UCOMPLAIN

Mini project submitted in partial fulfillment of the requirements for the award of the degree **of**

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

(2014-2018)

by

N. Venkat Ramana Reddy (14241A1243)

K. Venkata Raju (14241A1228)

K. Sri Vardhan (14241A1231)

J. Subash (14241A1225)

Under the Esteemed guidance of

Mrs. L. Sukanya

Assistant Professor

Department of Information Technology

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

DEPARTMENT OF INFORMATION TECHNOLOGY

April 2017

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

DEPARTMENT OF INFORMATION TECHNOLOGY

April 2017

CERTIFICATE

This is to certify that the Mini Project titled **"UCOMPLAIN"** is a bona fide work done by **N. Venkat Ramana Reddy(14241A1243), K. Venkata Raju(14241A1228), K. Sri Vardhan(14241A1231), J. Subash(14241A1225),** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology** and submitted to the Department of Information Technology, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad. The work embodied in this report has not been submitted earlier at any other University or Institute for award of any degree.

L. Sukanya **Dr. Y. Vijayalata**

(Guide) **(Head of Department)**

(External Examiner)

ACKNOWLEDGEMENT

We take the immense pleasure in expressing gratitude to our Internal Guide **L. Sukanya** Assistant Professor**,** Information Technology. We Express our sincere thanks for her encouragement, suggestions and support which provided the impetus and paved the way for the successful completion of the project work.

We are grateful to **Dr. Y. Vijayalata,** Head of Department of Information Technology and Members of Project Coordinators **Dr. Y. Prasanna Lakshmi, Mr. Y.J. Nagendra Kumar, Ms. V. Shailaja** and **Ms. P. Bharathi** for the valuable suggestions**.**

We express our sincere thanks to our Director, **Prof. P .S. Raju** and Principal, **Dr. Jandhyala N Murthy,** of Gokaraju Rangaraju Institute of Engineering and Technology for providing us the conductive environment for carrying through our academic schedules and project with ease. We also take this opportunity to convey our sincere thanks to the teaching and non- teaching staff of Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad for their kind cooperation throughout.

Thanking You

N. Venkat Ramana Reddy
venkat.narahari96@gmail.com
+91 9985535643

K. Venkata Raju
kanumurivenkata@gmail.com
+91 9581846677

K. Sri Vardhan
srivardhankandike@gmail.com
+91 739647946

J. Subash
subashjogu9999@gmail.com
+91 9542402075

DECLARATION

This is to certify that the Mini Project titled **"UCOMPLAIN"** is a bona fide work done by us in partial fulfilment of the requirements for the award of the degree Bachelor in Technology in Information Technology and submitted to the Department of Information Technology, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad.

We also declare that this project is a result of our effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references.

The work embodied in this report has not been submitted earlier at any other university or institute for the award of any degree.

N. Venkat Ramana Reddy (14241A1243)

K. Venkata Raju (14241A1228)

K. Sri Vardhan (14241A1231)

J. Subash (14241A1225)

ABSTRACT

Everyday we come across many grievances with our public service system and though we want to lodge a complaint to the concerned authorities we seldom do so because of the tedious red tape involved with it. **UCOMPLAIN** provides an online tool to solve problems faced by common public by saving time and demolishing the bridge of officers between the concerned authorities and the public.

The primary objective is to make complaints easier to coordinate, monitor, track and resolve. In the era of Facebook's like button influencing billions of people we are implementing a similar button to help a single complaint have thousand voices.

A customer can file a complaint by filling in the details. They can also view and respond to other complaints within their interests. It is also possible to add pictures which can much more easily show the severity of the situation. The complaint is forwarded to the concerned authorities on the basis of type of complaint. This helps in avoiding long queues for logging a complaint and helps in resolving the problem faced by people.

CONTENTS

TABLES

FIGURES

SCREENSHOTS

1. INTRODUCTION

Introduction

1.1 Motivation

With the dawn of the internet and digital age, websites have become a regular way of exchange of information. Today over a billion websites are influencing our lives in ways unnoticed. Websites have been growing rapidly in the digital age.

The goal of the project is a website to provide a pipe for easy and fast resolve of problems faced by common public who frequently complain but go unnoticed. To make a single individual site for notifying and viewing the complaints that the public faces.

1.2 Objective

The purpose of this document is to define requirements of the UCOMPLAIN to capture a complete set of requirements on the system. To improve the governance of officials by giving effective notices of aggravating problems faced by public .

Also giving a chance for public knowing other people's problems and knowing how the public administration is handling the issues.

2. LITERATURE SURVEY

Literature Survey

The website UCOMPLAIN provides the user to be able to complain to officials without any physical tedious tasks provided and also make other complaints get worked on by giving their voice of support by upvoting a complaint. This document furnishes all the necessary UML diagrams for the project UCOMPLAIN.

2.1 Feasibility Study

Feasibility study is an important phase in the software development process. It enables the developer to have an assessment of the product being developed. It refers to the feasibility study of the product in terms of outcomes of the product, operational use and technical support required for implementing it. Feasibility study should be performed on the basis of various criteria and parameters. The various feasibility studies are:1) Operational feasibility 2) Technical feasibility 3) Economic feasibility

2.1.1 Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.1.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.1.3 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3. ANALYSIS

Analysis

3.1 Existing System

In the previous most of the complaints were physically submitted. At present most of complaints come through social networking websites. Complaint management systems have been deployed at private sector consumer goods industry.

3.2 Proposed System

In this project we implement a separate exclusive website for public complaints on social issues. A simple post can generate upvotes from other users and also notify various levels of authorities related to the specific issue.

3.3 Requirement Specification

The system analysis is done keeping in mind the following key requirements:

1. Provide a simple user friendly interface for client for effective use .

2. Provide a quick notifications to officials.

3.3.1 Software Requirements

- Operating System : Windows 7/ Windows 8/Windows 10
- Code Editor: Sublime Text 3/ Notepad++
- Browser: Chrome 3.0/IE 11/Mozilla Firefox 45 and above
- Front End Framework: Bootstrap 4
- Web server: Apache 2.4
- Database Server: MySQL Server 5.7
- Libraries: PHPMailer
- Markup Language: HTML
- Styling Language: CSS
- Scripting Language: PHP
- Others: phpmyadmin

3.3.2 Hardware Requirements

- Processor: Intel dual core processors and higher.
- Main Memory: 2GB
- Hard Disk: 16GB

4. DESIGN

Design

A model is a simplification of reality. A model provides the blueprints of a system. A model may be structural, emphasizing the organization of the system, or it may be behaviour, emphasizing the dynamics of the system.

4.1 UML Diagrams

UML diagrams is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts of a software intensive system.

4.1.1 Use case Diagram

A use case specifies the behaviour of a system or a part of a system or a part of a system and is a description of set of sequences of actions, including variants, that a system performs to yield an observable result of value to an actor.

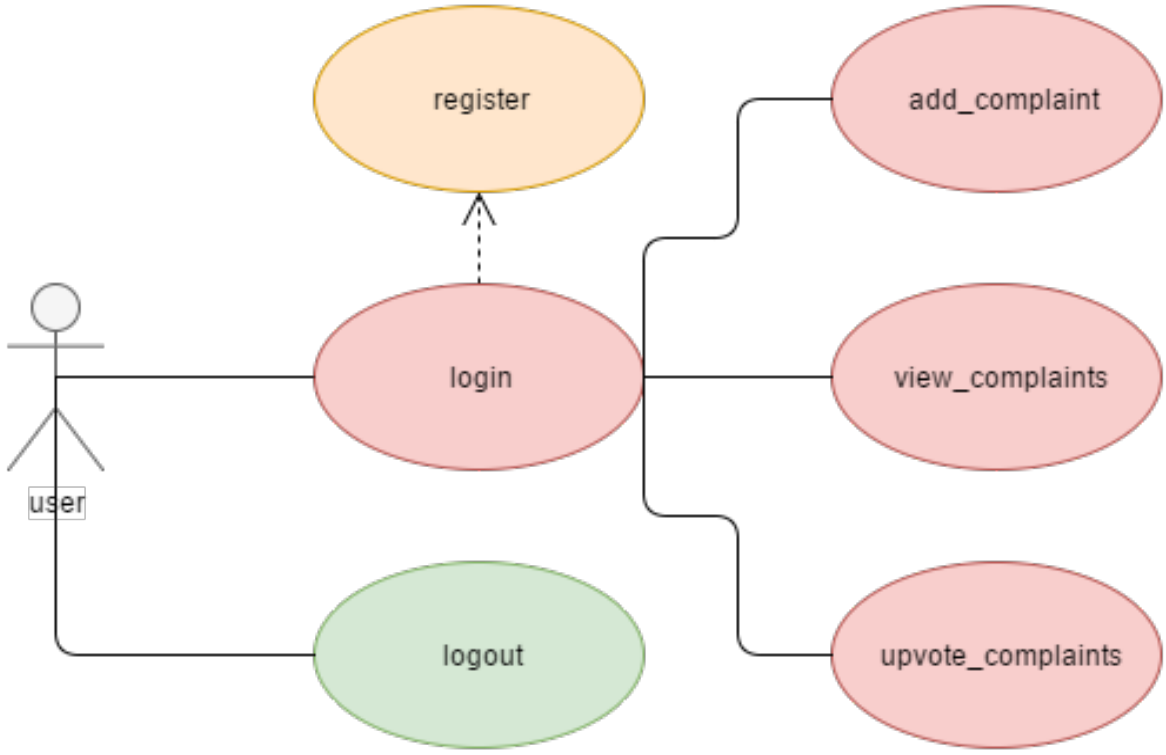


Fig. 4.1.1 Use case Diagram

The use case diagram depicts the user can login and logout of system if he is already registered. He can add complaint or view complaints and upvote complaints if he is logged in which are the use cases.

