IIT Delhi

Institute Level Complaint Management System

Design Document

Author:	*Entry No.:*
Ashley Jain	2014CS50280
Dasari Chandu	2013EE10449
M.Sreevatsa Saurabh	2013EE10475

Software design document:
for Web Server and android client app

on

March 2016

iit delhi

0.1 Introduction

- Purpose

  - The purpose of this project is implement a complaint management system for IITD, which allows its users to post complaints related to various aspects like the infrastructure, academics, sports, institute policy,etc. and also allows the institute administration to easily manage and resolve complaints, using various features of our app.

- Scope

  - This app allow user to post different levels of complaints like ***Institute level, Hostel Level and Individual complaint*** . The complaint-poster may find it useful to do so, for privacy reasons and also so that the complaint can be easily found by intended recipients (In this way it also benefits the admin-users).
  - Any group-level complaints can be seen by all the users belonging to that particular group (For example entire hostel can see the hostel level complaints of that particular hostel), and they can also ***comment*** *to discuss and also* ***vote*** complaints up or down.
  - The complaints should be given one or more departments to which the poster thinks the complaint belongs and this can be edited by the admins of that department so as to redirect the complaint elsewhere. Users can also filter all the complaints they have access to, using in-built filters for
    * particular user related complaints
    * hostel related complaints
    * a particular department related complaints.
  - Users can also search for complaints or users and view the user profiles and complaints posted by the user etc.
  - (IF time permits) User can chose to follow a particular complaint after which any activity related to the complaint sends a notification to the user.

- Project Overview

  - Server (Web2Py framework,Database)
    * web2py is web framework for web development with database-driven web applications in Python. This framework automatically respond to all request type created by API endpoints.

    * Database table collects all details of user,complaint,etc.It shows list of stored complaints or notifications while reponding to created request

  – Client (Android App)

    * Android is mobile devices operating system for app implementation in JAVA and android SDK. User can create complaint and comment and read status of complaint and other user's complaint posted through this app easily on smartphone or tablet devices.

0.2 System Interface

0.2.1 Login Page

- Username,password and login button
- signup for new user

0.2.2 SignUp Page

- new username,hostel_name,type,department,password,confirm password
- unique entry_no

0.2.3 Complaints List Page

- Starting page after successful login
- showing list of complaints with title,user,time,no. of votes up/down
- sort by complaint name,date and filter by type,user's hostel,username

0.2.4 Profile Page

- Details of user like username,entry_no,etc
- Complaints list posted by user with title,time,no. of votes up/down

0.2.5 Complaint Page

- Showing selected complaint with all details(title,body,images,etc)
- other user can vote up or down in this page
- resolving complaint and editing complaint can be done by user who posted this complaint

