

CHAPTER-1

INTRODUCTION

1.1 About the Project

COMPLAINT MANAGEMENT SYSTEM is a web-based application that is used for maintain all the complaints of customers who are belongs to the particular communication network. It helps the customer to raise complaint about their issues like network issue, validity issue, deduction issue and value-added services issues and etc

This project allows customer to register a complaint and it helps to the customers to view the status of their complaints whether it is solved or pending.

This project consists of the following module.

1. Admin module
2. Registration module
3. Complaint module
 - Network issue
 - Validity issue
 - Deduction issue
 - Value added services
4. Process module
5. User dashboard

This project will help to telecom industry to maintain all the complaints of the customers in easy manner.

CHAPTER-2

SYSTEM STUDY

2.1 Existing system

The existing system is manual work-based system. If the customer had any complaint, they have to approach respective circle dealer and raise a complaint. That circle dealer is kept note of all these complaints and update them to higher officials. This the main problem here we are facing, first thing is the customer have to reach the office and meet the circle dealer, in case the dealer is busy and the customer won't have the chance of meet the circle dealer and raise a complaint. Second thing is if the complaint was raised with a dealer there is no assurance of that complaint reaches the higher officials or the respective department of the problem cause of manual system.

2.2 Disadvantages of existing system

- In the previous manual work-based system the information are maintained on a notebook.
- The Customer details and area of complaint and department details are also maintained manually.
- Customer did not know whether the complaint is solved or not.

2.3 Proposed system

The proposed complaint management system helps the user to register the complaint clearly and get the exact values from the user and might provide better solution for user and maintain the complaint details on database. Once the customer registered their complaint it will be reflected on their portal. The customer can able to check the process of their complaint on their respective dashboard by login to the system.

2.4 Advantages of proposed system

- User friendly.
- Reduce the processing cycle of manual work.

- Sudden reaction for registered complaint.
- Get details according to the type of complaint.

2.5 Problem Definition and Description

This system will reduce manual work and it will store all the complaints into the database and process them when they are registered. It will notify the customer whether the problem is solved or not. As complaint details are registered in a paper it has been a difficult task to maintain all the records for long time, but this system provides long last of data and provide access to authorized people. The customer should always be in control of the application and not the vice versa. The user interface should be consistent so that the customer can handle the application with ease and speed. The application should be visually, conceptually clear.

The modules and sub modules of the system

- **Admin module** has eight sub modules, it allows the admin to add new customer, new complaint, new employee, announcements and alter them.
- **Registration module** has no sub module it allows the customer to register their details into the system.
- **Complaint module** has four sub modules and it provide access to the customer to register their complaints into the system.
 - Network module is containing a list of network issues.
 - Deduction module contains a list of deduction issues.
 - Validity module contains a list of validity issues.
 - Value added services module is contain value added issues.
- **Process module** has no sub module it processes the complaint details which are given by the customer and send them to the respective department successfully.

- **User dashboard module** has no sub module it provides personal login portal for the user and view their details and their complaint details.

CHAPTER-3

SYSTEM ANALYSIS

3.1 Package selected

- Front End: PHP
- Back End: MySql Database

3.2 Resources required

3.2.1 Hardware sources

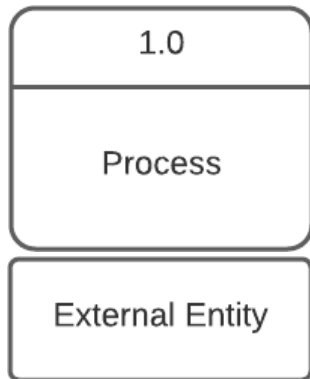
- Processor: Intel® Core™ i3-6006U CPU @ 2.00GHz 2.00GHz
- Memory: 2 GB
- Disk space: 1.5 GB of free disk space
- Screen Resolution: 1024 X 768

3.2.2 Software Resources

- Operating System: Windows 10
- Software: Visual Studio Code
- DataBase: My Sql Database

3.3 Data flow Diagram

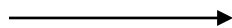
Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.



A process is a business activity or function where the manipulation and transformation of data take place. A process can be decomposed to a finer level of details, for representing how data is being processed within the process.



An external entity can represent a human, system or subsystem.



Data stores hold information for later use, like a file of documents that's waiting to be processed.

Data flow is the path the system's information takes from external entity to process and data stores.

