**CHAPTER 1**

**INTRODUCTION**

**1.1 GRIEVANCE**

Handling and responding the complaint is one of the challenge and responsibility by the Municipal Corporation (MC) or local governing body with in the town limit. To address these issues there has to be a platform, which makes people come forward to solve those problems by joining their hands with the government bodies, for which there proposed a platform with two main technical aspects: an innovative geo-social model to profile users along different variables, such as ward number, location and subject a new Android-based platform to collect sensing data from smart phones, automatically or with user help, and to deliver sensing/actuation tasks to government bodies. Thus, this platform will serve as a medium of communication for the public and the government bodies. All the complaints would be registered and the response would be generated by the government bodies and the status of the complaint would be intimated to the public regarding the complaint

Customers have the right to complaint or give comments on the services and facilities provided. Unfortunately, in most of cases customers face difficulties to channel their complaints and comments to the right party. The customers also have difficulties in knowing the status of their complaints. There has been extensive research in the area of eservice for municipal use. It is important to know the use and efficiency of services that are provided by a MC using latest and better technologies. There have also been research [1] which address the uses of e-services for physically challenged citizens. E-services were used in Europe for a while, but they have been extensively attracting far use up in India, in a big way in large cities (like Mumbai, Bangalore etc.) recently. There are separate departments within the MC to handle different problems of the city. It is important for the MC to know about the problems as and when they occur or come into existence in the city and respond to them, so that the problem areas can be dealt with quickly and efficiently. For easier handling of City upkeep, the city is divided into wards. The Bangalore city has about 198 wards. Any complaint relating to or originating from a ward is only handled by an officer associated with that ward. Addressing a complaint gets significantly delayed if the complaint corresponding to one ward is routed to a different ward. Citizens must participate actively which is very important for efficient functioning of all the utilities and services in the area under MC. MC allows the citizens to register their complaints using various interaction mechanisms. A correctly routed complaint can be made to handle promptly by the concerned departmental authority in that ward. An appropriate departmental authority is notified about the complaint for action once a complaint is registered by the citizen. The person who complains is notified of the status of the complaint.

**1.1.1 IMPACT OF INDIVIDUAL USER/COMMUNITY BEHAVIORS**

A participatory sensing application that uses location data sampled from everyday mobile phones to calculate personalized estimates of environmental impact and exposure [2]. It is an example of an important class of emerging mobile systems that combine the distributed processing capacity of the web with the personal reach of mobile technology. This paper documents and evaluates the running PEIR system, which includes server-side processing stages such as HMM-based activity classification (to determine transportation mode) automatic location data segmentation into "trips''; lookup of traffic, weather, and other context data needed by the models; and environmental impact and exposure calculation using efficient implementations of established models. Additionally, they described the user interface components of PEIR and present usage statistics from a two month snapshot of system use. The paper also outlines new algorithmic components developed based on experience with the system and undergoing testing for integration into PEIR, including: new map-matching and GSM-augmented activity classification techniques, and a selective hiding mechanism that generates believable proxy traces for times a user does not want their real location revealed.

**1.1.2 EXPERTISE**

An officer’s expertise demonstrates how capable he or she is at doing particular tasks. Two types of indicators point to worker expertise credentials and experience.

Credentials are documents or evidence from which the requesters or crowdsourcing platform can assess a worker’s capabilities as regards a particular crowd sourced task. Information such as academic certificates or degrees, spoken languages, or geographical regions that a worker is familiar with can be credentials. Experience refers to knowledge and skills a worker has gained while working in the system

**1.1.3 REPUTATION**

The trust relationship between a requester and a particular worker reflects the probability that the requester expects to receive a quality contribution from the worker. At the community level, because members might have no experience or direct interactions with other members, they can rely on reputation to indicate the community-wide judgment on a given worker’s capabilities. Reputation scores are mainly built on community members’ feedback about workers’ activities in the system. Sometimes, this feedback is explicit — that is, community members explicitly cast feedback on a worker’s quality or contributions by, for instance, rating or ranking the content the worker has created. In other cases, feedback is cast implicitly, as in Wikipedia, when subsequent editors preserve the changes a particular worker has made.

**1.1.4 TASK DEFINITION**

The task definition is the information the requester gives potential workers regarding the crowd sourced task. A main element is a short description of the task explaining its nature, time limitations, and so on. A second element is the qualification requirements for performing the task. These specify the eligibility criteria by which the requester will evaluate workers before accepting their participation.

For example, task requesters can specify that only workers with a specified percentage of accepted works (for example, larger than 90percent) can participate, or that only those workers living in the US can take part in a particular survey.

**1.1.5 TASK DESIGN**

Task design is the model under which the requester describes his or her task; it consists of several components. When the requester design task, he or she provides some information for workers. The requester might put a few criteria in place to ensure that only eligible people can do the task, or specify the evaluation and compensation policies. Four important factors are identified to contribute the quality task definition, user interface, granularity, and compensation policy.

**1.2 CURRENT STATUS OF THE E-GRIEVANCE**

As it is apparent from the evolution of the E-Grievance system - both the Public space Notification and Complaint system- the system is not stagnant but rather developed over time. As the status of e-grievance system in Amsterdam divers across the districts, the adoption of e-grievance was not planned at one shot and implemented in similar ways across the districts but rather the cultural, political and organizational difference played an important role. The recognition of complaints and notifications poses a critical view as there is a gap between the designer’s view and the user’s perceptions. A citizen may lodge a complaint believing it is legitimate complaint which the administration may disregard it as a complaint. Such misunderstanding and misconceptions are critical to the development of the system and day to day running of the e-grievance. In most cases, the recognition and registering of oral complaints have increased the number of complaints and put some pressure on the municipality to respond to citizen’s grievances. In General, the future development of the E-Grievance system are said to be towards reducing the increasing number of complaints and reports by avoiding the lessons learnt to reappear in the future. This would only be possible if post analysis of the system is done and inserted into the policy of the organization. Overall, the systems is in its early stage but an evolving and growing towards serving citizens in the best and speedy way. In view of the actual status of E-Grievance in Amsterdam, obviously the further development and sophistication of E-Grievance as such remains a challenge for the near future. The development of Public space notification seems to be ahead than the complaint handling in using ICT to enhance the e-participation and operational efficiency.

**1.3 E-GRIEVANCE SYSTEM: INFORMATION AND COMMUNICATION TECHNOLOGY**

There are varieties of ways citizens can complain and notify the local government in Amsterdam. Citizens can complain and notify by: by post mail, email, website form, visiting the front desk and filling complaint form, and calling telephone to the front office or directly to the complaint coordinator. In addition, it was possible to notify to the environmental police if the problem has to do with the environmental problems. Citizen’s preference varies widely between the complaint handling and Public space notification. In complaint handling the most preferred channel is calling the help desk of district office. Next preferred

Channel by citizens is visiting the help desk which citizens come to complain in the municipality building. The Internet channel was the third preference of citizens to lodge their complaint. On the other hand, Public space notification system receives almost 80% of citizen reports in digital form, while the rest use the other means. The highly preference of the digital channel in public space notification lies in the speed of notification and convenience compared to other means. Citizens notify government to require solving a problem with in the shortest period; therefore, the internet may facilitate this as well.

**1.4 CORRELATIONS AND SPATIAL ANALYSIS OF THE E-GRIEVANCE SYSTEM**

A complaint handling system should take the needs of different social groups and, even in an era of rapidly increasing computer literacy, recognizes that there are many people without access to the internet and/or the skills required to use it. In order to analyze this, the system took benefit of different database including the Public space notification database. Each reported incidents’ postcode was linked to the neighborhood postcodes of Amsterdam map by using Arc GIS software. The complaints are counted and double-checked. The total number of complaints was 4429, but only 3415 were included in our analysis. The rest (1014) could not be linked to the neighborhood postcodes. It may be that an officer has mistyped them or the citizen just put a wrong postcode. Nevertheless, they constitute 23% of our population, which is tolerable number in our studies.

**1.5 OBJECTIVE**

* To provide a new forum that facilitates the interaction between public and government bodies.
* To solve the social issues by joining the hands of public and government bodies.
* To make volunteer participation of public to make an effort to solve the social problems.
* To design an application console to allow the public to register themselves with the application and register a complaint.
* To design a console to allow the government bodies to view the list of complaints and their details in their sector and process it.
* To design a methodology to intimate the status of the complaint to the person who registered it.

**1.6 ORGANIZATION OF REPORT**

The overview of the subjects described in the underlying chapters are given below,

Chapter 2 describes the Problem Definition.

Chapter 3 describes the Design of Proposed System

Chapter 4 describes the Implementation of Proposed System

Chapter 5 describes Conclusion and Future Works

**CHAPTER 2**

**PROBLEM DEFINITION**

Handling and responding the complaint is one of the challenge and responsibility by the Municipal Corporation (MC) or local governing body with in the town limit. To address these issues there has to be a platform, which makes people come forward to solve those problems by joining their hands with the government bodies, for which there proposed a platform with various data: such as ward number, problem statement, location, picture of current state & city are collected from the user. A new Android-based platform to collect sensing data from smart phones, automatically or with user help, and to deliver sensing/actuation tasks to government bodies. Thus, this platform will serve as a medium of communication for the public and the government bodies. All the complaints would be registered and the response would be generated by the government bodies and the status of the complaint would be intimated to the public regarding the complaint.

