COP290: Complaint Management System

Karan Singh (2013EE10465)
Shantanu Kumar (2013EE10798)
Surag Nair (2013EE10504)

March 20, 2016

1 Introduction

In this assignment we aim to design and develop a complaint management system for the students, faculties and institute employees. We intend to develop a platform where the students can submit their complaints easily to the concerned authorities. The complaints are divided into three types :

1. Individual complaint
2. Hostel-level complaint
3. Institute-level complaint

We also intend to extend the scope of the system from Android clients to Web clients too. Our aim is to provide a seamless experience for both students and authorities, and make the system very flexible and easily extensible.

2 App Features

Our App has the following features, each being supported by the corresponding server API that will be called. There is one superuser account who would have the action user to edit the database i.e. add new users, add new user types and add new types of complaints. The responsibility and abilities of the superuser are explained later. He/She would only have a web interface to the system.

- Login to the system with the respective account credentials for different types of users
- View the list of all the complaints filed at the individual-level, hostel-level and institute-level.
- View all the notifications

  - Notifying all the concerned users in case a new complaint is created (all the hostelers in case of a Hostel-level complaint or all the app users in case of a Institute-level complaint).
  - Notifying the users once the complaint is resolved.
  - Notifying the users and concerned authorities if the complaint is edited.
  - The user may choose to notify the action user if the complaint has remained pending for some specified long time period.

- View all the resolved complaints (they will be cleared from the database after 30 days of resolution).

- View details about a particular complaint.

- Upvote or Downvote a hostel level or institute level complaint (this option available only once per user per complaint).

- Post the user's comment/opinion on a hostel level or institute level complaint (only one comment/opinion is allowed per user per complaint).

- Create a new complaint at individual, hostel or institute level
  - In case of a hostel or institute level complaint the user can choose to add more students as admins (along with himself), having the rights to edit the statement of the complaint.
  - User also has an option to attach an image file to the complaint.
  - A suggestion list of complaint types/categories is available to the user to select from while making a complaint. The list is defined by the super user who has the action user to add a new complaint category.
  - Selecting a particular option from the suggestions list will automatically send the complaint to the concerned action user defined by the super user.

- Edit the complaint depending on the type of user.

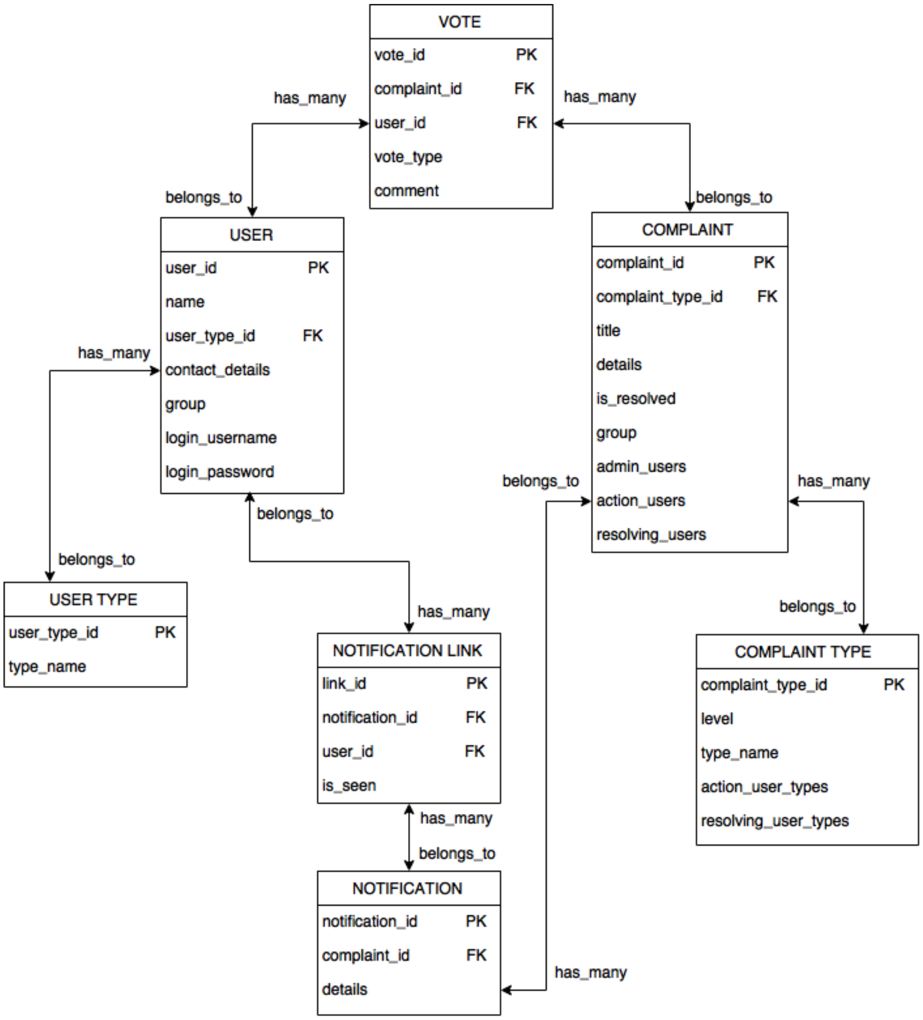
- Search a particular complaint among the list of all the complaints.