Level 0

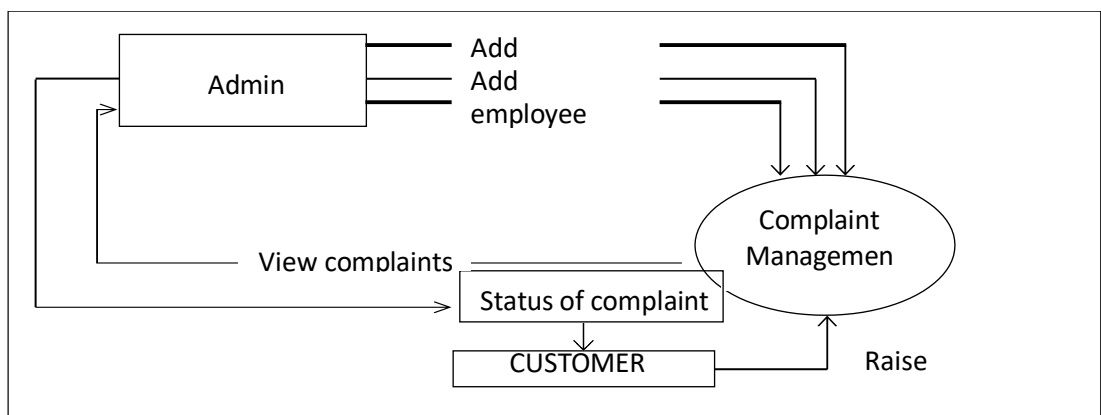


Fig. 3.1 Complaint Data

Level 1

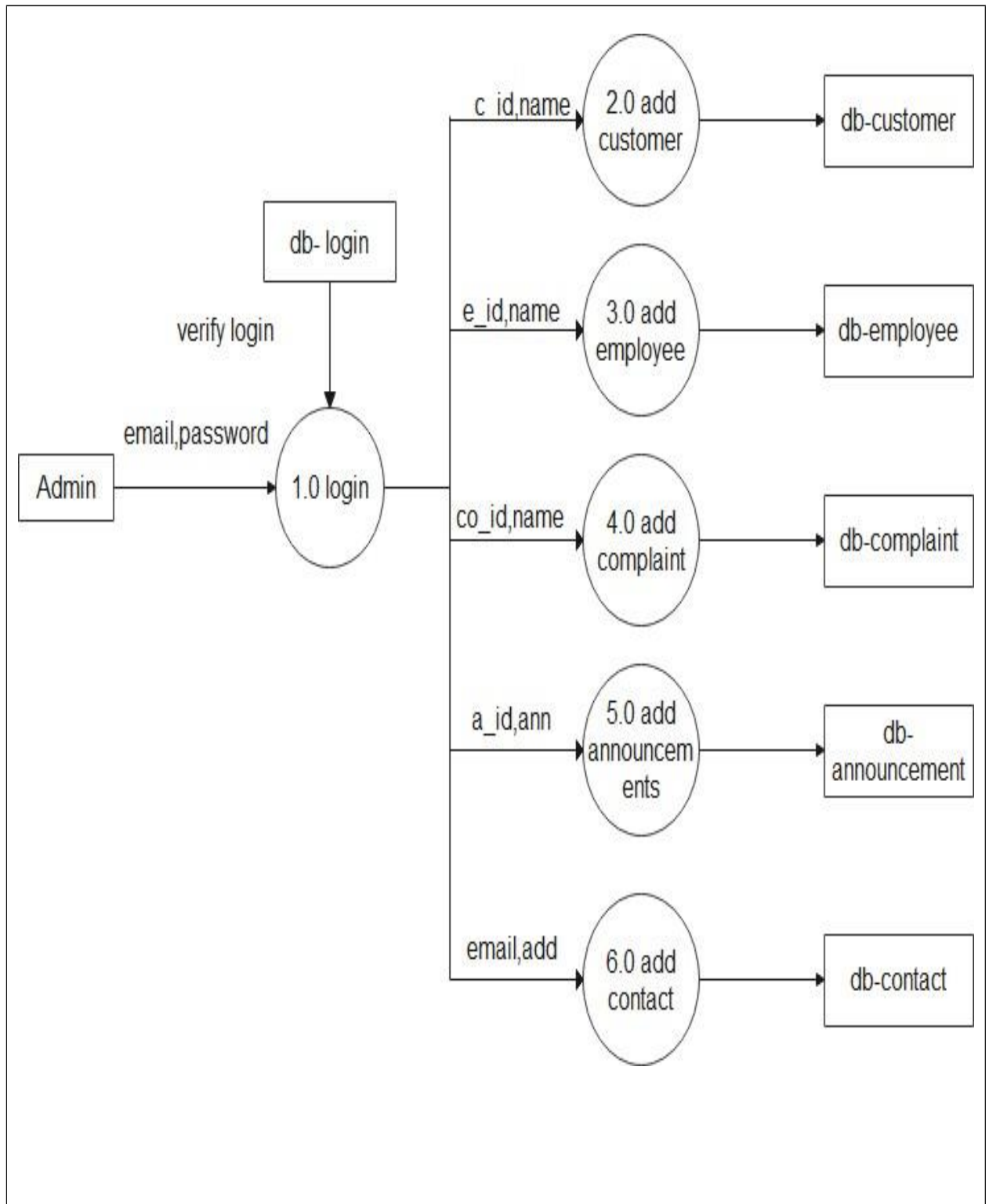


Fig. 3.2 Admin Data

3.4 Use Case Diagram

A Use Case Diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

- Use cases: Horizontally shaped ovals that represent the different uses that a user might have.
- Actors: Stick figures that represent the people actually employing the use cases
- Associations: A line between actors and use cases.

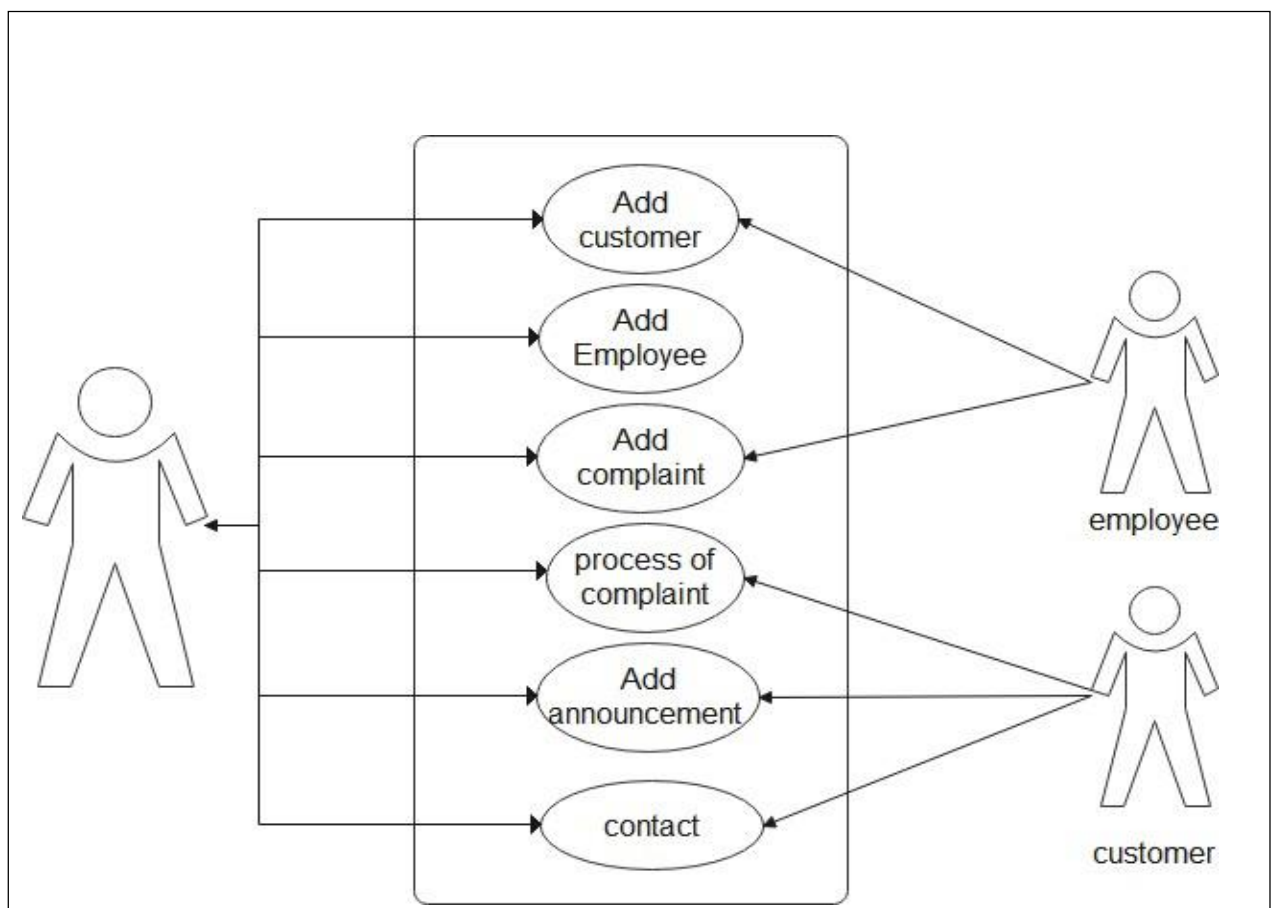


Fig. 3.2 Data Flow Diagram

CHAPTER-4

SYSTEM DEVELOPMENT

4.1 Functional Documentation

The “COMPLAINT MANAGEMENT SYSTEM” is a web based application developed for telecom industry. The main objective of the system is to maintain all the complaints of the particular telecommunication network and process of their complaints and maintain them. It will helps to the industry to maintain all the customer complaints and process their complaints according to their department.

This system contains following modules

1. Admin module
2. User dashboard
3. Complaint module
4. Registration module
5. Contact module

4.1.1 Admin module

This module is one of high priority module and it provides the access to the admin and by this module only the details of customer, complaint, employee, raised complaint and feedbacks can be altered.

4.1.2 User dashboard

This module contains user details and process of their complaint that is whether the raised complaint is solved or not and with this module customer only views their data and they can't alter the data.

4.1.3 Complaint module

The Complaint module contains list of complaints fixed by the admin and with this module only the customer can raise their complaints. A list of complaints is available in the following sub modules network, deduction, validity and value added.