Customers have the right to complaint or give comments on the services and facilities provided. Unfortunately, in most of cases customers face difficulties to channel their complaints and comments to the right party. The customers also have difficulties in knowing the status of their complaints. There has been extensive research in the area of eservice for municipal use. It is important to know the use and efficiency of services that are provided by a MC using latest and better technologies. There have also been research [1] which address the uses of e-services for physically challenged citizens. E-services were used in Europe for a while, but they have been extensively attracting far use up in India, in a big way in large recently. There are separate departments within the MC to handle different problems of the city. It is important for the MC to know about the problems as and when they occur or come into existence in the city and respond to them, so that the problem areas can be dealt with quickly and efficiently. For easier handling of City upkeep, the city is divided into wards. The Bangalore city has about 198 wards. Any complaint relating to or originating from a ward is only handled by an officer associated with that ward. Addressing a complaint gets significantly delayed if the complaint corresponding to one ward is routed to a different ward. Citizens must participate actively which is very important for efficient functioning of all the utilities and services in the area under MC. A correctly routed complaint can be made to handle promptly by the concerned departmental authority in that ward.

An appropriate departmental authority is notified about the complaint for action once a complaint is registered by the citizen. The person who complains is notified the status of the complaint.

**2.1 LIMITATIONS IN EXISTING SYSTEM**

* The people need to visit the organization and it consumes more time.
* The complete current system is a manual system and therefore it is very slow in process.
* Public details are not secured here so they might face any political problem.
* Clemency for the challenged people are not considered here, they can’t make a visit to the appropriate offices.

**CHAPTER 3**

**DESIGN OF PROPOSED SYSTEM**

The system makes use of the already available web portal infrastructure as complaint filing platform and assist user to file their complaint using their smart phone. The complaint is then forward to appropriate department or organizational officer. Then the response from the department is then fetched and sent back to the user. The status about the complaint is periodically checked and updated to the user. Our Grievance Redressal Policy focuses to improve customer satisfaction by collecting feedback from customers across all business units and action plans are put in place to address key issues which are assign ed to the relevant senior leaders to action. Public space is the concern of districts and its be auty.

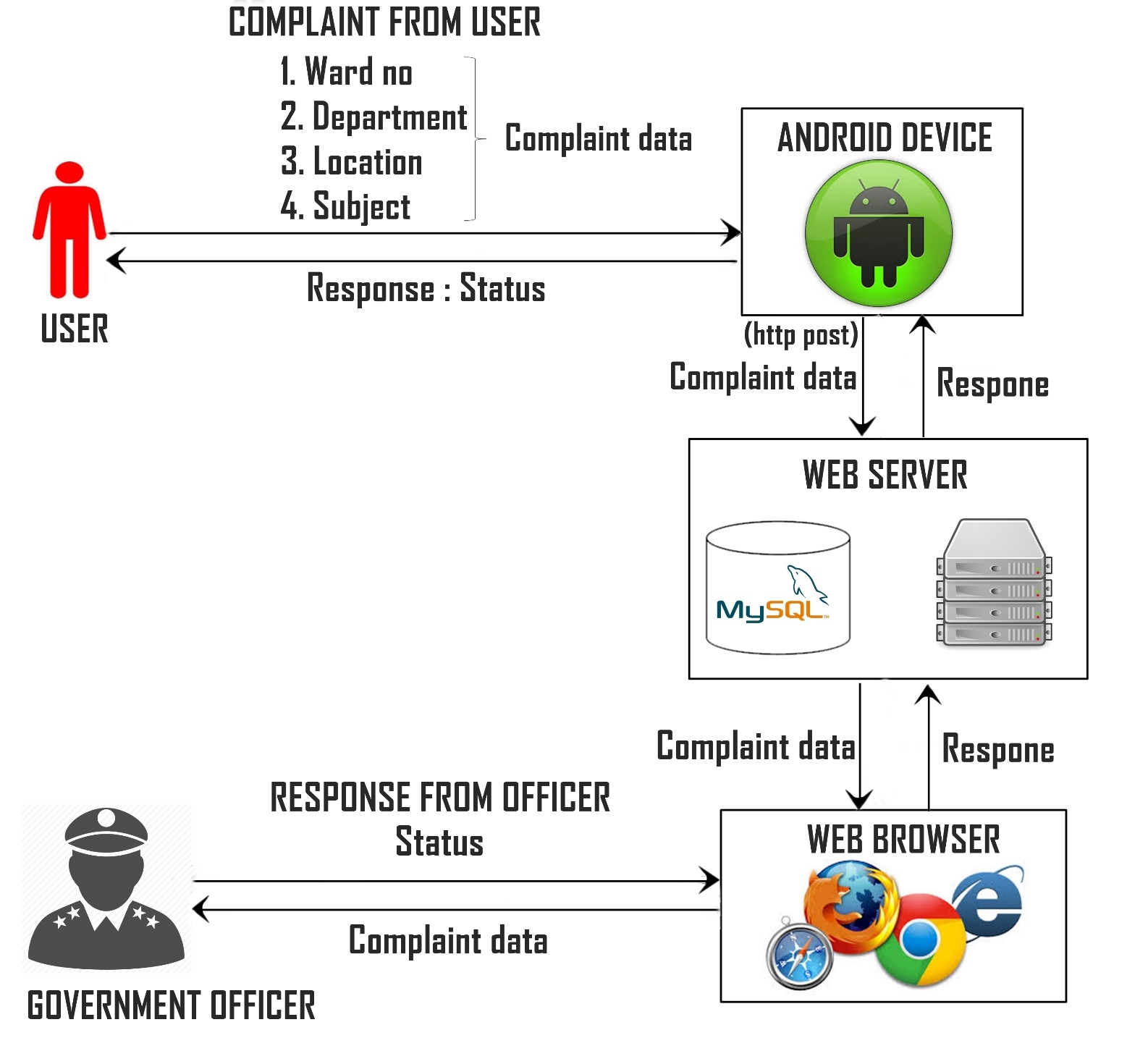
Citizens are encouraged to participate and notify the malfunctioning of public space to the district administration to fix it. In that respect, city administrators are confronted with an immediate problem which needs quick response. The public space notifications were open ended in the past, but recently the digital form is composed consisting of different categories and sub categories

Therefore E- grievance systems always take care of correcting what has gone wrong in dealings of government with its citizens perhaps through delays in handling a matter, neglect or other failures to conduct business properly.

**3.1 ARCHITECTURE OF PROPOSED SYSTEM**

The system make use of the already available web portal interface to complaint filing platform and assist user to file their complaint using their smart phone. The complaint is then forward to appropriate department or organization. Then the response from the department is fetched and sent back to the user.

The status about the complaint is periodically checked and updated to the user. The system makes use of the already available web portal interface to complaint filing platform and assist citizens to file their complaint using their mobile phone. The system enables the user to send his complaint in natural English. Response of the MC complaint handling portal is fetched and is parsed to determine the complaint number. This complaint number is then received by the user on his mobile phone. In the served complaint section the admin can view the processed query list so that he can know about the completed processes among the complaints which are registered in the portal.

****

**Figure 3.1** Architecture of proposed system

**3.2 SEQUENCE OF PROPOSED SYSTEM**

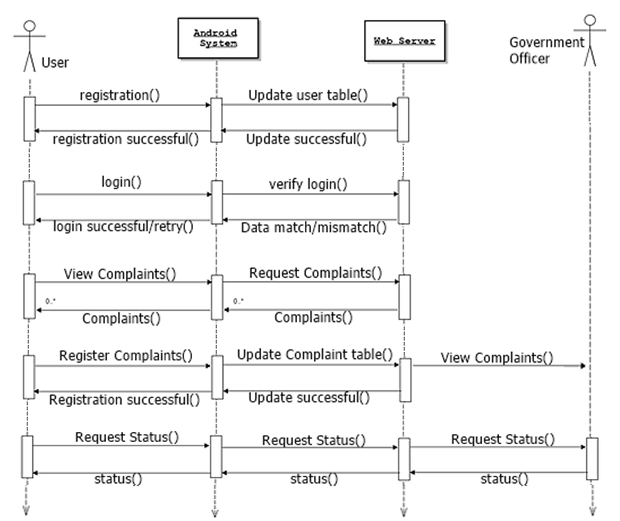
The sequence diagram of proposed system describes the sequence flow of data and action in a system. The initial step is the user registration it’s discussed follow

In registration process, user enters the required fields and send the data to webserver via android system. The required fields could be name, username, password, DOB etc., all these data are updated in the web server and acknowledgement is return to the user through android system.

In login process, user is asked to enter the username and password then the information is gathered by android system and then it is verified in the webserver. If data match, user login successful message is displayed to the user and the user is allowed to register a new complaint or he can view the already registered complaint by him.

View complaints, the user request the android system to view the registered complaints, the system then request the webserver for complaints information and webserver sends the same. Therefore the user can view all the complaints available in the system.

Register complaint, the user is asked to enter the necessary information regarding complaint and then it is send to the webserver and added in the table.

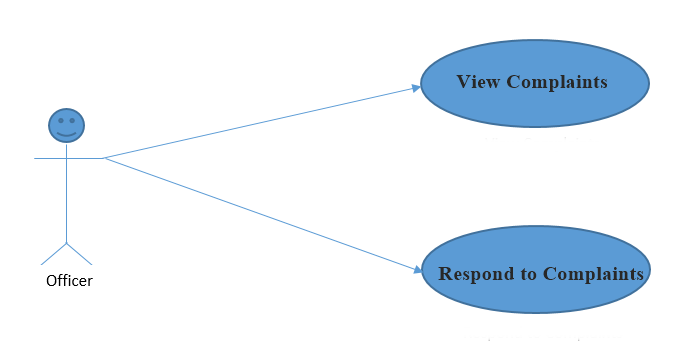
****

**Figure 3.2** Sequence diagram of proposed system

At first, user is registered on the portal then he/she could login with the verified username and password. Now, user is allowed to file new complaint and could view existing complaints which he registered so far. The complaints are stored in complaint data inventory. On the other hand, the Government officer has a login id and he could view the complaints which would be derived from the complaint repository and he could response for the complaints

The response to the complaint is done by the government officer and it is fetched by the android system and is made display to the user so that everyone can know what is the process going on and what is the current status or the reason for the delay.