- Poke the concerned action user if the complaint is not read by them within 5 days (this feature is available only once after every 5 days as long as the complaint has not been resolved).

- Mark a complaint as resolved (this feature can be availed by some particular users defined by the super user for each type of problem (e.g. warden for hostel level complaints).

- Logout of the system/app.

3 Database and Design Principles



- We are developing the server side application using the Ruby on Rails framework.
- At the core of our database lies extensibility in terms of user types and complaint types. This allows the super user to define new user types as and when required/available (action users such as electrician, plumber, admin, UG Section etc.) and to make new complaint types.

- These complaint types can be directed to the specified action user types so that all complaints of a particular type (e.g. lost items/thefts) will be sent to the designated action user types (security control room, in this case).

- Details of the users are stored in the USER database. Broadly, users are either students (who can post/edit/resolve complaints) and action users (who can view or resolve complaints, and take action).

- The complaints are stored in the COMPLAINT database, with attributes including list of student admins, list of action users and list of resolving users for a given complaint.

- We have a model to store all NOTIFICATIONS that are generated at various events in the work-flow. Each notification belongs to a complaint. Since the USERS and NOTIFICATIONS share a many-to-many association, this relationship is implemented using an intermediate NOTIFICATION LINK model. This model is also used to mark the notification as read or not.