Actors:	User
Description:	Use case diagram to make and view complaints
Trigger:	User should post a complaint.
Preconditions:	User must be registered
Postconditions:	User must be able to operate the website
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	1
Frequency of use:	No limit
Special Requirements:	None
Business Rules:	None
Assumptions:	The posted complaint gets notified to official
Notes and Issues:	None
Extends:	None

Table 4.1.1 Use case Diagram

4.1.2 Class Diagrams

A class diagram shows a set of classes, interfaces, and collaborations and their relationships. It is used to model the static design view of the system. Class diagrams are useful for constructing executable systems through forward and reverse engineering.

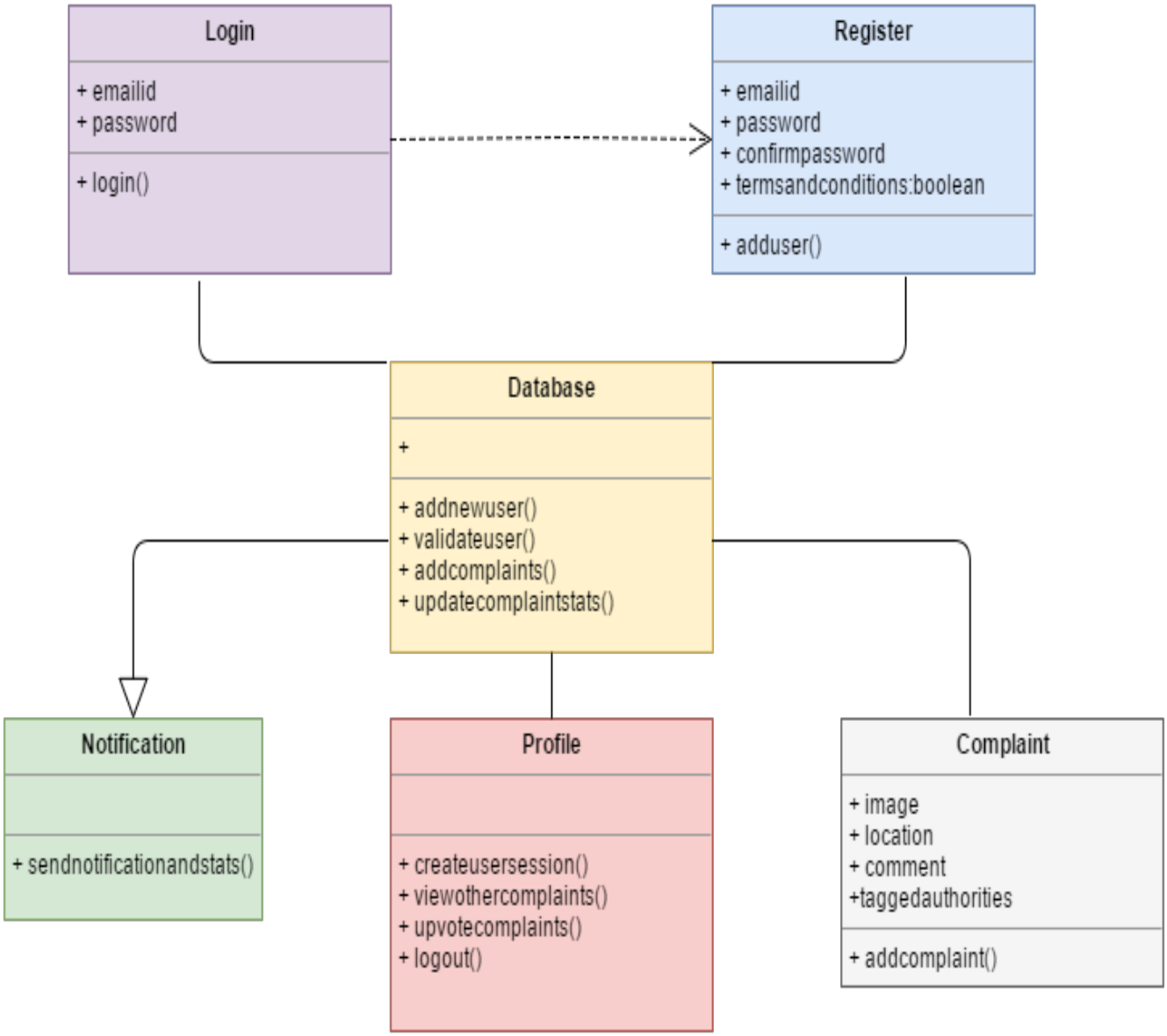


Fig 4.1.2 Class Diagram

The class diagram above gives the various relationships between the classes and of each class is related or dependent on other class.

Class	Attributes	Operations	Use
Login	Email id, password	login()	User to login
Register	Email id, password, confirm password.	adduser()	User to register
Database	-	addnewuser(), validateuser(), addcomplaints(), updatecomplaints()	To contain information about users and complaints
Notification	-	sendnotificationstats()	Sending notification to officials
Profile	-	createusersession(), viewothercomplaint(), upvotecomplaints(), logout()	Viewing user profile and upvoting complaints
Complaint	Image, location, comments, taggedauthorities	addcomplaint()	To post a complaint

Table 4.1.2 Class Diagram

The above class table gives the inferences about the projects class diagram and its uses.

4.1.3 Sequence Diagram

A Sequence diagram is an interaction diagram that emphasizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged in x-axis and messages ordered in increasing time along y-axis. They are called event diagrams or event scenarios. A sequence diagram shows as parallel vertical lines, different processes or objects that live simultaneously and horizontal arrows the messages exchanged between them in order which they occur.

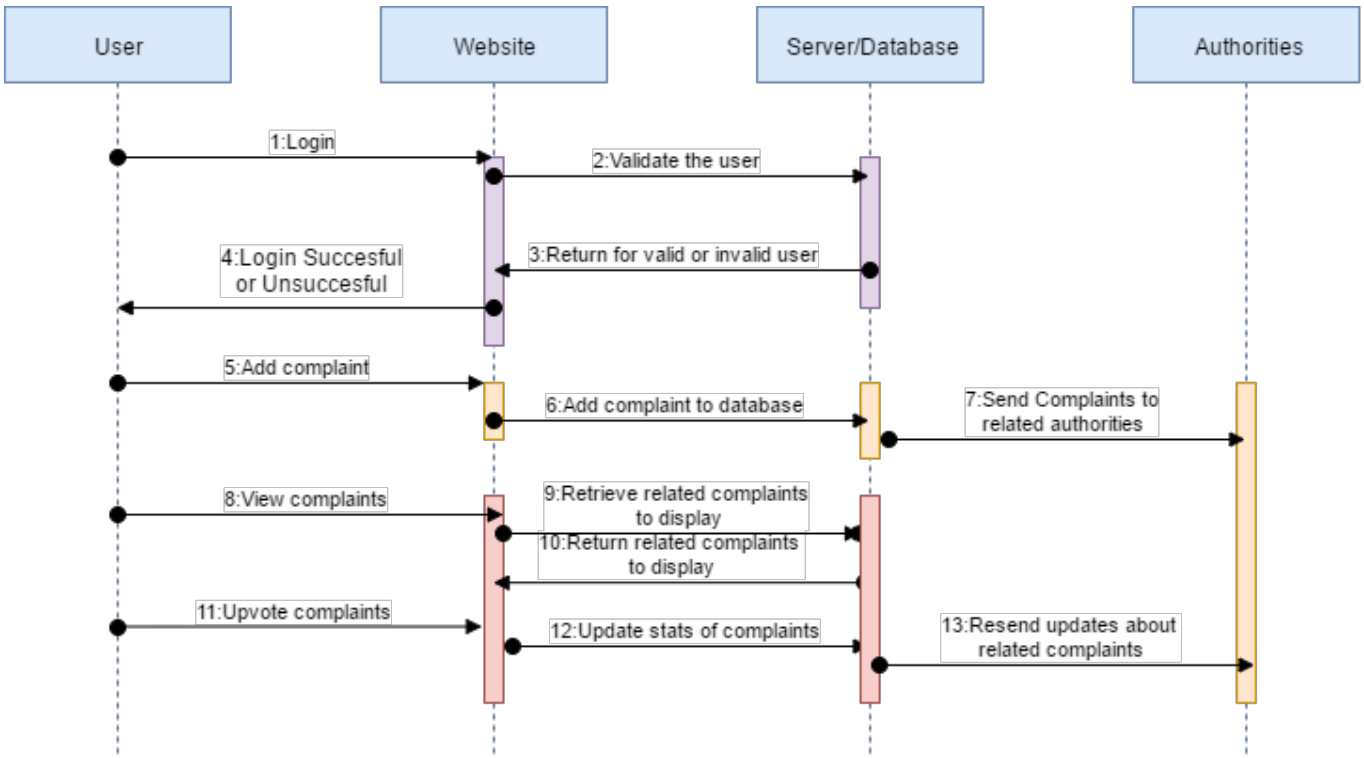


Fig 4.1.3 Sequence diagram

The above sequential diagram show the exchange of messages between four object ie: user, website, server/database and authorities and time is taken into account during flow of messages. The user sends messages login which on received by website verifies and sends a response if the login is successful or not. Then the user exchanges information from server via website for adding complaints, viewing and upvoting them. The server send the notification message to authorities object and also server continuously updates itself.

