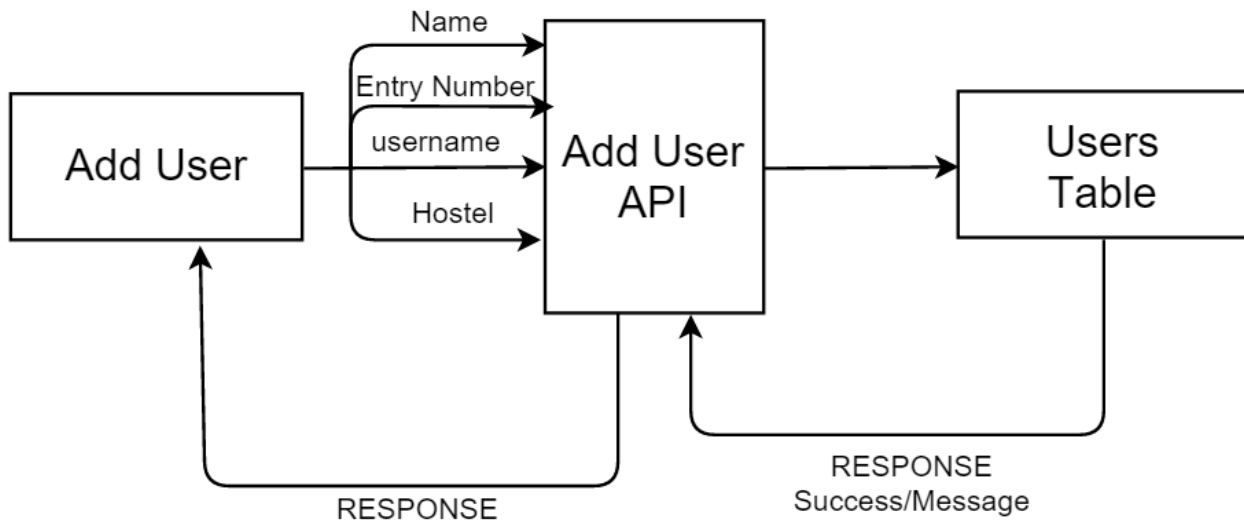


Figure 16: Add User: Flow of events when a specialised user adds a new user to the database

7 Features

1. Upvote/Downvote a complaint
2. Adding image with a complaint
3. Add comment to a complaint (Can include images)
4. Add tags to a complaint
5. Push Notifications using GCM

6. Tabs using Pager Sliding Tab Strip: Complaints are classified according to levels and are visible in different tabs
7. Navigation Drawer on each page activity which contains options to write a complaint, view all complaints, view all notifications, view one's account, add user(in case of special users) and logout.
8. Notifications
9. RecyclerView to display notifications and complaints.
10. Ability to mark the status of a complaint as Resolved/Pending by the complainant or House Warden.

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.

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/>.