**3.3 WORKING PROCESS OF GO SYSTEM**



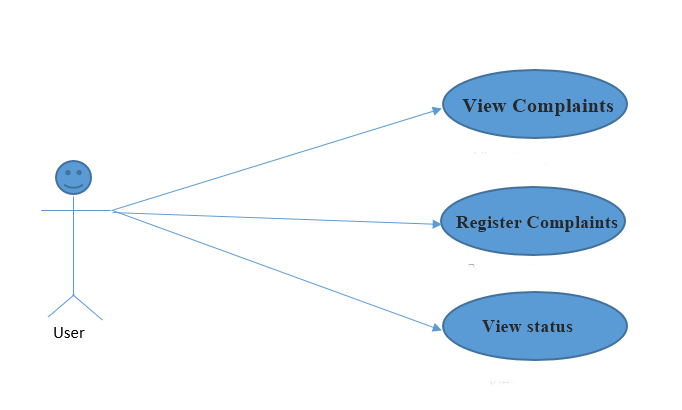
**Figure 3.4** Use case diagram of GO system

The figure 3.3describes the working process of Government officer, here the officer could able to view the complaints on his/her profile’s wall. After go through the complaint he should react to the complaints and make them serviced. After the problem is solved the officer need to forward the served complaint to the admin. The complaint is routed using the online channel to the system and complaint processing system then interprets the users complaint using natural language processing(NLP) techniques to determine the nature of the complaint(the Department and the Complaint Type) and other details (location, land mark, ward number) which are mandatorily required to lodge a complaint. The location would be pulled by the GPS system of the mobile phone, which calculates the latitude and longitude to know the location. Once this information has been inferred by the system, it sends the information to the MC complaint handling system in a compatible format. . There are separate departments within the MC to handle different problems of the city. It is important for the MC to know about the problems as and when they occur or come into existence in the city and respond to them, so that the problem areas can be dealt with quickly and efficiently.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.no** | **Uid** | **Uname** | **Password** | **Name** | **Department** |
| 1 | Off10 | Siva | Siv12 | Sivaraj | WaterLine |
| 2 | Off11 | Raj | Raj12 | Rajesh | RoadLine |
| 3 | Off12 | Rohit | Ro69 | Rohit | Electrical Line |
| 4 | Off13 | Praveen | Pra18 | Praveen | Sanitary |
| 5 | Off14 | Kumar | Ku56 | Kumar | RoadLine |

**Table 3.1** Officer Table

**3.4 WORKING PROCESS OF USER SYSTEM**



**Figure 3.5** Use case diagram of user system

The figure 3.4describes the working process of user system, the user has to register himself in the portal. Through the registration email id he can login into the portal, after successful authentication the user can register his/her complaints by providing required information in the form and he/she can trace the status of the complaint. The user is also facilitated to view the served complaints.

**3.5 MODULES**

* User registration
* Complaint registration
* Complaint forwarding
* Fetching Response

**3.5.1 USER REGISTRATION**

The registration process is the first time participation of the user where the user enters their details and a password of the user’s choice for further authentication. During the registration user is asked to enter an ID proof number such as license, voter ID in order to know that he is a genuine Indian citizen. The system keeps the user information highly confidential in order to avoid harm to the user. User provides the username and password for authentication in the login form.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.no** | **Username** | **Password** | **Reg. No** | **Address** | **City** |
| 1 | Siva | Siv12 | 12312 | xxxxxxx | Coimbatore |
| 2 | Raj | Raj12 | 75875 | xxxxxxx | Coimbatore |
| 3 | rohit | Roh12 | 86786 | xxxxxxx | Coimbatore |
| 4 | shyam | Shy12 | 45654 | xxxxxxx | Coimbatore |
| 5 | praveen | Prav12 | 45688 | xxxxxxx | Coimbatore |

**Table 3.2** User Table

**3.5.2 COMPLAINT REGISTRATION**

Whenever the public face any problem they can register their problem here with the required information as follows. Upload the picture of the problem situation such as opened drainage, damaged roads etc., Subject to the problem Required to choose the category. A brief description about the problem. Exact location where the problem occurs.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **s.no** | **Complaint id** | **Ward no** | **Category** | **Location** | **Subject** | **Replay** | **Officer id** |
| 1 | CMPL101 | 121 | WaterLine | Coimbatore | Xxxxxxxx | aaaaaa | Off10 |
| 2 | CMPL102 | 122 | RoadLine | Coimbatore | Xxxxxxxx | aaaaaa | Off11 |
| 3 | CMPL103 | 123 | Electrical Line | Coimbatore | Xxxxxxxx | aaaaaa | Off12 |
| 4 | CMPL104 | 124 | Sanitary | Coimbatore | Xxxxxxxx | aaaaaa | Off13 |
| 5 | CMPL105 | 125 | RoadLine | Coimbatore | xxxxxxxx | aaaaaa | Off14 |

**Table 3.3** Complaint Table

**3.5.3 COMPLAINT FORWARDING**

The registered complaints are updated in the server database. The spams are then removed in the spam control level 1. The complaints are then separated according to their categories in the server side and make appearthe admin panel. After analyzing the complaints the admin forwards the complaints to the appropriate GO’s profile.

* + 1. **FETCHING RESPONSE**

After forwarding the complaints, the officer can view the complaints which are received in their profile. The officer can respond to the complaints.

People are also facilitated to comment under the complaints and every comment is screened and then displayed in the portal. This system facilitates the easy interaction between public and government bodies.

**3.6 ADVANTAGES OF PROPOSED SYSTEM**

* There is no need to visit the organization regarding complaint registration.
* Proposed system is fully automated.
* It doesn’t consume more time.
* The user profile is secured here.

**CHAPTER 4**

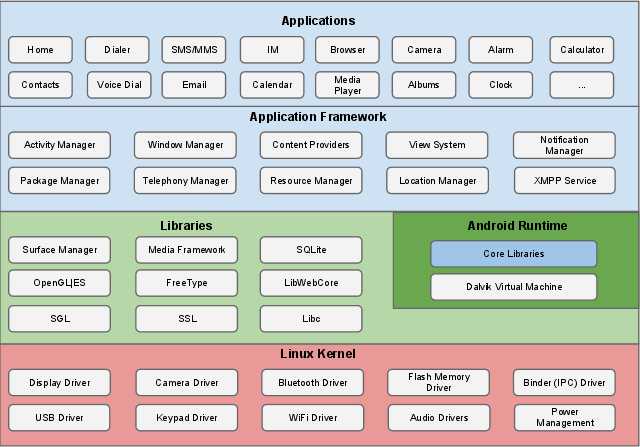
**IMPLEMENTATION OF PROPOSED SYSTEM**

**BREIF DESCRIPTION OF ANDROID**

The Android platform is a software stack for mobile devices including an operating system, middleware and key applications. Developers can create applications for the platform using the Android SDK. Applications are written using the Java programming language and run on Dalvik, a custom virtual machine designed for embedded use, which runs on top of a Linux kernel.

**4.1 ANDROID ARCHITECTURE**

The following diagram shows the major components of the Android operating system. Each section is described in more detail below.



**Figure 4.1**: Android system architecture

**4.1.1 APPLICATIONS**

Android will ship with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts, and others. All applications are written using the Java programming language.

**4.1.2 APPLICATION FRAMEWORK**

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.

Underlying all applications is a set of services and systems, including:

* A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even an embeddable web browser
* Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data
* A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files
* A Notification Manager that enables all applications to display custom alerts in the status bar
* An Activity Manager that manages the life cycle of applications and provides a common navigation back stack

**4.2 LIBRARIES**

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

* System C library - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices
* Media Libraries - based on PacketVideo's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG
* Surface Manager - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications
* LibWebCore - a modern web browser engine which powers both the Android browser and an embeddable web view
* SGL - the underlying 2D graphics engine
* 3D libraries - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer
* FreeType - bitmap and vector font rendering
* SQLite - a powerful and lightweight relational database engine available to all applications

**4.3 ANDROID RUNTIME**

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dex" tool.The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

**4.4 LINUX KERNEL**

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

* + 1. **JAVA JDK.1.7**

Computers connected to the net are from many different manufacturers, running on different operating systems and they differ in architecture, computing power and capacity. By considering this point SUN Microsystems Corporation felt the need for a new programming language suitable for this heterogeneous Environment and java was the solution. This breaks barriers between different computers, chips and operating systems.

The main properties of the Java, which made Java so popular, are as follows:

* Simple
* Secure
* Portable
* Object-Oriented
* Robust
* Multithreaded
* Interpreted
* High performance

The key that allows Java to solve both the security and the portability problems just described is that the output of a Java compiler is not executable code. Rather, it is Byte code. Byte code is a highly optimized set of instructions designed to be executed by the Java runtime systems, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for Byte code. This may come has a bit of surprise.

* + 1. **ANDROID SDK**

The Android SDK includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator (based on QEMU), documentation, sample code, and tutorials. Currently supported development platforms include x86-based computers running Linux (any modern desktop Linux Distribution), Mac OS X 10.4.6 or later, Windows XP or Vista. Requirements also include Java Development Kit, Apache Ant, and Python 2.2 or later. The officially supported integrated development environment (IDE) is Eclipse (3.2 or later) using the Android Development Tools (ADT) Plug-in, though developers may use any text editor to edit Java and XML files then use command line tools to create, build and debug Android applications.

