**Insurance Agent Samurai**

*A*

***Project Report***

*submitted*

*in partial fulfillment*

*for the award of the Degree of*

***Bachelor of Technology***

***in Department of Computer Science & Engineering***



Session-2015-16

**Supervisor** **Submitted By**

Mr. Shubham Gupta Vishal Chauhan

Lecturer, Department of Computer Science & Engineering Roll No: 12ESKCS762

Ajay Sharma

Roll No: 12ESKCS704

Amit Jangir

Roll No: 12ESKCS708

Manoj Kumar Gupta

Roll No: 12ESKCS732

**Department of Computer Science & Engineering**

**Swami Keshvanand Institute of Technology,**

**Management & Gramothan, Jaipur**

**SWAMI KESHVANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN, JAIPUR**

**CERTIFICATE**

I hereby declare that the project, which is being presented in the Project Report, entitled **“Insurance Agent Samurai”** in partial fulfillment for the award of Degree of “Bachelor of Technology” in Department of Computer Science & Engineering with Specialization in **Computer Engineering**, **and submitted to the Department of Computer Science &Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur**, Rajasthan Technical University is a record of my own project carried under the Guidance of **Mr. Shubham Gupta**, Lecturer, Department of Computer Science & Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur.

**Vishal Chauhan** B.Tech. (Computer Science &Engineering) Roll No: 12ESKCS762

**Ajay Sharma** B.Tech. (Computer Science &Engineering) Roll No: 12ESKCS704

**Manoj Kumar Gupta** B.Tech (Computer Science &Engineering) Roll No: 12ESKCS732

**Amit Jangir** B.Tech. (Computer Science &Engineering) Roll No: 12ESKCS708

.................................. ..................................

**Mr. Shubham Gupta Mrs. Rubal Gill**

Lecturer, Department of Computer Science Lecturer, Department of Computer

& Engineering Science & Engineering

Swami Keshvanand Institute of Technology, Swami Keshvanand Institute of

M & G, Jaipur Technology M & G , Jaipur

**ACKNOWLEDGMENT**

We feel immense pleasure in expressing my regards to the Chairman **Mr. Surja Ram Meel,** Director **Mr. Jaipal Meel,** Registrar **Mrs. Rachana Meel,** Director (Academics) **Prof.** (**Dr.) S. L. Surana,** Director (D&W) **Prof. (Dr.) S. K. Calla** &Principal **Prof. (Dr.) Ramesh Prachar** Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur for providing us necessary facilities during the various stages of this project.

We would like to thank **Dr. C. M. Choudhary,** Professor & Head, Department of Computer Science & Engineering and **Dr. Anil Chaudhary,** Professor & Head,Department of Information Technology for provide us opportunity to work in consistent direction and providing their valuable suggestions to improve Project Report.

We would like to thank our esteemed guide **Mr. Shubham Gupta, Mrs. Rubal Gill** and **Miss Neha Kapoor** Department of Computer Science & Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipurfor their valuable guidance, keen interest, constant encouragement, incessant inspiration and continuous help throughout this work. Especially we acknowledge their support when we were stuck and they suggested us new ideas to solve the problems. Their vast experience and realistic approach have been of great help to me. We are honored having **Mr. Shubham Gupta** as our project supervisor. Their excellent guidance has been instrumental in making this work a success.

We would also like to express our thanks to our parents for their support and blessings. In addition, a very special thanks to all our colleagues and friends for their support in the completion of this work.

**Vishal Chauhan**

B.Tech. (Computer Science &Engineering)

Roll No: 12ESKCS762

**Ajay Sharma**

B.Tech. (Computer Science &Engineering)

Roll No: 12ESKCS704

**Manoj Kumar Gupta**

B.Tech (Computer Science &Engineering)

Roll No: 12ESKCS732

**Amit Jangir**

B.Tech. (Computer Science &Engineering)

Roll No: 12ESKCS708

**TABLE OF CONTENTS**

**Certificate/declaration** i

**Acknowledgment** ii

**Contents** iii

**List of Tables** v

**List of Figures** vi

**Abbreviations and Acronyms** vii

**Abstract** 1

**Chapter 1: Introduction …………………………………………………………………..**2-4

1.1 Need of project 2

1.2 Scope of Project 2

1.3 Project Objective……………………………………………………………..2

1.4 Key Features…………………………………………………………………3

**Chapter 2: Project Discription ...**4-22

2.1 Application Requirements 4

2.2 User Characteristics 6

2.3 Data Flow Diagram…………………………………………………………..10

2.4 Database Design……………………………………………………………...12

2.5 Flow Procedure………………………………………………………………18

**Chapter 3:Technologies Used…………………………………………………………..23-26**

3.1 HTML………………………………………………………………………..23

3.2 JSP…………………………………………………………………………...23

3.3 Microsoft Sql Server………………………………………………………...24

3.4 Visual Studio 2010 Ultimate….……………………………………………..24

3.5 Java Script…………………………………………………………………..25

3.6 CSS…………………………………………………………………………25

3.7 Apache Tomcat Server….………………………………………………….26

3.8 Eclipse IDE………………………………………………………………....26

**Chapter 4:User Interface……………………………………………………………….27-39**

4.1 User Login………………………………………………………………….27

4.2 DashBoard………………………………………………………………….29

4.3 Administrator…………………………………………………………….....30

4.4 Agent……………………………………………………………………….31

4.5 Client Interface……..……………………………………………….……...34

**Chapter 5:Conclusion & Future Scope………………………………………………40-41**

5.1 Conclusion…………………………………………………………………40

5.2 Future Scope of the System………………………………………………..40

**Apendix………………………………………………………………………………………………………….**

**References…………………………………………………………………………………………………….**

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| Tables No. | Name | Page no. |  |
| 2.1 | **Hardware Description** | **4** |  |
| 2.2 | **Platform Support** | **4** |  |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| Figure No. | Name |  | Page no. |
| 2.1 | **Admin: Use Case** |  | **7** |
| 2.2 | **Agent: Use Case** |  | **8** |
| 2.3 | **Client: Use Case** |  | **9** |
| 2.4 | **DFD-0 level** |  | **10** |
| 2.5 | **DFD-1 level** |  | **11** |
| 2.6 | **Entity relationship Diagram** |  | **12** |
| 2.7 | **Admin: Activity** |  | **20** |
| 2.8 | **Customer: Activity** |  | **20** |
| 2.9 | **Agent: Activity** |  | **21** |
| 2.10 | **Sequence Diagram** |  | **21** |
| 2.11 | **Agent Sequence Diagram** |  | **22** |
|  |  |  |  |
|  |  |  |  |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| Abbreviation | Expanded Form |
| J2EE | **Java 2 Enterprise Edition** |
| JSP | **Java Server Pages** |
| IDE | **Integrated Development Environment** |
| CSS | **Cascading Style Sheet** |
| HTML | **Hyper Text Markup Language** |
|  |  |

**ABSTRACT**

The Insurance Premium calculator is a major component in the electronic Systems of Insurance Companies. The Insurance agent’s Samurai is the system meant for insurance agent that helps to automated most of the functionalities in the insurance policy related activities. This system should have the facility to enter the customer details who wants to take an insurance scheme. The system should be able to provide necessary illustrations and calculate the premium amount. This system should also provide the facility to show the specification of premium term as per the choice of premium value to be paid. If the customer decides to go for the selected insurance plan, the system should be able to generate the filled application form in MS word format.

**Tools & language used**: HTML, CSS, JavaScript, Advance Java (J2EE), MYSQL, Microsoft Visual Studio 2010, JSP, Eclipse IDE for J2EE, Apache Tomcat Server 7.0.

**CHAPTER 1: INTRODUCTION**

Insurance Agent Samurai is a web application. The application provides an ease to the agents to store the information about the customer, his policy and his illustration. The application also manages the information of agents, customers, policy, and gives instant status report of policy in activation.

* 1. **Need of Project:**

Insurance policy activities are too complex to handle and very time consuming .The customer have to face a uneasy procedure for associating with the policy through the agent. It consist selection, choosing, payment, verification and so many other operations that are traditionally being done manually and thus the process is exhausting. The agent has too many action that are to be performed for the each individual client and policy and it increases load and by this the insurance procedure doesn’t have good performance.

Thus to reduce the complexity and building an alternative to replace the exhaustive system, Insurance agent samurai comes into action. It tries to overcome the issues that are found in the manual process of insurance. Its main task is to bring in the most facilities in the automated form mechanisms to reduce the response time.

Our system solves this problem as it takes the data through the computer and store that on database. If one person wants to view, insert or edit data he just need to logon to the system. If there is a need of analysis it can also be done very efficiently.