0.2.6 Notification Page

- Showing all unread notification with complaint title,time,entry no

0.3 Work Flow

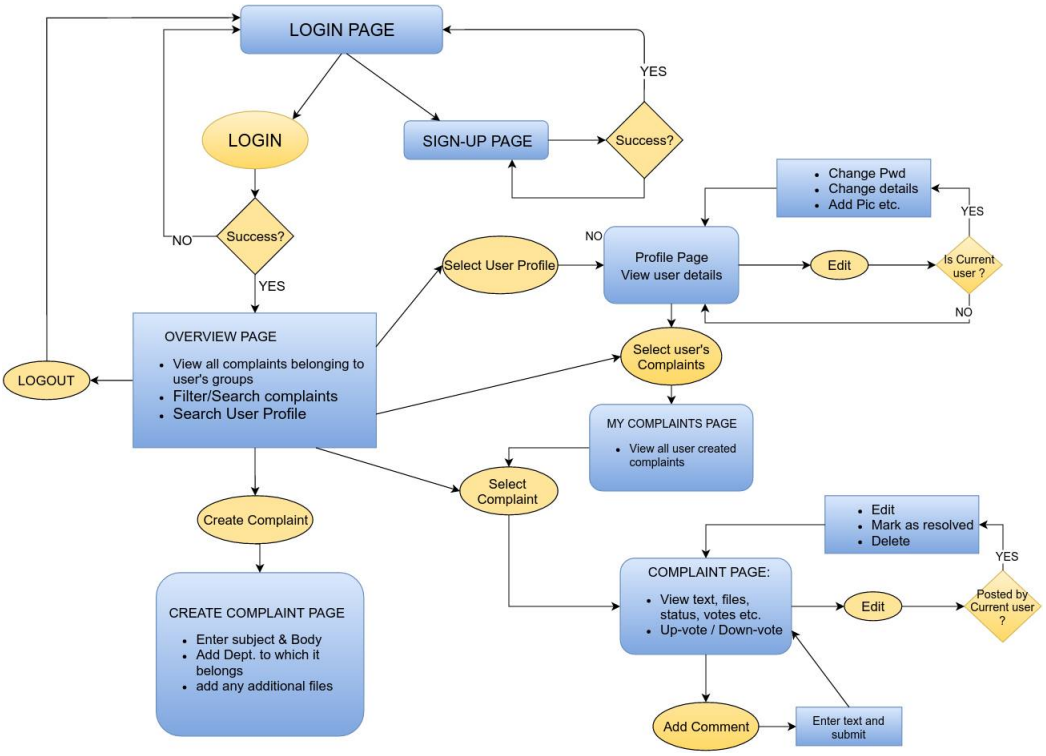

FIGURE 1: Work Flow

0.4 Entity Relationship Diagram

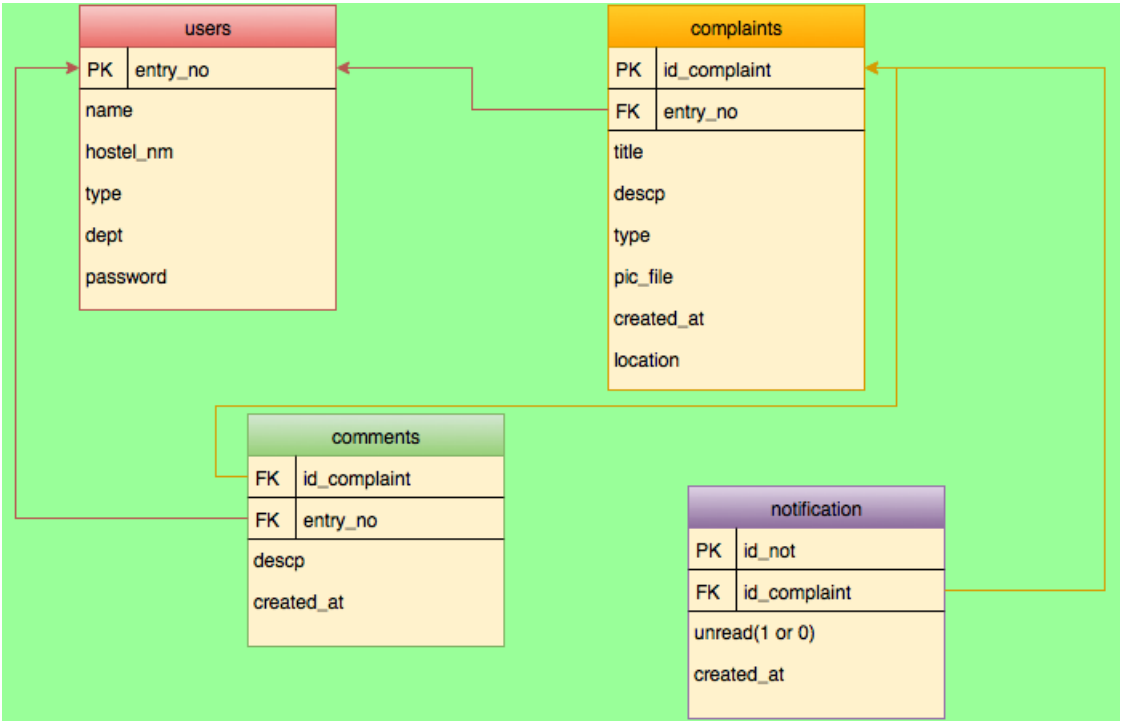

FIGURE 2: ERD

0.5 Event Flow (with expected API)

- Login API:
  - parameters to be passed: username,password
  - request type: GET
  - User-application takes these parameters and packages them into a POST request and posts them to the API, the server receives these parameters and checks if username exists in database. If not present, an error message is sent in the response back to the user-application. If the username is present then the password for that user is retrieved from the database to the server, where it is compared with the user-entered password. According to the result of the comparison, either an error message is sent back or the user is logged in and a session is started along with cookie being stored on the user device. If the session is created successfully the user is also taken to the overview screen.
- SignUp API:
  - parameters to be passed: username,entry no,confirm password,password,Hostel name
  - request type: POST

  - The server takes the parameters from the API and checks the database for any user with the same entry number if there is already such a user it doesn't allow account creation, server then checks if there is any user already registered with the same username, if there are none, it creates a new entry in user table with given parameters and sends a success message back to the user-app.

- Complaint List API (overview):
  - parameters to be passed : username, hostel, department
  - request type: POST
  - server gets the parameters from the API, it then cheeks if any of them is NULL, the NULL parameters are then ignored and the database is queried for complaints which have remaining attributes same as those given in parameter values.

- Particular complaint API:
  - parameters to be passed: Complaint ID
  - request type: GET
  - The servers just takes the complaint ID given in the API as parameter and queries the complaint database for complaint with the same id and returns the attributes as a JSON object to the user app where it is accordingly displayed. It also queries the comments table for all the comments with same complaint ID and sends them too in the JSON object.

- Profile Page API:
  - parameters to be passed: Entry no
  - request type: GET
  - The server gets the entry number from the API and queries the database in users table for user having the same entry number attribute. It then creates a JSON object with the user data and sends it back to the user-app which then displays the content in an intuitive way.