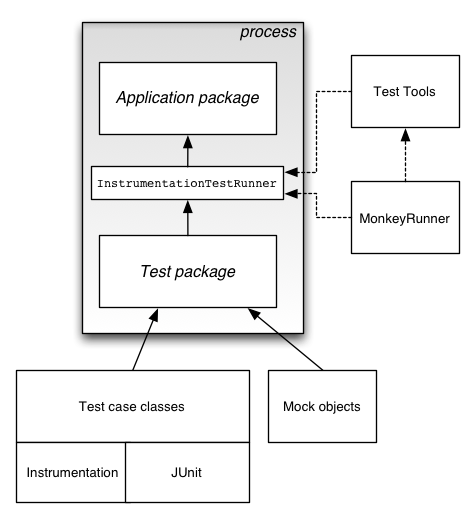
**4.5 ECLIPSE 3.7.1**

An open-source Java IDE and platform for rich client applications, an open source platform-independent software framework for delivering what the project calls or is known as "rich-client applications" (as opposed to "thin clients", this means the clients perform heavy-duty work on the host. So far this framework has typically been used to develop IDEs (Integrated Development Environments), such as the highly-regarded Java IDE called Java Development Toolkit (JDT) and compiler that come as part of Eclipse (and which are also used to develop Eclipse itself). However, it can be used for other types of client application as well, see the popular BitTorrent client for example.

**4.6 TESTING OF PROJECT**

**4.6.1 SYSTEM TESTING**

The Android testing framework, an integral part of the development environment, provides an architecture and powerful tools that help you test every aspect of your application at every level from unit to framework. The testing framework has these key features: Android test suites are based on JUnit. You can use plain JUnit to test a class that doesn't call the Android API, or Android's JUnit extensions to test Android components. If you're new to Android testing, you can start with general-purpose test case classes such as AndroidTestCase and then go on to use more sophisticated classes. The Android JUnit extensions provide component-specific test case classes. These classes provide helper methods for creating mock objects and methods that help you control the lifecycle of a component. Test suites are contained in test packages that are similar to main application packages, so you don't need to learn a new set of tools or techniques for designing and building tests. The SDK tools for building and tests are available in Eclipse with ADT, and also in command-line form for use with other IDEs. These tools get information from the project of the application under test and use this information to automatically create the build files, manifest file, and directory structure for the test package. The SDK also provides monkey runner, an API for testing devices with Python programs, and UI/Application Exerciser Monkey, a command-line tool for stress-testing UIs by sending pseudo-random events to a device. This document describes the fundamentals of the Android testing framework, including the structure of tests, the APIs that you use to develop tests, and the tools that you use to run tests and view results. The document assumes you have a basic knowledge of Android application programming and JUnit testing methodology.



**Figure 4.2:** Junit testing methodology.

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**4.6.2 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform

basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**4.6.3 FUNCTIONAL TEST**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**4.6.4 SYSTEM TEST**

System testing ensures that the entire integrated software system meets requirement. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**4.6.5 WHITE BOX TESTING**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**4.6.6 BLACK BOX TESTING**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

# 4.6.7 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**4.6.8 ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**4.7 REQUIREMENT ANALYSIS**

**4.7.1 HARDWARE REQUIREMENTS**

* Device: Android Smartphone with internet connection
* Memory required: 10MB of free space

**4.7.2 SOFTWARE REQUIREMENTS**

* Operating system: Android
* Version: 2.1 and above
* Development Platform: Eclipse 3.7.1
* Programming Language: Java
* Database: MYSQL.

**4.8 TEST CASES**

**4.8.1 TEST CASE 1: LOGIN VERIFICATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **TEST NAME** | **INPUT** | **EXPECTED OUPUT** | **OBTAINED OUTPUT** |
| 1 | Login | Wrong credential | User not found | User not found |
| 2 | Login | Correct credential | User found | User found |
| 3 | Login | Wrong password | Password mismatch | Password mismatch |

**Figure 4.1:** Login Verification

During login process the user enters his/her credentials such as username and password .The Android system sends the user credentials to the web server and the server synchronize the entered username and password , if the entered data doesn't match it displays an error message that user not found and if the data matches then user gets login to the portal and he could file new complaint or view existing complaints.

**4.8.2 TEST CASE 2**

**VALIDATION OF COMPLAINT REGISTRATION FORM.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **TEST NAME** | **INPUT** | **EXPECTED OUPUT** | **OBTAINED OUTPUT** |
| 1 | Complaint registration | Location field missed out | Fill all fields | Fill all fields |
| 2 | Complaint registration | Subject field missed out | Fill all fields | Fill all fields |
| 3 | Complaint registration | Ward number field missed out | Fill all fields | Fill all fields |

**Table 4.2:** Validation of Complaint Registration Form.

After login process, while filling the complaint field if the user left a field blank he is not allowed to submit the data .The system validates all the field and pops a message to fill all the fields if any field left blank. Once all the fields have been filled the system allows the user to submit the data on the server.

Whenever the public face any problem they can register their problem here with the required information as follows. Upload the picture of the problem situation such as opened drainage, damaged roads etc., Subject to the problem Required to choose the category. A brief description about the problem. Exact location where the problem occurs.

**4.9 HTTP POST**

In computing, POST is one of many [request methods](http://en.wikipedia.org/wiki/HTTP#Request_methods) supported by the [HTTP](http://en.wikipedia.org/wiki/HTTP) protocol used by the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web). The POST request method is designed to request that a [web server](http://en.wikipedia.org/wiki/Web_server) accepts the data enclosed in the request message's body for storage.[[1]](http://en.wikipedia.org/wiki/POST_(HTTP)#cite_note-RFC_def-1) It is often used when uploading a file or submitting a completed [web form](http://en.wikipedia.org/wiki/Form_(HTML)).

In contrast, the HTTP [GET](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Request_methods) request method is designed to retrieve information from the server. As part of a GET request, some data can be passed within the URI's [query string](http://en.wikipedia.org/wiki/Query_string), specifying

for example search terms, date ranges, or other information that defines the query. As part of a POST request, an arbitrary amount of data of any type can be sent to the server in a request message body. A [header field](http://en.wikipedia.org/wiki/List_of_HTTP_header_fields) in the POST request usually indicates the message body's [Internet media type](http://en.wikipedia.org/wiki/Internet_media_type).

The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. POST is designed to allow a uniform method to cover the following functions:

* Annotation of existing resources
* Posting a message to a bulletin board, newsgroup, mailing list or similar group of articles
* providing a block of data, such as the result of submitting a form, to a data-handling process
* Extending a database through an append operation

**4.9.1 PUBLIC CONSTRUCTORS**

* public HttpPost ()
* public HttpPost (URI uri)
* public HttpPost (String uri)

In this system the http client and http post method is used to send data from the android device to the web server the methods are as follows:

httpclient=newDefaultHttpClient();

httppost= newHttpPost("http://10.0.2.2/alumini/doc.php");

**DefaultHttpClient()-**Android includes two HTTP clients: HttpURLConnection and Apache HTTP Client. Both support HTTPS, streaming uploads and downloads, configurable timeouts, IPv6 and connection pooling. Apache HTTP client has fewer bugs in Android 2.2 (Froyo) and earlier releases. For Android 2.3 (Gingerbread) and later,[HttpURLConnection](http://developer.android.com/reference/java/net/HttpURLConnection.html) is the best choice. Its simple API and small size makes it great fit for Android. Transparent compression and response caching reduce network use, improve speed and save battery. See the [Android Developers Blog](http://android-developers.blogspot.com/2011/09/androids-http-clients.html) for a comparison of the two HTTP clients. **HttpPost("http://10.0.2.2/alumini/doc.php")** 10.0.2.2 is the IP address of the localhost. In the localhost the there is a folder named alumini where doc.php file is located. It contains the code to send and retrieve the data from the mysql server.

**CHAPTER 5**

**CONCLUSION AND FUTURE WORK**

**5.1 CONCLUSION**

E-Grievance redressal service system is proposed and built for users to lodge complaints about their city. This system analysis and maintains the complaint with in the corporation area. The system is accessible to everyone who has a smart phone, he/she can download this application and register their complaint all around the clock. The fully automated complaint registration service makes people to come forward to solve social problems The essential idea is to make use of the existing web portal infrastructure and provide an easy, cheap and quick mode of complaint registration and getting responses. E-services depend on three principal factors: Utility factors in such that the proposed E-service s must meet the expectations of the visually disabled persons and bring them a real added value through their use (by augmenting their ability to act, interact, and be informed). Ergonomic factors (usability and accessibility) in which the specificities of visually disabled persons as well as their level of expertise (with the internet and screen readers) are taken into consideration from the conception. Psychosocial factors of acceptability where the proposed E- services offer the possibility to truly compensate, assist, and valorise visually disabled persons.

**5.2 FUTURE WORK**

The future work of the system deals with the location would be pulled by the GPS system of the mobile phone, which calculates the latitude and longitude to know the location. Once this information has been inferred by the system, it sends the information to the MC complaint handling system in a compatible format and fetch the response from MC and send it back to the user. The current system is designed for particular city alone, depends on its need the system can be enhanced to whole country. Departments such as road line, waterline, electrical line, sanitary and health complaints will be processed and resolved, in the future more departments will added for better service. Easy integration will be added in order to bring the public and government bodies together to solve the social problems and make our city clean and safe.

