Institute Complaint Management System
COP290 : Assignment 2
Design Document

Anmol Gupta (2013EE10437)
Harshdeep Gupta (2013MT60597)
Tarun Agarwal (2013EE10505)

March 10, 2016

1 Databses

- The following tables will be used in the database:
  - **User Authentication:**
    This table contains the attributes *Username, User_id and Password*. Username is the primary key for this table. This is used for user authentication at login.
  - **User Info:**
    This table contains the attributes *User_id, Name, Hostel, Post and UserType*. User_id is the primary key for this table. This is used to display user info after login.
  - **Complaints:**
    This table contains the attributes *Complaint_id, User_id, Description, Type, Student_Visibility, Faculty_Visibility, Up-Vote, Down-Vote, isResolved and Created_At*. Complaint_id is the primary key for this table. This is used to store information about every individual complaint being made.

– **User_Complaint:**

  This table has attributes *User_id and Complaint_id*. User_id is the primary key for this table. This is used to keep a record about which complaints have been added by which user.

– **Hostel_Complaint:**

  This table has attributes *Hostel Name and Complaint_id*. Hostel Name is the primary key for this table. This is used to keep a record of all complaints for a particular hostel.

– **Institute_Complaint:**

  This table has only one attribute *Complaint_id*. This attribute is also the primary key for this table.This stores all the Complaint_id's which are at institute level.

– **Comments:**

  This table has attributes *Comments_id, Complaint_id, User_id, Description and Created_id*. Comments_id is the primary key for this table. This table stores the comments corresponding to the complaints and the User_id's for the users who posted the comments.

– **Resolver:**

  This table has attributes *Category, Supervisor and User_id,Post*. Category is the primary key for this table. This table stores the information about the people who can resolve hostel level and institute level complaint. The attribute Category has domain which include all Hostel names and the category areas of six deans of IITD. The attributes supervisor's and User_id have domains which include the names of people who can resolve the category's complaint and their User_id's respectively. People in the supervisor's domain include secretaries of hostels, wardens of hostels, deans of the various departments of IITD like Dean of Academics, Dean of Faculty etc.

2 API's

List of API's with their details

API Purpose	Request Type	URL Parameters	Server Response
Login	GET	**int** user_id **string** password	**bool** success
Logout	GET	**int** user_id	**bool** success
Up Vote a Complaint	PUT	**int** user_id **int** complaint_id	**bool** success **int** complaint_id **int** up_vote
Down Vote a Complaint	PUT	**int** user_id **int** complaint_id	**bool** success **int** complaint_id **int** down_vote
Resolve a Complaint	PUT	**int** user_id **int** complaint_id	**bool** success **int** complaint_id **bool** isResolved
User Information	GET	**int** user_id	**bool** success **string** use_name **int** user_id **string** hostel **string** user_type **string** post

API Purpose	Request Type	URL Parameters	Server Response
Resolver Information	GET	**string** user_type **string** hostel	**bool** success **int** warden_id **string** hostel_warden array of **int** dean_id and **string** dean_name
View Comments	GET	**int** complaint_id **string** hostel	JSON array of objects where each object contains **int** user_id **int** complaint_id **string** user_name **string** comment
Add Comment	POST	**int** user_id **string** user_name **int** complaint_id **string** comment **string** created_at	**bool** success **int** comment_id
Add Complaint	POST	**int** user_id **string** user_name **string** complaint_type **string** complaint **string** hostel **bool** visibilityStudent **bool** visibilityProf **string** created_at	**bool** success **int** complaint_id

API Purpose	Request Type	URL Parameters	Server Response
Lists of Complaints	GET	**int** user_id **string** hostel	JSON array of objects where each object contains **int** complaint_id **string** title **int** user_id **string** user_name **string** description **string** complaint_type **bool** visibilityStudent **bool** visibilityProf **bool** isResolved **int** resolver_id **int** up_vote **int** down_vote **string** created_at

API	End-point
Login	/default/login.json?userid=<username>&password=<password>
Logout	/default/logout.json
Up-Vote	/complaints/up_vote.json?complaint_id=<complaint_id>
Down-Vote	/complaints/down_vote.json?complaint_id=<complaint_id>
Resolve	/complaints/complaint_resolve.json?complaint_id=<complaint_id>
Add Complaint	/complaints/post_complaint.json?user_id=<user_id>& user_name=<user_name>&complaint_id=<complaint_id>& complaint_type=<complaint_type>&complaint=<complaint>& hostel=<hostel>&visibilityStudent=<visibilityStudent>& visibilityProf= <visibilityProf>&created_at=<created_at>
Add Comment	/complaints/post_comment.json?complaint_id=<complaint_id>& user_id=<user_id>& user_name=<user_name>& comment=<comment>& created_at=<created_at>
View Comment	/complaints/getcomment.json?comment_id=<comment_id>
User Information	/default/getuser_info.json?user_id=<user_id>
Resolver Information	/default/getresolver_info.json?hostel=<hostel>&usertype=<usertype>

