

# **MCMS: MBITS COMPLAINT MANAGEMENT SYSTEM**

A PROJECT REPORT

Submitted by

**ALEENA ANIL (MBI19CS006)**

to

The APJ Abdul Kalam Technological University in partial fulfillment of the requirements for  
the award of the Degree

Of

Bachelor of Technology

In

Computer Science and Engineering



**Department of Computer Science and Engineering**

Mar Baselios Institute of Technology & Science

Nellimattom

JULY 2022

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**MAR BASELIOS INSTITUTE OF TECHNOLOGY & SCIENCE**  
**NELLIMATTOM**



**CERTIFICATE**

This is to certify that the project report entitled **“MCMS : MBITS COMPLAINT MANAGEMENT SYSTEM”** submitted by, **ALEENA ANIL (MBI19CS006)** during the academic year 2022 towards the partial fulfillment of the requirement for the award of Bachelor of Technology in Computer Science & Engineering of **APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY** Trivandrum, Kerala is a bonafide work carried out by her in this department under our guidance and supervision.

**MRS. BONIA JOSE**

Assistant Professor

Project Guide

**MINNU ELIZABETH ITTAN**

Assistant Professor

Project Coordinator

**MR. MIDHUN MATHEW**

Head of the Department

Dept of CSE

External Examiner

Internal Examiner

## ACKNOWLEDGEMENT

First and the foremost, I thank **ALMIGHTY GOD** who gave me the inner strength, resource and ability to complete my work successfully, without which all my efforts would have been in vain.

I express my sincere gratitude to the principal, **Dr. P Sojan Lal** for providing me the provision to do the preliminary project in a successful way.

I stand grateful to **Mr. Midhun Mathew**, Head of the Department of Computer Science and Engineering, for his wholehearted support.

I express my sincere gratitude to my project coordinator, **Mrs. Minnu Elizabeth Ittan**, Asst.Prof. for her wholehearted support.

I extend my heartfelt gratitude to my project guide, **Mrs. Bonia Jose**, Asst.Prof. for her guidance and encouragement which were indispensable for the fulfillment of this preliminary project presentation.

I use this opportunity to thank all faculties in the department and the friends who have helped me to complete the project, with their inspiration and co-operation. Last but not the least I am thankful to my parents that without their prayers and blessings this mini project presentation would not have been a success.

## **ABSTRACT**

The COMPLAINT MANAGEMENT SYSTEM is web-based application and it is designed to keep track of complaints registered by the college department/lab staffs, so this system need to have distributed platform independent web application. The task of Administrator executives can control all the activities in the system. Identification of the drawbacks of the existing system led us to the designing of a computerized system that will be compatible to the existing system with the system which is more user-friendly and more GUI oriented. By designing the existing system of Complaint Management System we can improve the efficiency of the system and it also helps in overcoming the drawbacks of the old and traditional way of maintaining complaint handling.

# CONTENTS

ACKNOWLEDGEMENT	i
ABSTRACT	ii
LIST OF TABLES	vi
LIST OF FIGURES	vi
ABBREVIATIONS	vii
CHAPTER 1. INTRODUCTION	1
1.1 GENERAL BACKGROUND	1
1.2 PROPOSED SYSTEM	1
1.3 PRODUCT SCOPE	2
1.4 OBJECTIVES AND GOALS	2
1.4.1 Objectives	2
1.4.2 Goals	3
1.5 FEATURES OF COMPLAINT MANAGEMENT SYSTEM	3
1.5.1 Security and Confidentiality	3
1.5.2 Alerts and Notifications	3
1.5.3 Less manual effort	3
1.5.4 Accountability on Complaints	3
CHAPTER 2. LITERATURE SURVEY	4
2.1 RELATED WORKS	4
CHAPTER 3. SYSTEM ANALYSIS	5

3.1 SYSTEM ANALYSIS AND DESIGN	5
3.2 STAKEHOLDERS OF THIS PROJECT	6
3.3 SOFTWARE REQUIREMENT SPECIFICATION	6
3.4 FEASIBILITY STUDY	7
3.4.1 Economic Feasibility	8
3.4.2 Technical Feasibility	8
3.4.3 Operational Feasibility	8
3.5 SOFTWARE DEVELOPMENT LIFECYCLE MODEL	9
3.5.1 Agile Model	9
3.6 HARDWARE AND SOFTWARE REQUIREMENTS	11
3.6.1 Hardware Specifications	11
3.6.2 Software Specifications	11
CHAPTER 4. DESIGN	12
4.1 DFD	12
4.2 SYMBOLS USED IN DATA FLOW DIAGRAM	13
4.2.1 0-LEVEL DFD	14
4.2.2 1-LEVEL DFD	14
CHAPTER 5. IMPLEMENTATION DETAILS	16
5.1 SERVER: WAMP	16
5.2 IDE: VISUAL STUDIO CODE	16
5.3 DATABASE: MySQL	16
5.4 FRONT END	16
5.5 BACK END : PHP	17

5.6 CODE	18
5.6.1 Admin Codes	18
5.6.2 User Codes	20
CHAPTER 6. TESTING	24
6.1 TEST CASES	24
6.2 UNIT TESTING	25
6.3 INTEGRATION TESTING	26
6.4 BLACK BOX TESTING	26
6.5 WHITE BOX TESTING	27
6.6 VALIDATION TESTING	28
6.7 USER ACCEPTANCE TESTING	28
6.8 TEST CASE	29
CHAPTER 7. CONCLUSION	32
CHAPTER 8. FUTURE ENHANCEMENTS	33
CHAPTER 9. REFERENCES	34
CHAPTER 10. APPENDIX	35
10.1 Screenshots	35
10.1.1 User	35
10.1.2 Admin	39

## LIST OF TABLES

Figure No.	Title	Page No.
4.1	Data Flow Diagram Symbols	13
6.1	Test Case	29

## LIST OF FIGURES

Figure No.	Title	Page No.
4.2	Zeroth Level DFD	14
4.3	First Level DFD - Student	14
4.4	First Level DFD - Admin	15
10.1	User Main Page	35
10.2	User Profile	36
10.3	Lodge Complaint	37
10.4	Complaint History	38
10.5	Admin Home Screen	39
10.6	Manage Users	39
10.7	Manage Complaints	40
10.8	User Login Logs	41



## **ABBREVIATIONS**

CMS	-	COMPLAINT MANAGEMENT SYSTEM
OF	-	OPERATIONAL FEASIBILITY
TF	-	TECHNICAL FEASIBILITY
EF	-	ECONOMIC FEASIBILITY
SDLC	-	SOFTWARE DEVELOPMENT LIFE CYCLE
OS	-	OPERATING SYSTEM
RAM	-	RANDOM ACCESS MEMORY
DFD	-	DATA FLOW DIAGRAM
UML	-	UNIFIED MODELING LANGUAGE
CPU	-	CENTRAL PROCESSING UNIT

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 GENERAL BACKGROUND**

The significance of the study is to serve better than the existing system which is highly manual and therefore difficult in terms of monitoring complaints in the college, therefore improve database and enhance effectiveness, efficiency and security of the system. It is also intended that the study will help in development of a new and hopefully better computer aided system.