4.1.4 Collaboration Diagram

A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages Graphically the collaboration diagram is a collection of vertices and arcs. It portrays the roles, functionality and behaviour of individual objects as well as the overall operation of the system in real time. They are best suited for showing the simple interactions among objects.

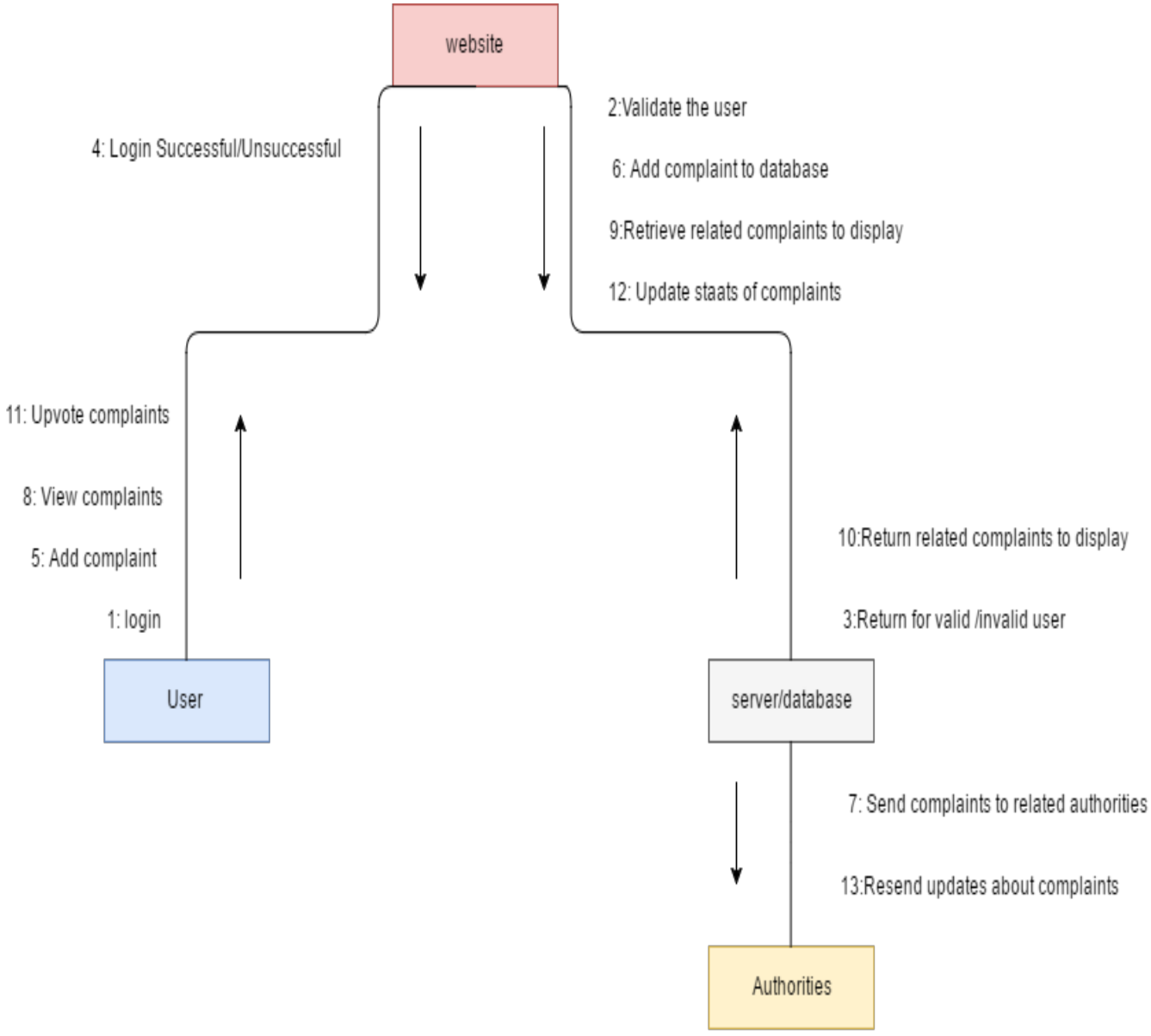


Fig 4.1.4 Collaboration diagram

The collaboration diagrams has four objects website, user, server/database, authorities where the message exchange take places which have a high structural organization and have similar messages as that of activity diagram.

4.1.5 Activity Diagram

An activity diagram show the flow of control from activity to activity. An activity is an ongoing execution within a state machine. It is essentially a flowchart modelling the dynamic aspects of the system.

The purpose of activity diagram is:

- Draw activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

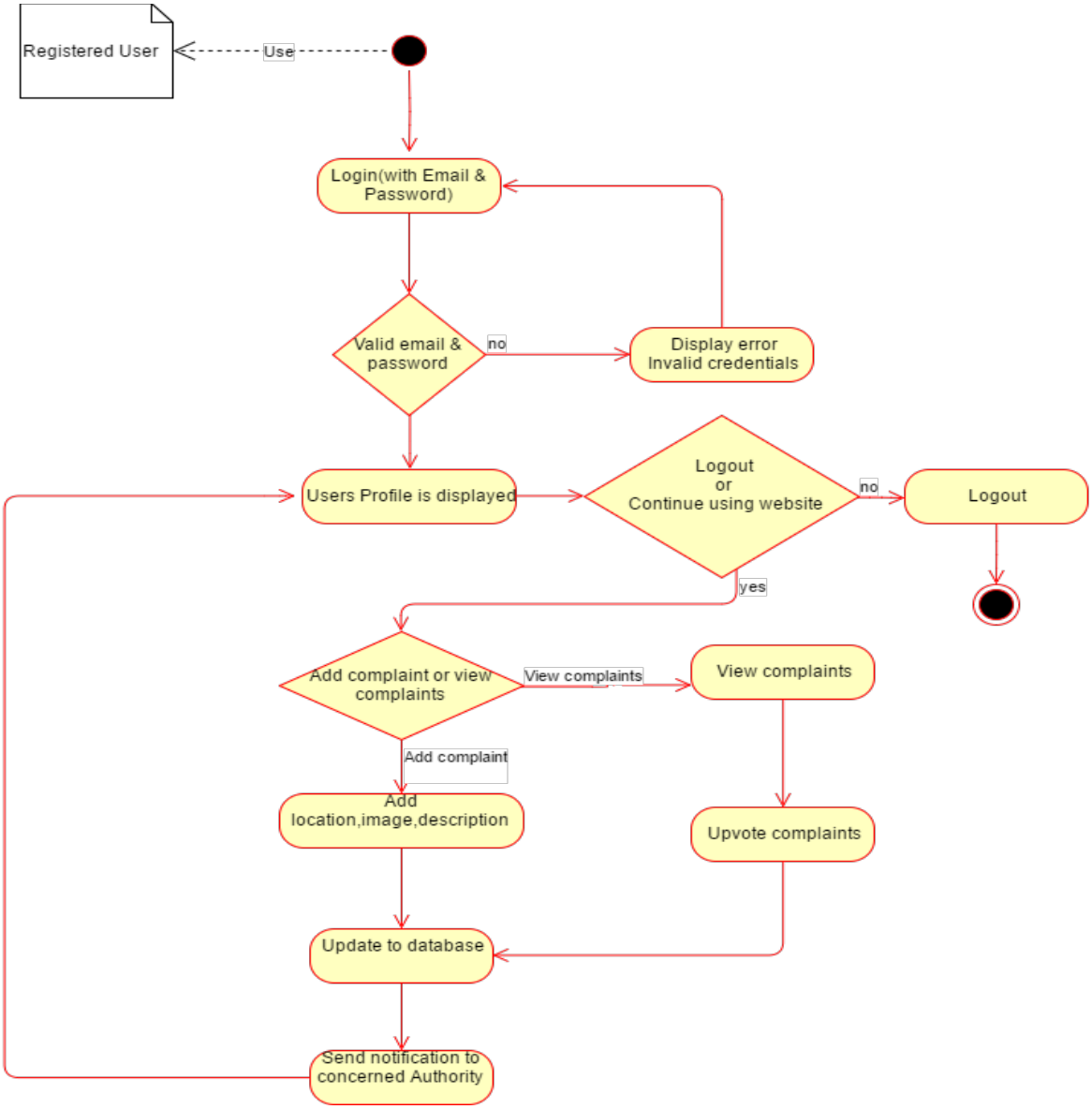


Fig 4.1.5 Activity diagram

The above diagram shows the different activities that flow in order to complete its main activity of complaining, viewing and updating database.