* 1. **Scope of Project:**

The scope of the project is limited insurance agencies or companies as it’s made for specific use. Along with its basic functionality the application also integrated with different features that will be further discussed in the report. The IAS system builds a narrow time activation bridge between the customer and the agent and as well as the company. It’s easy to process through this system for the agent.

* 1. **Project Objectives:**

To provide a user friendly environment that should perform following tasks:

* Store the records, customer details, associated policy details at a centralized data store.
* System should be able to process more than multiple policy of a single client.
* System should be scalable and flexible.
* There should be different views of system for different types of users.
* There may a scope of alteration in the policy activations.
* System should be aware of every client illustration of policy and deductibles.
  1. **Key Features:**
* Information of a customer for each policy is stored.
* Show the illustrations for the customer about his policy activations.
* Easy to handle for the managing agents.
* Auto & manual updating of illustrations.
* Each associated policy will be consistent to each client.



**CHAPTER 2: PROJECT DESCRIPTION**

* 1. **Application Requirements:**

[**Hardware Requirements**](javascript:void(0))**:**

|  |  |  |
| --- | --- | --- |
| **Hardware requirements** | **Version 4**  **Full** | **Version 4**  **Client** |
| **Processor** |  |  |
| Minimum | 1 GHz | 1 GHz |
| Recommended | 1 GHz | 1 GHz |
| **RAM** |  |  |
| Minimum | 512 MB | 512 MB |
| Recommended | 512 MB | 512 MB |
| **Disk space (minimum)** |  |  |
| 32-bit | 850 MB | 600 MB |
| 64-bit | 2 GB | 1.5 GB |

***Table 2.1: Hardware requirements***

**Platform Support:**

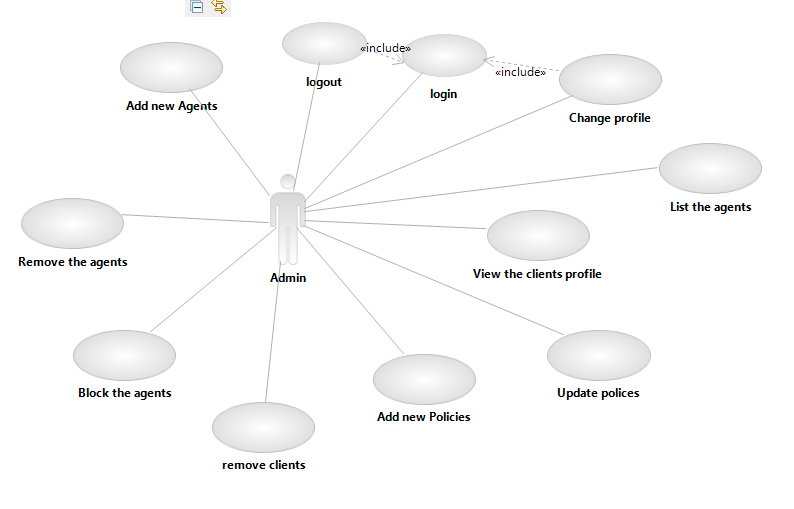
|  |  |
| --- | --- |
| **Operating system** | **Version 4 and**  **Version 4 client** |
| Windows 7 Ultimate x86 | √ |
| Windows 7 Ultimate N | √ |
| Windows 7 Ultimate x64 | √ |
| Windows 7 Enterprise x86 | √ |
| Windows 7 Enterprise N | √ |
| Windows 7 Enterprise x64 | √ |
| Windows 7 Professional x86 | √ |
| Windows 7 Professional N | √ |
| Windows 7 Professional x64 | √ |
| Windows 7 Home Premium x86 | √ |
| Windows 7 Home Premium N | √ |
| Windows 7 Home Premium x64 | √ |
| Windows7 Home Basic x86 | √ |
| Windows 7 Home Basic N | √ |
| Windows 7 Starter x86 | √ |
| Windows 7 Starter N | √ |
| Windows Vista Ultimate | √ |
| Windows Vista Ultimate x64 Edition | √ |
| Windows Vista Enterprise | √ |
| Windows Vista Enterprise x64 Edition | √ |
| Windows Vista Business | √ |
| Windows Vista Business x64 Edition | √ |
| Windows Vista Home Premium | √ |
| Windows Vista Home Premium x64 Edition | √ |
| Windows Vista Home Basic | √ |
| Windows Vista Starter Ed. Digital Boost | √ |
| Windows Vista Starter Ed. Digital Boost (x64) | √ |
| Windows Vista Starter | √ |
| Windows XP Professional | √ |
| Windows XP Professional x64 Edition | √ |
| Windows XP Home Edition | √ |

***Table 2.2: Platform Support***

* 1. **User Characteristics:**

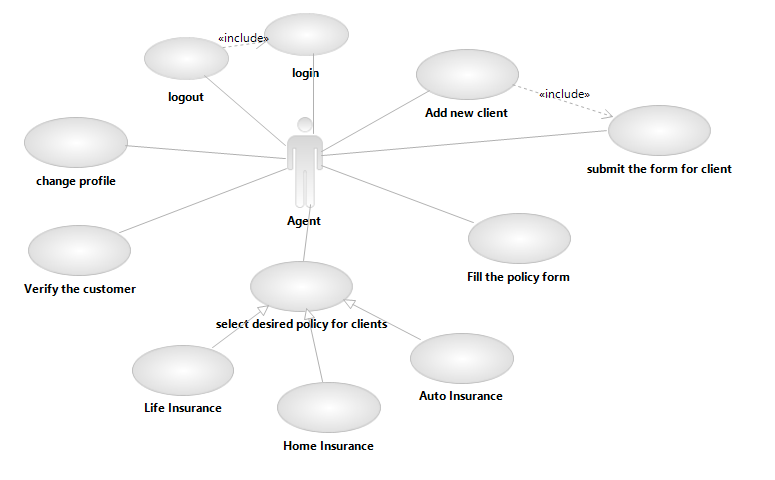
There are basically three users Client, Agent and Admin. More user types and there rights can also be added by the admin only. Login is required for all the users to access the system. The basic tasks of them are as following**:**

1. **Admin:** Admin can do each and every task in the system. The basic tasks of him/her are view customer details, create, update, list & delete agents , view agent details & create new agent login.
2. **Agent:**  Agent is the one who make the verification of customer at different levels according to the different type of insurance product for which the customer is applying.
3. **Customer:** Customer is the one who interact with the agent for application and activation of the insurance plan. His/her basic tasks are applying for the insurance product and see the status through the web portal, make premium payments etc.
   * 1. **Admin Use Case:**