**APPENDIX A1**

**SOURCE CODE**

**HOMEPAGE**

package com.example.alumini; import java.util.ArrayList; import java.util.List;

import org.apache.http.NameValuePair; import org.apache.http.client.utils.URLEncodedUtils;

import org.apache.http.impl.client.DefaultHttpClient; import org.apache.http.message.BasicNameValuePair;

import android.app.Activity; import android.content.Intent;

import android.database.sqlite.SQLiteDatabase; import android.os.Bundle;

import android.view.View; import android.view.View.OnClickListener;

import android.webkit.WebSettings; import android.webkit.WebView;

import android.webkit.WebViewClient; import android.widget.Button;

import android.widget.EditText; import android.widget.ImageView;

import android.widget.Spinner; import android.widget.Toast;

public class Homepage extends Activity{

ImageView imgg;

@Override

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.homepage);

imgg=(ImageView)findViewById(R.id.imageView1);

imgg.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) { Intent ittt = new Intent(Homepage.this,MainActivity.class);

startActivity(ittt); } } ); } }

**COMPLAINT REGISTRATION**

package com.example.alumini;

import java.io.BufferedInputStream; import java.io.ByteArrayOutputStream;

import java.io.File; import java.io.FileInputStream;

import java.io.FileOutputStream; import java.nio.channels.FileChannel;

import java.util.ArrayList; import java.util.List; import java.util.Random;

import org.apache.http.NameValuePair;

import org.apache.http.client.utils.URLEncodedUtils;

import org.apache.http.impl.client.DefaultHttpClient;

import org.apache.http.message.BasicNameValuePair;

import android.app.Activity; import android.content.Intent;

public class NewComplaint extends Activity{

private static final int CAMERA\_REQUEST = 1888;

private static int RESULT\_LOAD\_IMAGE = 1;

private ImageView imageView,imageView1;

ImageView immg,immg1;

Bitmap bMap;

String picturePath,uname,pwd;

String mSelectedImagePath;

TextView txt;

EditText ed1,ed2,ed3,ed4;

Spinner sp1;

WebView wbb;

private static String url\_create\_product = "http://10.0.2.2/alumini/complaint.php";

//private static String url\_create\_produc = "http://10.0.2.2/complaint/complaint.php";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.newcomplaint);

Button btncam=(Button)findViewById(R.id.button2);

Button btngall=(Button)findViewById(R.id.button3);

Button segmant=(Button)findViewById(R.id.btnLogin);

Button segmant1=(Button)findViewById(R.id.btnLogin1);

txt=new TextView(this);

this.imageView = (ImageView)this.findViewById(R.id.imageView1);

immg= (ImageView)this.findViewById(R.id.imageView1);

wbb = (WebView)this.findViewById(R.id.webView1);

//immg1= (ImageView)this.findViewById(R.id.imageView2);

ed1=(EditText)findViewById(R.id.editText1);

ed2=(EditText)findViewById(R.id.editText2);

ed3=(EditText)findViewById(R.id.editText3);

ed4=(EditText)findViewById(R.id.editText4);

sp1=(Spinner)findViewById(R.id.spinner1);

Bundle b = getIntent().getExtras();

if(b !=null)

{

uname=getIntent().getExtras().getString("name");

pwd=getIntent().getExtras().getString("pas"); }

btncam.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

Intent ittt = new Intent(NewComplaint.this,MainActivity.class);

startActivity(ittt);

}

});

btngall.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

Intent i = new Intent(

Intent.ACTION\_PICK,

android.provider.MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI);

startActivityForResult(i, RESULT\_LOAD\_IMAGE); } } };

segmant1.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

Intent ittt = new Intent(NewComplaint.this,CustomListViewAndroidExample.class);

Bundle bi = new Bundle();

//Inserts a String value into the mapping of this Bundle

bi.putString("name",uname);

bi.putString("pas", pwd);

//Add the bundle to the intent.

ittt.putExtras(bi);

startActivity(ittt); } } );

segmant.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

params.add(new BasicNameValuePair("name", ""+did));

params.add(new BasicNameValuePair("age", uu1));

params.add(new BasicNameValuePair("email", sp1.getSelectedItem().toString()));

params.add(new BasicNameValuePair("pass", uu3));

params.add(new BasicNameValuePair("mob", uu4));

params.add(new BasicNameValuePair("hosp", "1"));

params.add(new BasicNameValuePair("uname", uname));

if (requestCode == RESULT\_LOAD\_IMAGE && resultCode == RESULT\_OK && null != data) {

Uri selectedImage = data.getData();

String[] filePathColumn = { MediaStore.Images.Media.DATA };

Cursor cursor = getContentResolver().query(selectedImage,

filePathColumn, null, null, null);

cursor.moveToFirst();

int columnIndex = cursor.getColumnIndex(filePathColumn[0]);

picturePath = cursor.getString(columnIndex);

cursor.close();

imageView1 = (ImageView) findViewById(R.id.imageView1);

bMap = BitmapFactory.decodeFile(picturePath);

imageView1.setImageBitmap(bMap);

txt.setText(picturePath);

String ai=txt.getText().toString();

Toast.makeText(getApplication(),ai, Toast.LENGTH\_LONG).show(); } }

public String getPath(Uri uri) {

String[] projection = { MediaStore.Images.Media.DATA };

Cursor cursor = managedQuery(uri, projection, null, null, null);

startManagingCursor(cursor);

int column\_index = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);

cursor.moveToFirst();

return cursor.getString(column\_index); } }

**COMPLAINT RETRIEVAL**

package com.example.alumini;

import java.io.BufferedReader; import java.io.InputStream;

import java.io.InputStreamReader; import java.util.ArrayList; import java.util.List;

import java.util.Random; import android.app.AlertDialog;

import android.app.Notification; import android.app.NotificationManager;

import android.app.PendingIntent; import android.content.Context;

import android.content.DialogInterface; import org.apache.http.HttpEntity;

import android.widget.LinearLayout; import android.widget.ListView;

import android.widget.TextView; import android.widget.Toast;

public class RegularDetails extends Activity{

EditText username,phonu,locaarea;

TextView e1,e2,e3,e4,e5,e6;

Button btn,btn1;

String dname,area;

String result = null; InputStream is = null;

StringBuilder sb = null; ArrayList<String> al = new ArrayList<String>();

ArrayList<String> al1 = new ArrayList<String>();

ArrayList<String> al2 = new ArrayList<String>();

String targetmonth; String targetyear;

String targetamount,did,number;

String dame,sou,des,cno,ctype,avail,date,time,cost,did1,dsno;

JSONArray jArray; WebView wbb;

ListView list; // private static String url\_create\_product = "http://10.0.2.2/Car/docindex2.php";

CustomAdapter adapter;

public CustomListViewAndroidExample CustomListView = null;

public ArrayList<ListModel> CustomListViewValuesArr = new ArrayList<ListModel>();

private static final int NOTIFY\_ME\_ID=1337;

@Override

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.regdet);

e1=(TextView)findViewById(R.id.textView1);

e2=(TextView)findViewById(R.id.textView2);

e3=(TextView)findViewById(R.id.textView3);

e4=(TextView)findViewById(R.id.textView4);

e5=(TextView)findViewById(R.id.textView5);

e6=(TextView)findViewById(R.id.textView6);

wbb=(WebView)findViewById(R.id.webView1);

btn=(Button)findViewById(R.id.button1);

btn1=(Button)findViewById(R.id.button2);

Bundle b = getIntent().getExtras();

if(b !=null)

{

dname=getIntent().getExtras().getString("ee1");

//area=getIntent().getExtras().getString("ee2");

ArrayList<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();

nameValuePairs.add(new BasicNameValuePair("name", dname));

String U\_selectProcess;

U\_selectProcess="http://10.0.2.2/alumini/reportmissper1.php";

String paramString = URLEncodedUtils.format(nameValuePairs, "utf-8");

U\_selectProcess += "?" + paramString;

HttpClient httpclient = new DefaultHttpClient();

try{

HttpPost httppost = new HttpPost(U\_selectProcess);

StringEntity se = new StringEntity("envelope",HTTP.UTF\_8);

httppost.setEntity(se);

HttpParams httpParameters = new BasicHttpParams();

// Set the timeout in milliseconds until a connection is established.

int timeoutConnection = 3000;

HttpConnectionParams.setConnectionTimeout(httpParameters, timeoutConnection);

// Set the default socket timeout (SO\_TIMEOUT)

// in milliseconds which is the timeout for waiting for data.

int timeoutSocket = 3000;

HttpConnectionParams.setSoTimeout(httpParameters, timeoutSocket);

httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

HttpResponse response = httpclient.execute(httppost);

HttpEntity entity = response.getEntity();

is = entity.getContent();

}

catch(Exception e){

Toast.makeText(getApplicationContext(), e.toString(), Toast.LENGTH\_LONG).show();

}

try

{

BufferedReader reader = new BufferedReader(new InputStreamReader(is, "iso-8859-1"), 80);

sb = new StringBuilder();

sb.append(reader.readLine() + "\n");

String line = "0";

avail = json\_data.getString("subject");

date = json\_data.getString("reply");

time = json\_data.getString("officerid");

//cost = json\_data.getString("cost");

//did1= json\_data.getString("did");