Online Complaint Management System provides an online way of solving the problems faced by the students and staffs by saving time and eradicate corruption. The objective of the complaints management system is to make complaints easier to coordinate, monitor, track and resolve, and to provide company with an effective tool to identify and target problem areas, monitor complaints handling performance and make business improvements. Online Complaint Management is a management technique for assessing, analyzing and responding to customer complaints. Complaints management software is used to record resolve and respond to customer complaints, requests as well as facilitate any other feedback.

An outdated or ineffective complaint management system can lead to issues such as:

- Improper recording of complaints
- Failure to track complaints
- Poor decision making
- Data loss

### **1.2 PROPOSED SYSTEM**

This project is a web based application that can be accessed throughout the campus. This work was designed to aid the framework for an existing system which is manually based having demerit of inaccuracy, insecurity and poor performance. The obvious merit of online complaint handling system prompted this project to see ways of integrating our appeal process to the

internet for speed, efficiency, security, accuracy and reliability for student satisfaction. Validation is done for unique user id generation and registration. Validation of user id and password is done with user database. A unique complaint number would be generated to a particular complaint. User would be able to check the status of his/her complaint. Basically this would be a one stop solution for both student and faculty member to make authority aware of the existing issue at a particular time. Option for registering and signing up is available for first time users. Return option is available to cancel the registration process in midway. There is no time timelag between problem identification and action taken to resolve it.

### **1.3 PRODUCT SCOPE**

The significance of this study is to serve better than the existing system which is highly manual and therefore difficult in terms of monitoring the complaint in the University, improve database and enhance effectiveness, efficiency and security of the system. It is also intended that the study will help in the development of a new and hopefully and standard better computer aided system. The new system will save time, reduce improper handling of the complaint system and also improve relationship between student, lecturer and management. The system is expected to be easy as student can login their complaint anytime, staff and management can also equally response to student complaint in a more easy way.

### **1.4 OBJECTIVES AND GOALS**

#### **1.4.1 Objectives:**

The basic CMS objectives are as follows:

- Report Complaints
- Monitor Complaints
- Track Complaints
- Resolve Complaints

### **1.4.2 Goals:**

The goals of the system are to accept these problems in an effective and optimal manner by centralizing the database and thus solving the problems faced by the public by saving time and eradicate corruption. Make the system more user friendly by providing an intensive user interface. Easy access through reports. Restricted data access to employees thus providing additional security to data.

## **1.5 FEATURES OF COMPLAINT MANAGEMENT SYSTEM**

### **1.5.1 Security and Confidentiality**

The software entirely automates the student fees collection process. With the hostel management software login portal, students from any geographical region can apply to avail hostel facilities online.

### **1.5.2 Alerts and Notifications**

In Web notification facility is integrated into the complaint management website and allows them to send alerts, Progress, Action taken, etc to students.

### **1.5.3 Less manual effort**

Current scenario there is very less system operating for the registration of complaints. Presently complaint register process is done through telephone where a complaint is raised by calling the administration. This requires a manual effort and can lead to confusions for the students.

### **1.5.4 Accountability on Complaints**

There is no accountability for the complaints as to when a complaint was raised and when it was resolved . Most of the time students call in the morning or afternoon and at that time there is no one to register the complaint. There is usually a lot of time lag between problem identification and the action taken to resolve it.

## CHAPTER 2

### LITERATURE SURVEY

#### 2.1 RELATED WORKS

In an attempt to review the existing literature on this innovation, some of the existing complaint management systems were noted down as a part of Literature Survey and is listed below:

- There are grievance redressal cells in colleges which aims to look into complaints lodged by any student and redress it as per requirement. Such a student grievance cell is active in **FISAT** called student grievance redressal committee (SGRC). Thus they made a web page for entering the grievances or complaints of students.
- Tough revised complaint mechanism was in place for past 5-6 years with the substantial growth in number of subscribers and a variety of services offered in **BSNL**. The existing system was inadequate to address increasing number of complaints. So robust backend process for effective resolution of complaints is very crucial for customers satisfaction.
- **e-Regulation System for Kerala State Pollution Control Board**. Processing of various services & requests by Industries / Establishments in compliance with Pollution Control Acts, Rules and Notifications. The key areas covered are Consent Management, Waste & Returns Management, Laboratory Management and Complaints Management.

## **CHAPTER 3**

### **SYSTEM ANALYSIS**

#### **3.1 SYSTEM ANALYSIS AND DESIGN**

Systems analysis is a process of collecting factual data, identifying problems, recommending feasible suggestions for improving the functionality of the system. This involves studying the business processes, entity relationships, gathering operational data, understanding the information flow, finding out bottlenecks and evolving solutions for overcoming the weaknesses of the system so as to achieve the organizational goals.

System Analysis also includes decoupling of complex processes that make up the entire system, identification of data stores and manual processes. System design is an attempt towards creation of a new system. This phase focuses on the detailed implementation of the system. System design has two phases: logical design and physical design.

During the logical design phase, the analyst describes inputs (sources), outputs (destinations), databases (data stores) and procedures (data flows) all in a format that meets the user's requirements.

The analyst also specifies the user needs and at a level that virtually determines the information flow into and out of the system and the data resources. Logical design is done through data flow diagrams and database design. The logical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which tell the programmers exactly what the candidate system must do.

The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data through call and produce the required report on a hard copy or display it on the screen.