****

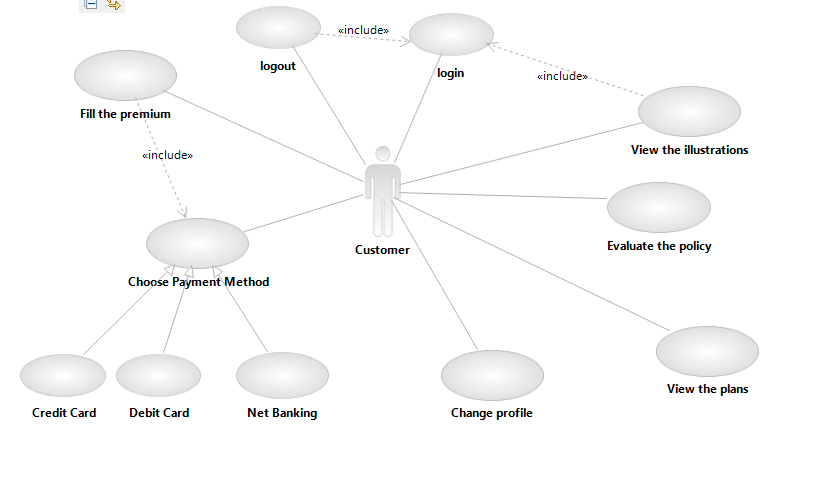
***Fig 2.1: Admin :Use case diagram***

* + 1. **Agent Use Case:**

****

***Fig 2.2: Agent :Use case diagram***

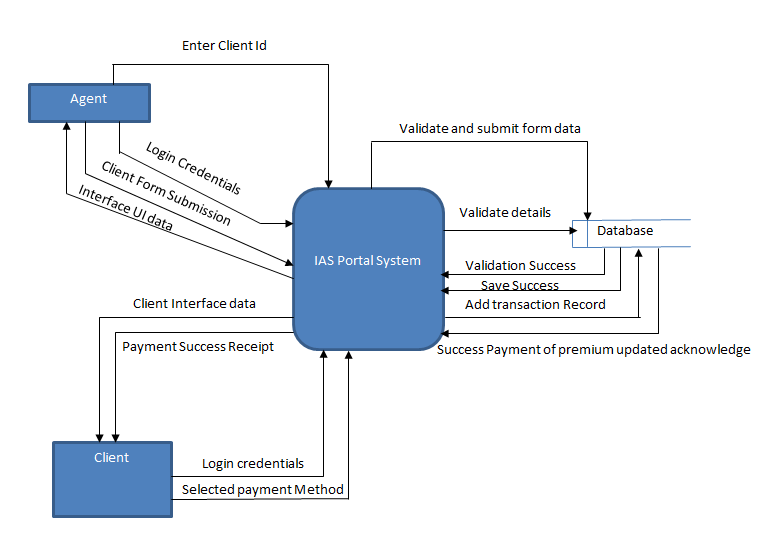
* + 1. **Client Use Case:**

****

***Fig 2.3: Client:Use case diagram***

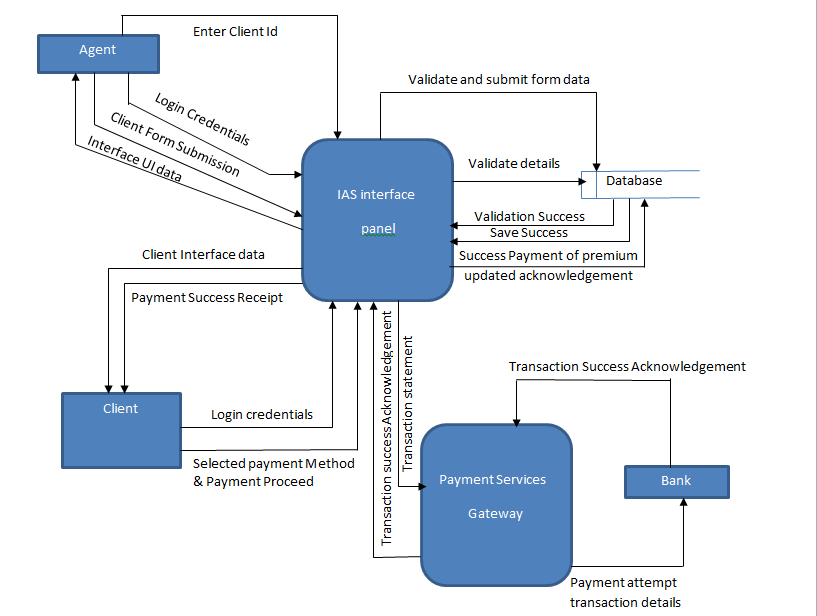
* 1. **Data Flow Diagram:**

**Level 0:**

****

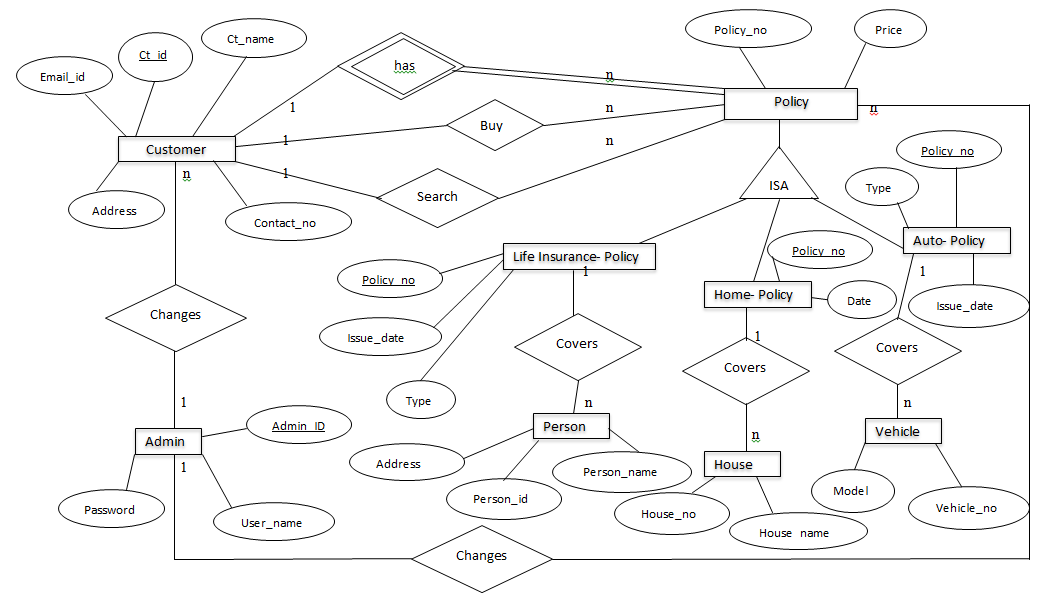
***Fig 2.4: Level-0 Data Flow Diagram***