4.1.4 Registration module

By this module customer register themselves with a system and login to the system. When the email id of the customer already available and that is detected by the system the customer won't register with that email.

4.1.5 Process module

From this module the complaints are processed and move to the respective department and after completion that is reflected on the user dashboard. The process module will check the complaint's department and send them to the respective departments.

4.2 Special features of the language

4.2.1 PHP

The main features of PHP are it is open source scripting languages so you can free download this and use. PHP is a server side scripting language. It is open source scripting language. It is widely used all over the world. It is faster than other scripting language.

4.2.2 SIMPLE:

It's popularly known for its simplicity, familiarity and easy to learn the language as the syntax is similar to that of C or Pascal language. So the language is very logical and well organized general purpose programming language. Even people with a normal programming background can easily understand and capture the use of language. PHP is very advantageous for new users as it's a very reliable fluent, organized, clean, demandable and efficient.

4.2.3 Loosely Typed Language:

PHP encourages the use of variables without declaring its data types. So this taken care at the execution tie depending on the value assigned to the variable. Even the variable name can be changed dynamically.

4.2.4 Flexibility:

PHP is known for its flexibility and embedded natures as it can be well integrated with HTML, XML, JAVASCRIPT and many more. PHP can run on multiple operating systems like Windows, UNIX, Mac, OS, Linux, etc. The PHP scripts can easily run on any device like laptops, mobiles, tables and copter. It is very comfortable integrated with various Databases. The executable PHP can also be run on command-line as well as directly on the machine.

4.2.5 Open Source:

All PHP frameworks are open sources, no payment is required for the users and it's completely free. User can just download PHP and start using for their applications or projects. Even in companies, the total cost is reduced for software development providing ore reliability and flexibility.

4.2.6 Cross-Platform Compatibility:

PHP is multi-platform and known for its portability as it can run on any operating system and windows environments. The most common are XAMPP and LAMP. As PHP is platform-independent, it's very easy to integrate with various databases and other technologies with various databases and other technologies without reimplementatation. It effectively saves a lot energy, time and money.

4.2.7 Error Reporting and Exception:

PHP supports much error reporting constants to generate errors and relevant warnings at run time. PHP5 supports exception handling which is used to throw errors which can be caught at any time.

4.2.8 Active Community Support:

PHP is very rich with many diverse online community developers to help beginners for web-based application. These worldwide volunteers contribute many features as well as new version for PHP libraries. Even that contributes a translation in different languages to help out programmers. There is a bundle of third-party open-source libraries which provide basic functionalities. Even the documentation given by the official site helps in implementing new features providing access to a variety of creative imagination.

4.2.9 Fast and Efficient Performance:

Users generally prefer fast loading websites. For any web development, speed becomes an important aspect which is taken care of by PHP. PHP scripts are faster than other scripting languages like ASP.NET, PERL and JSP. The memory manager of PHP7 is very optimized and fast as compared to older versions of PHP. Even connecting to the database and loading of required data from tables are faster than other programming languages. It provides a built in module for easy and efficient database management system. The high speed of PHP is advantageous for users for its server administration and mail functionality. Also, it supports session management and removing of unwanted memory allocation.

4.2.10 Features of MYSQL:

The following list shows the most important properties of MYSQL.

4.2.11 Relational Database System:

Like almost all other database system on the market, MYSQL is a relational database system.

4.2.12 Client/Server Architecture:

There is a database server (MYSQL) and arbitrarily many clients, which communicate with the server that is, that query data, save changes, etc. The clients can run on the same computer as the server or on another computer (communication via a local network or the internet). Almost all of the familiar large database system

(Oracle, Microsoft SQL Server, etc.) are client/server systems. These are in contrast to the file-server systems, which include Microsoft Access, dBase and FoxPro. The decisive drawback to file-server system is that when run over a network, they become extremely inefficient as the number of users grows.

4.2.13 SQL Compatibility:

MYSQL supports as its database language as its name suggests – SQL (Structured Query Language). SQL is a standardized language for querying and updating data and for administration of a database. There are several SQL dialects. MYSQL adheres to the current SQL standard, although with significant restrictions and a large number of extensions. Though the configuration setting sql-mode you can make the MYSQL server behave for the most part compatibly with various database systems. Among these are IBM DB/2 and Oracle.

4.2.14 Sub SELECT:

Since version 4.1 MYSQL is capable of processing a query in the form
SELECT * FROM table1 WHERE x IN (SELECT y FROM table2).

4.2.15 Views:

Put simply, views relate to an SQL query that is viewed as a distinct database object and makes possible particular view of the database. MYSQL has supported views since version 3.23.0.

4.2.16 Stored Procedures:

Here we are dealing with SQL code that is stored in the database system. Stored procedures are generally used to simplify certain steps, such as inserting or deleting a data record. For client programmers this has the advantage that they did not have to process the tables directly, but can rely on SPs. Like views, SPs help in the administration of larger database projects. SPs can also increase efficiency. MYSQL has supported SPs since version 3.23.0.

4.2.17 Triggers:

Triggers are SQL commands that are automatically executed by the server in certain database operation. MYSQL has supported triggers in a limited form from version 5.0 and additional functionality is promised for version 5.1.

4.2.18 Unicode:

MYSQL has supported all conceivable character sets since version 4.1, including Latin1, Latin-2 and Unicode.

4.2.19 Transactions:

In the context of a database system, a transaction means the execution of several database operations as a block. The database system ensures that either all of the operations are correctly executed or none of them. This holds even if in the middle of a transaction there is a power failure, the computer crashes or some other disaster occurs. Transactions also give programmers that possibility of interrupting a series of already executed commands.