4.1.6 Statechart Diagram

A statechart diagram shows a state of a machine. It is used to model the dynamic aspects of a system. For most part this involves modelling the behaviour of reactive objects.

Purpose of using this is:

- To model the dynamic aspect of a system.
- To model lifetime of reactive system.
- To describe different states of an object during its lifetime.
- Define a state machine to model the states of an object.

Statechart diagrams are used to model the states and also the events operating on the system. It is mainly used to analyse the object states influenced by events. This analysis is helpful to understand the system behaviour during its execution.

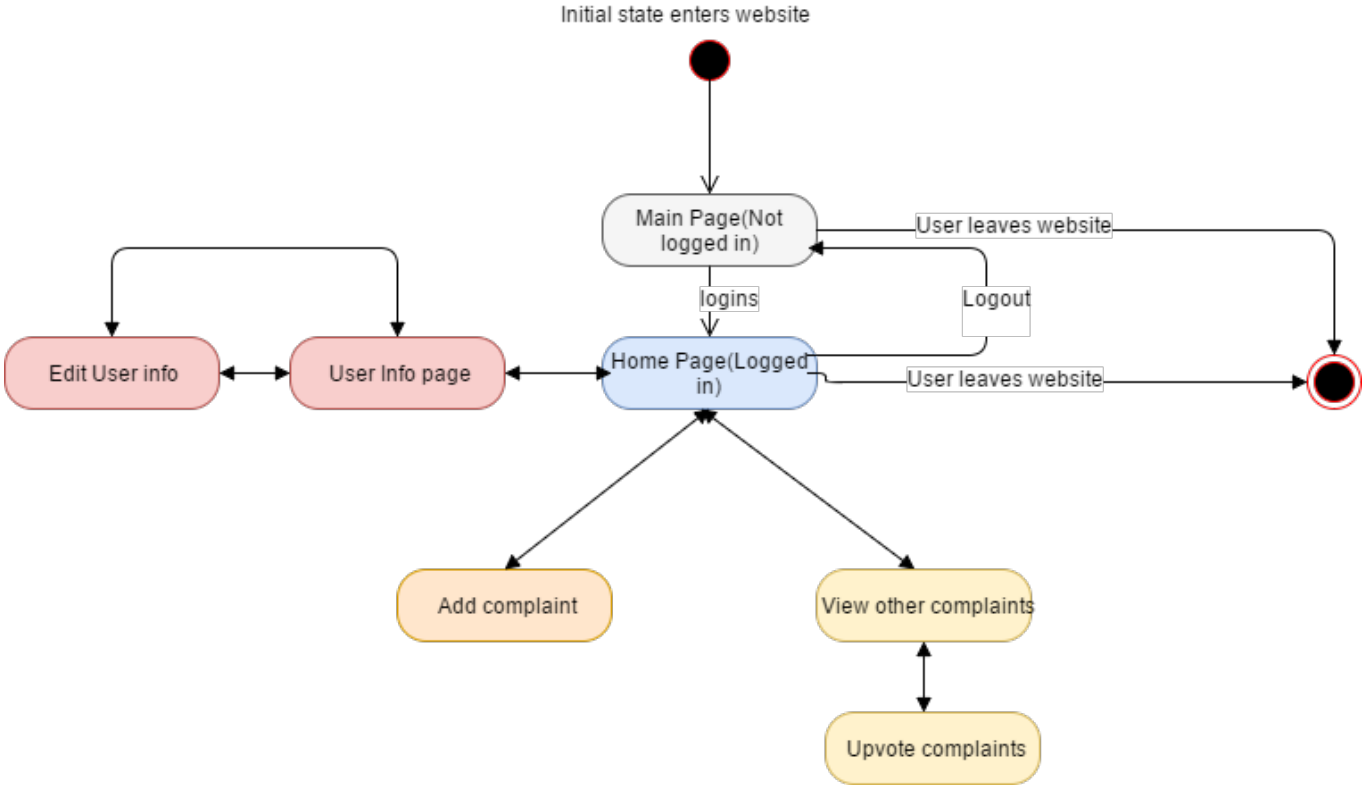


Fig 4.1.6 Statechart diagram

The diagram show the states of how the user logins and logs out of the system. The depiction of initial and final states and how the other states are getting changes on different triggers.

4.1.7 Component Diagram

Component diagrams are used to model the static implementation view of a system. This involves modelling the physical things that reside on a node, such as executables, libraries, tables, files and documents.

Component diagrams can be used to:

- Model the components of system.
- Model the database schema.
- Model the executables of an application.
- Model the system's source code.

The organization of these components help the developers to better understand the semantics of the system and flow of resources and data.

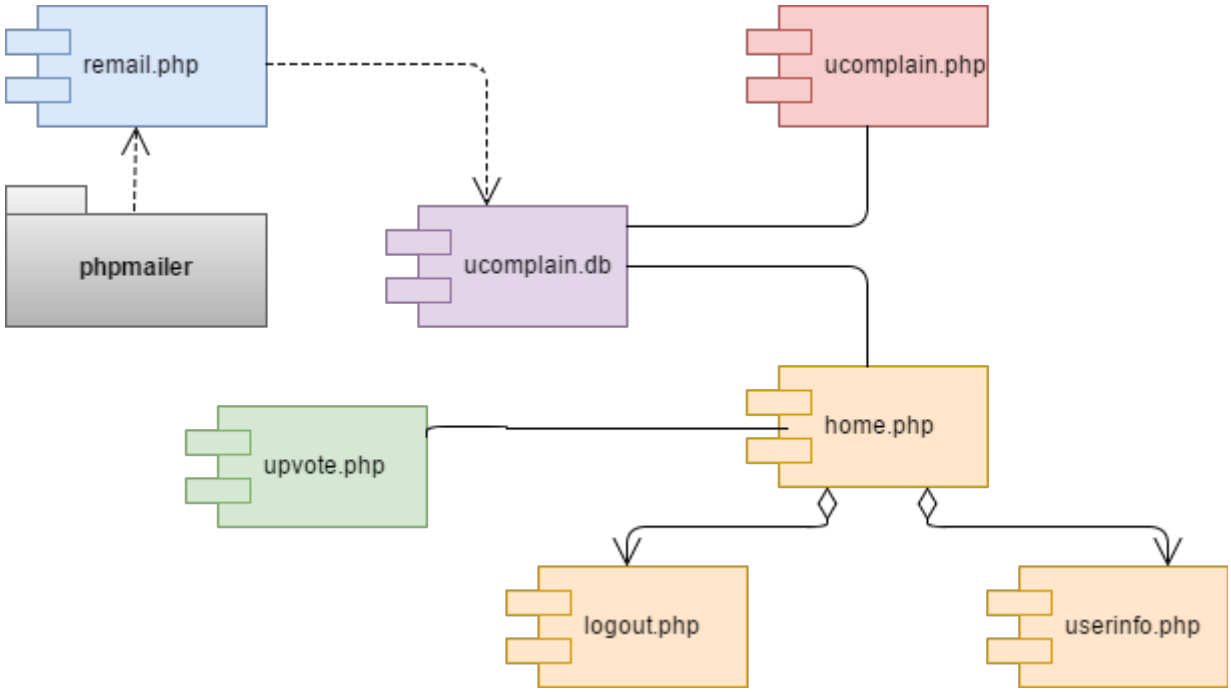


Fig 4.1.7 Component diagram

Here the components are the executable php files and the database file. The main website page is ucomplain which on logging goes to home where you can view complaints. Here you can add, view, upvote complaints or logout. PHP mailer represents a library component.

4.1.8 Deployment Diagram

Deployment diagrams show the configuration of runtime processing nodes and the components that live on them. They model the topology of the hardware on which your system executes.

The purpose of deployment diagrams are:

- Visualize the hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe the runtime processing nodes.

They are mainly used by system engineers. To meet software applications need of hardware resources, the hardware components must be designed efficiently and in a cost effective way.