**Level 1:**

******

***Fig 2.5: Level-1 Data Flow Diagram***

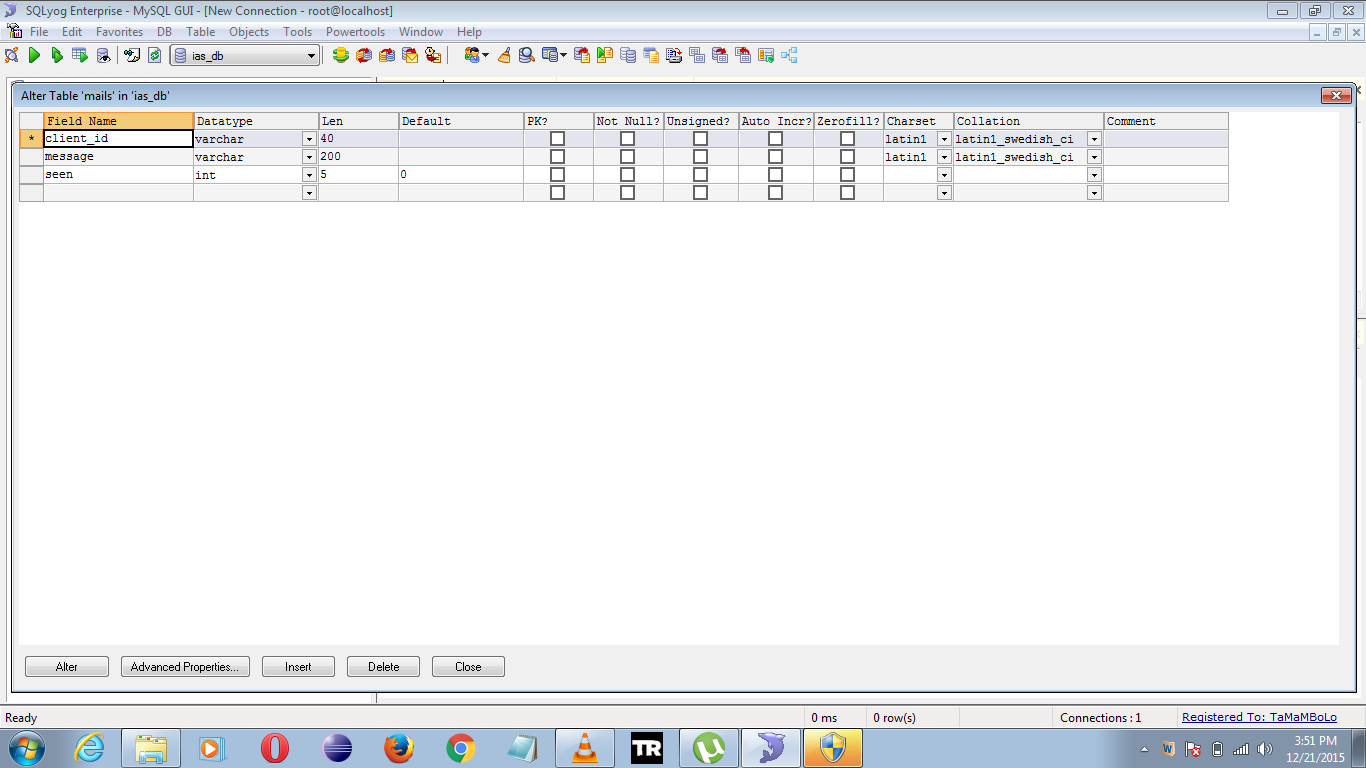
* 1. **Database Design:**

****

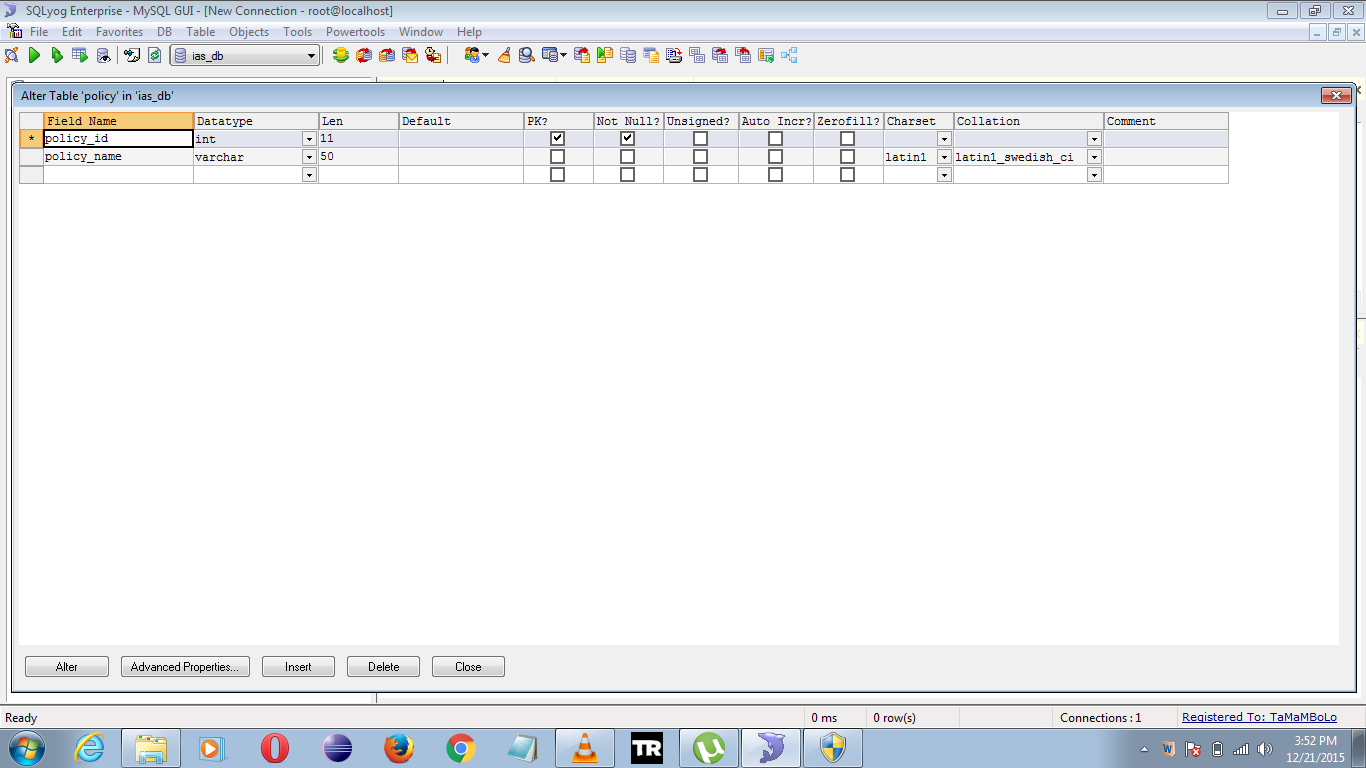
***Fig 2.6: Entity Relationship Diagram***

**Tables in Database:**

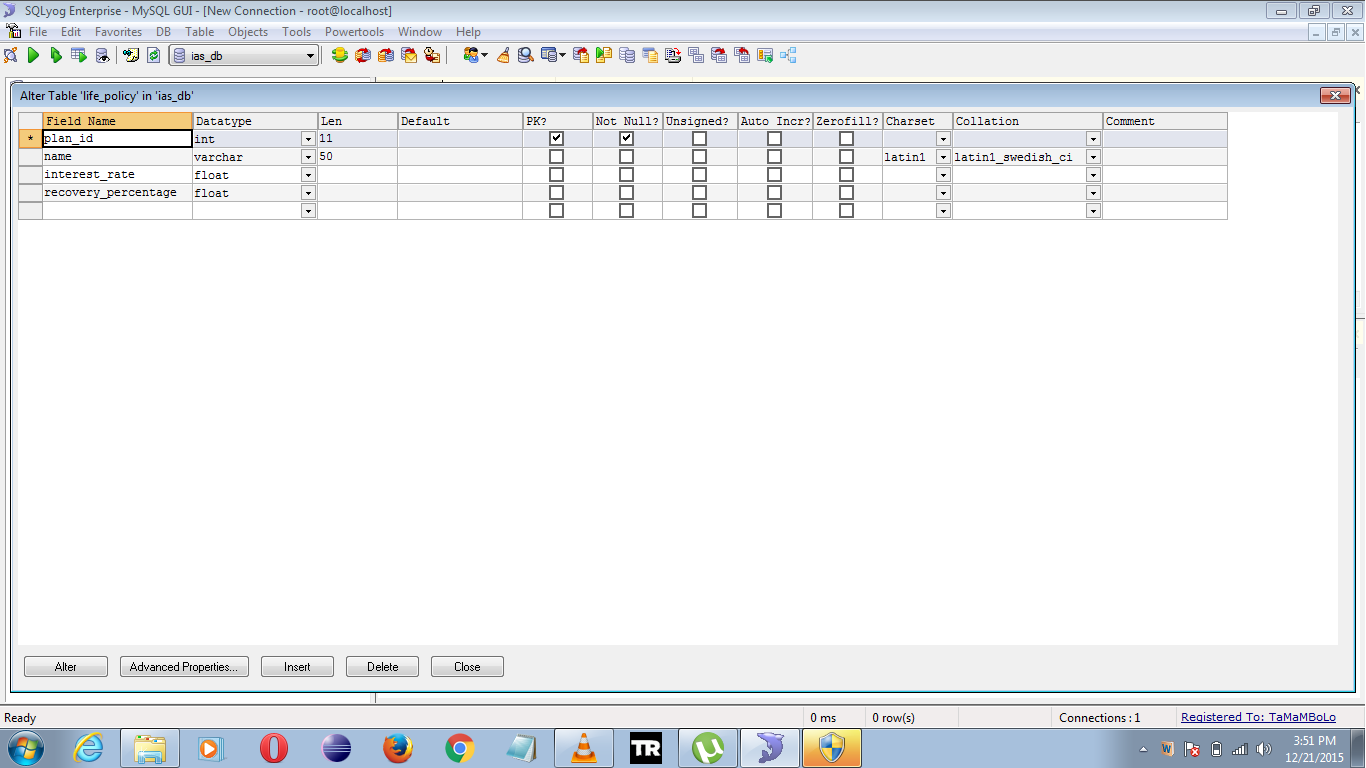
1. **Mails:** Store the brief information of mails.

******

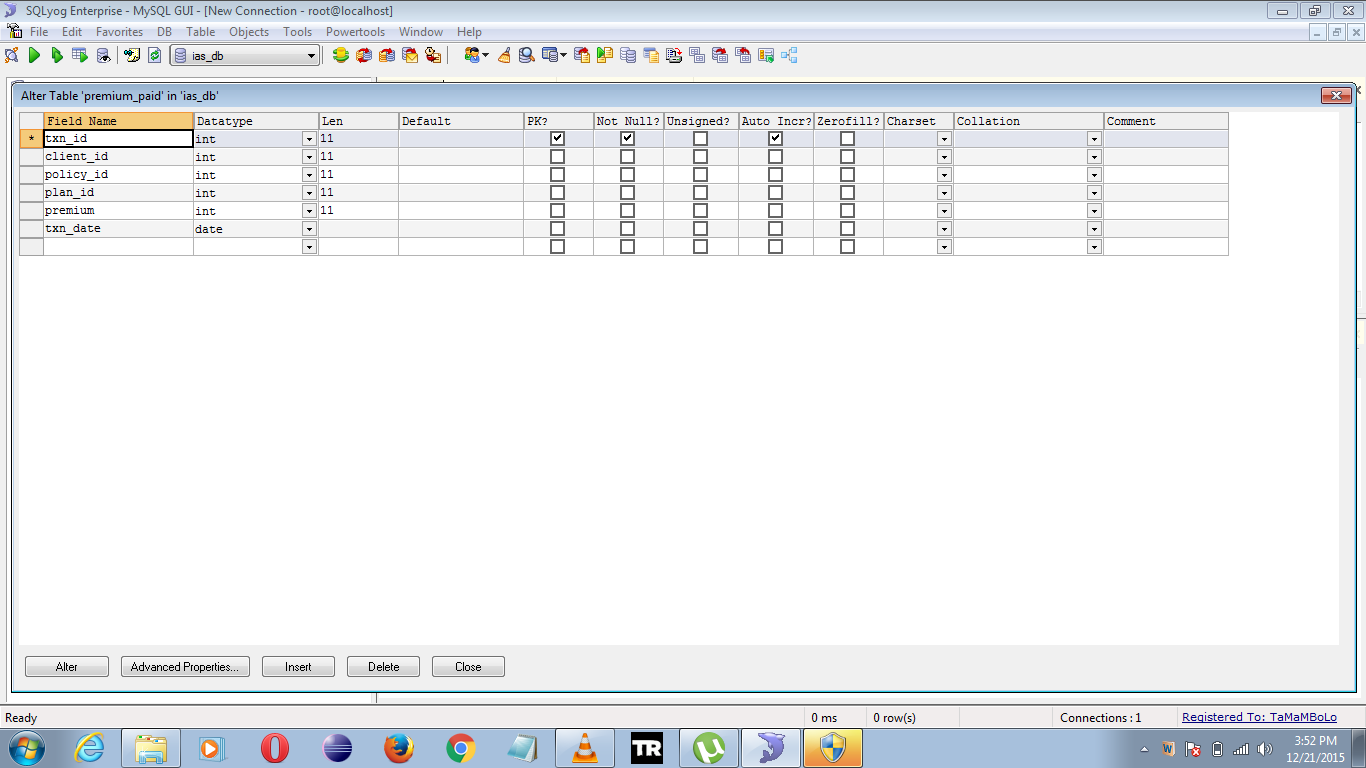
1. **Policy :** Store the information about of type of policies.

