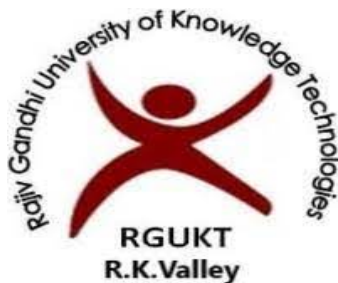


“LIFE INSURANCE MANAGEMENT SYSTEM”

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING



Rajiv Gandhi University of Knowledge Technologies

R.K VALLEY

submitted by

P.Dhanalakshmi - R171137

J.Akhila - R171117

Under the Esteemed guidance of

Mr.Santhosh Kumar P

RGUKT RK Valley.

DECLARATION

We hereby declare that the report of the B.Tech Major Project Work entitled **“LIFE INSURANCE MANAGEMENT SYSTEM”** which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for award of any degree.

P Dhanalakshmi – R171137

J Akhila – R171117

Dept. Of Computer Science and Engineering.

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES



RGUKT

RGUKT, RK VALLEY

Department of Computer Science and Engineering

CERTIFICATE FOR PROJECT COMPLETION

This is certify that the project entitled **“LIFE INSURANCE MANAGEMENT SYSTEM”** submitted by **P Dhanalakshmi(R171137),J Akhila(R171117)** under our guidance and supervision for the partial fulfillment for the degree Bachelor of Technology in Computer Science and Engineering during the academic semester -2 , 2021-2022 at RGUKT, RK VALLEY.To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma.

Project Internal Guide

Mr.Santhosh Kumar P

RGUKT, RK Valley

Head of the Department

Mr.Harinadha P

HOD Of CSE

RGUKT, RK Valley

ABSTRACT

This Life Insurance Management System project is the web based applications software programmed using web technology. It manages the activities related to insurance of the people. All the information and data catchup which providing insurance policy to people and can be saved in to database for which Sql is used as backend. Later on, the insurance holder can view the details using their login credentials. For rest of the users, the information of each other will keep confidential.

The insurance company needs to keep track of details of its agents, policyholders, their premium payments. Hence it is under tremendous pressure maintaining their day-to-day activities, which is currently being done manually. Entire records have to be updated timely, even a slight mistake could complicate things. It is very difficult to handle bulk data since human memory is weaker than electronic counter part. It is time consuming to summarize these details to produce the reports. Agents can control all the activities in the website.

He is responsible for maintaining the records of their clients such as adding clients, editing details of them, deleting the clients, adding payments etc. This portal is designed for the company to maintain the records. This website is user friendly to agents and the clients. We are offering a robust web based insurance solution, which has the flexibility of the software that is designed keeping in mind to make it one of the best system for keeping all the tracks of the details of their clients and for providing best services to them.

INDEX

1. Abstraction	4
2. Introduction	7
2.1 Purpose	8
2.2 Intended Audience	8
2.3 Product Vision	8
2.4 Technologies	8
3. HTML	9
4. CSS	9
5. PHP	9
6. MY SQL	9
7. Requirements	10
7.1 Non-Functional Requirements	11
7.2 Functional Requirements	11
8. System Design	11
8.1 Context Diagram	12
9.UML Diagrams	12

9.1 Usecase Diagram	12
9.2 Activity Diagram	14
10. ER Diagram	15
11. Agile Development	16
11.1 Agile	16
11.2 Agile Methodology	17
12.Coding	18
13.Testing	21
14.Output	24
15.References	26
16.Conclusion	27

SRS Document

Introduction:

This document has the requirements of Life Insurance Management System. It is used to maintain the information of customers and their payments. This system maintains profile management of all policy holders. This system providing interface to customer that helps him to know his policy details. The clients information files can be stored in centralized database which can be maintained by the system. User Friendliness is provided in the application with various

- Agent
- client

Agent Module:

An agent is nothing but a registered user who provides information regarding the policies. Agent adds clients by verifying their profile manually.

- Agent login module
Agent can login to website using their login credentials.
- Add client Module
Agent can add client details
- Add Nominee module
Agent can add nominees for the clients
- Add payments module
Agent can add the payments details
- Add policy module
Agent can add the clients policies

Client Module:

An client is the person who holds the policy and his details are adding by the agent

- Client login module
client can login website using their login credentials
- Client Details module

- This module is for clients where clients can check their details and payment status.

1.1: Purpose:

The purpose of this document is to gather the requirements that are needed for implementing the Life Insurance Management System. It also focuses on various key features, product vision and scope, product overview. This website provides easy access of information regarding the clients.

1.2: Intended Audience:

The intended audience will be the client who can access the platform to get information about the policies and agents can add, edit, update clients information and payments.

Users:

1. agents
2. clients

Product Vision:

Vision Statement:

The product vision is to manage the insurance policies, which is user friendly and easily accessible. This website helps to access the data for authorized people using their login details.

Technologies:

- HTML
- CSS
- PHP
- MY SQL

HTML:

HTML stands for Hyper Text Markup Language. It is not a programming language as the name itself tells it is markup language used for creating the web pages. It describes the structure of web page and consists of a series of elements. These elements tell the browser how to display the content. HTML elements label pieces of content such as "this is heading", "this is a paragraph", "this is a link", etc.

CSS:

CSS stands for Cascading Style Sheets. It describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple webpages all at once. External sheets are stored in CSS style. It is used to style the web pages such as to alter the font, color, size, and spacing of content, split into multiple columns, or add animations and other decorative features.

PHP:

PHP is a widely used, open source scripting language. PHP scripts are executed on the server. It is free to download and use. PHP is powerful enough to be at the core of the biggest blogging system on the web. PHP files can contain text, HTML, CSS, JavaScript, and PHP code. PHP code is executed on the server and the result is returned to the browser as plain HTML and these files have extension “.php”.

PHP can generate dynamic page content and can create, open, read, write, delete, and close the files on the server. It can collect form data, can send and receive cookies. It also can add, delete, modify data in our database. It can be used to control user-access and can encrypt the data.

MYSQL:

MySQL is a database server and is ideal for both small and large applications. It supports standard SQL. It is free to download and use. It creates a database for storing and manipulating data, defining relationship of each table. Clients can make requests by typing specific SQL statements on MySQL. The server application will respond with the requested information and it will appear on the client side. SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application-programming interfaces (APIs).

To access MySQL:

<http://localhost/phpmyadmin>

Connecting Database:

```
<?php
//EDIT Your Database name, Username and Password here
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "lms";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname) or die("User/password is wrong");
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
?>
```

3. SYSTEM REQUIREMENTS SPECIFICATION

3.1 NON FUNCTIONAL REQUIREMENTS

3.1.1 Software Requirements

This web site requires the following software in Server (computer), clients(computers)

Server-side Requirements:

Operation System : Windows 11

Web Server : XAMPP

Database : MY SQL

Client-side Requirements:

Browser : Any HTML 4.0 or prior version compliant browser with a Minimum
Screen resolution of 800X600 pixels (best ewed in 1024 x 768 resolution)

3.1.2 Hardware Requirements

The following is a list of minimum requirements on server side

Hard Disk : 40GB Hard disk with minimum 4GB free space

Interface : Mouse, Keyboard

On client side any hardware that can run a web browser.