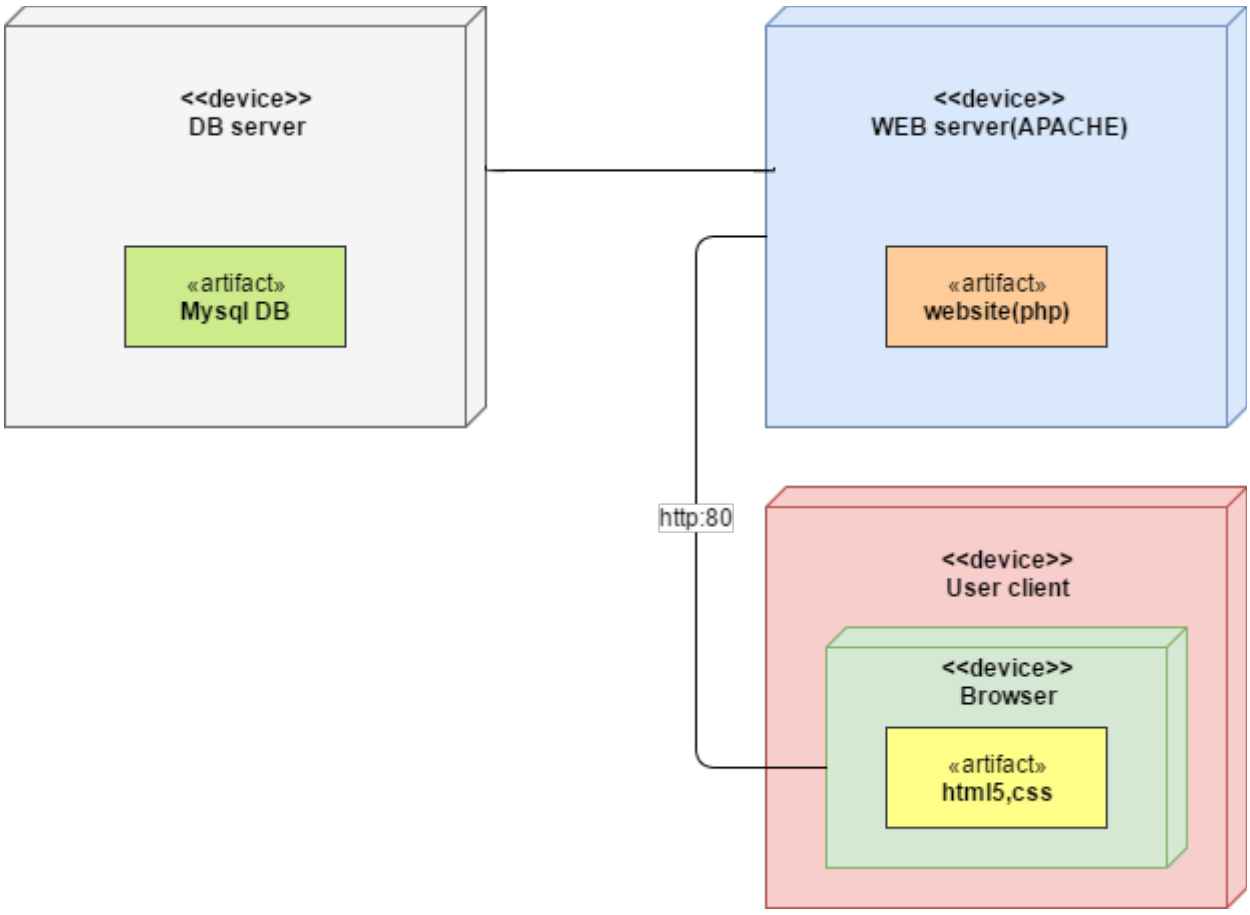


Fig 4.1.8 Deployment diagram

The figure illustrates the two servers. The web server which serves requests that come from the client node from the browser node using port number 80 by http request and also utilizes the data from the database server.

5. IMPLEMENTATION

Implementation

Implement all the modules considering all specifications and its hardware requirements. Finally integrate and test them with requirements connected to the system**.**

5.1 Modules

In this website we are having 6 modules. Each module has its own unique functionality.

- Register module
- Login module
- Add complaint module
- Upvote complaint module
- Change password/username module
- Logout module

5.2 Module Description

5.2.1 Register

The new user can create an account where the validation of the user is done by checking the database and if the validation is successful then the user's account is created.

5.2.2 Login

The login module validates the form data using the data from database and if mail and password match the data then he is redirected to his user home page and a session is created.

5.2.3 Add complaint

This module is used to take in the form data, validate it and add it to the database. Then the complaint is forwarded to the respective authorities and officials.

5.2.4 Upvote complaint

The complaints posted from other users which can be viewed can be upvoted by this.

5.2.5 Change password/username

The user can change password or username by updating database.

5.2.6 Logout

The logout module is used to end the session and redirect to the main page.

5.3 Code Modules

5.3.1 Register Code module

```
 $conn = new mysqli("localhost","root","narahari96","ucomplain"); //check
connection
 if($conn->connect_error)
 {
  die("Connection failed: " . $conn->connect_error);
 }

 $_SESSION["is_logged_in"]= false;

if(isset($_POST["reg_btn"]))
{ $username=$_POST['username'];
 $password=$_POST['password'];
 $password2=$_POST['password2'];
 $email=$_POST['email'];
 $rmessage="";


 /* Password Matching Validation */
 if($password != $password2)
 {
  $message = "Passwords not same!";
 }
 if($password === $password2)
 {
  $query1="SELECT * from users where email='$email'";
  $result = $conn->query($query1);
  if($result->num_rows > 0)
  {
   $rmessage = "User already exists with this email!";
  }
```

```
  else
  {
   $query2= "INSERT INTO users (username,email,password) VALUES
('$username','$email','$password')";
   if ($conn->query($query2) === TRUE)
   {
    $rmessage = "You have registered successfully!";
   }
   else
   {
    $rmessage = "Error in registering!";
   }
   unset($_POST);
  }
 }
}
```

5.3.2 Login Code Module

```
 $conn = new mysqli("localhost","root","narahari96","ucomplain");
 //check connection
 if($conn->connect_error)
 {
  die("Connection failed: " . $conn->connect_error);
 }

 $_SESSION["is_logged_in"]= false;if(isset($_POST['lgn_btn']))
{    $email=$_POST['email'];
    $password=$_POST['password'];
    $lmessage="";
    $query1="SELECT * FROM users WHERE email='$email' AND
password='$password'";
    $result = $conn->query($query1);
```

```
    $row = $result->fetch_assoc();
    if($result->num_rows > 0)
    {
      //$lmessage="You are logged in!";
      $_SESSION["is_logged_in"]= true;
      $_SESSION["username"]=$row["username"];
      $_SESSION["email"]=$row["email"];
      $_SESSION["password"]=$row["password"];
      header("location:home.php");
    }
   else
   {
      $lmessage="Incorrect Email/Password";
    }
}
```

5.3.3 Add complaint Code Module

```
session_start();
//connect to database
$conn = new mysqli("localhost","root","narahari96","ucomplain");
if($_SESSION['is_logged_in'] === true)
{
 if(isset($_POST["cmp_btn"]))
 {
  $cmpdetails=$_POST["cmpdetails"];
  $cmpaddress=$_POST['cmpaddress'];
  $cmplocation=$_POST['areas'];
  $cmptype=$_POST['type'];
  $cmessage="";
  $cmpemail=$_SESSION['email'];
  $cmpimage=addslashes(file_get_contents($_FILES['image']['tmp_name']));
  $cmpimagen="abc";
```

```
  //Get the content of the image and then add slashes to it

  // Insert the image name and image content in image_table
  $query4="INSERT INTO
complaints(complaintno,email,location,type,address,image,details,upvotes,ts
,counter) VALUES
(NULL,'$cmpemail','$cmplocation','$cmptype','$cmpaddress','$cmpimage','$cmp
details',0,CURRENT_TIMESTAMP,0)";

  if($conn->query($query4)===TRUE)
  {
      $cmpno=$conn->insert_id;
      $query4a="CREATE TABLE `$cmpno` (email VARCHAR(100) PRIMARY KEY)";
    if($conn->query($query4a)===TRUE)
    {
      $query6="SELECT * FROM `area` WHERE type='".$cmptype."' AND circle IN
(SELECT circle FROM `locationtoarea` where location='".$cmplocation."')";
      $result=$conn->query($query6);
      if ($result->num_rows > 0)
      {
       while($row = $result->fetch_assoc())
      {
       require('PHPMailer-master/PHPMailerAutoload.php');
       $subject = "You have a compliant!";
       $mail = new PHPMailer();

       $mail->isSMTP();
       $mail->Host = "ssl://smtp.gmail.com";
       $mail->SMTPAuth = true;
       $mail->Username = "ucomplaincc@gmail.com";
       $mail->Password = "ucomplain6";
       $mail->SMTPSecure = 'ssl';
       $mail->Port = 465;
```

```
  //     $headers = "MIME-Version: 1.0\n";
  //     $headers .= "Content-Type: multipart/form-data";
  //     $headers .= "Content-Type: text/html";
  //     $mail->AddCustomHeader($headers);
       $today = date("F j, Y, g:i a");
       $mail->setFrom('ucomplaincc@gmail.com', 'UCOMPLAIN');
       $mail->addAddress($row['c_in_email'], $row['c_incharge']);
       $mail->addAddress($row['z_in_email'], $row['z_incharge']);
       $mail->Subject = $subject;
       $mail->AddEmbeddedImage($_FILES['image']['tmp_name'],
"cimg","altimg","base64");
       $mail->Body    = '<html><body>
                          <h3>Complaint:</h3><br>
                          <p>Details: '.$cmpdetails.
                          '<br>Type: '.$cmptype.
                          '<br>Location: '.$cmpaddress.
                          '<br>Area: '.$cmplocation.
                          '<br>Time: '.$today.
                          '<br>Photo: <img alt="image" src="cid:cimg"
width="50%" height="50%">'
                          .'</p>'
                          .'</body></html>';
        $mail->IsHTML(true);

       if(!$mail->send())
       {
          $cmessage="Message could not be sent.</br>Mailer Error: ".$mail-
>ErrorInfo;
       }
       else
       {
        $cmessage="Compliant successfully posted!";
        header("location:home.php");
```

```
       }
      }
      }
      else
      {
       $cmessage= "Could not post compliant to specified authorities due to
unknown restriction!";
      }
    }

    else
    {
      $cmessage=$conn->error;
    }
  }
  else
  {
   $cmessage="Error in posting complaint.Please try again!";
  }
 }

}
```