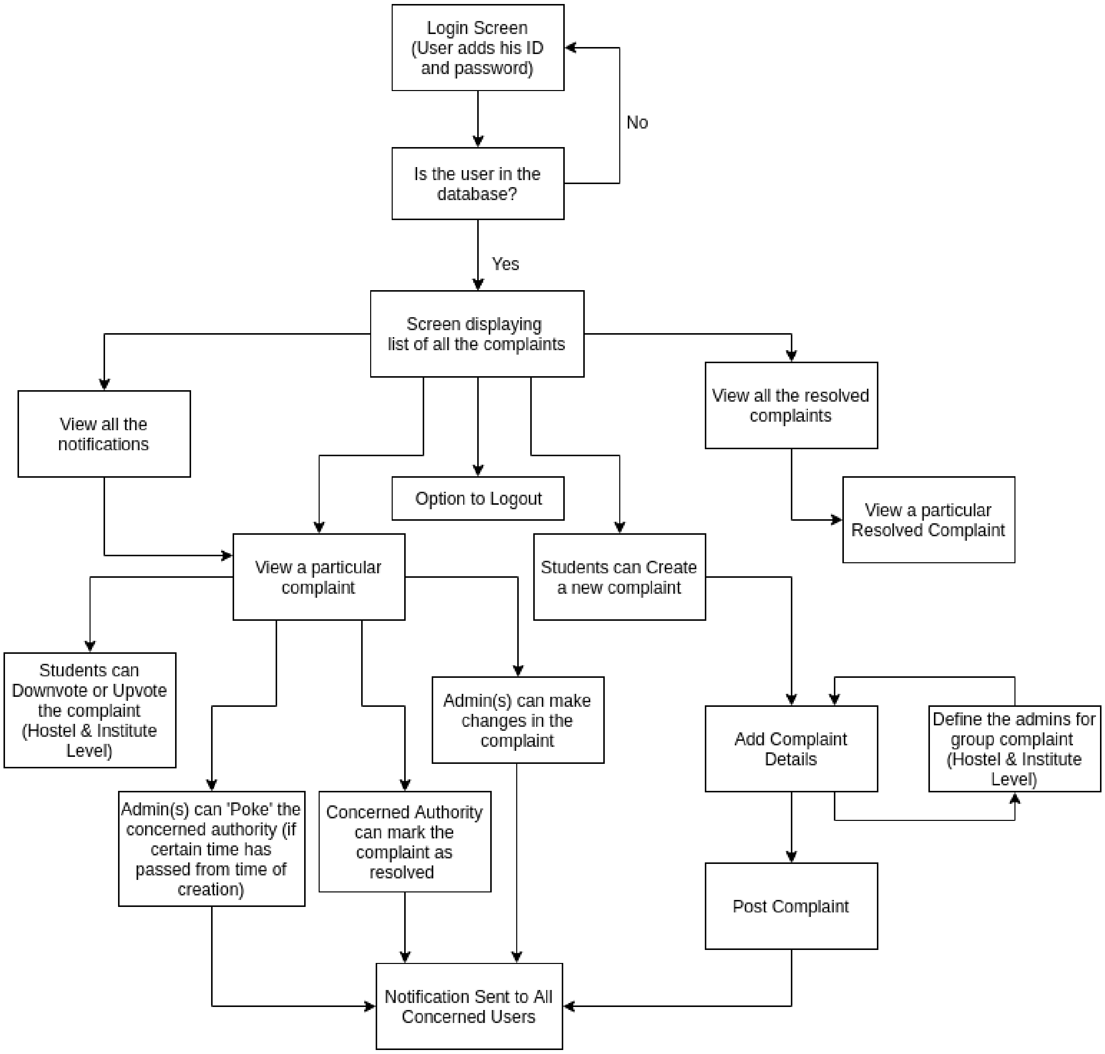
- We have a separate VOTE database to maintain the upvotes, downvotes and comments on complaints. We allow only one comment per user, following the model of 'change.org'.

4 Web Interface for Super User

We will also provide a web interface for the super user to allow great flexibility and extensibility to the authorities. The web interface will allow the super user to add new users, create new user types (e.g. electrician, plumber, UG section, securty control room, etc.) and create new complaint types. Each new complaint type (e.g. theft) will have the option to choose the complaint level (individual, hostel or institute; individual in this case), choose the set of action user types for the complaint type (security control room) and the set of user types who can mark it as resolved (student who posted the complaint in this case). These complaint types would be visible to the students in a suggestion list when they try to make a new complaint on the app.

5 Work Flow

The work-flow shown below depicts the order of events occurrences in the general usage of the app.


Login Screen (User adds his ID and password)
No
Is the user in the database?
Yes
Screen displaying list of all the complaints
View all the notifications
Option to Logout
View all the resolved complaints
View a particular Resolved Complaint
View a particular complaint
Students can Create a new complaint
Students can Downvote or Upvote the complaint (Hostel & Institute Level)
Admin(s) can make changes in the complaint
Add Complaint Details
Define the admins for group complaint (Hostel & Institute Level)
Admin(s) can 'Poke' the concerned authority (if certain time has passed from time of creation)
Concerned Authority can mark the complaint as resolved
Post Complaint
Notification Sent to All Concerned Users