## 3.2 STAKEHOLDERS OF THIS PROJECT

### → Admin

- Admin is the person who manages the software and focuses on the data and reports of the software. Admin can be logged in to the system by using the email and password provided. They can view User details, Manage Complaints, Manage Users, and view User Login Logs.

### → Student/Faculty

- Student/Faculty is the person who uses the software. He/She is the person who uses the web app for registering complaints and can also view the alerts generated by Admin.

## 3.3 SOFTWARE REQUIREMENT SPECIFICATION

### → Student/Faculty

- The system should have provision to enter Username and Password for every user.
- The system should have provision to register as a new user by providing the basic details of the user such as Name, Email id, Password, Phone no etc.
- Password length should be a minimum 8 characters and it should contain letters, digits and special characters.
- Second stage, after initial registration the user can login to the system using Email id and password.
- Users should be able to register complaints from the lodge complaint section in the web app.

**→ Admin**

- The system shall allow a provision to login the Administrator by entering Username and Password.
- Admin should be able to view User details.
- Admin should be able to view the lodged complaints.
- Admin should be able to manage the users and remove unwanted users if needed.
- Admin should be able to generate alerts that will be shown in the user dashboard.

### **3.4 FEASIBILITY STUDY**

Feasibility Study is basically the test of the proposed system in the light of its workability, meeting user's requirements, effective use of resources and of course, the cost effectiveness. As the name implies, feasibility study is an analysis of the viability of an idea. It ensures that a project is legally and technically feasible and economically justifiable. Moreover this study can be used in various ways with focus on the proposed business. It tells us whether a project is worth doing or not? Feasibility study is a must because a feasibility study evaluates the project's potential for success.

The main goal of feasibility study is not to solve the problem but to achieve the scope. In the process of feasibility study, the cost and benefits are estimated with greater accuracy. It has three types:

**→ Operational Feasibility**

**→ Technical Feasibility**

**→ Economic Feasibility**



### **3.4.1 Economic Feasibility**

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis. This procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise, further justification or alterations in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

### **3.4.2 Technical Feasibility**

The technical feasibility in the proposed system deals with the technology used in the system. It deals with the hardware and software used in the system whether they are of the latest technology or not. It happens that after a system is prepared a new technology arises and the user wants the system based on that technology. This system uses the HTML, CSS and JavaScript as Front end technology, MySQL as database and php as Back end technology. So MCMS is technically feasible. Technical feasibility centres on the existing computer system (hardware, software, etc.,) and to what extent it can support the proposed addition. If the budget is a serious constraint, then the project is judged not feasible. Technical feasibility speaks about the existing hardware and the software that we are using and the deviations that we have to make from the existing one, as we are developing the application using php there is no change in the hardware that we are using. So we say that this application is technically feasible as there is no change in the configuration moreover it is cost effective.

### **3.4.3 Operational Feasibility**

The project has been developed in such a way that it becomes very easy even for a person with little computer knowledge to operate it. It is understandable that the introduction of a candidate system requires special effort to educate, sell, and train the staff on new ways of conducting business. As this project is a user friendly version there is not much training required for the people to use. This product is not only making the task of the administrator easy but it is reducing the time that is taken otherwise. So we say that this product is operationally feasible.

### **3.5 SOFTWARE DEVELOPMENT LIFECYCLE MODEL**

One of the basic notions of the software development process is SDLC models which stand for Software Development Life Cycle models. SDLC – is a continuous process, which starts from the moment, when it's made a decision to launch the project, and it ends at the moment of its full remove from the exploitation. Software development lifecycle (SDLC) is a framework that defines the steps involved in the development of software. It covers the detailed plan for building, deploying and maintaining the software. SDLC defines the complete cycle of development i.e. all the tasks involved in gathering a requirement for the maintenance of a Product.

Some of the common SDLC models are Waterfall Model, V-Shaped Model, Prototype Model, Spiral Model, Iterative Incremental Model, Big Bang Model, Agile Model. We used Agile Model for our Project.

#### **3.5.1 Agile Model**

Agile Model is a combination of the Iterative and incremental model. This model focuses more on flexibility while developing a product rather than on the requirement. In the agile methodology after every development iteration, the client is able to see the result and understand if he is satisfied with it or he is not. Extreme programming is one of the practical use of the agile model. The basis of this model consists of short meetings where we can review our project. In Agile, a product is broken into small incremental builds. It is not developed as a complete product in one go. At the end of each sprint, the project guide verifies the product and after his approval, it is finalized. Client feedback is taken for improvement and his suggestions and enhancement are worked on in the next sprint. Testing is done in each sprint to minimize the risk of any failures.

Advantages of Agile Model:

- It allows more flexibility to adapt to the changes.
- The new feature can be added easily.
- Customer satisfaction as the feedback and suggestions are taken at every stage.
- Risks are minimized thanks to the flexible change process.

Disadvantages of Agile Model:

- Lack of documentation.
- If a customer is not clear about how exactly they want the product to be, then the project would fail.
- With all the corrections and changes there is possibility that the project exceeds time.

## **3.6 HARDWARE AND SOFTWARE REQUIREMENTS**

### **3.6.1 Hardware Specifications**

The selection of hardware configuring is a very task related to the software development, particularly inefficient RAM may adversely affect the speed and corresponding on the efficiency of the entire system. The processor should be powerful to handle all the operations. The hard disk should be sufficient to solve the database and the application.

#### **Hardware used for development:**

CPU: Intel Core i5-7200U

Memory: 4 GB DDR4

Cache: 20 MB

Storage: 1 TB HDD

#### **Minimum hardware required:**

CPU: Snapdragon 425

Memory: 256 MB Above

Cache: 512 KB Above

Storage: 1 GB Above

### **3.6.2 Software Specifications**

This project is built upon the latest technology software.