//dsno=json\_data.getString("sno");

e1.setText("Id :"+dname);

e2.setText("Wardno :"+sou);

e3.setText("Category :"+cno);

e4.setText("Location :"+ctype);

e5.setText("Status :"+date);

e6.setText("Subject:"+avail); } }

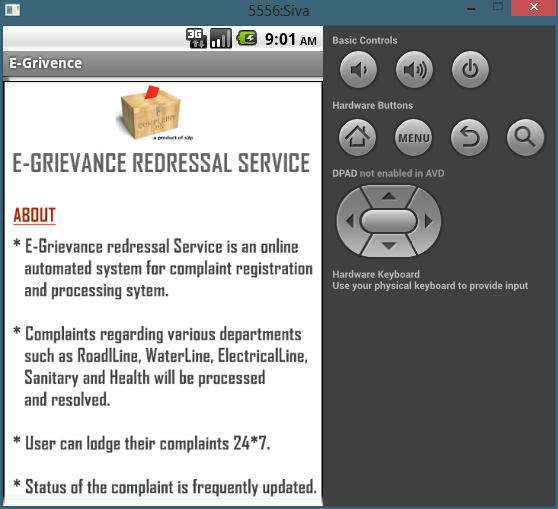
catch(JSONException e){

Toast.makeText(getApplicationContext(), e.toString(), Toast.LENGTH\_LONG).show();

} }}}

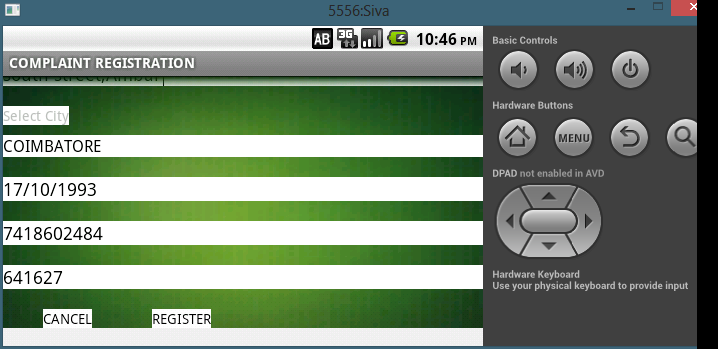
**APPENDIX A2**

**SCREEN SHOTS**



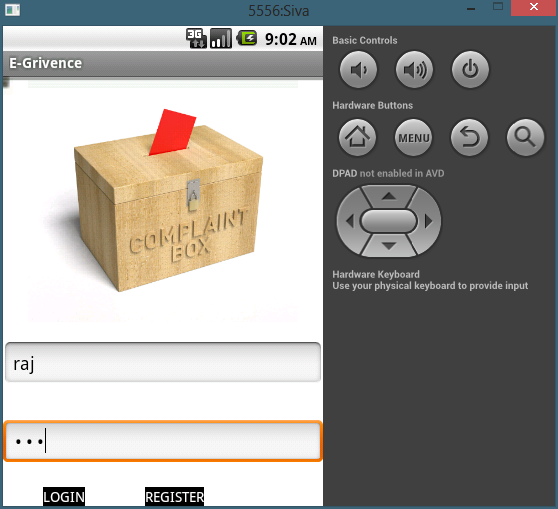
**Figure A2.1:** Start Screen

Start screen of the android device, it describes the about of the application.



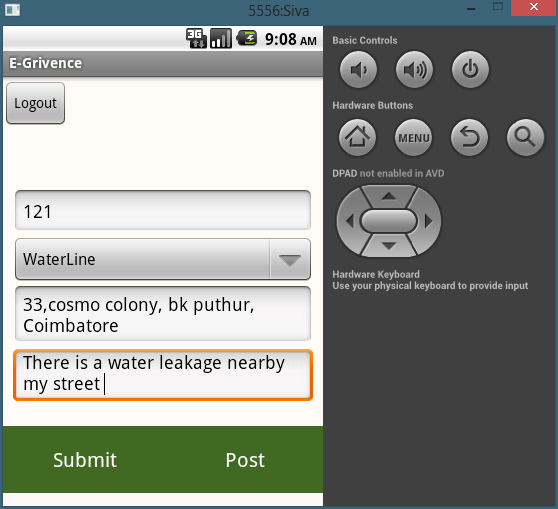
**Figure A2.2:** Registration Form

Registration form of the android device, it’s used to get the user’s information from the user.



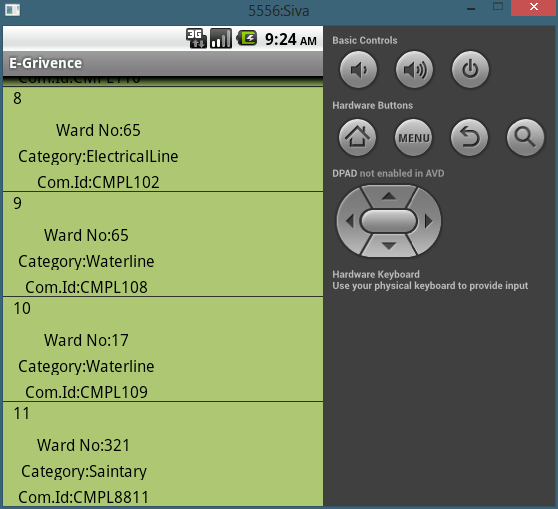
**Figure A2.3:** Login Screen

Login form of the android system, it’s used to get login credentials of the user.



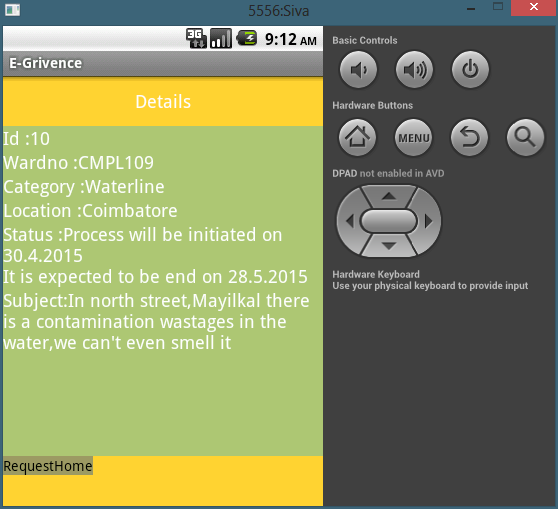
**Figure A2.4:** Complaint Form

Complaint form of the android system, it’s used to get the complaint data from the user.

****

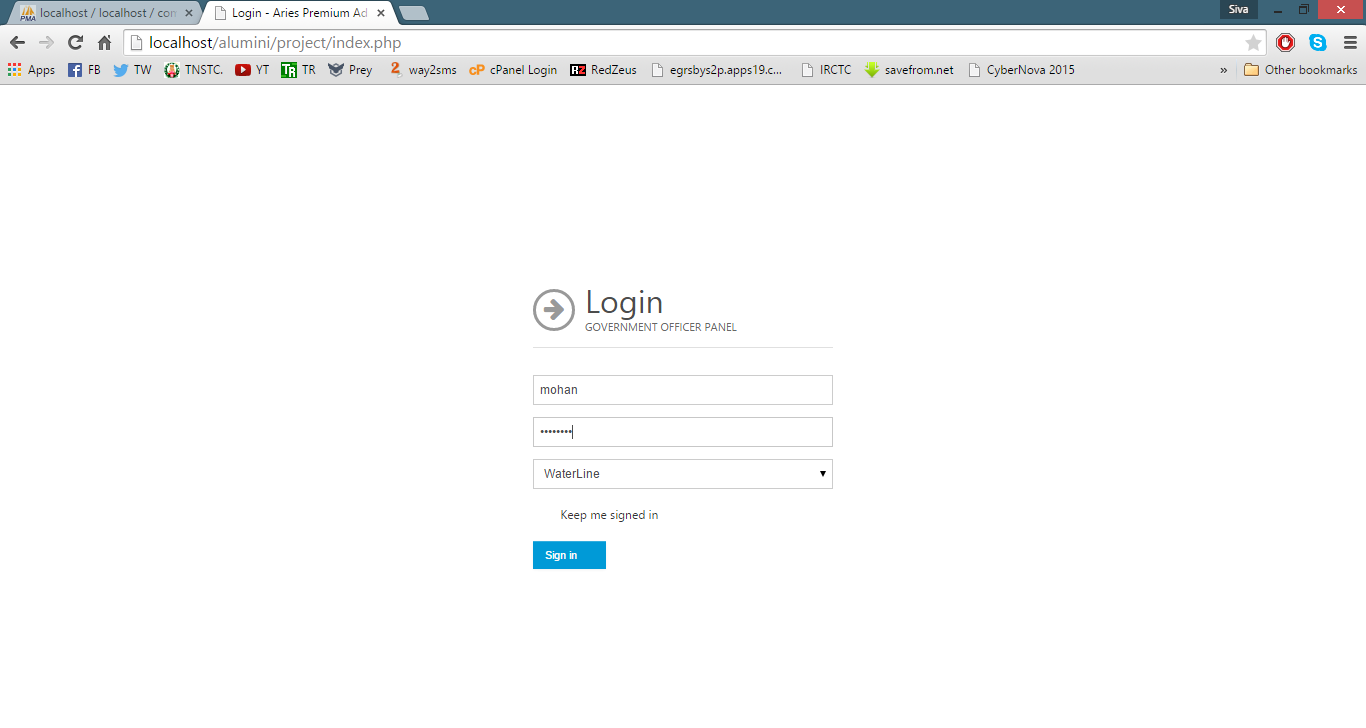
**Figure A2.5:** Registered Complaints

Registered complaints of the user, therefore every user can view their registered complaint.



**Figure A2.6:**  Status View

Status view of the complaints, Every user can view the status of his/her complaint.