3.2 FUNCTIONAL REQUIREMENTS

3.2.1 Product Requirements

This web site is an online website that provides the following features

Login to agent panel.

Clients logins and management

Add clients including nominees

update delete clients and payments

Search for clients/insured persons

3.1.2 User Requirements

The web site provides easy adding ,editing and deleting the details of clients and the payments .A visitor with minimum knowledge of web browsing/surfing can access the site very easily. Due to dynamic nature of features, the members, Admin members should be able to understand the provided facilities.

3.1.3 Performance Requirements

The following performance requirements should be maintained in the project.

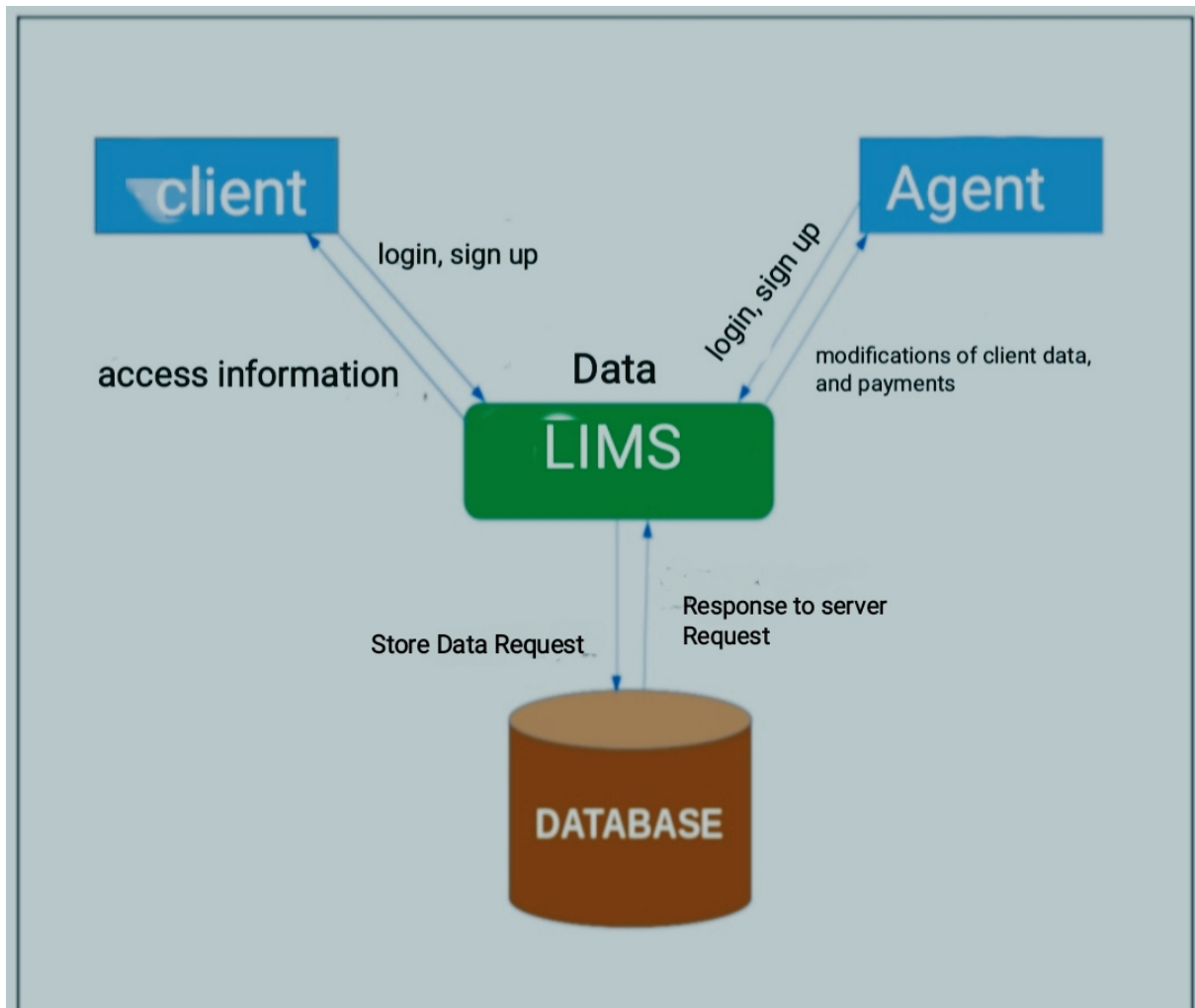
Each page in the site needs to load in a reasonable amount of time.

Latest web techniques like Caching should be implemented to speed up the loading of dynamic pages. This will also improve on the number of simultaneous users, as connections are freed faster.

System Design:

This Life Insurance Management System helps the agents to add clients by taking the details of nominees too and to edit and delete clients and payments of the clients. This website also helps the user to access their data.but they can just view and check thier data such as their details,their payment status,policy details,payment details etc.

Context Diagram:



UML Diagrams:

Actor:

A coherent set of roles that users of use cases play when interacting with the use cases. An observable result of value of an actor.

Use case:

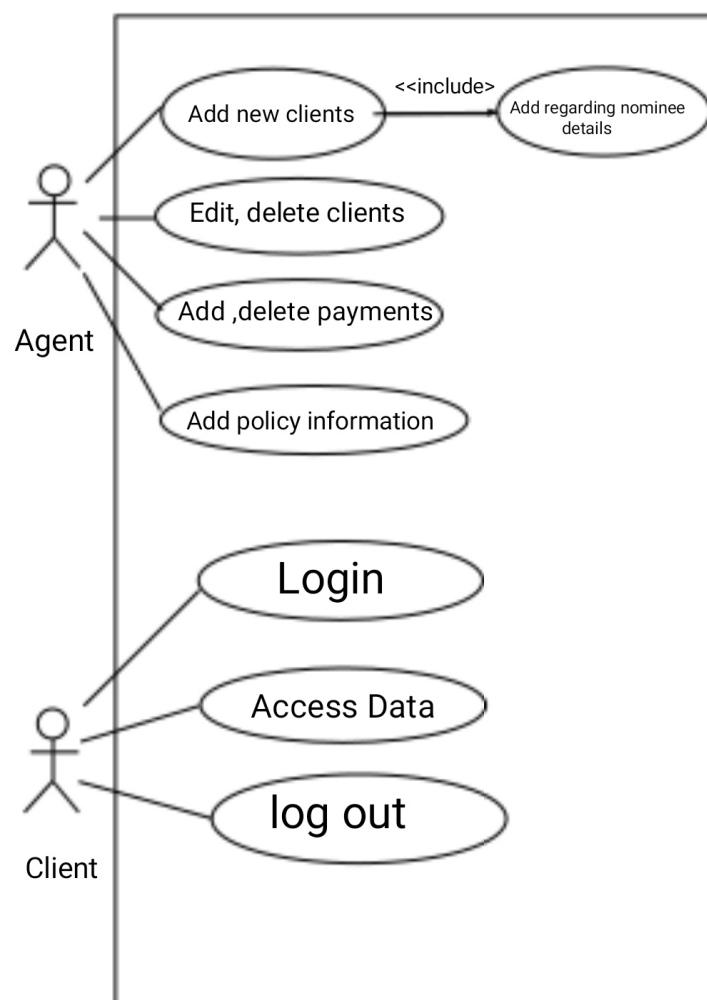
A description of sequence of actions, including variants, that a system performs

yields an observable result of value of an actor. actor diagram is drawn in a eclipse shape
UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after nalysis.