Front End: HTML, CSS, JavaScript

Back End: PHP

Database: MySQL

OS: Windows, Linux

## CHAPTER 4

### DESIGN

#### 4.1 DFD

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs storage points and the routes between each destination. It is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops. There are several notations for displaying data-flow diagrams. For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes. The data-flow diagram is a tool that is part of structured analysis and data modelling. When using UML, the activity diagram typically takes over the role of the data-flow diagram. A special form of data-flow plan is a site-oriented data-flow plan. DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. The DFD belongs to structured-analysis modeling tools. DFD uses hierarchy to maintain transparency thus multi level DFD's can be created. Levels of DFD's are as follows:

- 0 -Level DFD
- 1-Level DFD
- 2-Level DFD

## 4.2 SYMBOLS USED IN DATA FLOW DIAGRAM





	<p>A data flow is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a process and from data store or process. An arrow line depicts the flow, with arrowhead pointing in the direction of the flow.</p>
	<p>Circles stands for process that converts data in to information. A process represents transformation where incoming data flows are changed into outgoing data flows.</p>
	<p>A source or sink is a person or part of an organization, which enters or receives information from the system, but is considered to be outside the contest of data flow model.</p>
	<p>A data store is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names. If a process merely uses the content of store and does not alter it, the arrowhead goes only from the store to the process. If a process alters the details in the store then a double-headed arrow is used.</p>

Table 4.1 Data Flow Diagram Symbols

### 4.2.1 0-LEVEL DFD

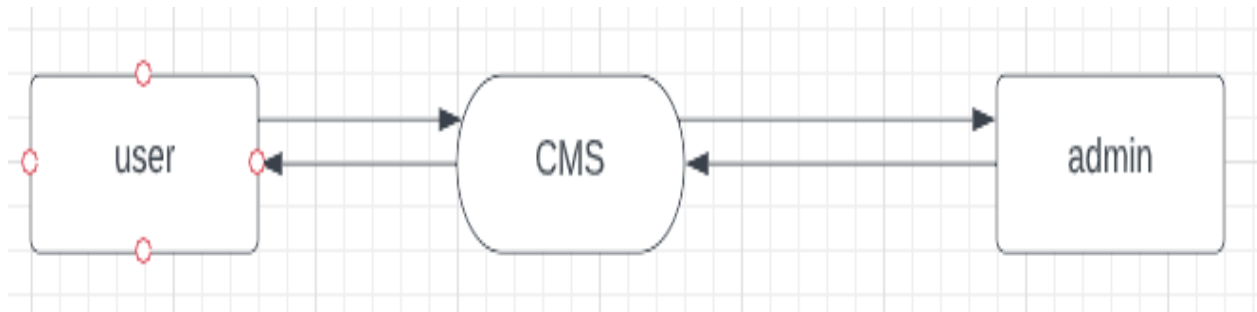


Fig 4.2 Zeroth level DFD

This is the simple view of our entire system processes. In figure 4.2 we can see that, three entities in our system; user, admin and CMS. Each user needs to register in our system once and through the credentials; they can login into the system.