CHAPTER-5

SYSTEM DESIGN

5.1 Architecture Design

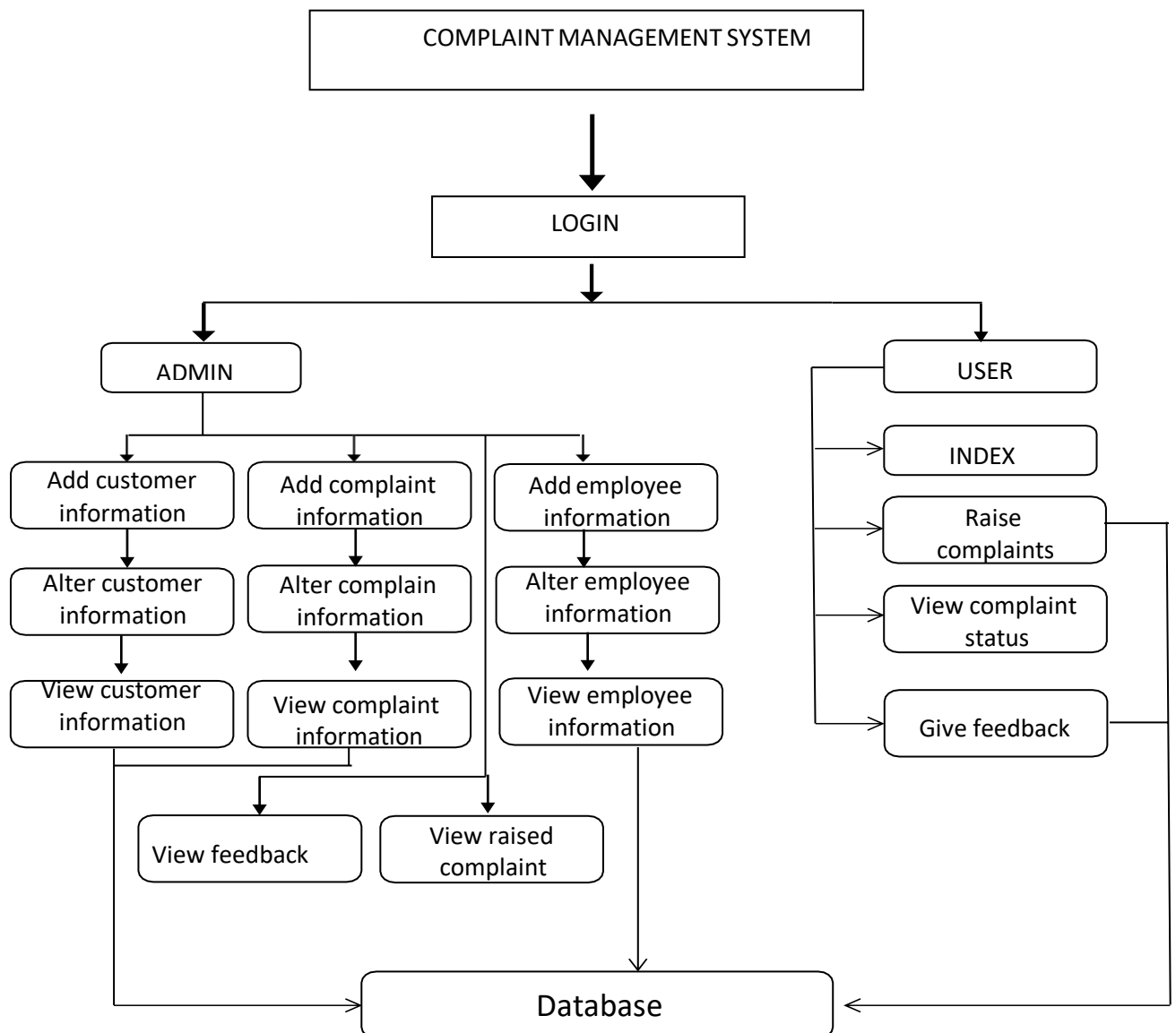


Fig. 5.1 System Architecture

5.2 I/O Form Design

5.2.1 Registration form

• Name	:	<input type="text" value="MANI"/>
• Email	:	<input type="text" value="mani@gmail.com"/>
• Phone	:	<input type="text" value="8754759915"/>
• Address	:	<input type="text" value="Bangalore"/>
• Password	:	<input type="text" value="Mani1234"/>
• Confirm password	:	<input type="text" value="Mani1234"/>
		<input type="text" value="4G"/>
<input type="button" value="Register"/>		

Fig. 5.2 Consumer Registration Form

5.2.2 Complaint registration form

• Name of the:	<input type="text" value="No network"/>	
• complaint:	<input type="text" value="No network in mobile"/>	
• departm	:	<input type="text" value="net"/>
<input type="button" value="Register"/>		

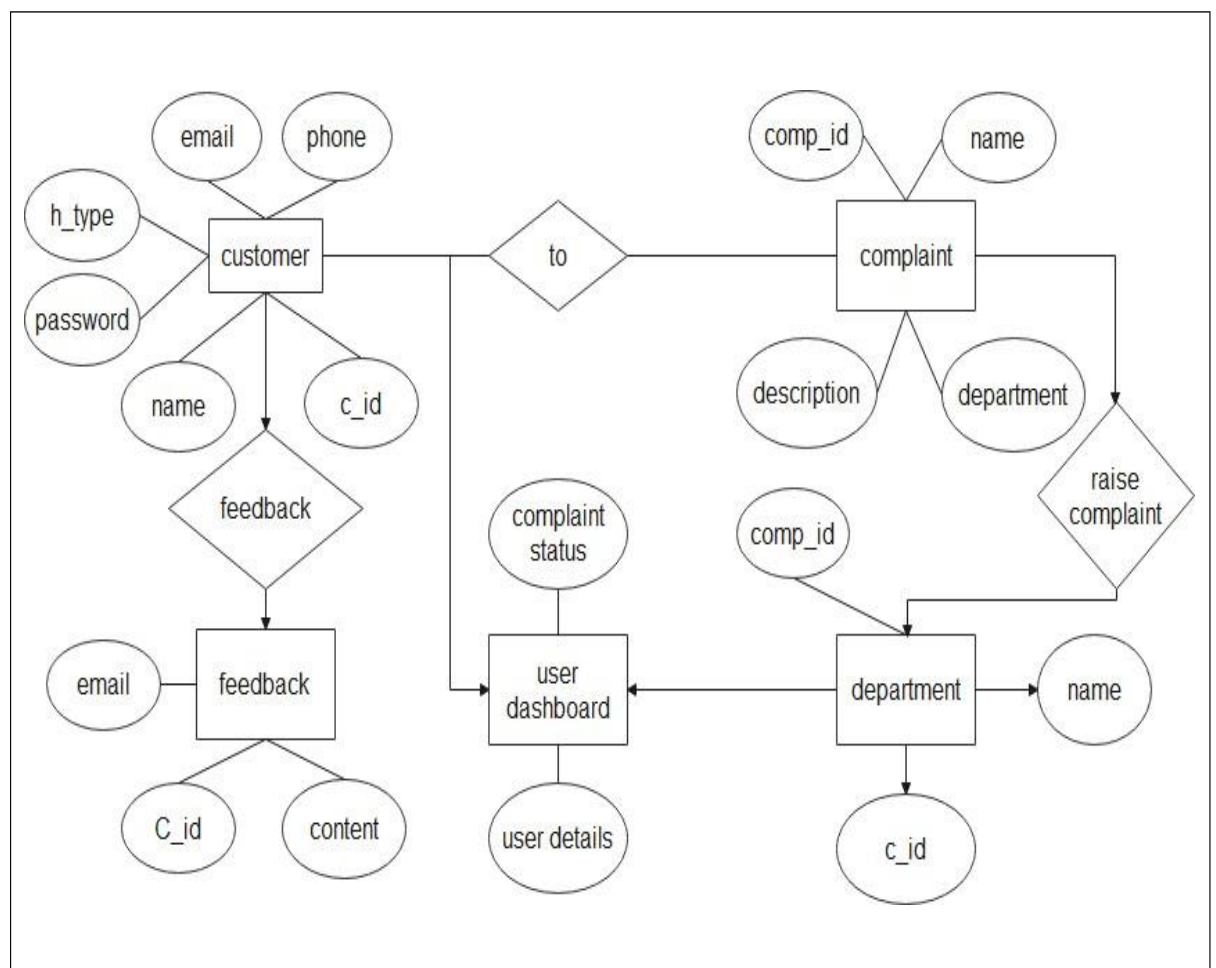
Fig. 5.3 Complaint Form

5.3 Entity Relationship Diagram

An entity–relationship model is the result of using a systematic process to describe and define a subject area of business data. It does not define business process; only visualize business data. The data is represented as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. The three schema approach to software engineering uses three levels of ER models that may be developed.