The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

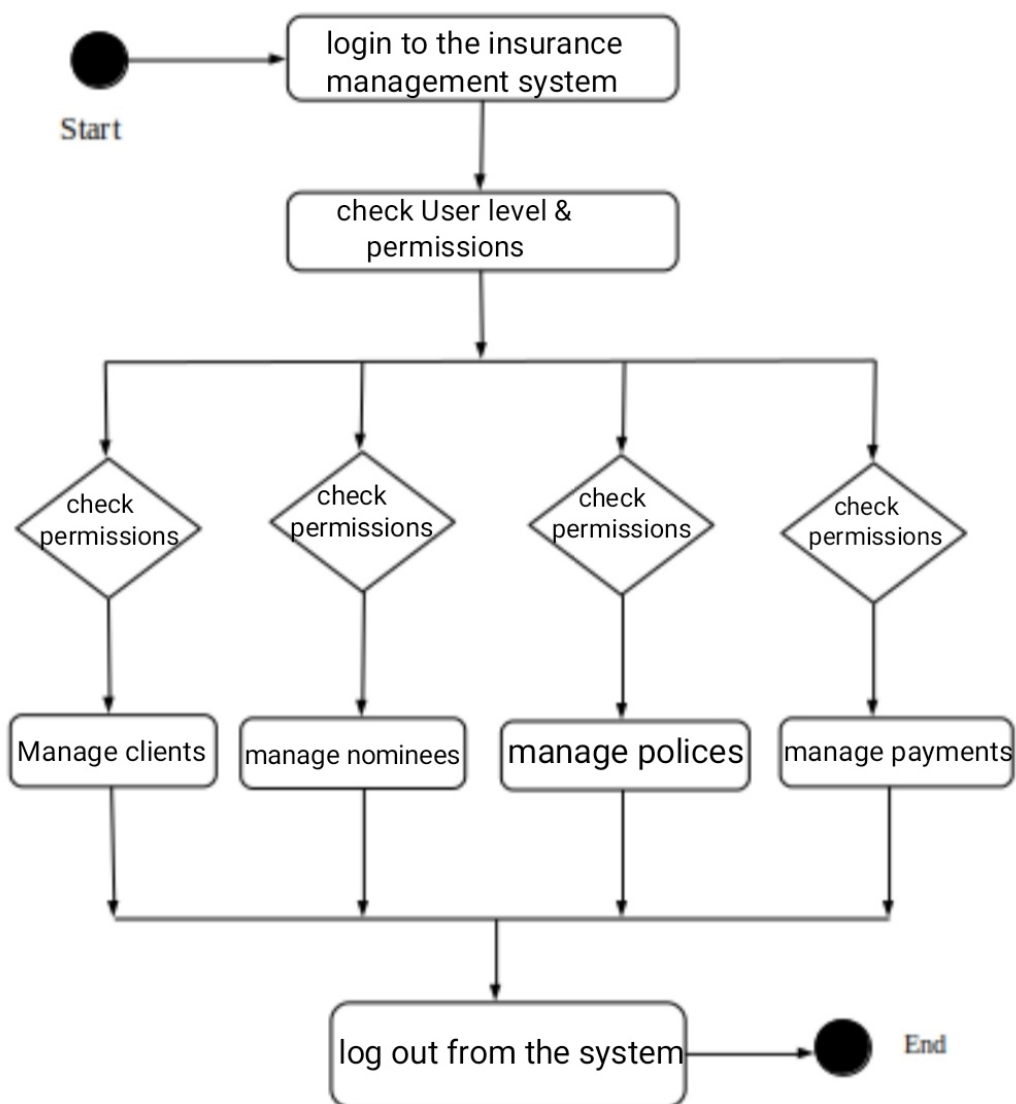
USECASE DIAGRAMS:

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying that can do and more importantly what they can't do. Use case diagram consists of use cases and actors and shows the interaction between the use case and actors. The purpose is to show the interactions between the use case and actor. To represent the system requirements from user's perspective. An actor could be the end-user of the system or an external system.



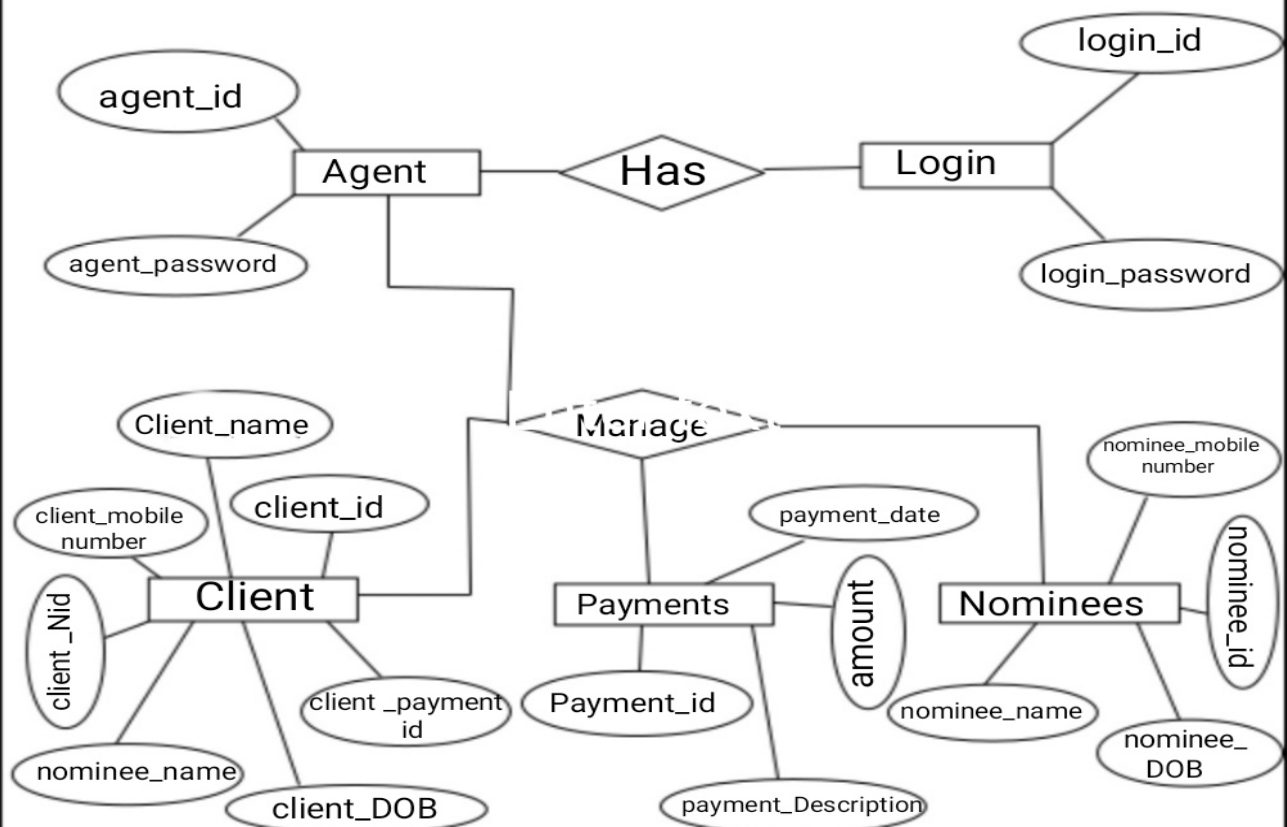
ACTIVITY DIAGRAM:

An activity Diagram is a behavioral of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.