6 API Calls for the App Client

Function	URL	Request Type	Returns (JSON)
User Login	/user/login.json (User credentials in Post params)	POST	User details
User Logout	/user/logout.json	DELETE	Success Token
Get all complaints	/complaints.json	GET	List of complaints for the user
Create a new complaint	/complaints.json (Complaint details in Post params)	POST	Success token
Get all notifications	/notifs.json	GET	List of notifications for the user
Get details of a complaint	/complaints/<id>.json	GET	Details of a complaints
Upvote/Downvote complaint	/complaints/<id>/vote.json (Vote type in Post params)	POST	Success token
Add Comment to Complaint	/complaints/<id>/comment.json (Comment text in Post params)	POST	Success token
Remove vote from complaint	/complaints/<id>/remove_vote.json	DELETE	Success token
Mark a complaint as resolved	/complaints/<id>/mark_resolved.json	PUT	Success token
Poke Action Users	/complaints/<id>/poke.json	POST	Success token
Edit a complaint	/complaints/<id>.json	PUT	Success token
Get complaint types	/complaint_types.json	GET	List of complaint types

Table 1: API calls

The following actions take place on the server when the APIs are called:

- Login: A user login session is created and a cookie is returned in the header of the response. The client is required to send this cookie with every subsequent API call.
- Get all complaints: Depending on the type and details of the user, the corresponding list of complaints visible to the user are retrieved from the database.
- Create a new complaint: This is allowed only for students. The complaint details are obtained from the POST parameters and a new record is added in the complaint database. The complaint type database is looked up to determine the user types of the action users. Using the group and user type of the action users, the USER id of the appropriate action users

is retrieved. Appropriate notifications are generated for all concerned students and action users.

- Get all notifications: Retrieves the notifications of the given user from the NOTIFICATION database.
- Upvote/Downvote complaint: A new record is created in the vote database linking the given user to the related complaint.
- Comment on complaint: The comment column of the vote database is populated for the given complaint and given user if it doesn't exist.
- Mark as resolved: Only the user with the right to mark a complaint as resolved is allowed to use this. The complaint is marked as resolved and notifications are sent to the relevant users.
- Poke action users: Creates a new notification reminding the relevant action users of the complaint.
- Get complaint type: This is called from the Create New Complaint activity to populate the list of complaint types.

7 App Interface

We intend to divide the client side app interface into five activities with some relevant fragments. These activities are common for all types of users (students and action users). Depending on the type of user corresponding visibility criteria are defined to display only the relevant data and enable only required features. The details of different activities are shown below :

- Login Activity : The Login Activity is invoked on startup when the app icon is clicked. Here the user can log into his/her account using the login credentials.

- Main Activity : On successful login, the user is brought to the Main Activity. On the top right corner, we have a notification icon which switches the fragment to show the list of notifications. The Main Activity also contains a Navigation Menu which can be accessed by clicking on the menu button in the toolbar. The Navigation Menu contains options to view the resolved complaint and logout from the app. This activity has three more fragments, each for different complaint type (individual, hostel-level & institute-level), listing all the complaints in the list-view. These fragments are controlled by view-pager. User also has a button at the bottom right corner to create a new complaint.

- Complaint Activity : This activity contains all the relevant details of a particular complaint. Complaint title, description, date of creation, level and type. User has the option to upvote or downvote the complaint (only once) depending on the complaint type. User can also remove his/her vote. Additional feature to poke the complaint resolving authorities if the complaint is pending for a long time(available once after some specified time period).

- New Complaint Activity : This activity is invoked when the user presses the '+' button on the home screen. In this activity user add all the relevant details about the complaint (title, description, level, type, admins, etc.). A suggestion menu occurs when the user clicks on the complaint type text-field. It contains different complaint categories (Electrician related, Hostel Mess related, etc.) defined by the super user.

- Notification Activity : On clicking the notifications button on the top right corner of the screen, this fragment is invoked enlisting all the notification pertaining to all the relevant complaints of the user. The unread notifications are shown with a highlight and the user can tap on any notification to go to the related thread of the notification.

References

[1] Android network tutorial. `http://developer.android.com/training/basics/network-ops/index.html`.

[2] Codeschool ruby on rails tutorials.

[3] Ror tutorial. `https://www.railstutorial.org/book/`.

[4] Ruby on rails guide. `http://guides.rubyonrails.org/`.

[5] Stackoverflow. `http://www.stackoverflow.com/`.

[6] Vogella android tutorials. `http://www.vogella.com/tutorials/android.html`.