3 Event Flow

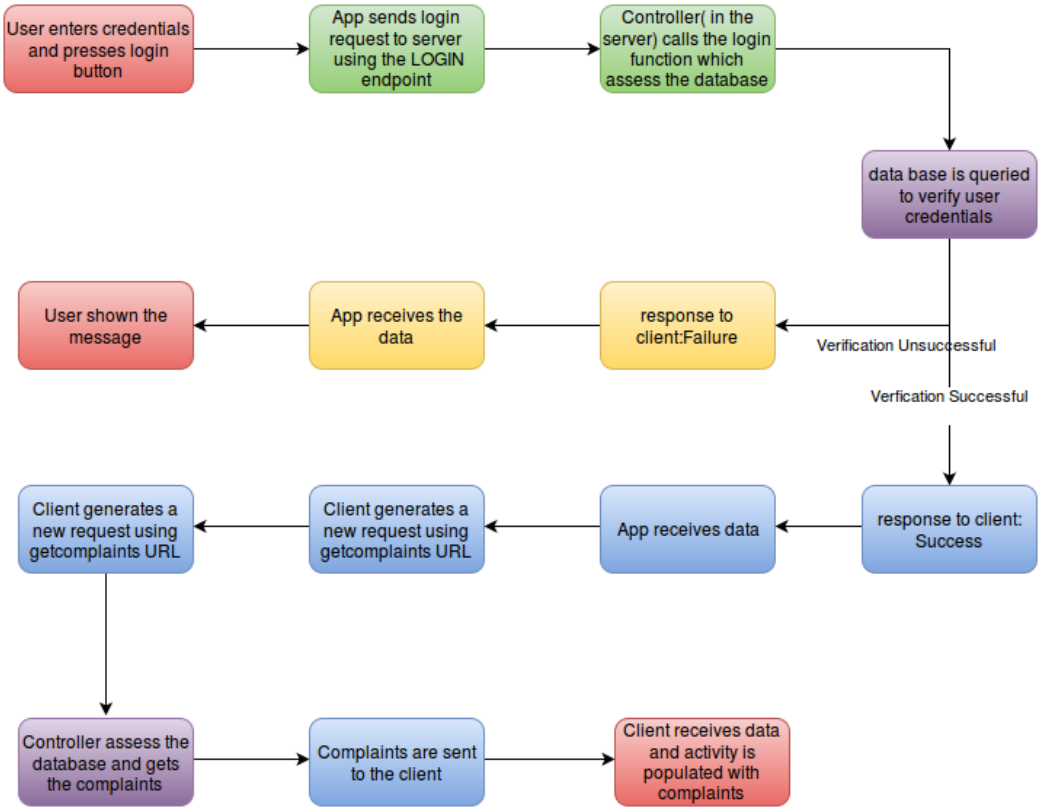

Figure 1: Event Flow for Login

State Color	Code
Red	Terminal State
Green	Client to Server
Yellow	Server to Client
Purple	Database Transaction
Blue	Successful Login