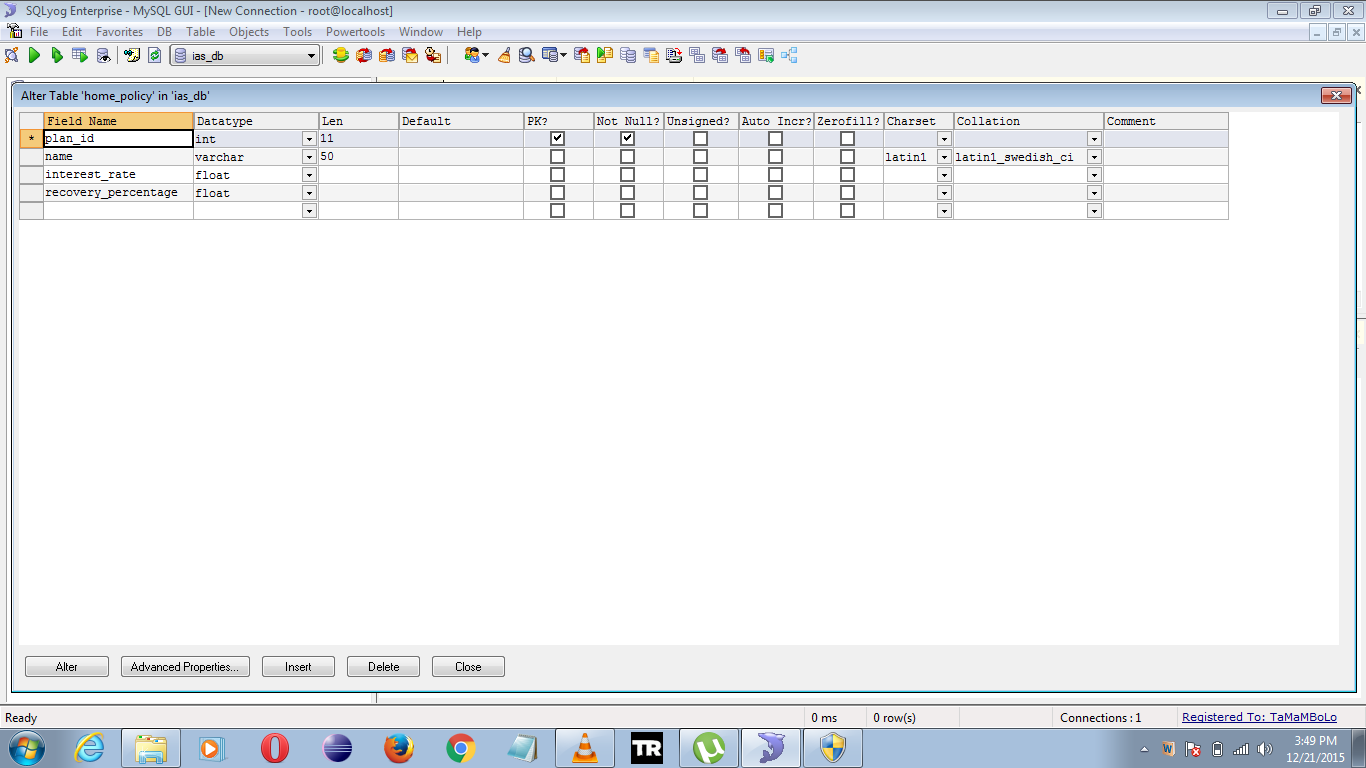
******

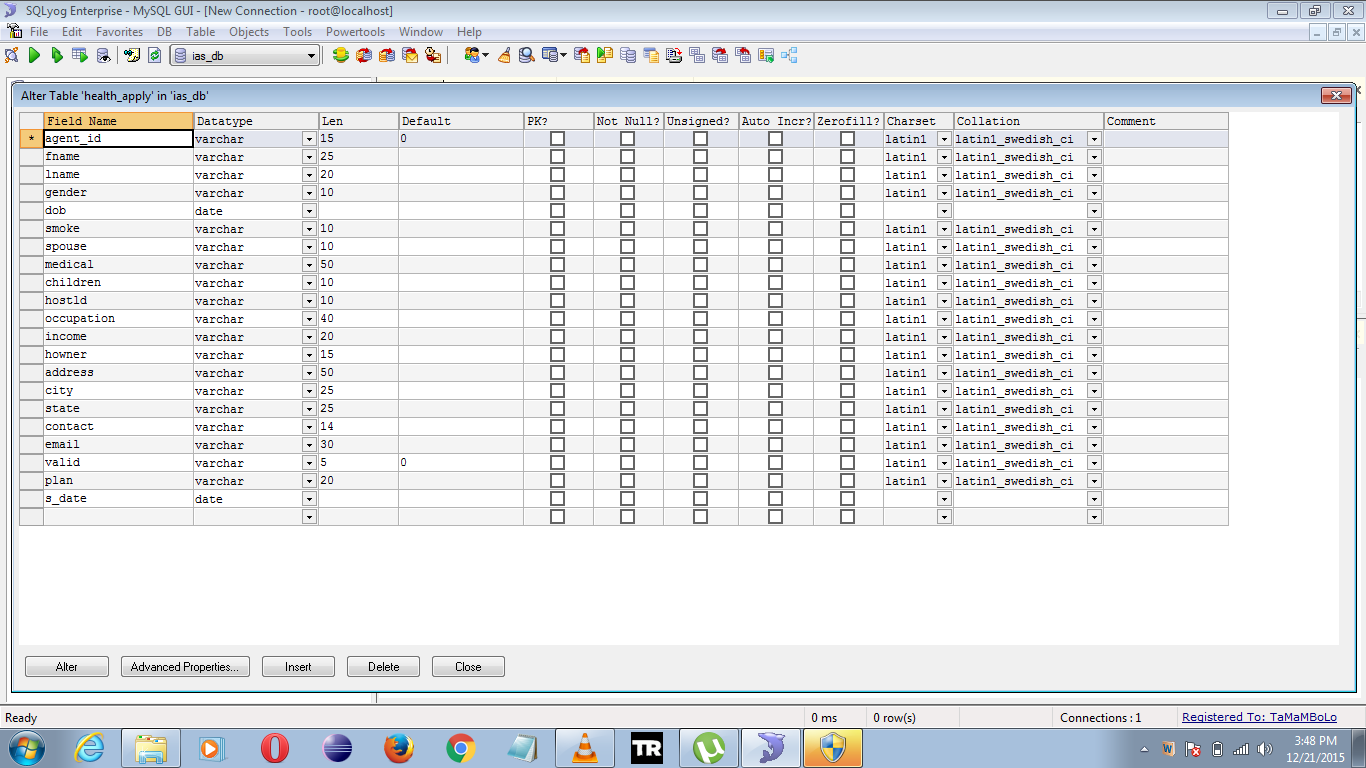
1. **Life policy:** This table contains the plan’s info available in the life policy department.