### 4.2.2 1-LEVEL DFD

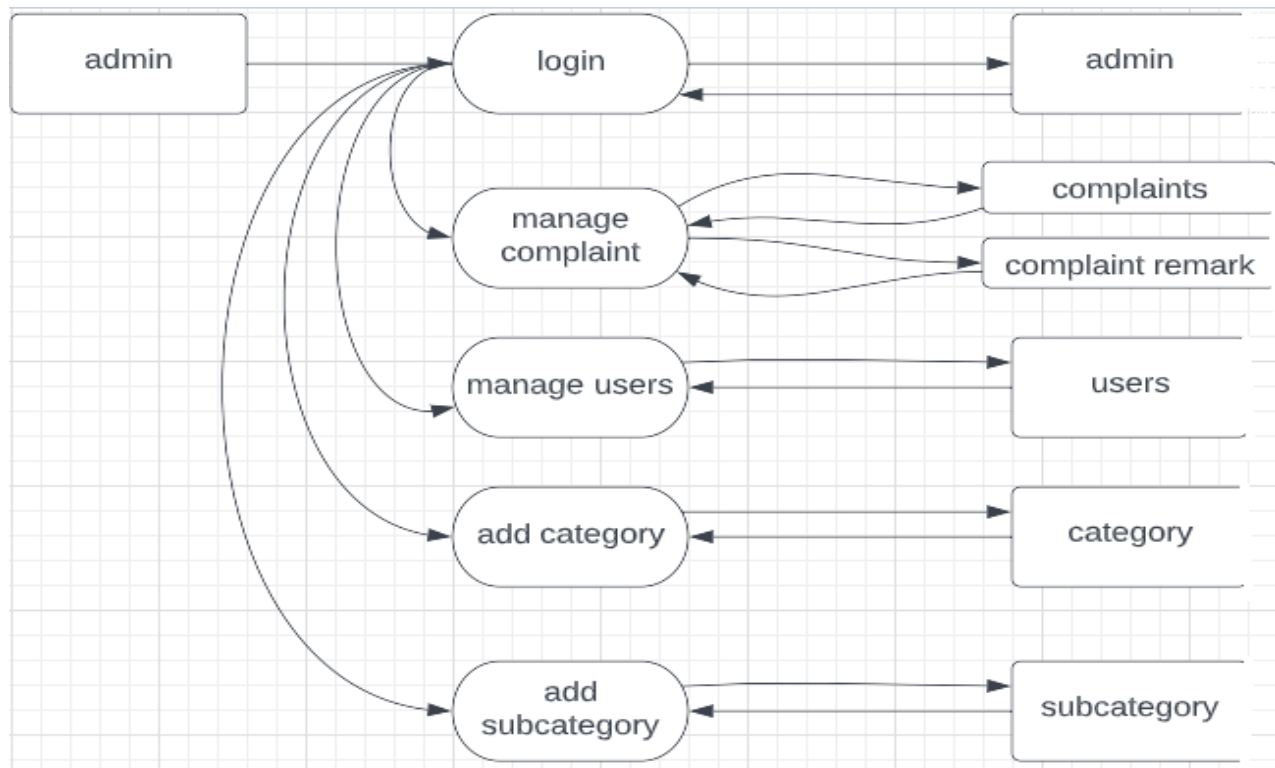


Fig 4.3 First level DFD- Admin

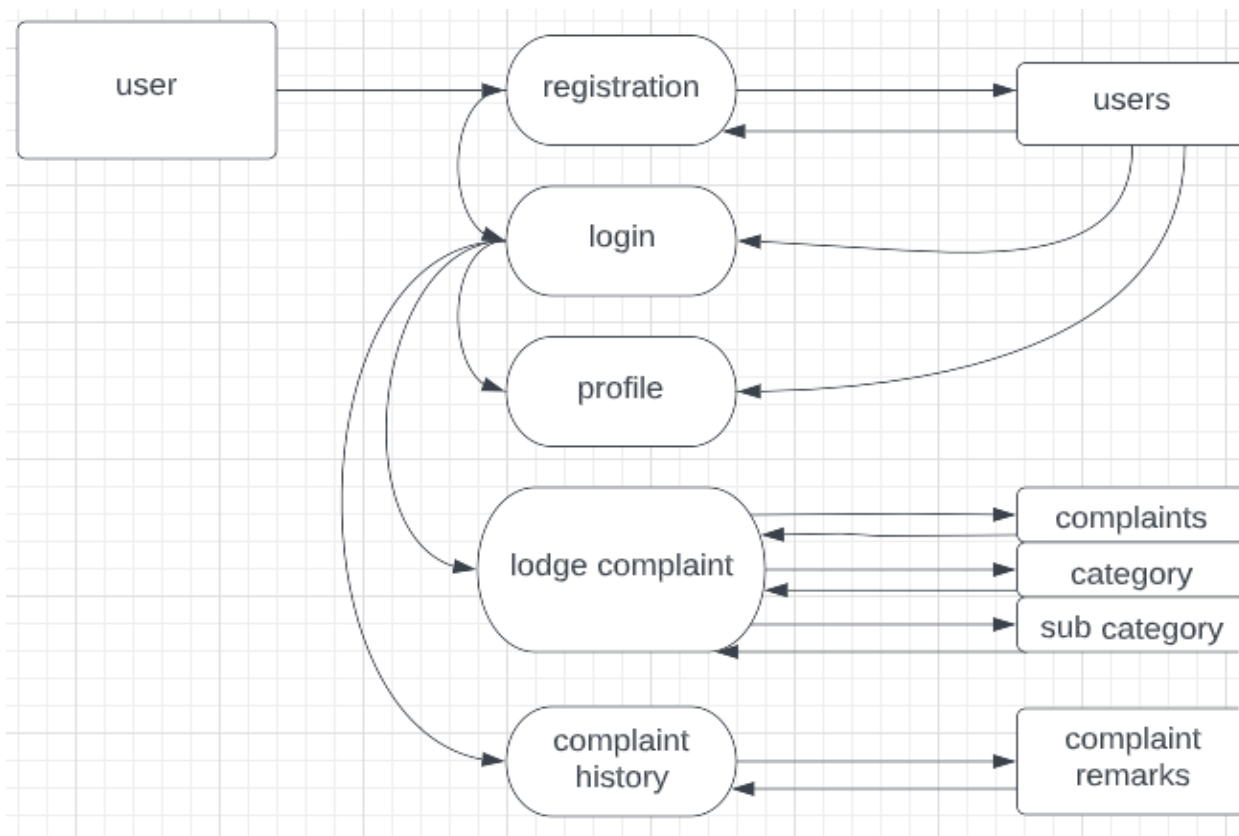


Fig 4.4 First level DFD- User



## **CHAPTER 5**

### **IMPLEMENTATION DETAILS**

#### **5.1 SERVER: WAMP**

WampServer refers to a solution stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language.

#### **5.2 IDE: VISUAL STUDIO CODE**

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDE.

#### **5.3 DATABASE: MySQL**

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses.

#### **5.4 FRONT END:**

- **HTML:**

The HyperText Markup Language or 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 and scripting languages such as JavaScript.

- **CSS:**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.

- **JavaScript:**

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based objectorientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles.

## **5.5 BACK END : PHP**

PHP is a general-purpose scripting language geared toward web development. PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable.

## 5.6 CODE

### 5.6.1 Admin Codes

- **Index**

```
<?php
session_start();
error_reporting
(0);
include("include/config.php");
if(isset($_POST['submit']))
{

$username=$_POST['username'];
$password=md5($_POST['password']);
$ret=mysqli_query($con,"SELECT * FROM admin WHERE username='$username' and
password='$password'");
$num=mysqli_fetch_array($ret);
if($num>0)
{

$extra="change-password.php";//
$_SESSION['alogin']=$_POST['username'];
$_SESSION['id']=$num['id'];
$host=$_SERVER['HTTP_HOST'];
$url=rtrim(dirname($_SERVER['PHP_SELF']),'/\\');
header("location:http://$host$url/$extra");
exit();
}
```

```
else
{

$_SESSION['errmsg']="Invalid username or password";
$extra="index.php";
$host = $_SERVER['HTTP_HOST'];
$uri      =      rtrim(dirname($_SERVER['PHP_SELF']),'/\\');
header("location:http://$host$uri/$extra");
exit();
}

}
```

?>

- **Complaint Details**

```
<?php
session_start();
include('include/config.php');
if(strlen($_SESSION['alogin'])==0)
{

header('location:index.php');
}

else{

?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Admin| Complaint Details</title>
<link type="text/css" href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link type="text/css" href="bootstrap/css/bootstrap-responsive.min.css" rel="stylesheet">
<link type="text/css" href="css/theme.css" rel="stylesheet">
<link type="text/css" href="images/icons/css/font-awesome.css" rel="stylesheet">
<link type="text/css" href='http://fonts.googleapis.com/css?family=Open
+Sans:400italic,600italic,400,600' rel='stylesheet'>
<script language="javascript" type="text/javascript"> var
popUpWin=0;
function popUpWindow(URLStr, left, top, width, height)
{

if(popUpWin)
{

if(!popUpWin.closed) popUpWin.close();
}

popUpWin = open(URLStr,'popUpWin',
'toolbar=no,location=no,directories=no,status=no,menubar=no,scrollbars=yes,resizable=no,co
pyhistory=yes,width='+600+',height='+600+',left='+left+', top='+top+',screenX='+left
+',screenY='+top+'");
}
```