- Create complaint API
  - parameters to be passed: Subject, Body(Text), Images, User Entry No., Group, Dept
  - request type: POST
  - The server creates a new entry in the complaints table with the attributes taken from the API and then assigns a complaint ID to it. which from then acts as a unique identifier for that complaint.

- API to update changes in the complaint:
  - parameters to be passed: Complaint ID, Subject, Body(Text), Images, User Entry No., Group, Dept
  - request type: POST
  - This is similar to the create complaint API. The difference is that here, instead of creating a new complaint entry and assigning a complaint ID, the server first queries the complaint table for the complaint with same complaint ID and then updates the attributes with non-NULL new attributes.
- create comment API:
  - parameters to be passed: user entry number, body, complaint id.
  - request type: POST
  - The server creates a new entry in the comments table with the received complaint id and body and user entry number. and returns a success or failure message to the user-app.
- Profile Update API
  - Parameters : Entry No, Name, password, Image File
  - request type: POST
  - The server first searches for the user with given Entry no and then updates all the entry attributes whose received values are non-NULL.
- Notification API:
  - request type: GET
  - The server create new push notification after posting new complaint or updating existed complaint and send to all user in case of institute level,to all hostel resident in case of hostel level or to individual user.
- Logout API
  - request type: GET
  - Server deletes the cookies in the user-device and ends the session.

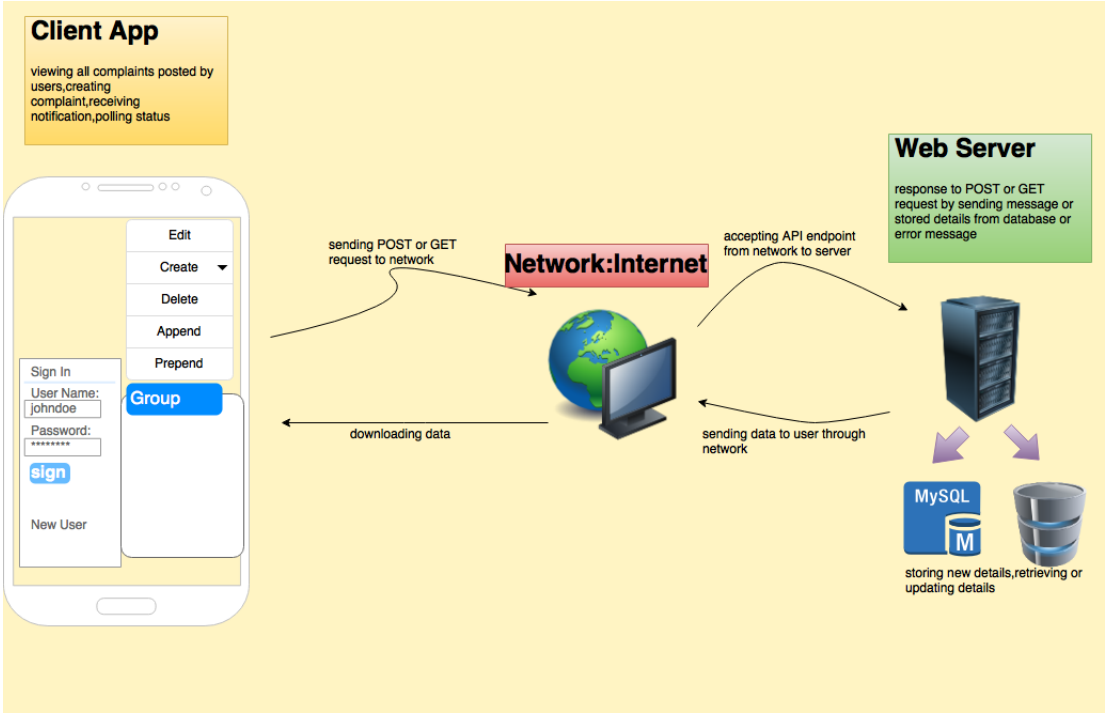

FIGURE 3: Event Flow

0.6 Feature

- Type of complaints:-
  - Individual Level: This comprises of the complaints at his hostel room level
  - Hostel Level: This is public complaint of the hostel which most of the resident are facing
  - Institute level: This is institute level of complaint and it should be visible to all user
- Add complaints: User can create a new complaint of individual level ,hostel level (his respective hostel) and an Institute level complaint. He or she can add an image regarding the complaint.
- Filter/Search : We can search the complaints by the keywords and filter the complaints by the things we need.
- Sort : We can sort the complaints by the following categories like -Level(i.e Hostel,Institute) ,Date Added , Pending/Resolved , Name.
- Up-vote/Down-vote : Every user can up-vote or down-vote the complaints if they agree or disagree with the complaint respectively
- Notify: User can add a option notify me about that complaint then we will notify him about the constant updates on the complaint

- Poll : Users can add polls about the things they want the management to change and other users can vote them