******

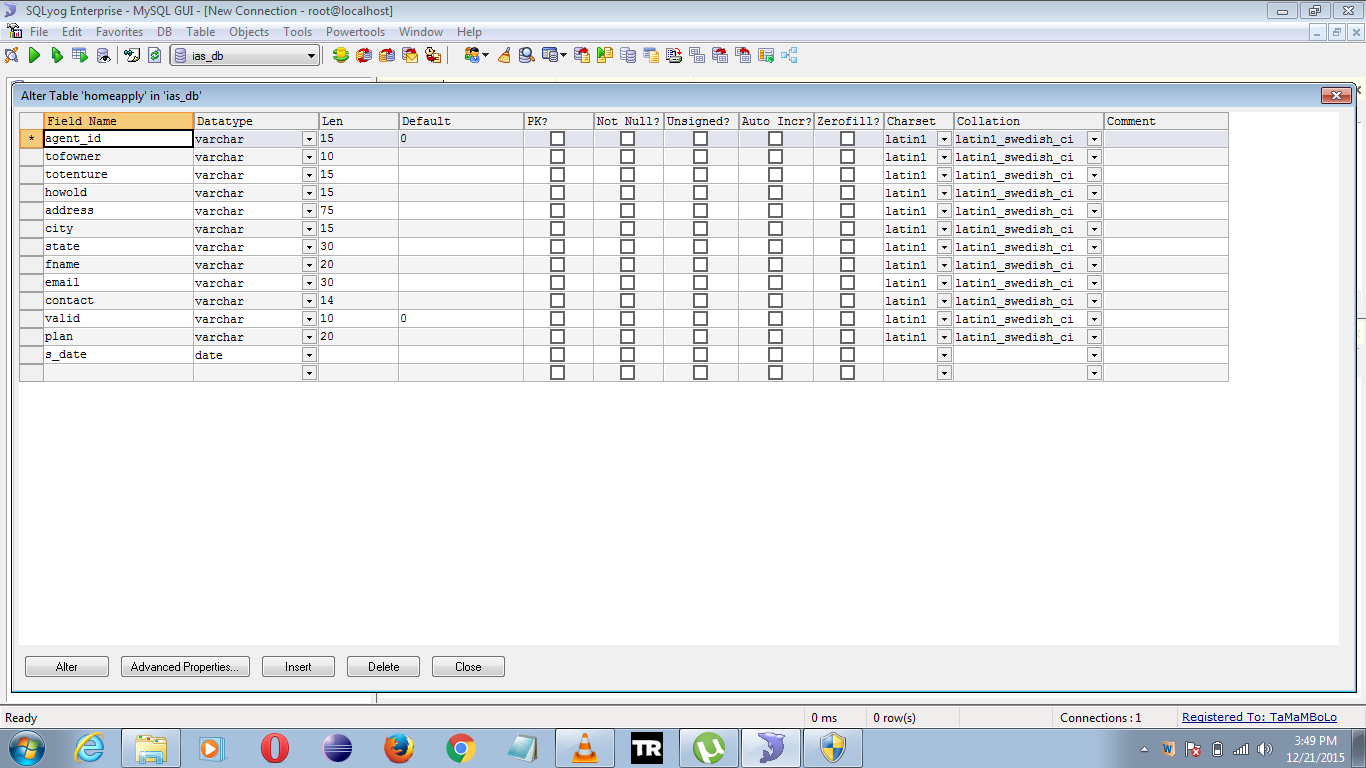
1. **Premium Paid:** This table contains the information of the premium transaction of all the clients.

******

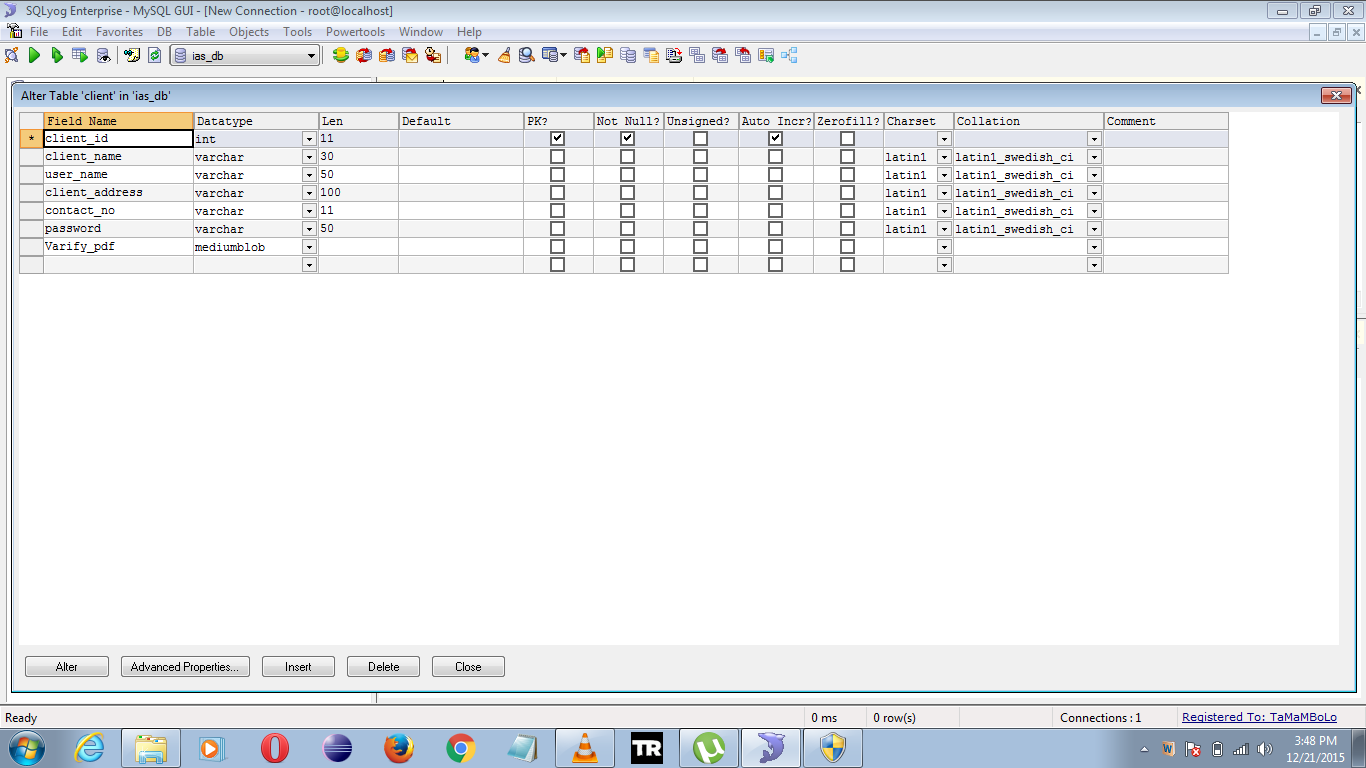
1. **Home policy:** This table contains the plan’s info of home policy. ******
2. **Health apply:** contains the detailed information related to home policy clients.