ER DIAGRAM

Fig. 5.4 Entity Relationship Diagram



CHAPTER-6

SYSTEM IMPLEMENTATION

6.1 TOOLS AND LANGUAGES USED:

Frontend: Html, PHP, CSS software used for this is Notepad++

Backend: Xampp Controller v3.2.2

Database Connectivity: PhpMyAdmin

Languages used:

1. SQL
2. PHP HTML
3. CSS

6.2 IMPLEMENTATION DETAILS:

6.2.1 HTML



Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured

documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

6.2.2 CSS



Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in

a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium. The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. The process is called cascading.

6.2.3 PHP



PHP is a server-side scripting language that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages. PHP scripts can only be interpreted on a server that has PHP installed. The client computers accessing the PHP scripts require a web browser only. A PHP file contains PHP tags and ends with the extension ".php".

The term PHP is an acronym for PHP: Hypertext Pre-processor. PHP is a server-side scripting language designed specifically for web development. PHP can be easily embedded in HTML files and HTML codes can also be written in a PHP file. The thing that differentiates PHP with client-side language like HTML is, PHP codes are executed on the server whereas HTML codes are directly rendered on the browser.

PHP: Hypertext Pre-processor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994. PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common

Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control.

6.2.4 MY SQL



MySQL is an open-source relational database management system (RDBMS) based on Structured Query Language (SQL). It is one part of the very popular LAMP platform consisting of Linux, Apache, My SQL, and PHP. Currently My SQL is owned by Oracle. My SQL database is available on most important OS platforms. It runs on BSD Unix, Linux, Windows, or Mac OS. Wikipedia and YouTube use My SQL. These sites manage millions of queries each day. My SQL comes in two versions: My SQL server system and My SQL embedded system.

RDBMS Terminology

Before we proceed to explain MySQL database system, let's revise few definitions related to database.

- **Database:** A database is a collection of tables, with related data.
- **Table:** A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- **Column:** One column (data element) contains data of one and the same kind, for example the column postcode.
- **Row:** A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.

- **Redundancy:** Storing data twice, redundantly to make the system faster.
- **Primary Key:** A primary key is unique. A key value cannot occur twice in one table. With a key, you can find at most one row.
- **Compound Key:** A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- **Index:** An index in a database resembles an index at the back of a book.
- **Referential Integrity:** Referential Integrity makes sure that a foreign key value always points to an existing row.

6.3 Source Code

Index.php

```
<?php

include'connection.php';

$mysqli= new mysqli('localhost','root','','cms');

session_start();

if (!isset($_SESSION['id'])) {

    header("Location: login.php");

}

else

{

    $a=$_SESSION['id'];

}

?>

<!DOCTYPE HTML>

<html>
```

```
<head>

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></scr
ipt>

</head>

<title>

    welcome to online complaint management sytem

</title>

<body>

<div class="container-fluid " style="margin-top:70px;">

    <?php include'header.php';?></div>

    <style>

        <?php include'main.css';?>

    </style>

    <div class="container-fluid navbar-fixed-top" style="margin-top:20px">

        <?php include'nav.php';?>

    </div>

    <form>

        <div class="row1">

            <div class="columnan1">
```

```
<a href="comp_home.php">

</img></a><P align="center">RAISE COMPLAINT</P>

</div>

</div>

</form>

<script src="js/bootstrap.min.js">

<script src="js/jquery.min.js">

</body>

</html>
```

Login.php

```
<?php

include'connection.php';

$mysqli= new mysqli('localhost','root','','cms');

session_start();

$a=("")

?>

<!DOCTYPE HTML>

<html>

<center>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
```



```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></scr
ipt>

</head>

<body>

<div class="container-fluid fixed-top" style="margin-top:50px;">

<?php include'header.php';?></div>

<style>

<?php include'main.css';?>

</style>

<div class="container-fluid navbar-fixed-top" style="margin-top:50px">

</div>

<form method="POST" class="responsive" style="margin-top:50px"
action="login_2.php?id="loginf"><br>

<div class="row1">

<div class="column">

<table id="login" align="center">

<tr>

<td>



</img>

</td>
```

```
</tr>

</table>

</div>

<div class="column1">

<table border="0" id="login">

    <th colspan="3" class="th1">

        LOGIN

    </th>

    <tr class="row1">

        <td class="c1">

            <label>USER NAME</label>

        </td>

        <td class="c2">

            :

        </td>

        <td class="c3">

            <input

type="email" name="uname" required

                                placeholder="Enter the username

here" title="Enter the user name">

        </td>

    </tr>

    <tr class="row1">

        <td class="c1">
```



```
</tr>

</table>

</div>

</div>

</form>

<script src="js/bootstrap.min.js">

<script src="js/jquery.min.js">

</body>

</center>

</html>

<?php

session_start();

include'connection.php';

if(isset($_POST['login']))

{

    $email=$_POST['uname'];

    $password=$_POST['pass'];

    $sel="SELECT * FROM log WHERE email='$email' &&

password='$password'";

    $table=mysqli_query($link,$sel);

    $row=mysqli_num_rows($table);

    $usel="SELECT * FROM customer WHERE email='$email' &&

password='$password'";
```

```
$utable=mysqli_query($link,$usel);

$urow=mysqli_num_rows($utable);

if($row==1)

{

    $_SESSION['id']=$email;

    header("location:admin.php");

}

elseif($urow==1)

{

    $_SESSION['id']=$email;

    header("location:index.php");

}

else{

    echo "<script>alert('Invalid Email and Password')</script>";

    echo "<script>window.location='login.php'</script>";

}

}

?>
```

Register.php

```
<?php

include'connection.php';

mysqli= new mysqli('localhost','root','','cms');

?>
```

```
<html>

<center>

  <title>

</title>

<head>

  <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></scr
ipt>

</head>

  <title>

    registration form

  </title>

<body>

  <div class="container-fluid " style="margin-top:70px;">

    <?php include'header.php';?></div>

    <style>

      <?php include'css_register.css';?>

      <?php include'main.css';?>

    </style>
```

```
<div class="container-fluid navbar-fixed-top" style="margin-top:20px">

<?php include'nav.php';?>

</div>

<form method="POST" name="register" onsubmit="return
validateform()">

<table border="0">

<th colspan="3">NEW CUSTOMER</th>

<tr>

<td class="l"><label class="a1">CUSTOMER
NAME</label></td>

<td class="r"><input
type="text"name="name"required placeholder="ENTER THE CUSTOMER
NAME HERE" title="ENTER THE CUSTOMER NAME HERE"></td>

</tr>

<tr>

<td class="l"><label class="a1">MOBILE
NUMBER</label></td>

<td class="r"><input type="text"name="mno"
required placeholder="ENTER THE CUSTOMER MOBILE NUMBER HERE"
title="ENTER THE CUSTOMER MOBILE NUMBER HERE"/></td>

</tr>

<tr>

<td class="l"><label class="a1">E-MAIL
ID</label></td>
```

```
<td class="r"><input
type="email"name="email"required placeholder="ENTER THE CUSTOMER
EMAIL ID HERE" title="ENTER THE CUSTOMER EMAIL ID
HERE"/></td>
```

```
</tr>
```

```
<tr>
```

```
<td class="l"><label
class="a1">ADDRESS</label></td>
```

```
<td class="r"><input
type="text"name="address"required placeholder="ENTER THE CUSTOMER
ADDRESS HERE" title="ENTER THE CUSTOMER ADDRESS
HERE"/></td>
```

```
</tr>
```

```
<tr>
```

```
<td class="l"><label class="a1">HANDSET
TYPE </label></td>
```

```
<td class="r"><input type="radio"name="type"
value="2G"/>2G
```

```
<input type="radio"name="type"
value="3G"/>3G
```

```
<input type="radio"name="type" value="4G"
title="ENTER THE TYPE OF DEVICE HERE">4G</td>
```

```
</tr>
```

```
<tr>
```



```

<td class="l"><label class="a1">SELECT
NETWORK</label></td>

<td>

<select name="network">

<option selected hidden
value="">SELECT NETWORK</option>

<option
value="AIRTEL">AIRTEL</option>

<option value="JIO">JIO</option>

<option value="VI">VI</option>

<option value="BSNL">BSNL</option>

</select>

<td>

</tr>

<tr>

<td class="l"><label
class="a1">PASSWORD</label></td>

<td class="r"><input
type="password" name="pass" required placeholder="ENTER THE
PASSWORD HERE" title="ENTER THE PASSWORD HERE"/></td>

</tr>

<tr>

<td class="l"><label class="a1">CONFIRM
PASSWORD</label></td>