### 5.6.2 User Codes

- **Registration**

```
<?php
include('includes/config.php');
error_reporting(0);
if(isset($_POST['submit']))
```

```
{

$fullname=$_POST['fullname'];
$email=$_POST['email'];
$password=md5($_POST['password']);
$contactno=$_POST['contactno'];
$status=1;
$query=mysqli_query($con,"insert into users(fullName,userEmail,password,contactNo,status)
values('$fullname','$email','$password','$contactno','$status')");
$msg="Registration successfull. Now You can login !";

}

?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="">
<meta name="author" content="Dashboard">
<meta name="keyword" content="Dashboard, Bootstrap, Admin, Template, Theme, Responsive,
Fluid, Retina">

<title>CMS | User Registration</title>
<link href="assets/css/bootstrap.css" rel="stylesheet">
<link href="assets/font-awesome/css/font-awesome.css" rel="stylesheet" />
<link href="assets/css/style.css" rel="stylesheet">
<link href="assets/css/style-responsive.css" rel="stylesheet">
<script>
```

- **Register complaint**

```
<?php
session_start();
error_reporting(0)
;
include('includes/config.php');
if(strlen($_SESSION['login'])==0)
{

header('location:index.php');
}

else{

if(isset($_POST['submit']))
{

$uid=$_SESSION['id'];
$category=$_POST['category'];
$subcat=$_POST['subcategory'];
$complainttype=$_POST['complainttype'];
$state="Kerala";
$noc=$_POST['noc'];
$complaintdetials=$_POST['complaindetails'];
$compfile=$_FILES["compfile"]["name"];
```

```
move_uploaded_file($_FILES["compfile"]["tmp_name"],"complaintdocs/".
$_FILES["compfile"]["name"]);
$query=mysqli_query($con,"insert                                into
tblcomplaints(userId,category,subcategory,complaintType,state,noc,complaintDetails,complai
ntFile)
values('$uid','$category','$subcat','$complainttype','$state','$noc','$complaintdetials','$compfile')
");

// code for show complaint number
$sql=mysqli_query($con,"select  complaintNumber  from  tblcomplaints  order  by
complaintNumber desc limit 1");
while($row=mysqli_fetch_array($sql))
{

$cmpn=$row['complaintNumber'];
}

$complainno=$cmpn;
echo '<script> alert("Your complain has been successfully filled and your complaintno is "+"'.
$complainno.'"</script>';
}

?>
```



## **CHAPTER 6**

### **TESTING**

Coding conventions are a set of guidelines for a specific programming language that recommend Programming style, practices and methods for each aspect of a piece program written in this language. These conventions usually cover file organization, indentation, comments, declarations, statements, white space, naming conventions, programming practices, programming principles, programming rules of thumb, architectural best practices, etc. These are guidelines for software structural quality. Software programmers are highly recommended to follow these guidelines to help improve the readability of their source code and make software maintenance easier.

#### **6.1 TEST CASES**

The objective of system testing is to ensure that all individual programs are working as expected, that the programs link together to meet the requirements specified and to ensure that the computer system and the associated clerical and other procedures work together. The initial phase of system testing is the responsibility of the analyst who determines what conditions are to be tested, generates test data, produces a schedule of expected results, runs the tests and compares the computer produced results with the expected results with the expected results. The analyst may also be involved in procedures testing. When the analyst is satisfied that the system is working properly, he hands it over to the users for testing. The importance of system testing by the user must be stressed. Ultimately it is the user must verify the system and give the go-ahead. During testing, the system is used experimentally to ensure that the software does not fail, i.e., that it will run according to its specifications and in the way, users expect it to.

Special test data is input for processing (test plan) and the results are examined to locate unexpected results. A limited number of users may also be allowed to use the system so analysts can see whether they try to use it in unexpected ways. It is preferable to find these surprises before the organization implements the system and depends on it. In many organizations, testing is performed by people other than those who write the original programs. Using persons who do

not know how certain parts were designed or programmed ensures more complete and unbiased testing and more reliable software. Parallel running is often regarded as the final phase of system testing. Since the parallel operation of two systems is very demanding in terms of user resources; it should be embarked on only if the user is satisfied with the results of testing -- it should not be started if problems are known to exist. Testing is the major quality control measure during software development. Its basic function is to detect errors in the software. Thus, the goal of testing is to uncover requirement design and coding errors in the program. Testing is the process of correcting a program with the intent of finding an error.

Different types of testing are,

1. Unit Testing
2. Integrated Testing
3. Black Box Testing
4. White Box Testing
5. Validation Testing
6. User Acceptance Testing

## **6.2 UNIT TESTING**

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. In this testing we test each module individually and integrate the overall system. Unit testing focuses verification efforts on the smaller unit of software design in the module. This is also known as module testing. The modules of the system are tested separately. The testing is carried out during the programming stage itself. In this testing step each module is found to work satisfactory as regard to the expected output from the module. There are some validation checks for verifying

the data input given by the user which both the formal and validity of the entered. It is very easy to find errors debugging the system.

## **6.3 INTEGRATION TESTING**

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed. Data can be lost across an interface; one module can have an adverse effect on the other sub functions when combined by, may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. This testing was done with sample data. The developed system has run successfully for this sample data.

The need for an integrated test is to find the overall system performance. Integration testing is a logical extension of unit testing. In its simplest form, two units that have already tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. Integration testing identifies problems that occur when units are combined. By using a test plan that requires you to test each unit and ensure the viability of each before combining units, you know that any errors discovered when combining units are likely related to the interface between units. This method reduces the number of possibilities to a far simpler level of analysis. Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

## **6.4 BLACK BOX TESTING**

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or working.

This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher-level testing, but can also dominate unit testing as well. In black box testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements or the specification of the program or module, and the internals of the module or program are not considered for selection of the test cases. In the Black Box testing tester only knows the input that can be given to the system and what output the system should give.

In other words, the basis of deciding test cases in functional testing is requirements or specifications of the system or module. This form of testing is also called functional or behavioral testing. One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight." Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case, or leaves some parts of the program untested.

## **6.5 WHITE BOX TESTING**

White-box testing (also known as clear box testing, glass box testing, and transparent box testing and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements. White Box testing is concerned with testing the implementation of the program.

The intent of this testing is not to exercise all the different input or output conditions but to exercise the different programming structures and data structures used in the program.

White-box test design techniques include:

- Control flow testing
- Data flow testing
- Branch testing
- Path testing
- Statement coverage
- Decision coverage

## **6.6 VALIDATION TESTING**

At the culmination of Black Box testing, software is completely assembled as a package, interface errors have been uncovered and corrected and the final series of software tests, Validation tests begin. Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably accepted by the customer. After the validation test has been conducted one of the two possible conditions exists:

- The function or performance characteristics conform to specification and are accepted.
- A derivation from specification uncovered and a deficiency list is created.

