

COP290: Complaint Management System

Parichay Jindal (2013TT10937)

Silky Jain (2013TT10970)

Ujjawal Malik (2013TT10974)

March 29, 2016

This is an application to register and manage complaints at different levels like individual level, hostel level and institute level. Both users and concerned user can view their details. A user can view the complaint, add comment to it or upvote/downvote that particular complaint related to him.

1 User Interface

1.1 *FirstScreen*

First screen demands to enter its credentials and select the user its type i.e. a student or a faculty. the screen has following:

- Edittext username with hint username written.
- Edittext password with hint password written.
- Login

You need to enter both username and password details compulsorily before you click on Submit button else a note "This field is required" will appear with cursor on the text field with missing field.

On clicking a field "username" or "password"; the field name appears on top of it. To login you need to tap the option of your user type, else an error message will appear.



(a) Login Screen



(b) Missing Field



(c) Wrong User

1.2 Menu Screen

After successful login, you reach the next screen where a dialog box appears.

At the top left corner three horizontal lines are visible. On tapping a navigation drawer comes up with

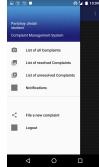
- List of All complaints
- List of Unresolved Complaints
- List of Resolved Complaints
- Notifications
- File a New Complaint
- Add a New User(Optional)
- Notification
- Logout

Only warden/dean will be eligible to add new user and thus that option will be given only to them.

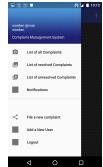
Clicking any of these options will open their respective activities.



(a) Welcome Screen



(b) Drawer1



(c) Drawer2

1.3 Complaints

On selection of list of all complaints , a new activity appears with all the complaints related to that user.



Figure 3: Complaints

Selecting a particular complaint will open a new activity with complete detail of that complaint and also with options:

- Upvote

- Downvote
- Resolve
- Add Comment



(a) Individual Complaint



(b) Hostel level

These options will vary according to the eligibility of user.

1.3.1 File A New Complaint

This will help the user to file a new complaint. It compulsorily requires type of complaint, concerned user to whom the complaint is made, title and description to file a new complaint.

1.3.2 Add a New User

This will help the eligible user to add a new user, for example warden can add a new student. The user is allowed to fill the details of the new user and the new user can setup his/her username and private password as well.

1.4 Notifications

On clicking "notifications" item in navigation bar; an activity displaying the notifications with the details of person who posted and the time stamp. On further clicking a particular notification, details of that particular complaint will appear.

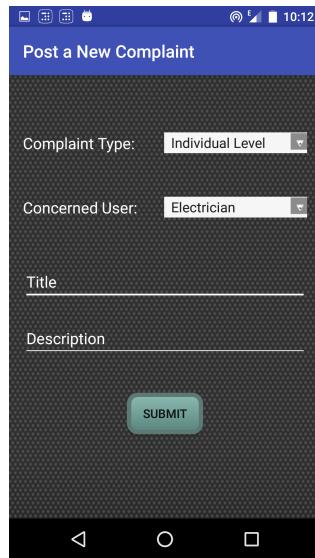


Figure 5: File a new Complaint



(a) Complaint filed successfully

1.5 Logout

This click will bring back the user to the main login screen and he can access no more information without logging again.

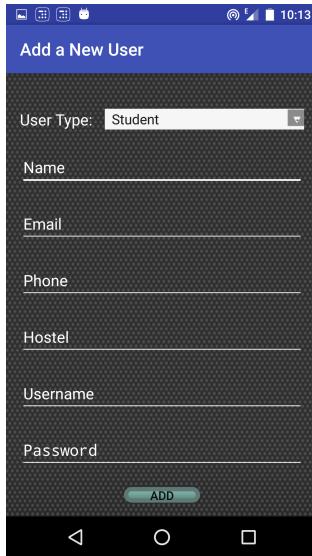


Figure 7: Add a new user

2 Implementation Details

- Login information is stored and the request is sent using the login details in the **login api**.**Volley** is used for network communication.
- **Custom Adapter** is used to display the details of information obtained from **parsing**.
- **Parsing**: This is used to send request and get JSON response which is stored in the string format.
- **Cookie**: When the login request is added to the Requestqueue, a cookie manger is also initialized and set to default until the person is logged in Moodle Plus which helps to remember the user which is logged in to fetch his courses,grades,etc.
- **PutExtra** : When the user presses submit button, the name and id are also sent to Home Screen as Extra items in a bundle.
- **Dialog Box** : To inform the user the validity of action performed.



Figure 8: Notifications

- **IP Address :** To connect with the local server generated from a PC using web2py, IP address of the PC which depends on its network connection is to be used to generate the url for every Stringrequest.
- **Attempt Login:** This ensures that none of the field of username and password is empty while logging in.
- **Attempt Submit:** This ensures that none of the field of title and description are left empty while creating a new thread.
- **mProductList:** mProductList is the array in which the parsed details are stores using functions showgrades, showcourses, showassignments
- **ivProduct:** This gets the id of the item clicked and opens the corresponding activity.