```

> | <label |

```
function validateform(){
```

```
var password=document.register.pass.value;
```

```
if (name==null || name==""){
    alert("Name can't be blank");
    return false;
}

else if(password.length<6){
    alert("Password must be at least 6 characters long.");
    return false;
}
} </script>
</html>

<?php
include'connection.php';

if(isset($_POST["reguser"]))
{
    $name  =$_POST['name'];
    $m_no  =$_POST['mno'];
    $email =$_POST['email'];
    $add   =$_POST['address'];
    $htype =$_POST['type'];
    $network=$_POST['network'];
    $pass  =$_POST['pass'];
    $rpass =$_POST['rpass'];

    $sel="SELECT * FROM customer WHERE email='$email'";
```

```
$match=mysqli_query($link,$sel) or die(mysqli_error($link));

$match1=mysqli_num_rows($match);

$z=date('E');

$y=('11');

$example=mysqli_query($link,"SELECT c_id FROM customer ");

$ex=mysqli_num_rows($example);

$invID = str_pad($ex,3,'0',STR_PAD_LEFT);

$c_id=$z.$y.$invID;

if($pass==$rpass){

if ($match1 != 0){

        echo '<script>alert("CUSTOMER ALREADY EXISTS")</script>';

    }

elseif ($match1 == 0)

    {

mysqli_query($link,"insert into customer

values('$c_id','$name','$m_no','$email','$add','$htype','$network','$pass')") or

die(mysqli_error($link));

        echo '<script>alert("SUCCESSFULLY REGISTERED")</script>';

        $_SESSION['id']=$email;

    }

}

else

{

        echo '<script>alert("PASSWORD IS NOT MATCH")</script>';

    }
```

}

}

?>

cms.sql

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

```
SET AUTOCOMMIT = 0;
```

```
START TRANSACTION;
```

```
SET time_zone = "+00:00";
```

```
--
```

```
-- Database: `cms`
```

```
-- Table structure for table `announce`
```

```
--
```

```
CREATE TABLE `announce` (
```

```
  `announce_id` varchar(100) NOT NULL,
```

```
  `type` varchar(100) NOT NULL,
```

```
  `content` varchar(200) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-- Dumping data for table `announce`
```

```
INSERT INTO `announce` (`announce_id`, `type`, `content`) VALUES
```

```
('AN66000', 'for login', 'login.php'),
```

```
('AN66001', 'put feedback ', 'feedback.php');
```

```
-- Table structure for table `customer`
```

```
CREATE TABLE `customer` (
```

```
`c_id` varchar(100) NOT NULL,

`name` varchar(200) NOT NULL,

`m_no` bigint(15) NOT NULL,

`email` varchar(200) NOT NULL,

`address` varchar(200) NOT NULL,

`h_type` enum('2g','3g','4g') NOT NULL,

`password` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table `customer`

INSERT INTO `customer` (`c_id`,`name`,`m_no`,`email`,`address`,`h_type`,`password`) VALUES

('8098135709','mani ', 8098135709, 'mani@gmail.com', 'bangalore', '3g', 'mani'),

('8765876578','PETER', 8765876578, 'peter@gmail.com', 'bangalore', '',

'peter123'),

('89898980098','peter1', 89898980098, 'peter1@gmail.com', 'jsks', '', 'peter123'),

('9090909090','nisheeth', 9090909090, 'nish@gmail.com', 'kanniyakumari', '',

'nish123'),

('E11003','peter1', 89898980098, 'peter1@gmail.com', 'jsks', '4g', 'peter123'),

('E11004','', 0, '', '', '', ''),

('E11006','kss', 0, 'iss@kjsk.slks', 'skjsj', '', 'kikiki00'),

('E11007','kss', 0, 'iss@kjsk.slks', 'skjsj', '', 'kikiki00'),

('E11008','', 0, '', '', '', '');

-- Table structure for table `employee`

CREATE TABLE `employee` (
```

```
`E_id` varchar(20) NOT NULL,  
  
`name` varchar(200) NOT NULL,  
  
`email` varchar(200) NOT NULL,  
  
`phone` bigint(15) NOT NULL,  
  
`address` varchar(1000) NOT NULL,  
  
`pass` varchar(200) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
-- Dumping data for table `employee`  
  
INSERT INTO `employee` (`E_id`, `name`, `email`, `phone`, `address`, `pass`)  
VALUES  
  
('E33002', 'gjj', 'rwrew', 0, 'jjds', 'kikikikiki');  
  
-- Table structure for table `log`  
  
CREATE TABLE `log` (  
  
  `id` int(11) NOT NULL,  
  
  `email` varchar(50) NOT NULL,  
  
  `password` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
-- Dumping data for table `log`  
  
INSERT INTO `log` (`id`, `email`, `password`) VALUES  
  
(1, 'mani@gmail.com', 'mani');  
  
-- Table structure for table `logu`  
  
CREATE TABLE `logu` (  
  
  `id` varchar(100) NOT NULL,  
  
  `email` varchar(200) NOT NULL,
```

```
`pass` varchar(200) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `network`

CREATE TABLE `network` (

  `net_id` varchar(100) NOT NULL,

  `reg_id` varchar(100) NOT NULL,

  `complaint_id` varchar(100) NOT NULL,

  `user_id` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table `network`

INSERT INTO `network` (`net_id`, `reg_id`, `complaint_id`, `user_id`)
VALUES

('NET66000', 'UC44000', 'COM99000', 'peter@gmail.com');

-- Table structure for table `net_com`

CREATE TABLE `net_com` (

  `com_id` varchar(20) NOT NULL,

  `name` varchar(300) NOT NULL,

  `description` varchar(300) NOT NULL,

  `department` varchar(30) NOT NULL DEFAULT 'network'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table `net_com`

INSERT INTO `net_com` (`com_id`, `name`, `description`, `department`) VALUES

('COM99000', 'slow data', 'network speed is too slow', 'network'),

('COM99001', 'balance deduction', 'suddenly balance has drop', 'deduction'),
```



```
('COM99002', 'caller tune', 'de activate caller tune', 'valad'),

('COM99003', 'no network', 'network is not available', 'network'),

('COM99004', 'indoor network slow', 'network speed is poor while inside of
bildeing', '');

-- Table structure for table `raised_com`

CREATE TABLE `raised_com` (

  `ucom_id` varchar(100) NOT NULL,

  `com_id` varchar(100) NOT NULL,

  `user_id` varchar(200) NOT NULL,

  `department` varchar(200) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table `raised_com`

INSERT INTO `raised_com` (`ucom_id`, `com_id`, `user_id`, `department`)
VALUES

('UC44000', 'COM99000', 'peter@gmail.com', 'network');

-- Table structure for table `validity`

CREATE TABLE `validity` (

  `val_id` varchar(100) NOT NULL,

  `reg_id` varchar(100) NOT NULL,

  `complaint_id` varchar(100) NOT NULL,

  `user_id` varchar(200) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `value_added`

CREATE TABLE `value_added` (
```

```
`va_id` varchar(100) NOT NULL,  
  
`reg_id` varchar(100) NOT NULL,  
  
`complaint_id` varchar(100) NOT NULL,  
  
`user_id` varchar(200) NOT NULL  
  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
-- Indexes for dumped tables  
  
-- Indexes for table `announce`  
  
ALTER TABLE `announce`  
  
    ADD PRIMARY KEY (`announce_id`);  
  
-- Indexes for table `customer`  
  
--ALTER TABLE `customer`  
  
    ADD PRIMARY KEY (`c_id`,`email`);  
  
-- Indexes for table `employee`  
  
ALTER TABLE `employee`  
  
    ADD PRIMARY KEY (`E_id`),  
  
    ADD UNIQUE KEY `email` (`email`);  
  
-- Indexes for table `log`  
  
ALTER TABLE `log`  
  
    ADD PRIMARY KEY (`id`);  
  
-- Indexes for table `logu`  
  
ALTER TABLE `logu`  
  
    ADD PRIMARY KEY (`email`);  
  
-- Indexes for table `network`  
  
ALTER TABLE `network`
```

```
ADD PRIMARY KEY (`net_id`);

-- Indexes for table `raised_com`

ALTER TABLE `raised_com`

ADD PRIMARY KEY (`ucom_id`);

-- Indexes for table `value_added`

ALTER TABLE `value_added`

ADD PRIMARY KEY (`reg_id`);

-- AUTO_INCREMENT for dumped tables

-- AUTO_INCREMENT for table `log`

ALTER TABLE `log`

MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=2;

COMMIT;
```