ER Diagram:

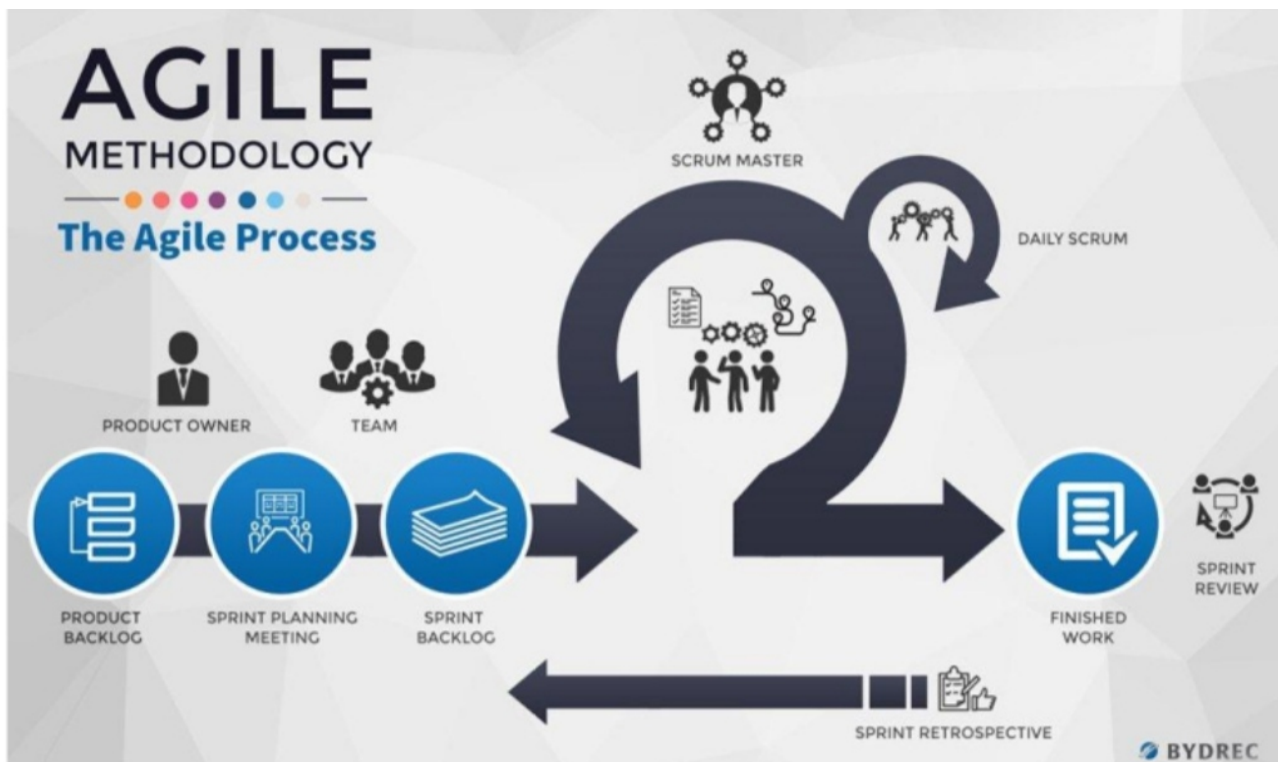
The Entity-Relationship (ER) model was originally proposed by Peter as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer.



Agile Development

Agile

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of **planning, executing, and evaluating**. Continuous collaboration is vital, both with team members and project stakeholders.



Agile methodology

It's a process for managing a project that involves constant collaboration and working in iterations. Today, the word Agile can refer to these values and the frameworks for implementing them, including Scrum, Kanban, Extreme Programming(XP), and Adaptive Project Framework (APF).

Agile

A project management methodology characterized by building products using short cycles of work that allow for rapid production and constant revision.

Kanban

A visual approach to project management where teams create physical representations of their tasks, often using sticky notes on whiteboards (or online apps). Tasks are moved through predetermined stages to track progress and identify common roadblocks.

Scrum

A PM methodology in which a small team is led by a Scrum master, whose main job is to clear away all obstacles to completing work. Work is done in short cycles called sprints, but the team meets daily to discuss current tasks and roadblocks.

Adaptive Project Framework (APF)

A project management methodology that grew from the idea that most IT projects can't be managed using traditional PM methods. Work is done in stages and evaluated after each one.

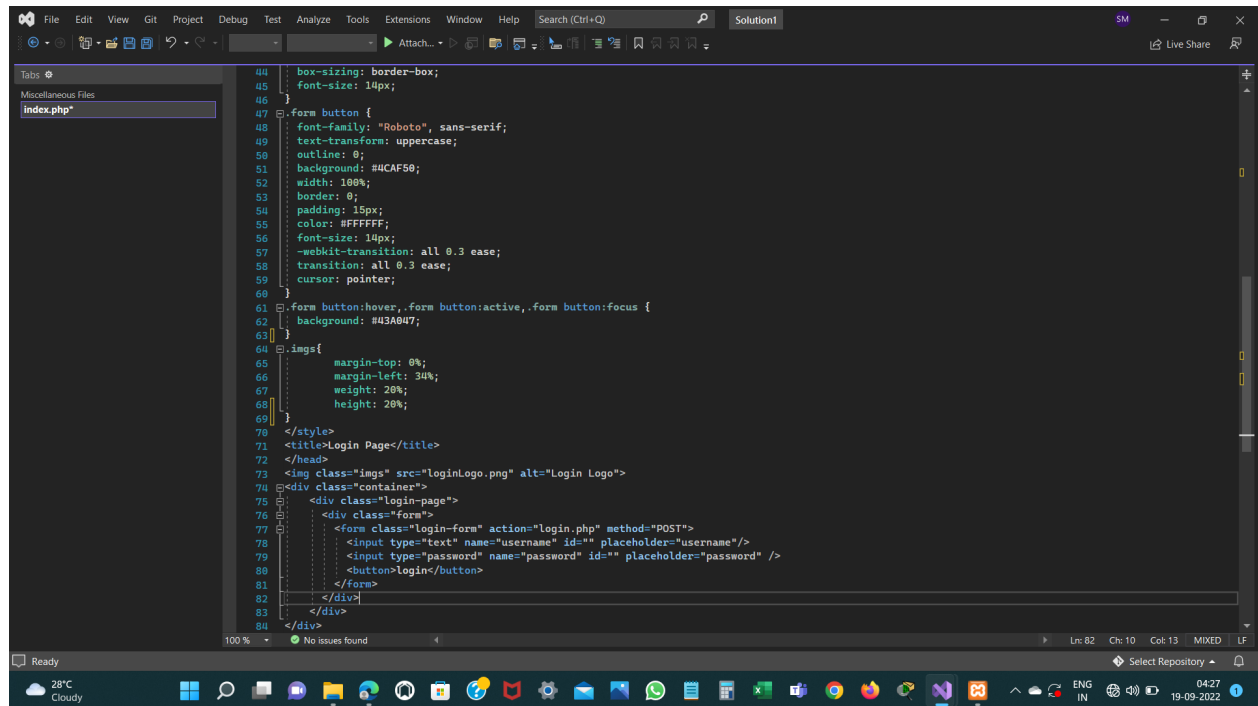
Extreme Project Management (XPM)

A PM methodology where the project plan, budget, and final deliverable can be changed to fit evolving needs, no matter how far along the project is.