******

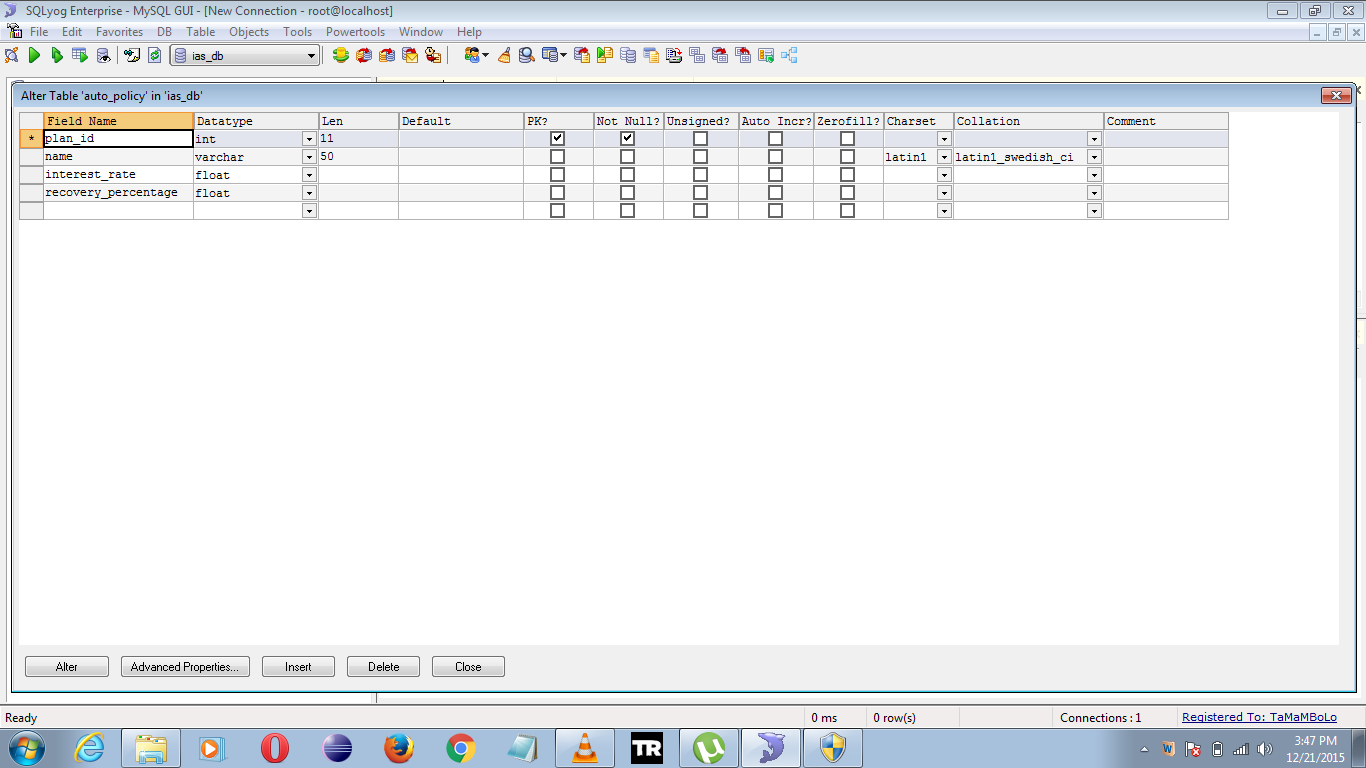
1. **Home apply:** contains the detailed info about the home policy clients

******

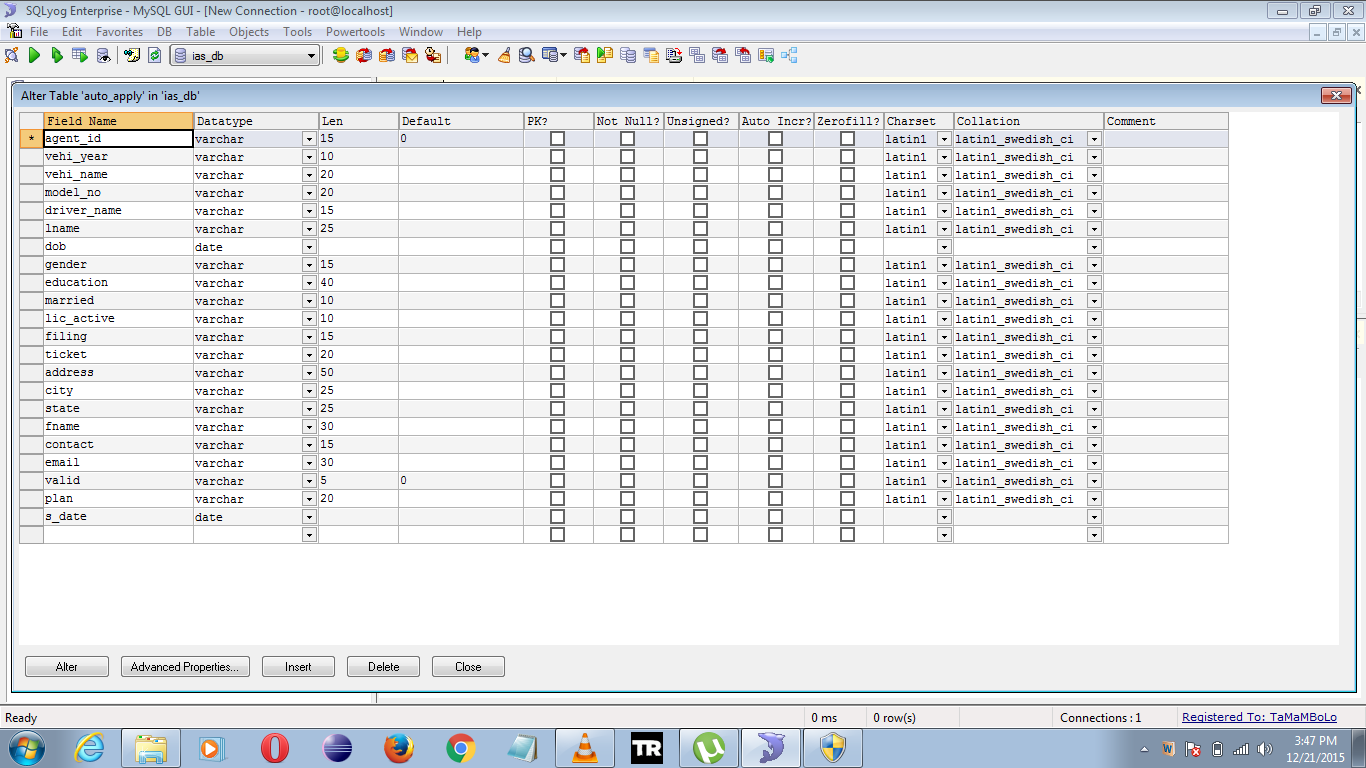
1. **Client:** Stores the general registration details of client.

******

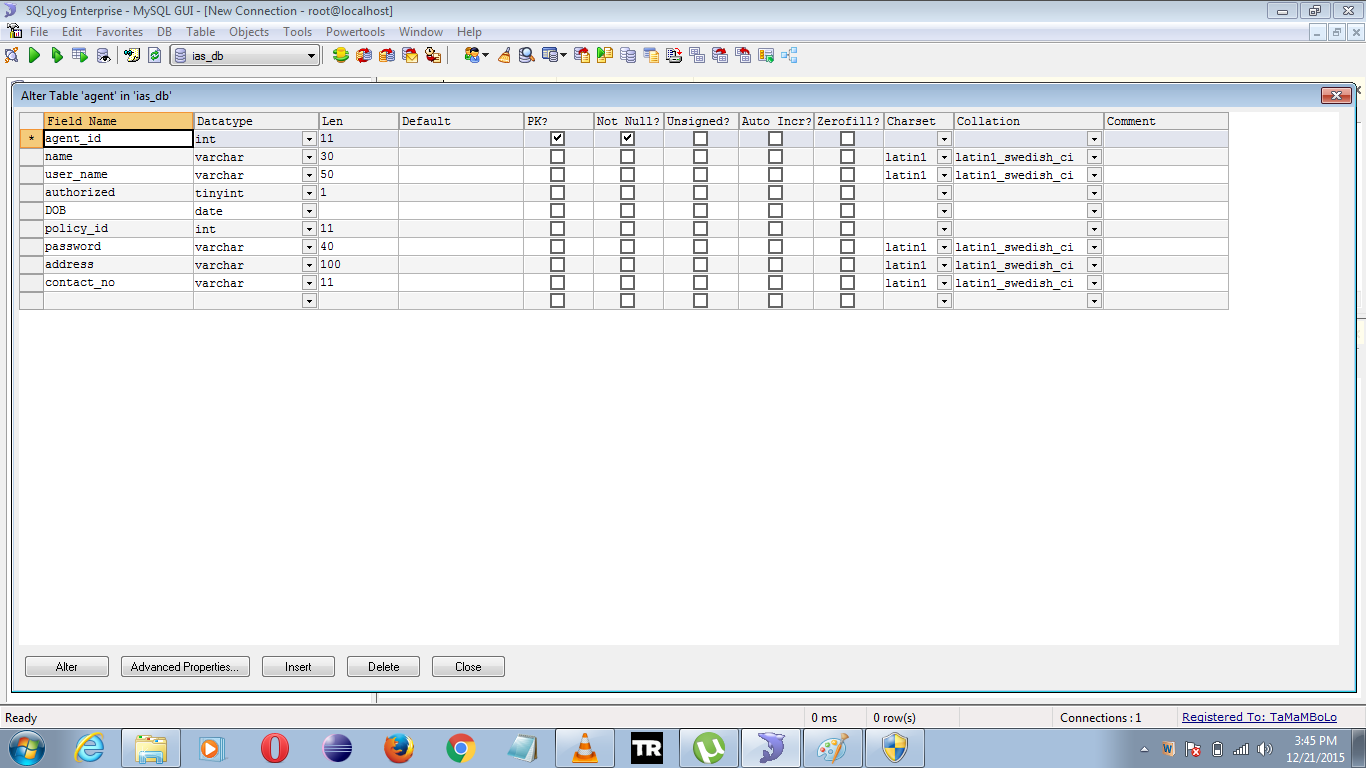
1. **Auto Policy:** contains the auto policy plans information.