5.3.4 Upvote Code Module

```
<?php
session_start();
//connect to database
$conn = new mysqli("localhost","root","narahari96","ucomplain");
if($conn->connect_error)
 {
  die("Connection failed: " . $conn->connect_error);
 }
```

```
if($_SESSION['is_logged_in'] === true)
{
  if(isset($_POST["upvote"]))
  {
    $complaintno=$_POST['cmpno'];
    $email=$_SESSION['email'];

      $query4="SELECT * FROM  `$complaintno` WHERE email='".$email."'";
      $result1=$conn->query($query4);
      if ($result1->num_rows >0)
      {
        $status=1;
      }
      else
      {
        $status=0;
      }

      echo $status;
   if($status==0)
   {
    $query5="SELECT upvotes FROM complaints where complaintno=".
$complaintno."";
    $result3=$conn->query($query5);
    if($result3->num_rows> 0)
    {
     while($row=$result3->fetch_assoc())
     {
      $u=$row['upvotes'];
      $u=$u+1;
      $query6="INSERT INTO `$complaintno`(email) VALUES('".$email."')";
      $conn->query($query6);
      $query7="UPDATE complaints SET upvotes=".$u." WHERE complaintno=".
```

```
$complaintno."";
      echo $query7;
      $conn->query($query7);
     }
    }
   }
   if($status==1)
   {
    $query5="SELECT upvotes FROM complaints where complaintno=".
$complaintno."";
    $result3=$conn->query($query5);
    if($result3->num_rows > 0)
    {
     while($row=$result3->fetch_assoc())
     {
      $u=$row['upvotes'];
      $u=$u-1;
      $query6="DELETE FROM `$complaintno` WHERE email='".$email."'";
      echo $query6;
      $conn->query($query6);
      $query7="UPDATE complaints SET upvotes=".$u." WHERE complaintno=".
$complaintno."";
      echo $query7;
      $conn->query($query7);
     }
    }
   }

   header("location:userinfo.php");
  }
}
else
{
```

```
  header("location:ucomplain.php");
}

?>
```

5.3.5 Change password/username Code Module

```
<?php
session_start();
//connect to database
$conn = new mysqli("localhost","root","narahari96","ucomplain");
if($conn->connect_error)
 {
  die("Connection failed: " . $conn->connect_error);
 }
if($_SESSION['is_logged_in'] === true)
{

//Username reset php

 if(isset($_POST["un_btn"]))
 {
  $username=$_POST['username'];
  $email=$_SESSION['email'];
  $umessage="";
  $query1 = "UPDATE users SET username='$username' WHERE email='$email'";
  if ($conn->query($query1) === TRUE)
   {
    $umessage="Username changed successfully changed!";
    $_SESSION['username']=$username;
    header("location:userinfo.php");
   }
  else
   {
```

```
    $umessage="Error in changing username. Please try again!";
   }
 }

//Password reset php

 if(isset($_POST["pwd_btn"]))
 {
  $email=$_SESSION['email'];
  $password=$_POST['password'];
  $password1=$_POST['password1'];
  $password2=$_POST['password2'];
  $umessage="";
  if($password === $_SESSION['password'])
  {
   if($password1 === $password2)
   {
    $query2 = "UPDATE users SET password='$password2' WHERE
email='$email'";
    if ($conn->query($query2) === TRUE)
    {
     $umessage="Password successfully changed!";
     $_SESSION['password']=$password1;
     header("location:userinfo.php");
    }
    else
    {
     $umessage="Error in changing password.Please try again!";
    }
   }
   else
   {
    $umessage="New passwords do not match!";
```

```
   }
  }
  else
  {
   $umessage="Old password is incorrect";
  }
 }



}
else
{
  header("location:ucomplain.php");
}
?>
```

5.3.6 Logout Code Module

```
<?php
session_start();
session_unset();
session_destroy();
unset($_SESSION);
header("location:ucomplain.php");
?>
```

5.4 Libraries

Phpmailer:

PHPMailer is a class library for PHP that provides a collection of functions to build and send email messages. PHPMailer supports several ways of sending email: mail(), Sendmail, qmail & direct to SMTP servers. You can use any feature of SMTP-based e-mail, multiple recipients via to, CC, BCC, etc. In short: PHPMailer is an efficient way to send e-mail within PHP. It is hosted, developed and maintained on github.

5.5 Frameworks

Bootstrap:

Bootstrap is the most popular open source HTML, CSS and JS framework for mobile first responsive projects. It makes frontend development faster and easier. It is hosted, maintained and developed on github. Bootstrap is modular and consists of a series of less style sheets that implement various components of the toolkit. Each Bootstrap component consists of HTML structure, CSS declarations and in some cases accompanying JS code. It is supported by a wide variety of browsers.

5.6 Languages

HTML:

Hyper Text Markup Language is a most popular markup language for developing web pages and applications. It describes the structure of the web page semantically and originally included cues for appearance of the document. HTML that web browsers use to interpret and compose text, images and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser and these characteristics can be altered and enhanced by web page designer's additional use of CSS.

CSS:

Cascading Style Sheets is a style sheet language used for describing the presentation of the document written in markup language. It is primarily designed for separation between document content from document presentation, including aspects such as the layout, colours and fonts. This can provide more accessibility and flexibility and control in the specification of presentation characteristics.

PHP:

Hypertext preprocessor is a server side scripting language designed primarily for web development. It is one of the most widely used and can be put onto any operating system and server. PHP code may be embedded into HTML markup or it can be used in combination with various web content systems and frameworks.

5.7 Servers and Database

Web Server(Apache HTTP Server):

Apache is the world's most used HTTPd web server. It is developed and maintained by community of developers of Apache software foundation. It played a key role in initial growth of world wide web.

Database Server(MySQL Community Server):

A database server is a computer program that provides database services to other computer programs or to computers, as defined by the client server model. The term may also refer to a computer dedicated to a running such a program. Database management systems frequently provide database server functionality, and some database management systems such as MySQL rely exclusively on client server model for database access. Users access a database a server either through a front end running on user's computer which displays requested data or through back end which runs on the server ad handles such as data analysis and storage.

Database(MySQL):

It is an open source relational database management used by many high profile large scale web based firms. It is one of the most widely used relational database. It is a central component of LAMP(Linux Apache MySQL PHP, Perl, etc.) open source web application stack and other popular AMP stacks.

6. TESTING

Testing

Testing your documents should be an integral part of your plain language writing process. It should not just be something you do after the final version to see if it worked. This is especially important if you're writing to hundreds, thousands or even millions of people. The information gained in testing can save time in answering questions about your document later.

6.1 System Testing

The process of executing a system with the intent of finding an error. Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction. Quality is defined as justification of the requirements.

Defect is nothing but deviation from the requirements. Defect is nothing but bug. Testing can demonstrate the presence of bugs, but not their absence. Debugging and Testing is not the same thing. Testing is a systematic attempt to break a program or the AUT.

6.2 Testing Methodologies

6.2.1 Functionality Testing

Test for all the links in web pages, database connection, forms used for submitting or getting information from user in the web pages, Cookie testing etc.

6.2.2 Usability Testing

Used to test the navigation of user, checking the content such as user design and information for user help.

6.2.3 Interface Testing

Used to check the interactions between servers and client. It also checks if the errors returned by servers are handled.

6.2.4 Compatibility Testing

This testing is used to check browser, operating system, responsiveness and printing option compatibility.

6.2.5 Performance Testing

Website should be able to handle heavy load on it. Hence load test for handling huge data or user requests and stress tests for handling huge number of sessions must be made.

6.2.6 Security Testing

Url check, inputs checking and database validations come under security testing.

6.3 Validation

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

6.3.1 Test cases

A test case is a condition which accepts inputs, processes it and gives expected output. A test case in software engineering normally consists of a unique identifier, requirement references from a design specification, preconditions, events, a series of steps (also known as actions) to follow, input, output and it validates one or more system requirements and generates a pass or fail.

The mechanism for determining whether a software program or system has passed or failed such a test is known as a test oracle. Test cases are often referred to as test scripts, particularly when written. Written test cases are usually collected into test suites. 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.