How to Develop a project using Agile methodology

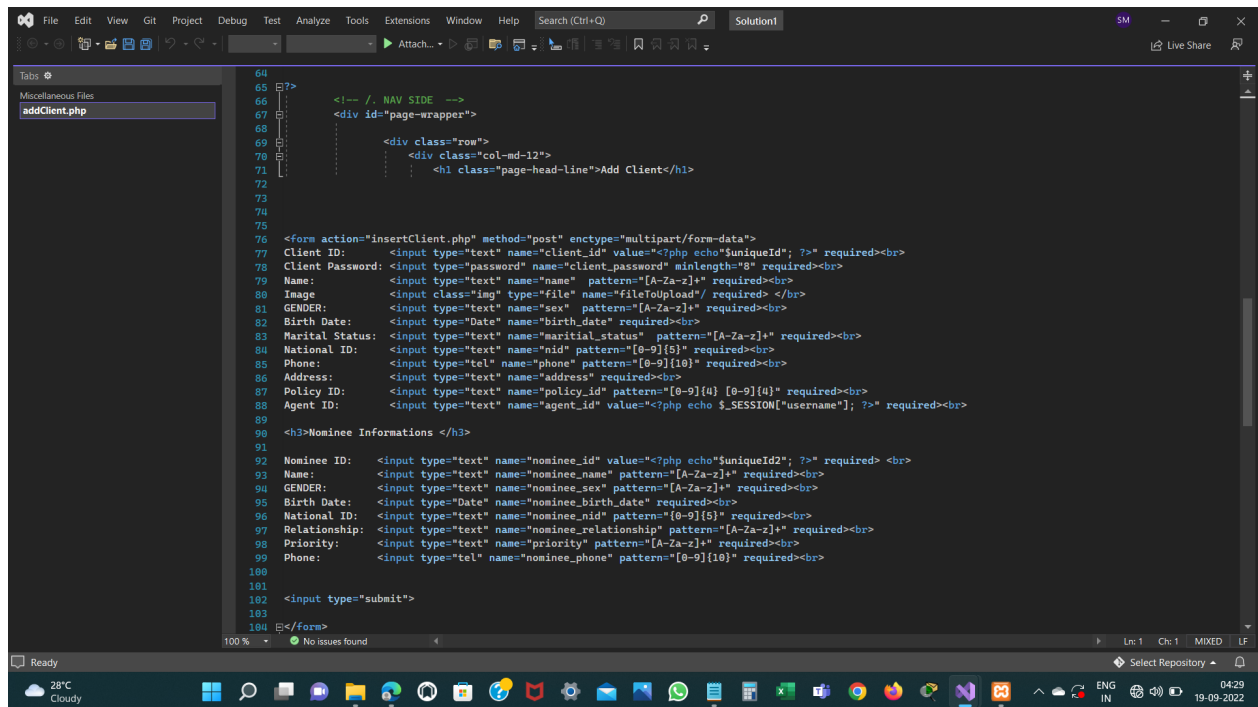
Agile Development is a Continuous Integration(CI) from Requirements gathering to testing the code..We start the project Development with Requirements analysis and Gathering. In this we collect the data from the project description and draw the UML diagrams like ER diagram for database tables and Use Case diagram for implementation functionalities.After we create a short stories like Login, Signup, Homepage Design, Database Creation From the requirements file.

CODING:



The screenshot shows the VS Code editor with the file `index.php` open. The code is a combination of CSS and HTML. The CSS part defines styles for a login form button and its container. The HTML part includes a title, a login logo, and a form with username and password fields.

```
44  box-sizing: border-box;
45  font-size: 14px;
46
47  .form button {
48    font-family: "Roboto", sans-serif;
49    text-transform: uppercase;
50    outline: 0;
51    background: #4CAF50;
52    width: 100%;
53    border: 0;
54    padding: 15px;
55    color: #FFFFFF;
56    font-size: 14px;
57    -webkit-transition: all 0.3 ease;
58    transition: all 0.3 ease;
59    cursor: pointer;
60  }
61  .form button:hover, .form button:active, .form button:focus {
62    background: #43A047;
63  }
64  .imgs{
65    margin-top: 0%;
66    margin-left: 34%;
67    width: 28%;
68    height: 28%;
69  }
70
71  </style>
72  <title>Login Pages</title>
73  </head>
74  
75  <div class="container">
76    <div class="login-page">
77      <div class="form">
78        <form class="login-form" action="login.php" method="POST">
79          <input type="text" name="username" id="" placeholder="username"/>
80          <input type="password" name="password" id="" placeholder="password" />
81          <button>login</button>
82        </form>
83      </div>
84    </div>
85  </div>
```