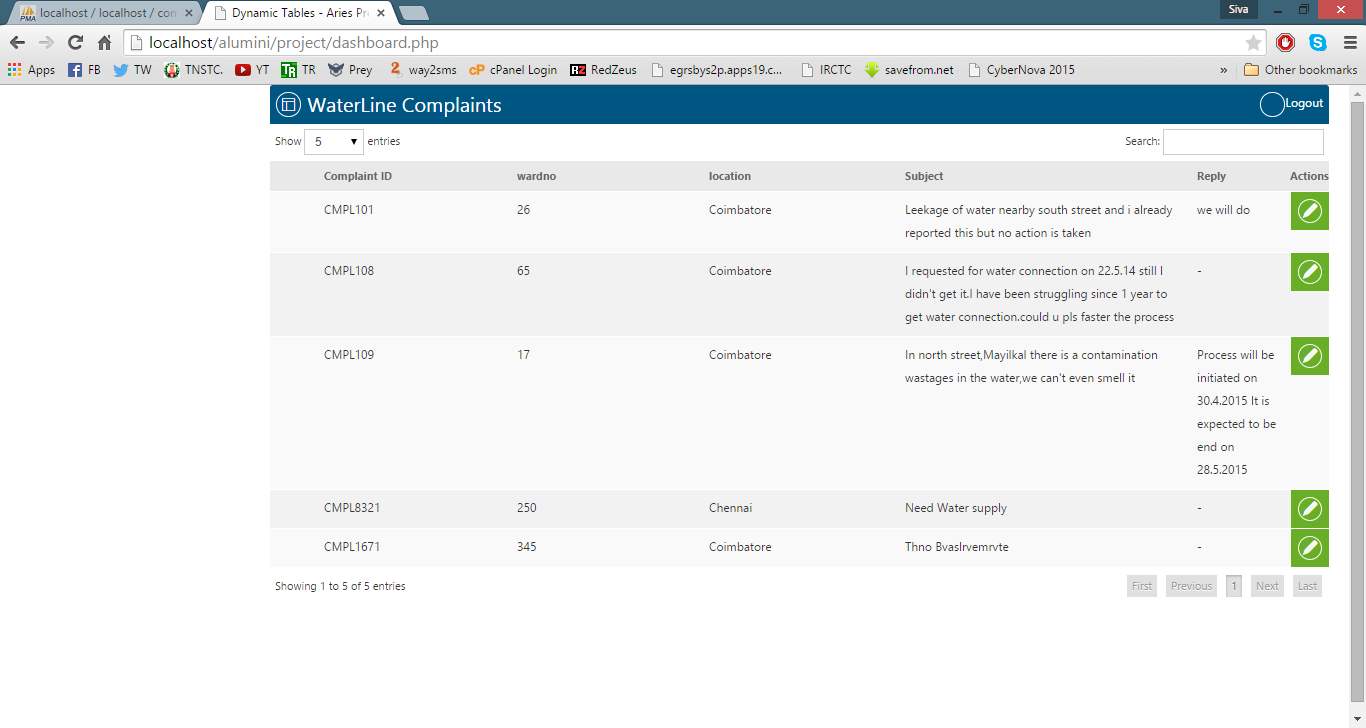


**Figure A2.7:** Officer Login Page

Login page of the Government officer where officer can login with his/her credential.

FGHGFHGFHG

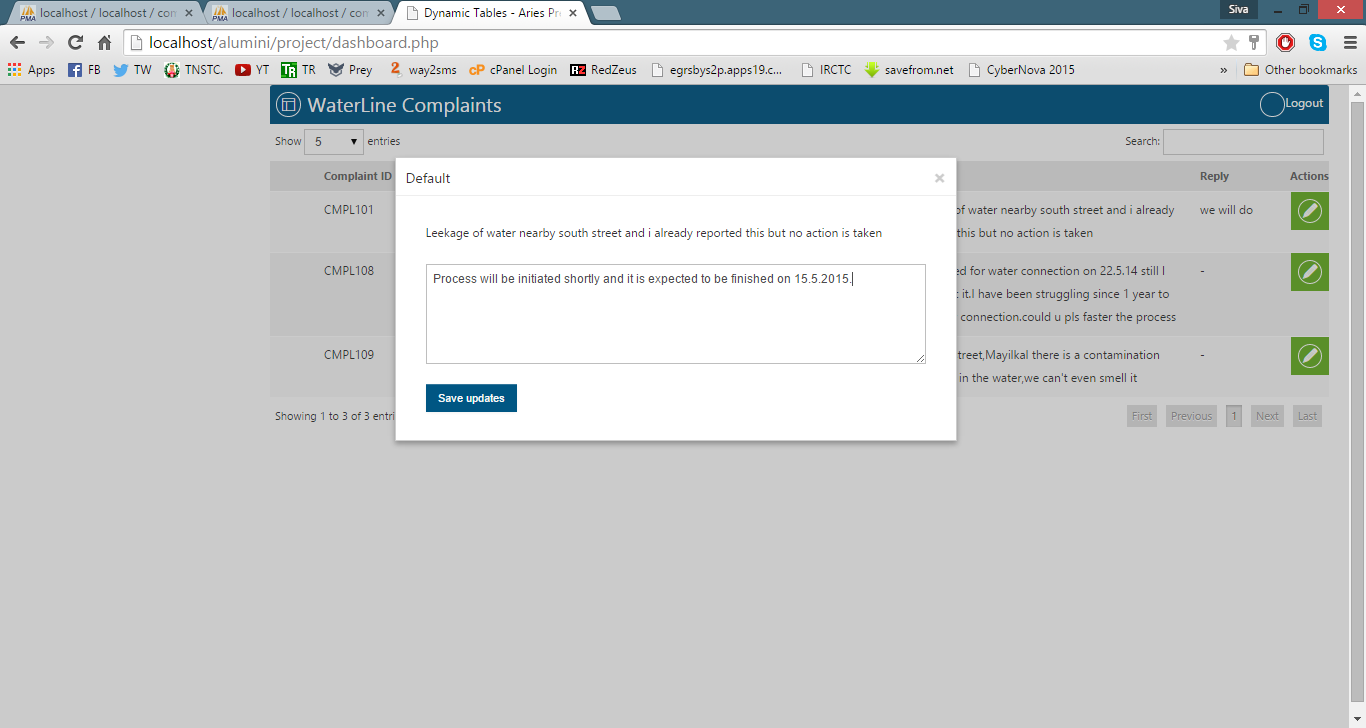
GGDGHGHGHGH



**Figure A2.8:** Complaint View Page

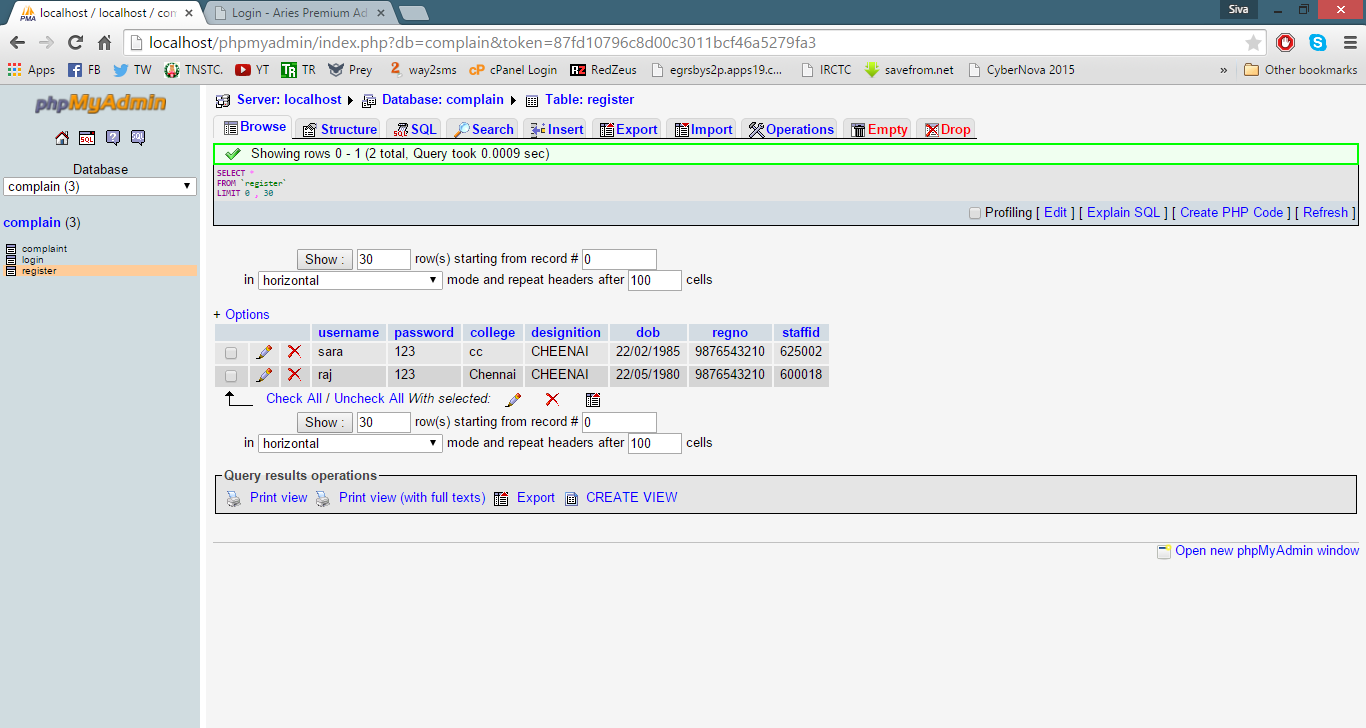
Complaint view page, where Government officer can view the registered complaints by the user.