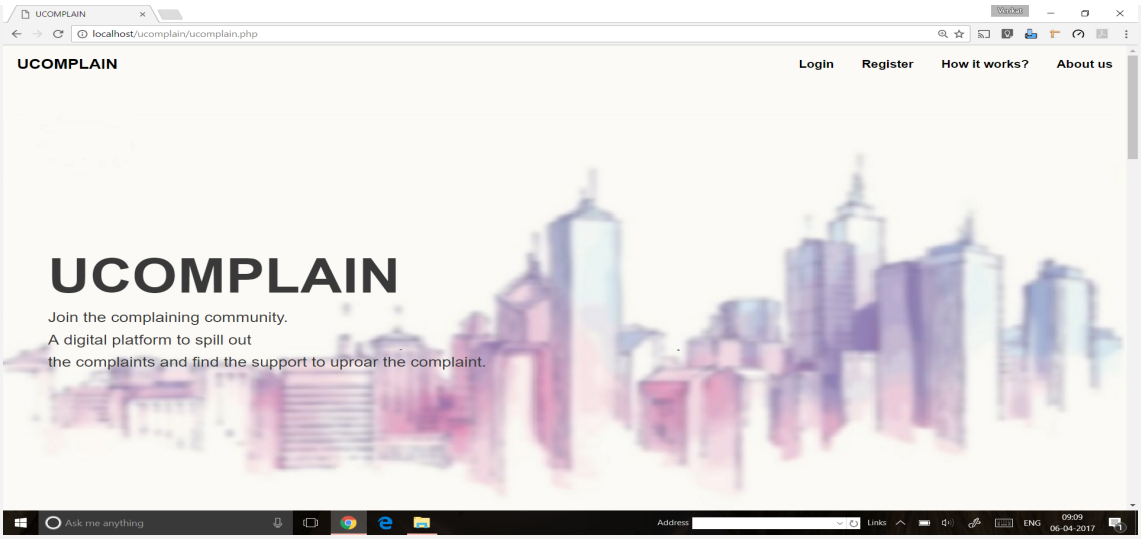
Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

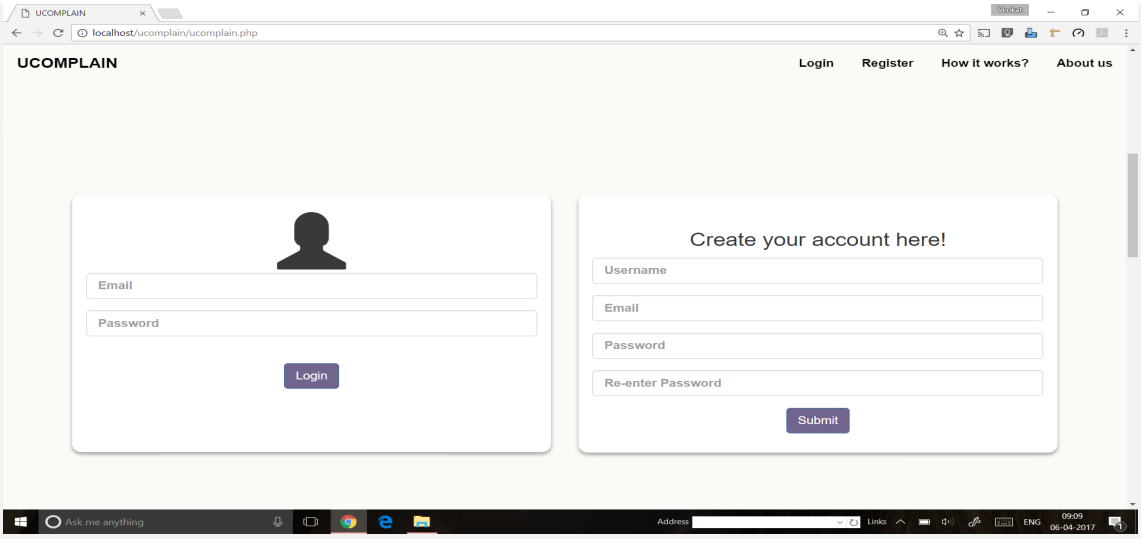
Test Case Id	Test Case Description	Step	Expected Result	Actual Result	Pass/Fail
1.	Register	To create a new user.	Create an account into the database.	• New Account created. • Error in registering the user.	Pass Fail
2.	Login	To create a user session and logged in.	Redirect to users home page.	• Session created and users home page displayed. • Error in logging in.	Pass Fail
3.	Change username / password	Change username or password.	Updated the username and password in database.	• Changed • Error could not change.	Pass Fail
4.	Post complaint	To fill complaint details and submit.	Submit it to database.	• Complaint submitted and notification sent to authority. • Error in submitting the complaint.	Pass Fail
5.	Upvote complaint	To click upvote button.	Increase or decrease number of upvotes.	• Upvoted successfully • No change.	Pass Fail
6.	Logout	To end session and come back to main website.	Redirect to the main website by ending session.	• Successfully logged out. • Error in logging out	Pass Fail