The screenshot shows the VS Code editor with the file `addClient.php` open. The code is an HTML form for adding a new client. It includes fields for Client ID, Password, Name, Image, Gender, Birth Date, Marital Status, National ID, Phone, Address, Policy ID, and Agent ID. There is also a section for Nominee Information.

```
64  <!-- /. NAV SIDE -->
65  <div id="page-wrapper">
66
67    <div class="row">
68      <div class="col-md-12">
69        <h1 class="page-head-line">Add Client</h1>
70      </div>
71    </div>
72
73    <form action="insertClient.php" method="post" enctype="multipart/form-data">
74      Client ID: <input type="text" name="client_id" value="php echo $uniqueId; ?" required><br>
75      Client Password: <input type="password" name="client_password" minlength="8" required><br>
76      Name: <input type="text" name="name" pattern="[A-Za-z]+" required><br>
77      Image: <input class="img" type="file" name="fileToUpload" required> <br>
78      GENDER: <input type="text" name="sex" pattern="[A-Za-z]+" required><br>
79      Birth Date: <input type="Date" name="birth_date" required><br>
80      Marital Status: <input type="text" name="maritalStatus" pattern="[A-Za-z]+" required><br>
81      National ID: <input type="text" name="nid" pattern="[0-9]{5}" required><br>
82      Phone: <input type="tel" name="phone" pattern="[0-9]{10}" required><br>
83      Address: <input type="text" name="address" required><br>
84      Policy ID: <input type="text" name="policy_id" pattern="[0-9]{4} [0-9]{4}" required><br>
85      Agent ID: <input type="text" name="agent_id" value="php echo $_SESSION['username']; ?" required><br>
86
87      <h3>Nominee Informations </h3>
88
89      Nominee ID: <input type="text" name="nominee_id" value="php echo $uniqueId2; ?" required> <br>
90      Name: <input type="text" name="nominee_name" pattern="[A-Za-z]+" required><br>
91      GENDER: <input type="text" name="nominee_sex" pattern="[A-Za-z]+" required><br>
92      Birth Date: <input type="Date" name="nominee_birth_date" required><br>
93      National ID: <input type="text" name="nominee_nid" pattern="[0-9]{5}" required><br>
94      Relationship: <input type="text" name="nominee_relationship" pattern="[A-Za-z]+" required><br>
95      Priority: <input type="text" name="priority" pattern="[A-Za-z]+" required><br>
96      Phone: <input type="tel" name="nominee_phone" pattern="[0-9]{10}" required><br>
97
98      <input type="submit">
99    </form>
100  </div>
```

```
File Edit View Git Project Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Solution1
Miscellaneous Files
addNominee.php

53 <link href="assets/css/font-awesome.css" rel="stylesheet" />
54 <!--CUSTOM BASIC STYLES-->
55 <link href="assets/css/basic.css" rel="stylesheet" />
56 <!--CUSTOM MAIN STYLES-->
57 <link href="assets/css/custom.css" rel="stylesheet" />
58 <!-- GOOGLE FONTS-->
59 <link href="http://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet" type="text/css" />
60 </head>
61 <?php
62 include 'header.php';
63
64 $uniqueId2 = time().'-'.mt_rand();
65
66 if(isset($_GET['client_id'])){
67     $client_id
68 }else{ $client_id="";
69 }
70
71 <!-- /. NAV SIDE -->
72 <div id="page-wrapper">
73
74     <div class="row">
75         <div class="col-md-12">
76             <h1 class="page-head-line">Add Nominee</h1>
77
78
79
80
81
82 <form action="insertNominee.php" method="post">
83 Nominee ID: <input type="text" name="nominee_id" value="<?php echo $uniqueId2; ?>" required><br>
84 Client ID: <input type="text" name="client_id" value="<?php echo $client_id; ?>" required><br>
85 Name: <input type="text" name="name" pattern="[A-Za-z]+" required><br>
86 GENDER: <input type="text" name="sex" pattern="[A-Za-z]+" required><br>
87 Birth Date: <input type="Date" name="birth_date" required><br>
88 MID: <input type="text" name="mid" pattern="[0-9]{5}" required><br>
89 Relationship: <input type="text" name="relationship" pattern="[A-Za-z]+" required><br>
90 Priority: <input type="text" name="priority" pattern="[A-Za-z]+" required><br>
91 Phone: <input type="tel" name="phone" pattern="[0-9]{10}" required><br>
92
93 <input type="submit">
```

```
File Edit View Git Project Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Solution1
Miscellaneous Files
addPayment.php

48 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
49 <title>Add Payment</title>
50 <!-- BOOTSTRAP STYLES-->
51 <link href="assets/css/bootstrap.css" rel="stylesheet" />
52 <!-- FONTAWESOME STYLES-->
53 <link href="assets/css/font-awesome.css" rel="stylesheet" />
54 <!--CUSTOM BASIC STYLES-->
55 <link href="assets/css/basic.css" rel="stylesheet" />
56 <!--CUSTOM MAIN STYLES-->
57 <link href="assets/css/custom.css" rel="stylesheet" />
58 <!-- GOOGLE FONTS-->
59 <link href="http://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet" type="text/css" />
60 </head>
61 <?php include 'header.php';
62 $uniqueId = time().'-'.mt_rand();
63 if(isset($_GET['client_id'])){
64     $client_id = $_GET['client_id'];
65 }else{ $client_id="";
66 }
67
68 <!-- /. NAV SIDE -->
69 <div id="page-wrapper">
70
71     <div class="row">
72         <div class="col-md-12">
73             <h1 class="page-head-line">Add Payment</h1>
74
75
76
77
78 <form action="insertPayment.php" method="post">
79
80 Receipt No: <input type="text" name="receipt_no" value="<?php echo $uniqueId; ?>" required><br>
81 Client ID: <input type="text" name="client_id" value="<?php echo $client_id; ?>" required><br>
82 Month: <input type="Date" name="month" required><br>
83 Amount: <input type="text" name="amount" pattern="[0-9]{1,}" required><br>
84 Due: <input type="text" name="due" pattern="[0-9]{1,}" required><br>
85 Fine: <input type="text" name="fine" required><br>
86 Agent Id: <input type="text" name="agent_id" value="<?php echo $_SESSION['username']; ?>" required><br>
87
88 <input type="submit">
```

```
File Edit View Git Project Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Solution1
Miscellaneous Files
addAgent.php

43 <tr:nth-child(even) {
44     background-color: #dddddd;
45 }
46 </style>
47 <meta charset="utf-8" />
48 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
49 <title>Add Agent</title>
50 <!-- BOOTSTRAP STYLES-->
51 <link href="assets/css/bootstrap.css" rel="stylesheet" />
52 <!-- FONTAWESOME STYLES-->
53 <link href="assets/css/font-awesome.css" rel="stylesheet" />
54 <!-- CUSTOM BASIC STYLES-->
55 <link href="assets/css/basic.css" rel="stylesheet" />
56 <!-- CUSTOM MAIN STYLES-->
57 <link href="assets/css/custom.css" rel="stylesheet" />
58 <!-- GOOGLE FONTS-->
59 <link href="http://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet" type="text/css" />
60 </head>
61 <?php include 'header.php';
62 <?>
63 <!-- /. NAV SIDE -->
64 <div id="page-wrapper">
65
66     <div class="row">
67         <div class="col-md-12">
68             <h1 class="page-head-line">Add Agent</h1>
69
70
71
72
73 <form action="insertagent.php" method="post">
74 Agent ID: <input type="text" name="agent_id" pattern="[A-Za-z]+" required><br>
75 Agent Password: <input type="password" name="agent_password" required><br>
76 Name: <input type="text" name="name" pattern="[A-Za-z]+" required><br>
77 Branch: <input type="text" name="branch" pattern="[A-Za-z]+" required><br>
78 Phone: <input type="tel" name="phone" pattern="[0-9]{10}" required><br>
79
80 <input type="submit">
81 </form>
82
83
```

```
File Edit View Git Project Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Solution1
Miscellaneous Files
logout_template.php

1 <form action="logout.php" method="POST">
2     <table>
3     <tr>
4         <td><input type="submit" value="Logout"/></td>
5     </tr>
6     </table>
7 </form>
```

Testing

After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

System Testing

The goal of the system testing process was to determine all faults in our project .The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing
- 2 .Integration testing

Unit Testing

Unit testing is commenced when a unit has been created and effectively reviewed .In order to test a single module we need to provide a complete environment i.e. besides the section we would require The procedures belonging to other units that the unit under test calls Non local data structures that module accesses .A procedure to call the functions of the unit under test with appropriate parameters

1. Test for the admin module

Testing admin login form-This form is used for log in of administrator of the system. In this form we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask the details.

Report Generation: admin can generate report from the main database.