CHAPTER-7

SYSTEM TESTING

7.1 Unit testing

Unit Testing is a software testing technique by means of which individual units of software i.e group of computer program modules, usage procedures and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which every independent module is tested to determine if there is any issue by the developer himself. It is correlated with functional correctness of the independent modules. Unit testing is defined as a type of software testing where individual components of a software are tested. Unit Testing of software product is carried out during the development of an application. An individual component may be either an individual function or a procedure. Unit testing is typically performed by the developer.

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Fig. 7.1 Unit Testing 1

Test Id	UT01
Unit Tested	To test add Customer
Purpose	To check whether details are added or Not
Test Status	Record Inserted
Test Result	Pass

Fig. 7.2 Unit Testing 2

Test Id	UT02
Unit Tested	To test update Customer
Purpose	To check whether Customer details updated or not.
Test Status	Record Updated
Test Result	Pass

7.2 Validation Testing

Validation testing is the process of ensuring if the tested and developed software satisfies the client/user needs. The business requirement logic or scenarios have to be tested in detail. All the critical functionalities of an application must be tested here.

Fig. 7.3 Validate Testing

Test Id	VT01
Unit Tested	To test Add Customer
Purpose	To check whether customer details is already exists or not
Test Data	Customer ID
Test Status	Customer already Exists
Test Result	Pass

7.3 Module Tested: User Access

In this test case, we test the login credentials of the user by providing login ID and password. If the access is provided for the credentials, then the test case is passed, if the access is denied, then the test case fails.

Fig. 7.4 Log IN Testing

Test Case	MT01
Name of the Test	Working of Login Credentials
Module being tested	User access
Sample input	Login Credentials
Expected output	Should open home page
Actual output	Home page is accessed
Remarks	Pass

CHAPTER-8

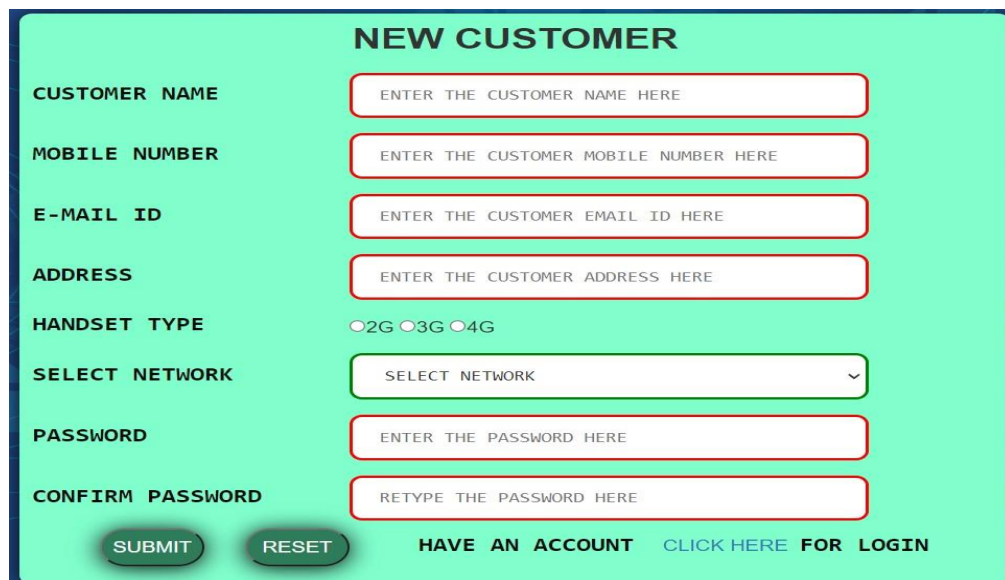
RESULT

Fig. 8.1 Log IN Page of Complain Management System



Shows the login page through which you can login to either admin or customer accounts.

Fig. 8.2 Registration for new Consumer



Shows the registration page for new user can register in the portal

Fig. 8.3 User Details

CREATE USER DETAILS			UPDATE USER DETAILS		DELETE USER DETAILS		DISPLAY USER DETAILS	
ID	NAME	MOBILE NUMBER	EMAIL	ADDRESS	HANDSET TYPE	NETWORK	PASSWORD	
8754759915	mani	8098135709	mani@gmail.com	bangalore	3g		mani	
8765876578	PETER	8765876578	peter@gmail.com	bangalore			peter123	
89898980098	peter1	89898980098	peter1@gmail.com	jsks			peter123	
9090909090	nisheeth	9090909090	nish@gmail.com	kanniyakumari			nish123	
E11003	peter1	89898980098	peter1@gmail.com	jsks	4g		peter123	
E11004		0						
E11006	kss	0	iss@kjsk.silks	skjsj			kikiki00	
E11007	kss	0	iss@kjsk.silks	skjsj			kikiki00	
E11008		0						
E11009	Ashik	8754759912	ashik@gmail.com	Thiruvapur			ashik123	
E11010	MANIMARAN	8754759915	manimaran1209mmm@gmail.com	ANDAVOORANI POST OFFICE	3g		maran1	
E11011	abi	9980865432	abi@gmail.com	Amaravathi	4g		abi123	
E11012	Raed	9807654456	raed@gmail.com	Kerala	4g	JIO	raed123	
E11013	Jesly	8765432190	jesly@gmail.com	Goa	4g	JIO	jesly123	
E11014	Gowtham	9876543210	gowtham@gmail.com	Bangalore	4g	AIRTEL	gow123	

Shows the list of User Details

Fig. 8.4 Complaint Customize

The screenshot shows a web application titled "ONLINE COMPLAINT MANAGEMENT SYSTEM". The interface includes a navigation bar with links: HOME, COMPLAINTS, MYQUIRIES, ABOUT US, LOGOUT, and FAQ. Below the navigation bar, there are tabs for different complaint categories: NETWORK, DEDUCTION, VALIDITY, VALUE ADDED SERVICES, and OTHERS. The "COMPLAINTS" tab is currently selected. In the center, there is a form to create a new complaint. The form has two input fields: "Complaint_name" and "Description". Below these fields are two buttons: "SUBMIT" and "RESET". The background of the page features a blue and green abstract design with circuit-like patterns.