## **6.7 USER ACCEPTANCE TESTING**

Acceptance Testing is a level of the software testing process where a system is tested for acceptability. User Acceptance testing is the software testing process where the system tested for

acceptability & validates the end to end business flow. Such type of testing executed by client in separate environment & confirms whether system meets the requirements as per requirement specification or not.

UAT is performed after System Testing is done and all or most of the major defects have been fixed. This testing is to be conducted in the final stage of Software Development Life Cycle (SDLC) prior to the system being delivered to a live environment. UAT users or end users are concentrating on end to end scenarios & typically involves running a suite of tests on the completed system. User Acceptance testing also known as Customer Acceptance testing (CAT), if the system is being built or developed by an external supplier. The CAT or UAT are the final confirmation from the client before the system is ready for production. The business customers are the primary owners of these UAT tests. These tests are created by business customers and articulated in business domain languages. So ideally it is collaboration between business customers, business analysts, testers and developers. It consists of test suites which involve multiple test cases & each test case contains input data (if required) as well as the expected output. The result of the test case is either a pass or fail.

## 6.8 TEST CASE

A test case is a set of conditions or variables under which a tester will determine whether the system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

A sample of the test case document format is given below.

TC No	Test Steps	Expected Result	Actual Result	Status	Comment
1	Run the web app and navigate to the login screen.	Login screen is displayed. A field for entering Email and a	Login screen has been displayed, fields for entering email and password	Pass	

		field for entering password and a button to login should be present.	together with a login button is available.		
2	Enter an invalid email and invalid password and press the button.	A message should be displayed stating that email and password are invalid.	A message has been displayed stating that email and password are invalid.	Pass	
3	Enter a valid Email and password and press the button.	Users must successfully login to the respective screen.	A message has been displayed stating that the login successful and navigate into home page.	Pass	Using 'Admin@gmail.com' as E-mail and Admin @123 as password.
4	Enter a valid E-mail and leave password and press the button.	A message should be displayed stating that please enter the user Email and password.	A message has been displayed stating that please enter user Email and password.	Pass	Password field is outlined in red colour stating invalid entry
5	Leave Email and password and press the button.	A message should be displayed stating that please enter the user Email and password.	A message has been displayed stating that please enter user Email and password.	Pass	Email and password field is outlined in red colour stating invalid entry.
6	Leave E-mail and enter a	A message should be	A message has been	Pass	Email field is outlined in

	valid password and press the button Leave.	displayed stating that please enter the user Email and password.	displayed stating that please enter user Email and password.		red colour stating invalid entry.
--	--	--	--	--	-----------------------------------

Fig 6.1 Test Case



## **CHAPTER 7**

### **CONCLUSION**

MCMS complaint management system is a user friendly computer based system for managing complaints in institution. It has be designed to automate, manage complaints and look after the overall processing of records of student complaints. The MCMS offers security, confidentiality and usability. Since the web application allows user to login, register complaints, track complaints and complaint history. MCMS is user friendly it can be made flexible for future enhancement.

The project was successfully completed within the time span allotted. All the modules are tested separately and put together to form the main system. Finally, the modules are tested with real data and it worked successfully. The system has fulfilled the entire objective defined.

## **CHAPTER 8**

### **FUTURE ENHANCEMENTS**

- This system is found tested and examined for its successful processing. Future change in the environment or processing can be easily adopted by having simple change in coding. It is very user friendly, cost effective, feature rich and it provides very high level of security. It protects the unauthorized users.
- Moreover, the system coding is so well designed that new operations can be easily incorporated without much modification. A facility to inform through SMS or Email on landing of the consignment can be added in future.

## CHAPTER 9

### REFERENCES

- [https://www.researchgate.net/publication/329363039\\_Design\\_and\\_Implementation\\_of\\_Online\\_Students'\\_Complaint\\_Case\\_Study\\_of\\_English\\_Study\\_Program\\_at\\_Victory\\_University\\_Sorong](https://www.researchgate.net/publication/329363039_Design_and_Implementation_of_Online_Students'_Complaint_Case_Study_of_English_Study_Program_at_Victory_University_Sorong)
- <https://www.scribd.com/document/428462427/DESIGN-AND-IMPLEMENTATION-OF-ONLINE-STUDENT-COMPLAINT-MANAGEMENT-SYSTEM-doc>
- <https://www.campcodes.com/projects/php/complaint-management-system-project-in-php-mysql-free-download/>
- <https://www.sourcecodester.com/php/14206/complaint-management-system.html>
- <http://www.educationmanagementsoftware.com>
- [http://nptel.iitm.ac.in/courses/Webcourse-contents/IIScBANG/System%20Analysis%20and%20Design/pdf/Lecture\\_Notes/LNm5.pdf](http://nptel.iitm.ac.in/courses/Webcourse-contents/IIScBANG/System%20Analysis%20and%20Design/pdf/Lecture_Notes/LNm5.pdf)
- <http://www.bbnisys.com/images/eschool.pdf>
- [https://www.academia.edu/13516291/Students\\_Complaints\\_Management\\_System](https://www.academia.edu/13516291/Students_Complaints_Management_System)