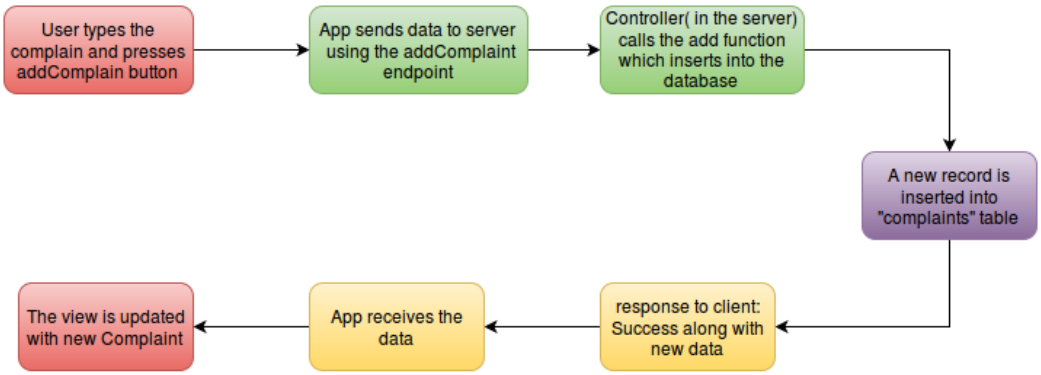

Figure 2: Event Flow for adding a Complaint

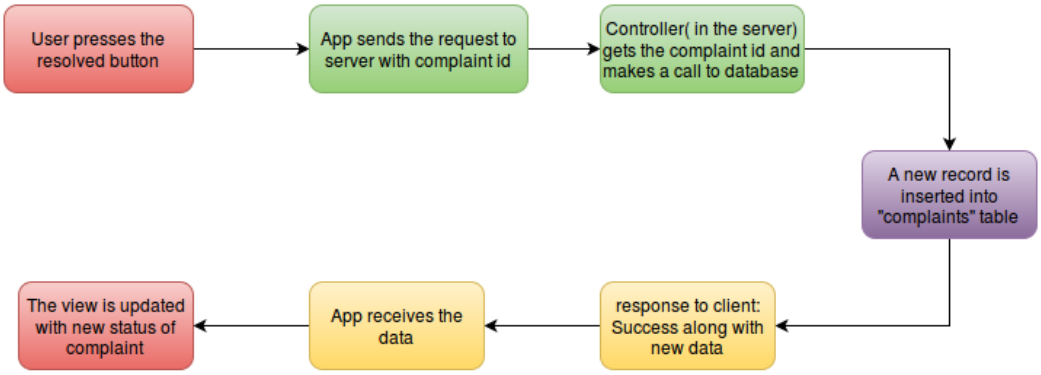

Figure 3: Event Flow for resolving a Complaint

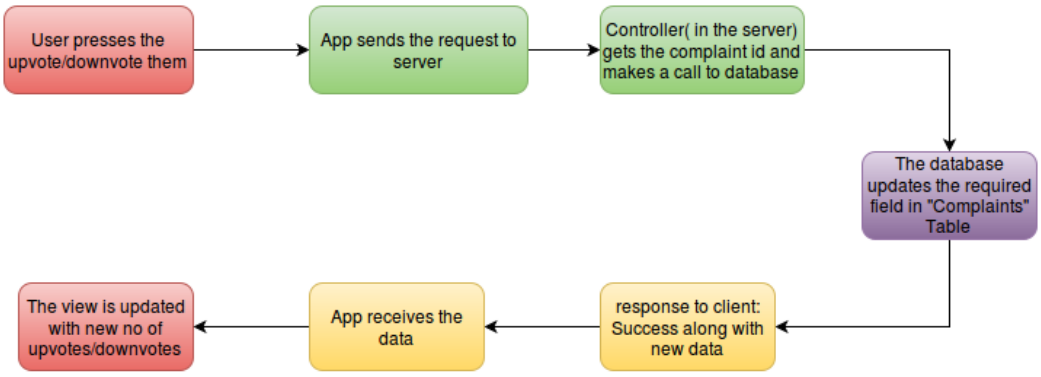

Figure 4: Event Flow for Upvoting/Downvoting a Complaint

4 Bugs Foresight

- **Notifications to Complaint Receiver**
  - **Individual Complaints**
    While making an individual complaint, the user will be presented with an additional option to select the receiver for his complaint. The options provided to him would include Electricity (received by electrician), Furniture (carpenter), LAN port (Computer Cell), Ragging (anti-ragging cell) and Sexual (sexual harassment cell).
  - **Hostel Complaints**
    While making an hostel level complaint, similarly the options provided to him would include selecting the department of his complaint from Maintenance, Cultural, Sports and Messing. A complaint will be received by the concerned secretary from hostel.
- **Incorrect Information**
  An incorrect info for logging in would be replied with a "Login failed" response. A lot of information is categorized to reduce discrepancy in submitted data, like hostel, complaint section, user type, complaint

type and many boolean data which will be received by presenting drop-down menus.

- **Missing Information**
  All information will be marked as necessary fields so that the user would be unable to submit any request with missing data fields.

5 Modularity

The entire work can be divided into

- **Admin Application** This is a simple application which is to be constructed to help in populating the database for the user application. Admin app has privileges to add users to the application.
- **User Application** The user application can be divided into server and client.
  - **Android Client :-** The entire app on android can be divided into the following modules
    - Data Models - Different data models are constructed for different entities like a complaint, a comment, global information in the application about the user.
    - Populate Adapters - Different adapters are constructed to populate appropriate layout with data models constructed above.
    - Networking - A separate class is constructed to handle network request to the server.
  - **Server :-** Following controllers have to be implemented for the server
    - **Default :-** This is the default controller for the web2py server. This implements login, logout, user information, lists of Complaints API's.
    - **Complaints :-** This is the controller for complaints. This implements Add Complaint,Add Comment, View Comment, Up Vote a Complaint, Down Vote a Complaint, Resolve a Complaint API's.

6 Interface of Application

The android interface of the application of the app is made up of the following activities:

- **Login Screen**
  This is the activity which is displayed when the user opens the app for the first time. This activity asks the user his username and password and has a login button in it. After this activity, the user is taken to notifications activity.

- **Notifications Activity**
  This activity is a tabbed activity with three tabs, one for user's personal complaints, one for hostel level complaints and one for institute level complaints. All these pages have an add complaint button. Also, each complaint has a add comment button in it. The appropriate fragment is displayed when the user presses any of this button

- **Add Complaint/Comment**
  This fragment asks user for the description along with asking some fields such as complaint type, visible to students or/and professors etc.

- **Add user**
  This activity is only shown to those users who have admin privileges. This activity asks the details of the user to be added (such as name, hostel , entry no. etc...) and displays the admin an button to execute the action. When the button is pressed, the app sends the data to the server and after receiving response updates