USER TA



**Figure A2.9: S**tatus update

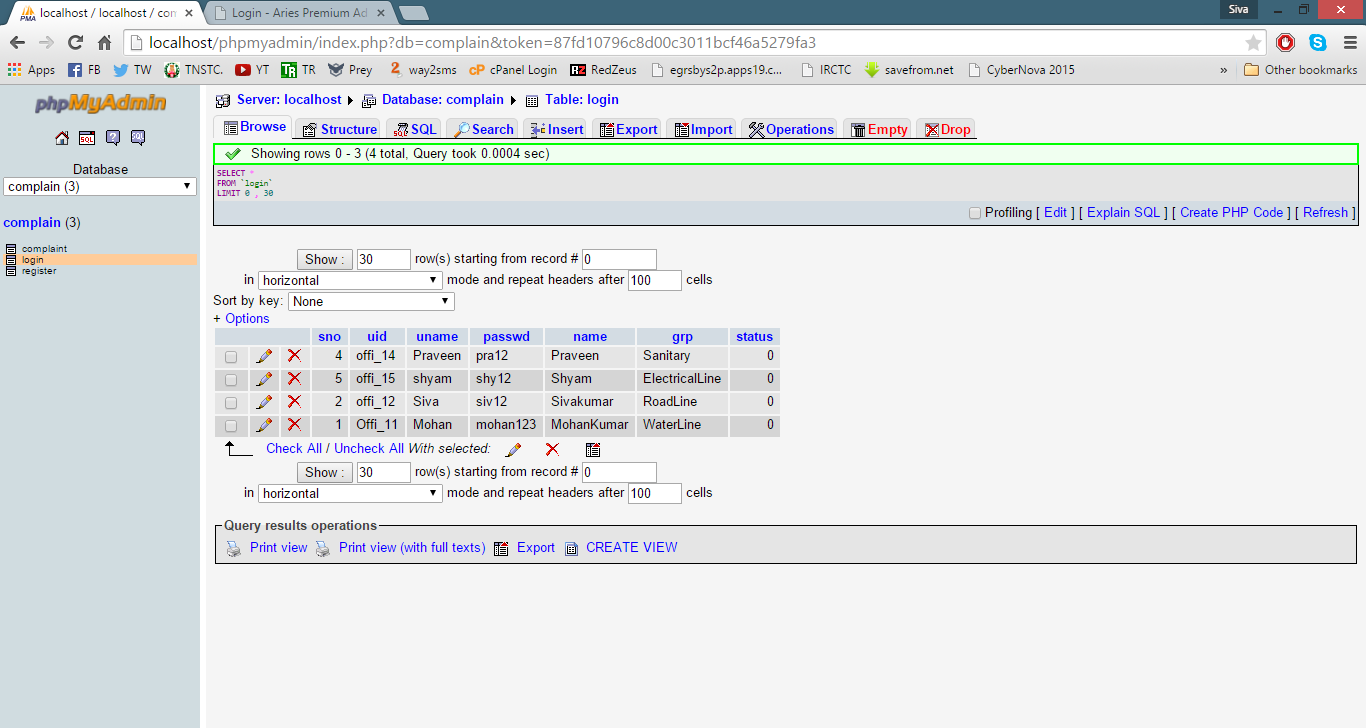
Status of the complaint is updated by the offcier in his/her panel and it is made viewable to the user.



**Figure A2.10:** User Table

User table, this table is maintained in the complaint database, holds user’s information

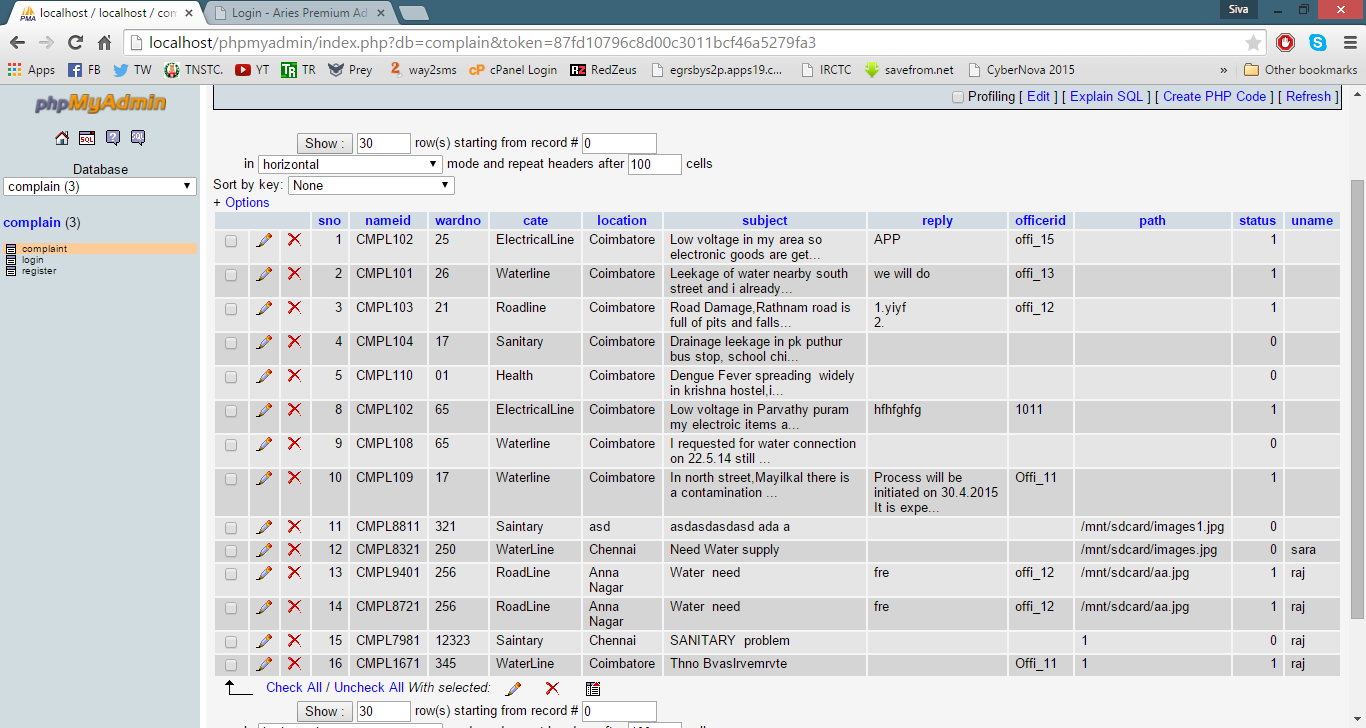
wfeweeff



e

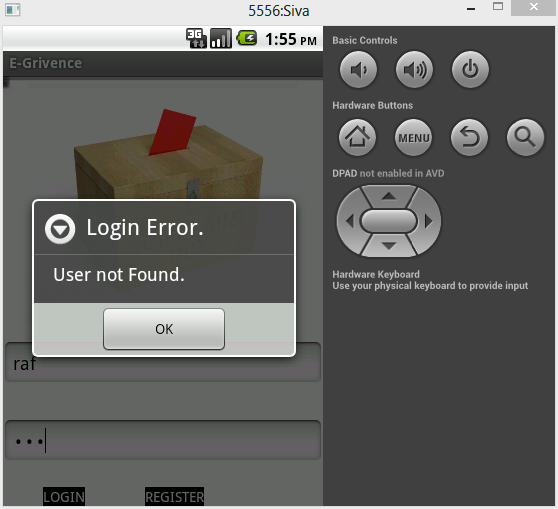
**Figure A2.11:** Officer Table

Officer table holds the Government Officer’s informations such as name,department etc.,



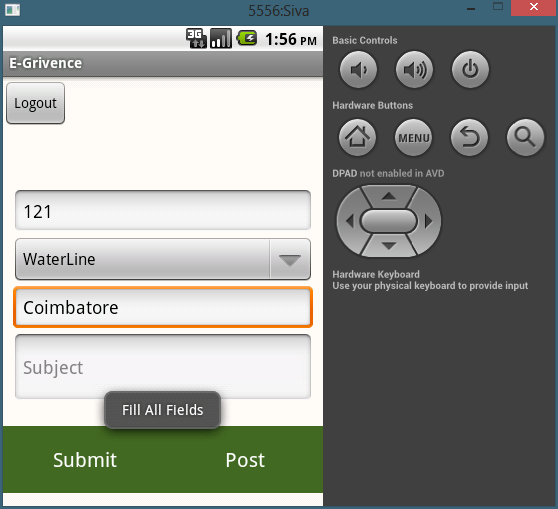
**Figure A2.12:** Complaint Table

Complaint table holds the complaint data which is accessed by both user and officer.



**Figure A2.13:** Login Verification

Login verification is performed by the system so that unauthorized person can’t access the user’s personal data.



**Figure A2.14:** Validation of Complaint Registration Form

Complaint registration form validation is performed in order to avoid insufficient data

**REFERENCES**

[1] Chaumon, B. and Guermond. S. Study of conditions of use of e-services accessible to visually disabled persons. In Proceedings DEGAS 2007.

[2] Mun, M., Sokij, T “PEIR, the Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research,” Proc. 7th Int’l. Conf. Mobile Systems, Applications, and Services, June 2009.

[3] B.P. Vuurens , "Obtaining High-Quality Relevance Judgments Using Crowdsourcing,"Sept.-Oct. 2012.

[4] M.-R. Ra, “Medusa: A Programming Framework for Crowd-Sensing Applications,”2012

[5] T. Yan*.*, “mCrowd: A Platform for Mobile Crowdsourcing,” 2009

[6] Allahbakhsh, M. Sulain, J. proposed a framework for characterizing various dimensions of quality control in crowdsourcing systems March-April 2013.

[7] R. K. Ganti et al., “Mobile Crowd sensing: Current State and Future Challenges,” 2011