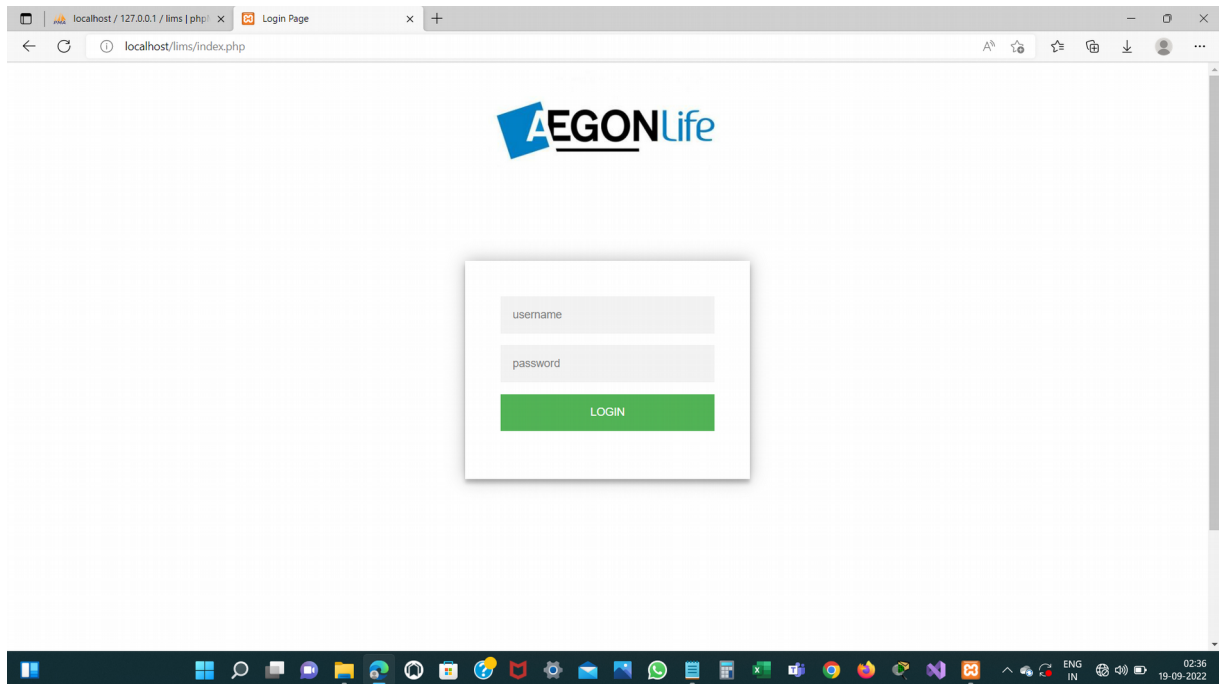
Integration Testing

In the Integration testing we test various combination of the project module by providing the input.

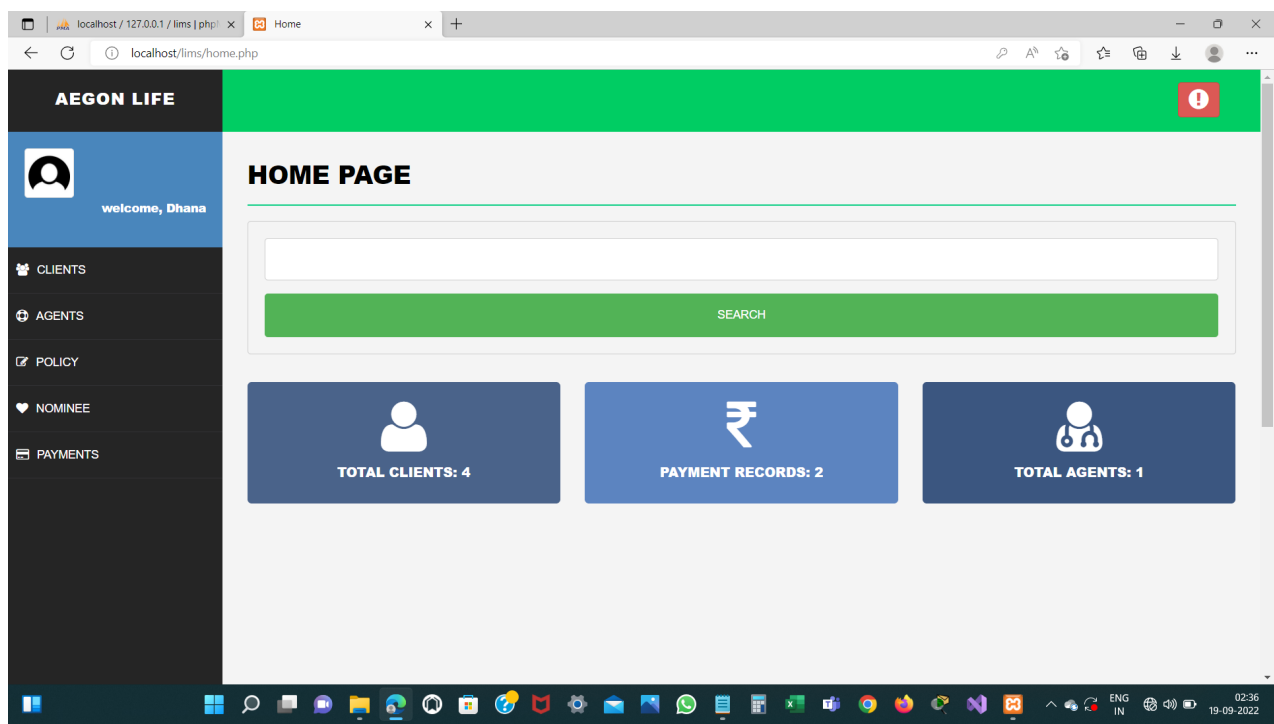
The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

OUT PUT:

Login page:



Home page:



Clients:

AEGON LIFE

welcome, Dhana

CLIENTS INFORMATION

[Add Client](#)

CLIENT ID	NAME	Birth Date	NID	PHONE	ADDRESS	STATUS	UPDATE
1659014389	Akhila	24-7-2001	67543	9087253816	honnampalli	Client Status	Edit
1659391373	Nandu	05-12-1996	78654	9023849173	Banglore	Client Status	Edit
1663507563	D.pavani	03-08-2002	14561	7993767846	kadapa	Client Status	Edit
1663520242	Raju	2001-12-09	78762	9727982974	nmear tirupati	Client Status	Edit

Adding client details:

AEGON LIFE

welcome, Dhana

ADD CLIENT

Client ID:

1663535222

Client Password:

Name:

Image

DSC_0606.JPG

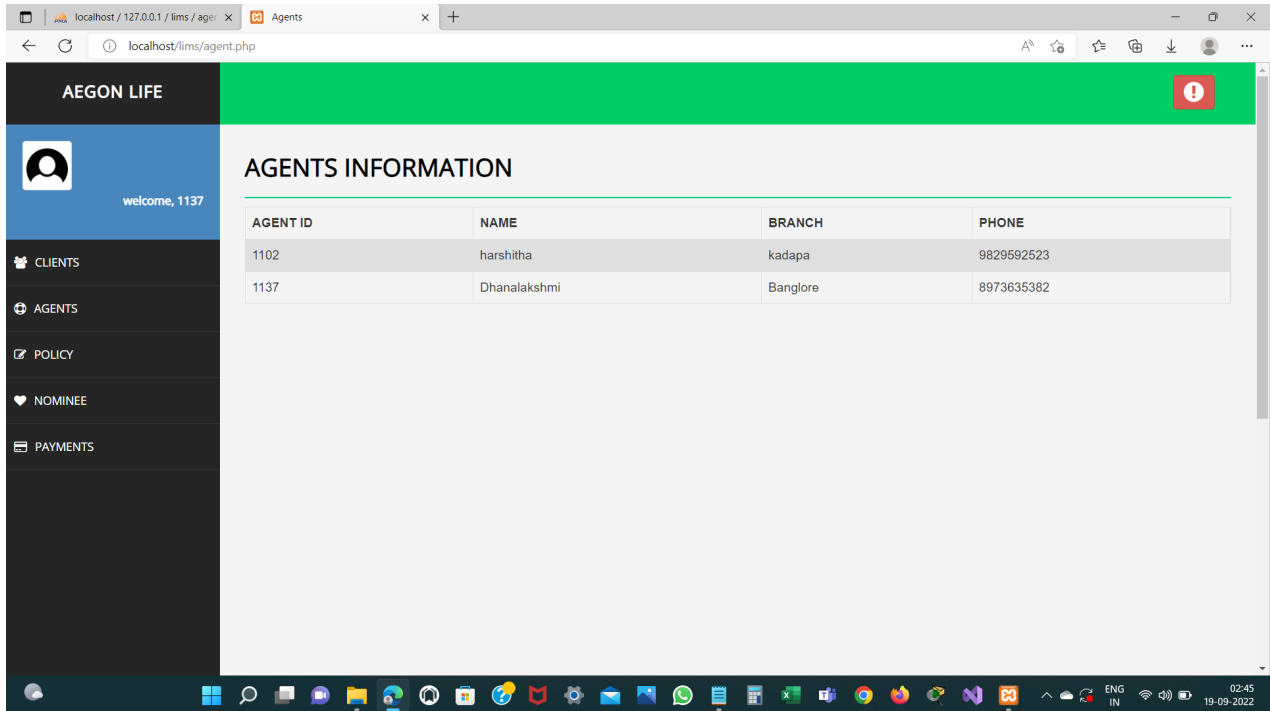
GENDER:

Birth Date:

Marital Status:

National ID:

Agent:



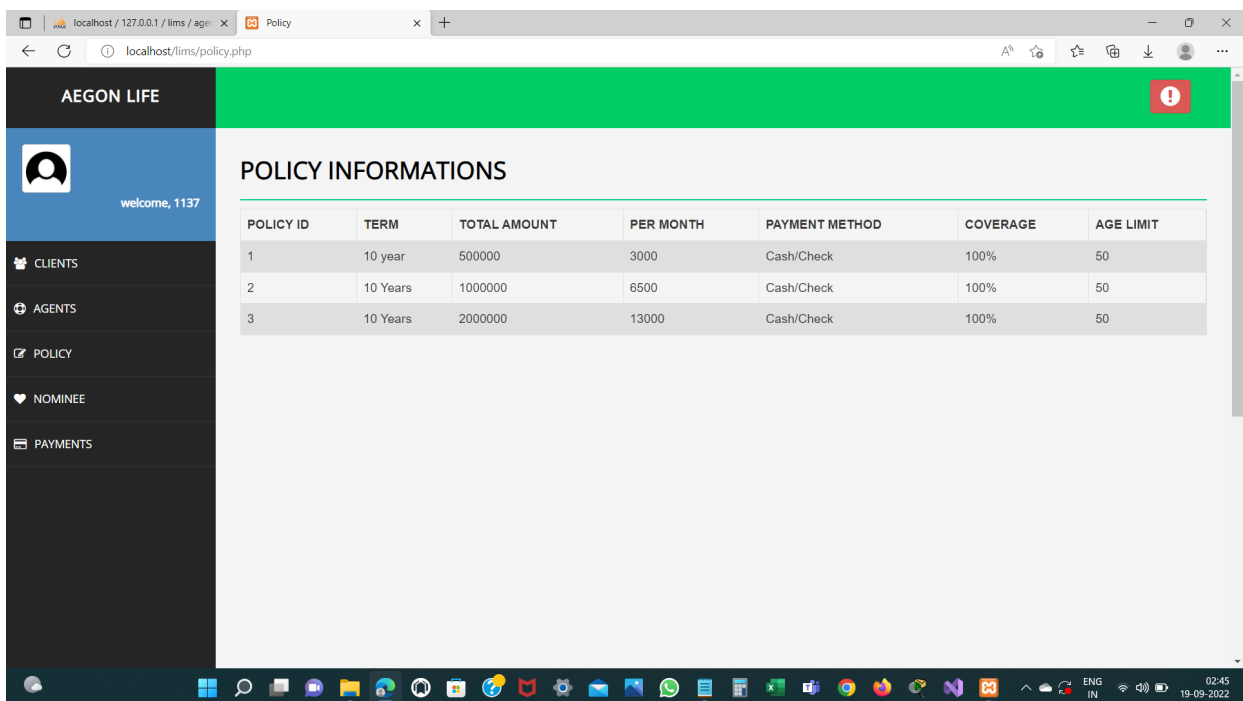
AEGON LIFE

welcome, 1137

AGENTS INFORMATION

AGENT ID	NAME	BRANCH	PHONE
1102	harshitha	kadapa	9829592523
1137	Dhanalakshmi	Banglore	8973635382

Policy:



AEGON LIFE

welcome, 1137

POLICY INFORMATIONS

POLICY ID	TERM	TOTAL AMOUNT	PER MONTH	PAYMENT METHOD	COVERAGE	AGE LIMIT
1	10 year	500000	3000	Cash/Check	100%	50
2	10 Years	1000000	6500	Cash/Check	100%	50
3	10 Years	2000000	13000	Cash/Check	100%	50

Adding Nominee Details:

AEGON LIFE

welcome, 1137

POLICY INFORMATION

POLICY ID	TERM	TOTAL AMOUNT	PER MONTH	PAYMENT METHOD	COVERAGE	AGE LIMIT
1	10 year	500000	3000	Cash/Check	100%	50
2	10 Years	1000000	6500	Cash/Check	100%	50
3	10 Years	2000000	13000	Cash/Check	100%	50

Payments:

AEGON LIFE

welcome, 1137

PAYMENT INFORMATION

Add Payment

RECEIPT NO	CLIENT ID	MONTH	AMOUNT	DUE	FINE	UPDATE
1663507014_118579622	1659014389	august	100000	1000	5 years	Edit
1663521056_159566691	954298670702	2022-09-18	12000	29	1000	Edit

Add payments:

ADD PAYMENT

Receipt No:
1663535915_1424878338

Client Id:
1659391373

Month: 16-09-2022

Amount:
200000

Due:
20000

Fine:
1000

Agent Id:
1137

References

For PHP

<https://www.w3schools.com/php/default.asp>
<https://www.sitepoint.com/php/>
<https://www.php.net/>

For MySQL

<https://www.mysql.com/>
<http://www.mysqltutorial.org>

For XAMPP

<https://www.apachefriends.org/download.html>

CONCLUSION

A computerized insurance management system has been developed and the system was tested with sample data. The system results in regular timely preparations of required outputs. In comparison with manual system the benefits under a computer system are considerable in the saving of man power working hours and Effort. Provision for addition , updation and deletion of customers is there in the system .It is observed that proper filing system has been adopted for future refernce . The entire project runs on windows environments. The system can be used to make better management described at appropriate time. The user gets amount and timely information system.