## CHAPTER 10

### APPENDIX

#### 10.1 Screenshots

##### 10.1.1 User

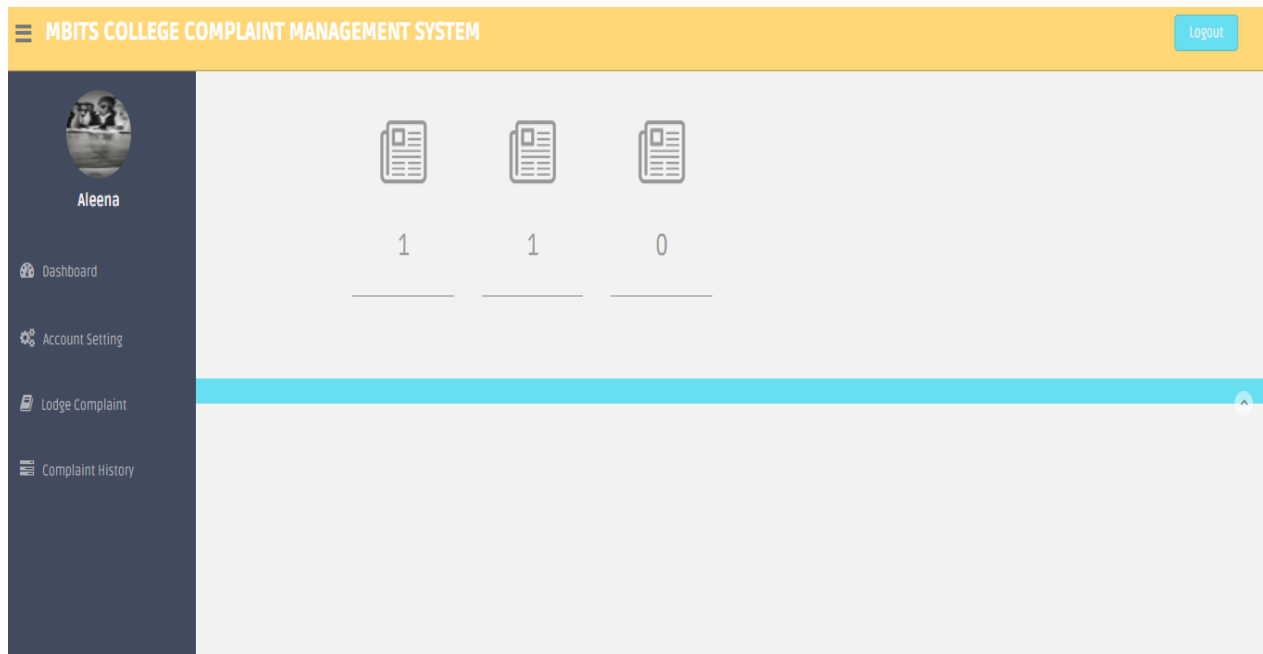


Fig 10.1 User Main Page

MBITS COLLEGE COMPLAINT MANAGEMENT SYSTEM [Logout](#)

**Aleena**

> Profile info


**Aleena's Profile**

Last Updated at :

Full Name  User Email

Contact  Address

Country  Pincode

Reg Date  User Photo 

[Change Photo](#)

Fig 10.2 User Profile

Fig 10.3 Lodge Complaint

> Register Complaint

Category	<input type="text" value="Select Category"/>	Sub Category	<input type="text" value="Select Subcategory"/>
Complaint Type	<input type="text" value="Complaint"/>		
Nature of Complaint	<input type="text"/>		
Complaint Details (max 2000 words)	<div></div>		
Complaint Related Doc(if	<div><input type="button" value="Choose File"/> No file chosen</div>		

Activate Windows  
Go to Settings to activate Windows.

Fig 10.4 Complaint History

>Your Complaint Hstory

Complaint Number	Reg Date	last Updation date	Status	Action
4	2022-07-22 13:03:47		Not Process Yet	<a href="#">View Details</a>
5	2022-07-23 19:22:40		In Process	<a href="#">View Details</a>

## 10.1.2 Admin

CMS | Admin

Admin ▾

Manage Complaint ▾

Manage users

Add Category

Add Sub-Category

User Login Log

Logout

### Admin Change Password

Current Password  Enter your current Password

New Password  Enter your new current Password

Current Password  Enter your new Password again

Submit

© 2022 MBITS CMS All rights reserved.

Fig 10.5 Admin Home Screen

Manage Users					
Show	10 ▾	entries	Search:	<input type="text"/>	
#	Name	Email	Contact no	Reg. Date	Action
1	Aleena	aleenaanil29@gmail.com	8078155171	2022-07-12 08:58:25	<a href="#">View Detials</a> <a href="#">Delete</a>
2	Parvathy	vsparvathyraj@gmail.com	9495671833	2022-07-14 11:50:03	<a href="#">View Detials</a> <a href="#">Delete</a>
3	anusree ps	anusreepsanu1004@gmail.com	8078155171	2022-07-14 12:19:39	<a href="#">View Detials</a> <a href="#">Delete</a>
4	Aleena Anil	aleena.9.2000@gmail.com	8078155171	2022-07-21 21:44:47	<a href="#">View Detials</a> <a href="#">Delete</a>
Showing 1 to 4 of 4 entries					<a href="#">&lt;</a> <a href="#">&gt;</a>

Fig 10.6 Manage Users



Fig 10.7 Manage Complaints

### Closed Complaints

Show  entries

Search:

Complaint No	complainant Name	Reg Date	Status	Action
4	Aleena	2022-07-22 13:03:47	Not process yet	<a href="#">View Details</a>

Showing 1 to 1 of 1 entries

< >

Fig 10.8 User Login Logs

Manage Users					
Show	10	▼	entries	Search:	<input type="text"/>
#	User Email	User IP	Login Time	Logout Time	Status
1	aleena.9.2000@gmail.com	127.0.0.1	2022-07-21 23:26:45	21-07-2022 11:27:20 PM	Successfull
2	aleenaanil29@gmail.com	127.0.0.1	2022-07-21 23:29:03	21-07-2022 11:30:23 PM	Successfull
3	vsparvathyraj@gmail.com	127.0.0.1	2022-07-21 23:30:31	21-07-2022 11:34:25 PM	Successfull
4	aleena.9.2000@gmail.com	127.0.0.1	2022-07-21 23:36:22	21-07-2022 11:37:17 PM	Successfull
5	aleenaanil29@gmail.com	127.0.0.1	2022-07-21 23:37:25	21-07-2022 11:41:55 PM	Successfull
6	vsparvathyraj@gmail.com	127.0.0.1	2022-07-21 23:42:08	21-07-2022 11:42:10 PM	Successfull
7	aleenaanil29@gmail.com	127.0.0.1	2022-07-21 23:42:17	21-07-2022 11:42:49 PM	Successfull
8	anusreepsanu1004@gmail.com	127.0.0.1	2022-07-21 23:44:02	21-07-2022 11:45:54 PM	Successfull
9	aleenaanil29@gmail.com	127.0.0.1	2022-07-21 23:46:08	21-07-2022 11:46:18 PM	Successfull
10	aleena.9.2000@gmail.com	127.0.0.1	2022-07-21 23:46:27	21-07-2022 11:47:37 PM	Successfull