Table 6.3.1 Test Cases

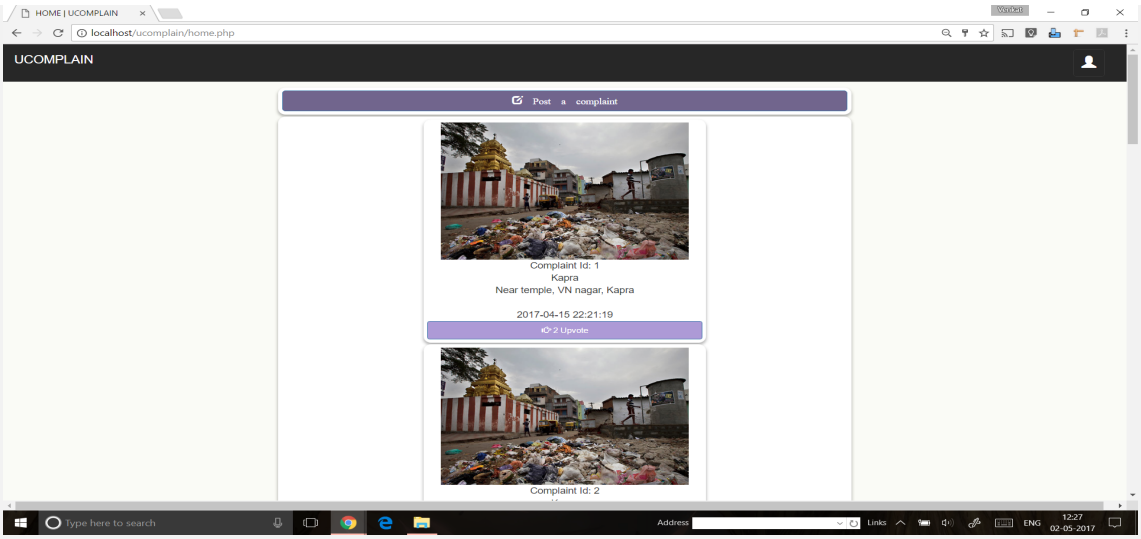
6.4 GUI OF THE SYSTEM



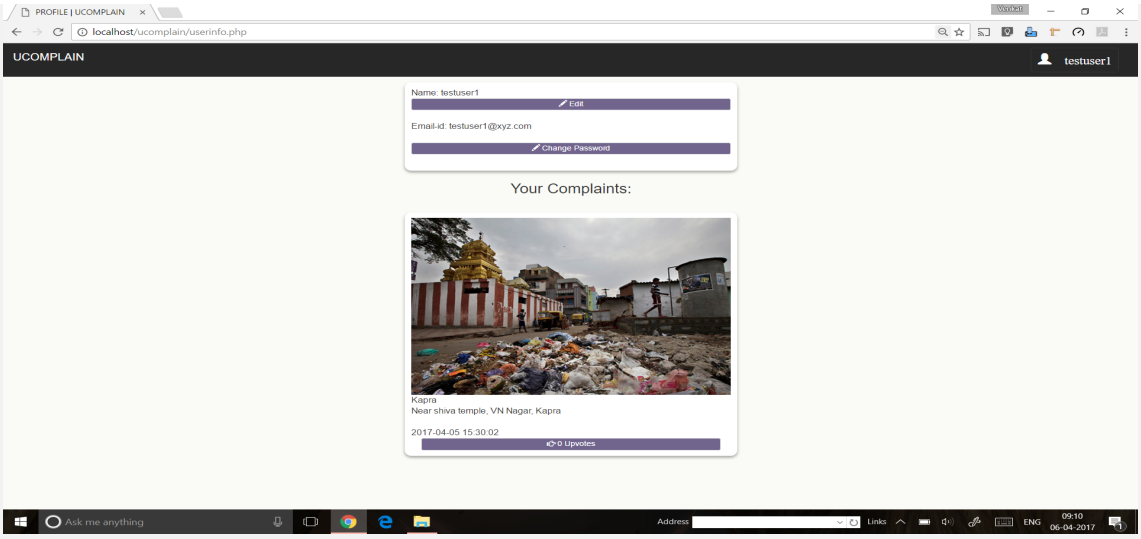
Screenshot 6.4.1 Main Website



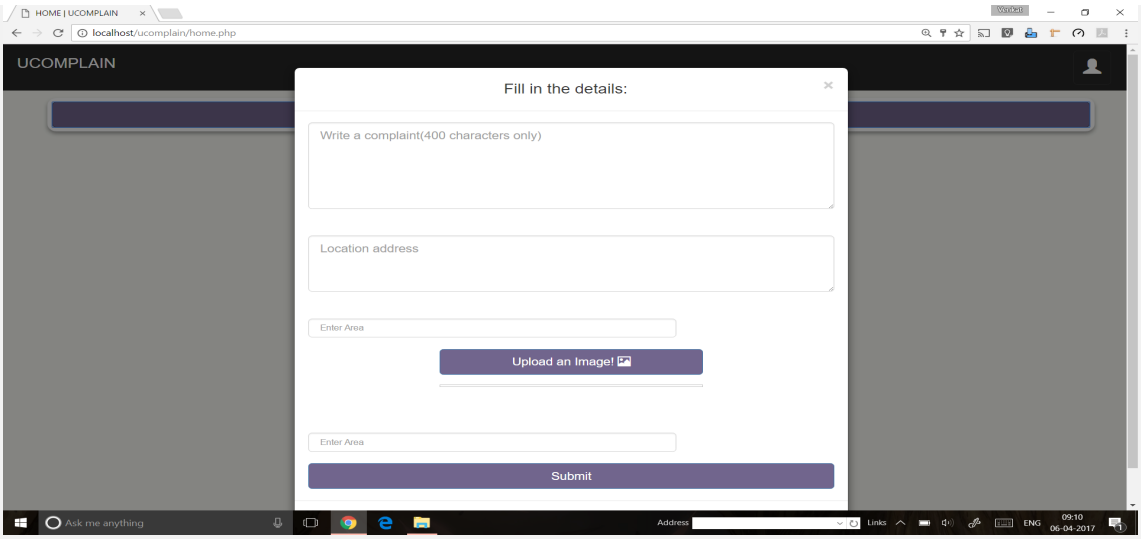
Screenshot 6.4.2 Login/Register Page



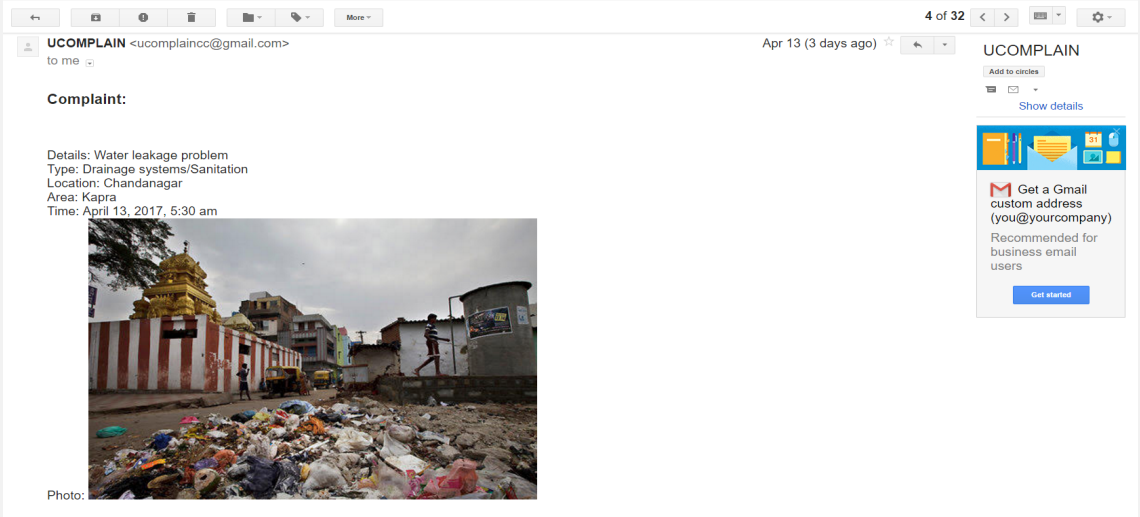
Screenshot 6.4.3 Home Page



Screenshot 6.4.4 Userinfo Page



Screenshot 6.4.5 Complaint Form



Screenshot 6.4.6 Email to the authorities

6.5 Database view

type	circle	c_incharge	c_in_email	zone	z_incharge	z_in_email
Roads	Kapra	Venkat	venkat.narahari96@gmail.com	East	Venkat	venkat.narahari96@gmail.com
Roads	Hayathnagar	Subash	subashjogu9999@gmail.com	East	Venkat	venkat.narahari96@gmail.com
Roads	LB Nagar	Vardhan	srivardhankandike@gmail.com	East	Venkat	venkat.narahari96@gmail.com
Roads	Uppal Kalan	Venkata Raju	kanumurivenkata@gmail.com	East	Venkat	venkat.narahari96@gmail.com
Roads	Saroornagar	Harsha	srikacharla@yahoo.co.in	East	Venkat	venkat.narahari96@gmail.com
Roads	Erstwhile1	Venkat	venkat.narahari96@gmail.com	South	Harsha	srikacharla@yahoo.co.in
Roads	Erstwhile2	Venkat Raju	kanumurivenkata@gmail.com	South	Harsha	srikacharla@yahoo.co.in
Roads	Erstwhile3	Subash	subashjogu9999@gmail.com	South	Harsha	srikacharla@yahoo.co.in
Roads	Rajendra Nagar	Venkat	venkat.narahari96@gmail.com	Central	Gajender	gajju7773@gmail.com
Roads	Erstwhile4	Venkat Raju	kanumurivenkata@gmail.com	Central	Gajender	gajju7773@gmail.com
Roads	Erstwhile5	Subash	subashjogu9999@gmail.com	Central	Gajender	gajju7773@gmail.com
Roads	Erstwhile6	Vardhan	srivardhankandike@gmail.com	Central	Gajender	gajju7773@gmail.com
Roads	Gachibowli	Venkat	venkat.narahari96@gmail.com	West	Surya	suryamopuru@gmail.com
Roads	Serilingampally	Venkat Raju	kanumurivenkata@gmail.com	West	Surya	suryamopuru@gmail.com
Roads	Ramachandra Puram/Patancheru	Subash	subashjogu9999@gmail.com	West	Surya	suryamopuru@gmail.com

Screenshot 6.5.1 Authorities Table

circle	location
Kapra	Kapra
Kapra	Dr AS Rao Nagar
Kapra	Cherlapalli
Kapra	Meerpet HB Colony
Kapra	Mallapur
Kapra	Nacharam
Uppal Kalan	Habsiguda
Uppal Kalan	Chilukanagar
Uppal Kalan	Ramanthapur
Uppal Kalan	Uppa
Hayathnagar	Nagole
Hayathnagar	Mansoorabad
Hayathnagar	Hayaat Nagar
Hayathnagar	BN Reddy Nagar
LB Nagar	Vanasthalipuram
LB Nagar	Hasthinapuram
LB Nagar	Champapet
LB Nagar	Lingojiguda
Saroornagar	Saroornagar

Screenshot 6.5.2 Areas to location table

username	email	password
testuser1	testuser1@xyz.com	testuser1
testuser2	testuser2@xyz.com	testuser2

Screenshot 6.5.3 User information table

complaintno	email	location	type	address	image	details	upvotes	ts	counter
3	testuser1@xyz.com	Kapra	Garbage Disposal	Near shiva temple, VN Nagar, Kapra	[BLOB - 65.3 KiB]	No garbage clearance in my locality.	0	2017-04-05 15:30:02	0

Screenshot 6.5.4 Complaints table

7. CONCLUSION

Conclusion

UCOMPLAIN which can be used to view and post complaints in online domain helps at better administration by officials through public views and their issues. This allows the removal of lot of red-tape unlike which is found in traditional complaint systems. The system is focused on from where the complaints are generated and how to forward them to respective officials providing an easy efficient pipeline. When a complaint is posted the message is forwarded to various levels of authorities based on the issue and hence gets a faster resolve. The others who can view the complaints can upvote them and show who are all facing the related issues.

Thus this creates a proper complaint managing and pipeline system available to general public. An efficient and faster resolving of public issues makes a better public administration and thus well being of the public.

8. FUTURE ENHANCEMENTS

Future Enhancements

Complaint management system have a great role in improving the flaws in any system. The scope of the project has a long unnoticed path ahead. The future enhancements that can be projected for the project are:

- A mobile application can be developed for the following purpose.
- More better designed user interface and also notification on how the complaints are being solved by the authorities.
- Data analytics and statistics can be shown about a particular area's administration and how the complaints are being handled.
- The complaint system can be deployed even at the private sector. Example: consumer electronics, food and beverages, etc.

9. REFERENCES

References

Textbooks:

- UML: The Unified Modeling Language User Guide by Grady Booch, James Rumbaugh, Ivar Jacobson - Pearson

- PHP: Beginning PHP and MySQL by W. Jason Gilmore - Dreamtech Press

- Bootstrap: Mastering Bootstrap 4 by Benjamin Jakobus, Jason Marah - Packt Publishing Limited

Web links:

- PHP: http://php.net

- HTML, CSS, PHP & SQL: https://www.w3schools.com/

- Boottrap: http://getbootstrap.com/