Shows the Availability of customizing complaints

Fig. 8.5 Complaint Details

RAISED COMPLAINTS	COMPLAINT	USER ID	DEPARTMENT
<input type="checkbox"/>	COM99000	peter@gmail.com	network
<input type="checkbox"/>	COM99000	peter@gmail.com	network
<input type="checkbox"/>	COM99003	jesly@gmail.com	network
<input type="checkbox"/>	COM99001	jesly@gmail.com	validity
<input type="checkbox"/>	Network Issue	jesly@gmail.com	very slow net work

APPROVE COMPLAINTS

Shows the list of User Complaints Details to Admin

Fig 8.6 User Complaint Status

project desktop x D:\MS 2007 Mail x CMS report form x (p3) x Model x Groovy (p3) x Download file x localhost/CMS x localhost/127.7 x

localhost/CMS/mycom.php

Apps Gmail YouTube Maps Download & Watch... Disney+ Hotstar ...

out feedback

peter@gmail.com HOME COMPLAINTS MYQUIRIES ABOUT US LOGOUT FAQ

ONLINE COMPLAINT MANAGEMENT SYSTEM

COMPLAINT ID	NAME OF COMPLAINT	DESCRIPTION	DEPARTMENT	STATUS OF COMPLAINT
COM99000	slow data	network speed is too slow	network	proggeress
COM99000	slow data	network speed is too slow	network	proggeress

RAISE COMPLAINT

Shows the status of the user complaint

Fig. 8.7 Add employee Detail by Admin

The screenshot displays the 'ADMIN PANEL' for the 'ONLINE COMPLAINT MANAGEMENT SYSTEM'. The top navigation bar includes links for HOME, LOGIN, COMPLAINTS, CONTACT US, ABOUT US, and LOGOUT. Below this, a secondary navigation bar lists various system modules: USER DETAILS, COMPLAINT DETAILS, EMPLOYEE, USER COMPLAINTS, ANNOUNCEMENTS, CONTACT, NEW OFFERS, and ABOUT US. The 'EMPLOYEE' module is currently selected, leading to a form titled 'CREATE EMPLOYEE DETAILS'. This form contains several input fields for employee information: Employee_name, Employee Mobile_number, Employee E-mail, Employee Address, Password, and Confirm_password. Each field has a placeholder text indicating where to enter the data. At the bottom of the form, there are two buttons: 'SUBMIT' and 'RESET'.

CREATE EMPLOYEE DETAILS	
Employee_name	ENTER THE EMPLOYEE NAME HERE
Employee Mobile_number	ENTER THE EMPLOYEE MOBILE NUM
Employee E-mail	ENTER THE EMPLOYEE EMAIL ID
Employee Address	ENTER THE EMPLOYEE ADDRESS HI
Password	ENTER THE PASSWORD HERE
Confirm_password	RETYPE THE PASSWORD HERE
<input type="button" value="SUBMIT"/> <input type="button" value="RESET"/>	

Shows Adding the employee details in database

CONCLUSION

The Complaint management System was designed in such a way that future modifications can be done easily. Automation of the manual work for maintaining user complaints is done efficiently. Updating of information becomes so easier using admin module. It provides a friendly graphical user interface to raise complaint which proves to be better when compared to the existing manual work-based system. It effectively overcomes the delay in process of complaint. System security, data security and reliability are the striking features.

Future Enhancements

In future, the live meeting feature will be added and live testing of user's devices remotely. In future mail alert system will be included. In future, the online payment facility will be added.

BIBLIOGRAPHY

BOOK REFERENCE

- 1 Gerken, Till, and Tobias Ratschiller. Web Application Development with PHP. New Riders Publishing, 2000.
- 2 Greenspan, Jay, and Brad Bulger. MySQL/PHP database applications. John Wiley & Sons, Inc., 2001.
- 3 Gutmans, Andi, Stig Bakken, and Derick Rethans. PHP 5 Power Programming (Bruce Perens' Open Source Series). Prentice Hall PTR, 2004.
- 4 Krasnick, Cheryl L. [Book Review] Dark Paradise: Opiate Addiction in America Before 1940. Canadian Society for the History of Medicine, 1984.
- 5 Scollo, Chris, and Sascha Shumann. Professional PHP programming. Wrox Press Ltd., 1999.
- 6 Ware, Brent. Open source development with LAMP: using Linux, Apache, MySQL and PHP. Addison-Wesley Longman Publishing Co., Inc., 2002.
- 7 Yank, Kevin. Build your own database driven website using PHP & MySQL. SitePoint Pty Ltd, 2004.
- 8 E.F. Combarro, E. Montan˜ E'S, I. Dı'Az, J. Ranilla, And R. Mones , “Introducing A Family Of Linear Measures For Feature Selection In Text Categorization,” Ieee Trans. Knowledge And Data Eng., Vol. 17, No. 9, Pp. 1223-1232, Sept. 2005

H K B K COLLEGE OF ENGINEERING

VISION

✦ To empower the students through wholesome education & enable the students to develop into highly qualified and trained professionals with ethics and emerge as responsible citizens to build a vibrant nation.

✦ Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

x

MISSION

✦ To enable our students to develop into outstanding professional with high ethical standards to face the challenges of 21st century.

✦ To provide educational opportunities to the deprived and weaker section of the society to uplift their socio-economic status.

**DEPARTMENT OF INFORMATION
SCIENCE AND ENGINEERING**

VISION

To train skilled and ethical professionals with the ability to plan, design, develop, organize and manage modern and traditional information systems with the knowledge of information technologies, services and organizations globally

MISSION

To impart high quality engineering education in the field of Information Science and Technology with strong theoretical and extensive practical training methodologies through innovation and research to make world-class Engineers.

HKBK COLLEGE OF ENGINEERING

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

Programme Outcomes [POs]

- **PO 1:** Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems.
- **PO 2:** Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- **PO 3:** Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
- **PO 4:** Conduct investigations of complex problems: use research based knowledge and research methods including design of Experiments, analysis & interpretation of data, and synthesis of the information to provide valid conclusions.
- **PO 5:** Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling to complex engineering activities, with an understanding of the limitations.
- **PO 6:** The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

- **PO 7:** Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- **PO 8:** Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- **PO 9:** Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- **PO 10:** Communication: Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- **PO 11:** Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- **PO 12:** Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.