3 Server Design:

- We have used **web2py** framework to design our local server.
- **Database storage** : 4 Database Tables have been used to store all the details. These are users, complaints, comments and valid user.
- **Table users:** This is used to store the username,password,name,hostel,type,phone number and email ids of all the users.
- **Table complaints:** This stores the complaint title,description,userid,concerned userid,comments,upvote users,downvote users,status,type etc
- **Table comments :** Stores all the comment descriptions along with the complaint id and user id for each comment.
- **Table validuser :** Stores the userid of currently logged in user and is updated with each login.
- **APIs used :** For each function that the application can do, a separate API has been designed. There are 2 .py controller files used for API designing.
- **default.py:** Within this the functions for login, logout and new user are written
- **complaints.py:** within this are functions for display of all complaints, resolved complaints, unresolved complaints, particular complaint, notifications, posting a new complaint, posting a new comment, upvote, downvote and resolve.

4 Server APIs

1. Login (API 1):

assignment2/default/login.json?userid =<username> &password =<password>

The server will check the username and password entered against the stored data in users table and respond.

Response : { "users": { "userdetails" }, "success": "true" } (if correct user details else will remain the default value 'false')

2. Logout (API 2):

assignment/default/logout.json

The cookie sessions and cache get cleared. Sets all tokens to there default values.

3. List of all complaints (API 3) :

assignment2/complaints/mainlist.json

All the complaints that the logged in user has created or can view will be sorted using the user id as key, from the complaints table and added in the JSONArray complaints.

Response : {"complaints": [], "users":{details}}

4. Resolved Complaints (API 4) :

assignment2/complaints/resolvedlist.json

All the complaints that the logged in user has created or can view will be sorted using the user id as key, from the complaints table and added in the JSONArray resolved after checking the status of each complaint.

Response : {"resolved": [], "users":{details}}

5. Unresolved complaints (API 5) :

assignment2/complaints/unresolvedlist.json

All the complaints that the logged in user has created or can view will be sorted using the user id as key, from the complaints table and added in the JSONArray unresolved after checking the status of each complaint.

Response : {"unresolved": [], "user":{details}}

6. Post a new complaint (API 6):

assignment2/complaints/new.json?title=< title >&description=< description >&comp

The new complaint posted will be stored into the complaints table in the database using the user key. Also, the notifications table will be updated.

Response: {"comment_id": "<id>", "success": "true"} or
{"err_msg": "<error>", "success": "false"}

Post: "notifications": "<name> has posted a new complaint titled <complaint_title>"

7. Post a comment to a complaint (API 7) :

assignment2/complaints/post_comment.json?complaint_id=<complaint_id>&description=<comment>

The posted comment details will be added to the database through JSONArray comments and the user details will be added to the JSONArray **comment_users**. Also, the notifications table will be updated.

Response:{comment:{}, "user":{}, "success":"true"} or

{"err_msg":"<error>","success":"false"}

Post: "comment_users":"<userdetails>","complaints":"<complaint details>","comments": "|commentdetails|"

8. Info of a particular complaint (API 8):

assignment2/complaints/complaint.json/ <complaint_id>

The **complaint_id** will be used to retrieve the details of the particular comment.

Response:{complaint:{details},comments:[],comment_users: [] }

9. All notifications(API 9) :

assignment2/complaints/notifications.json

All the notifications corresponding to the logged in user will be retrieved.

Response:{ "notifications":[] }

10. Upvotes(API 10) :

assignment2/complaints/upvote.json/ <complaintid>

The user details will be added to the JSONArray **upvote_users**.

Response:{ "users":{}, "success":"true" } or

{ "err_msg":"<error>","success":"false" }

Post: "upvote_users":"<userdetails>"

11. Downvote11 (API 11):

assignment2/complaints/downvote.json/ <complaintid>

The user details will be added to the JSONArray **downvote_users**.

Response:{ "users":{}, "success":"true" } or

{ "err_msg":"<error>","success":"false" }

Post: "downvotevote_users":"<userdetails>"

12. Resolve (API 12) :

assignment2/complaints/resolve.json/ < complaintid >

The status of complaint will change to resolved if the user has the power to do so. Response:{“complaints”:{}, “success”：“true”} or {“err_msg”：“<error>”, “success”：“false”}

13. New user (API 13) :

assignment2/default/new.json/username = < username > & password = < password > &

The new user will be inserted to users table Response:{“users”:{}, “success”：“true”} or {“err_msg”：“<error>”, “success”：“false”}

Except when posting a new complaint or when commenting/upvoting/downvoting a complaint, we use GET method to call the API

5 References

- Android Development tutorial :
 - <http://developer.android.com/training/index.html>
 - <http://vogella.com/tutorials/android.html>.
- Android window leak issues and debugging :<http://stackoverflow.com>
- Android network tutorial: <http://developer.android.com/training/basics/networkops/index.html>
- Web2py tutorial: <http://www.web2py.com/books/default/chapter/29/00>