******

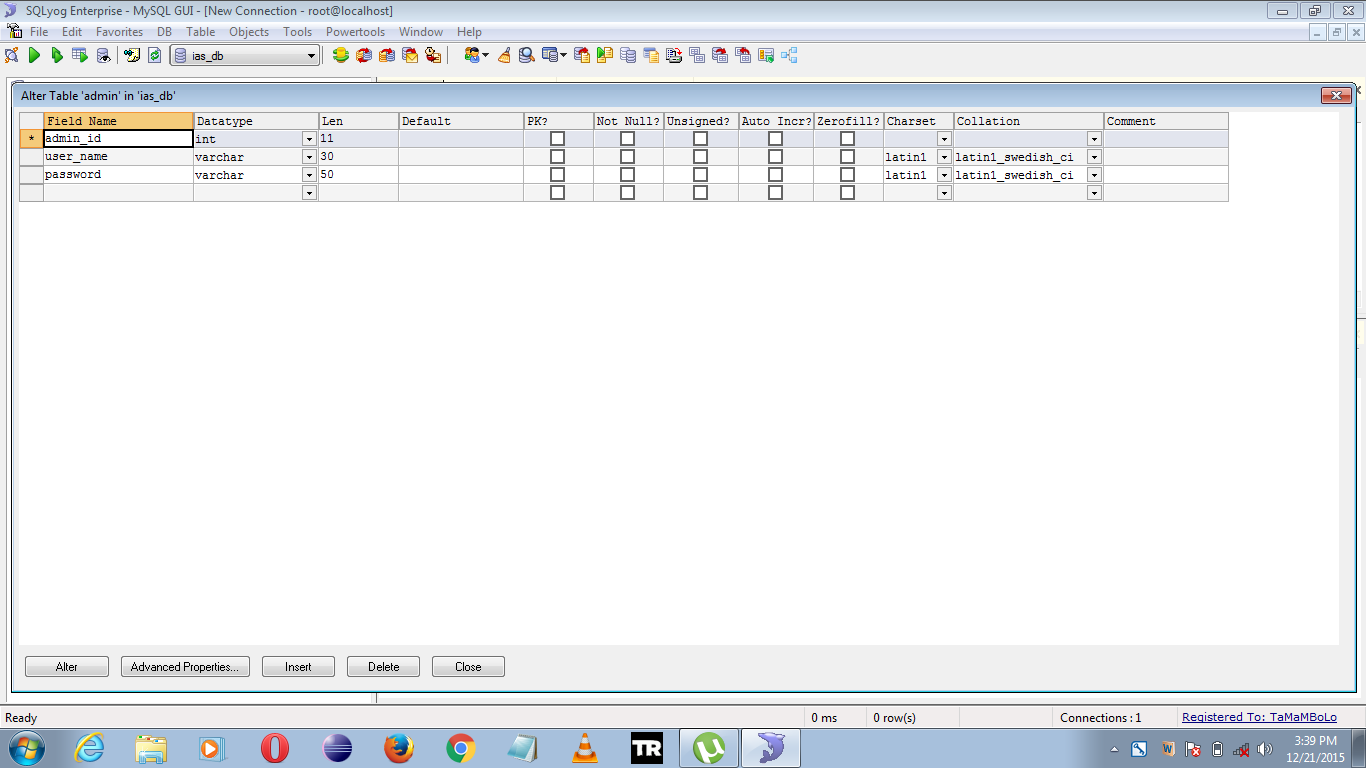
1. **Auto apply:** contains the detailed info of auto insurance clients.

******

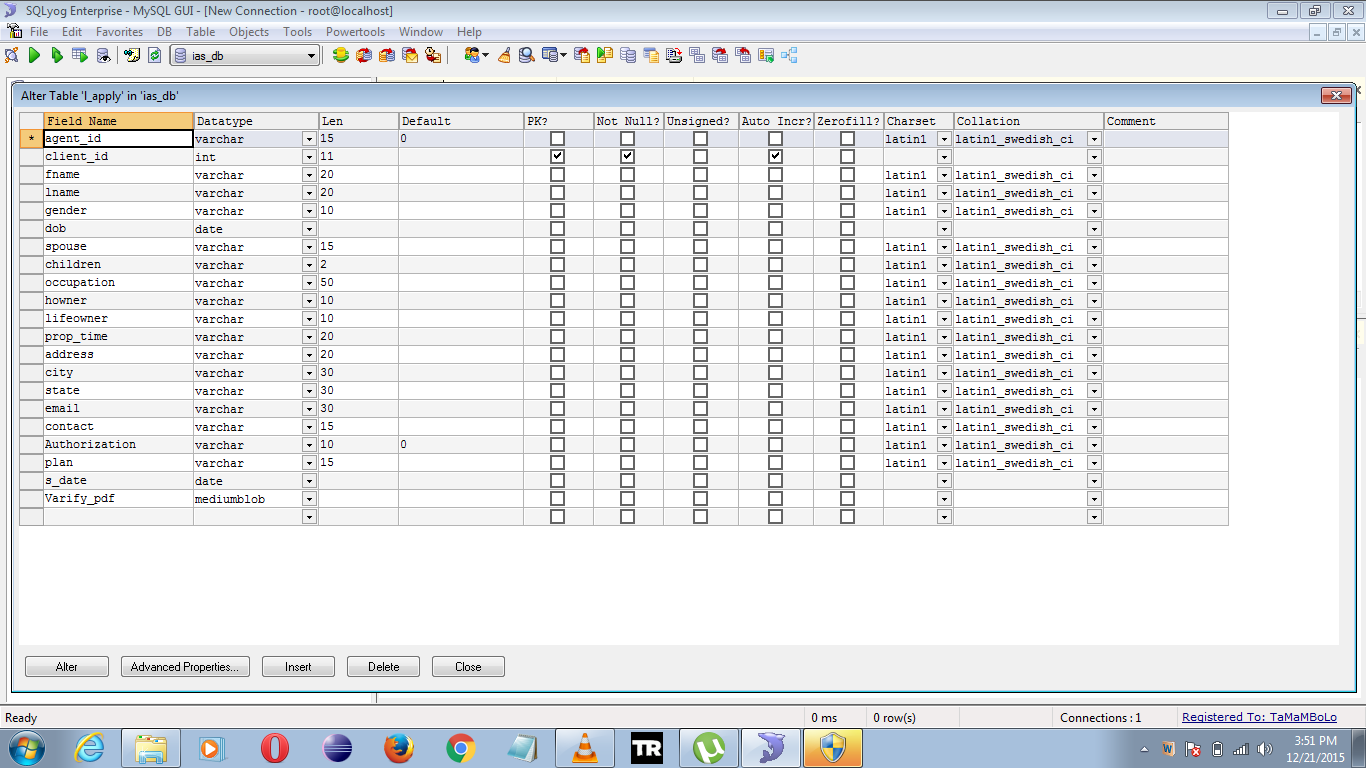
1. **Agent:** contains the registration details of the insurance agents of all departments.

******

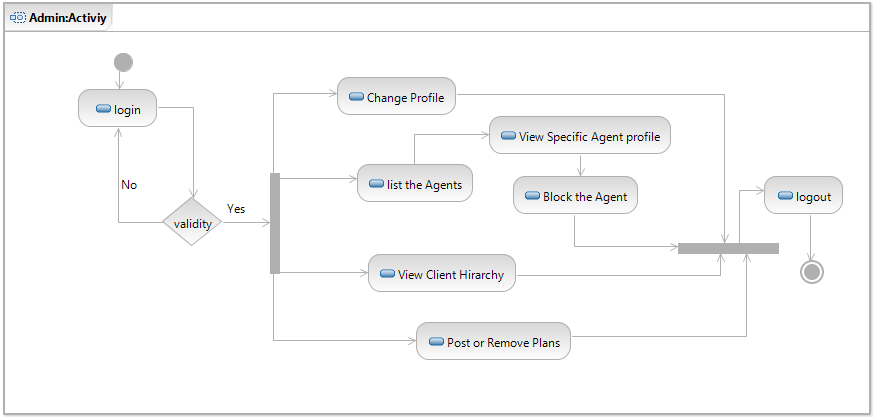
1. **Admin :** contains sensitive admin information.

******

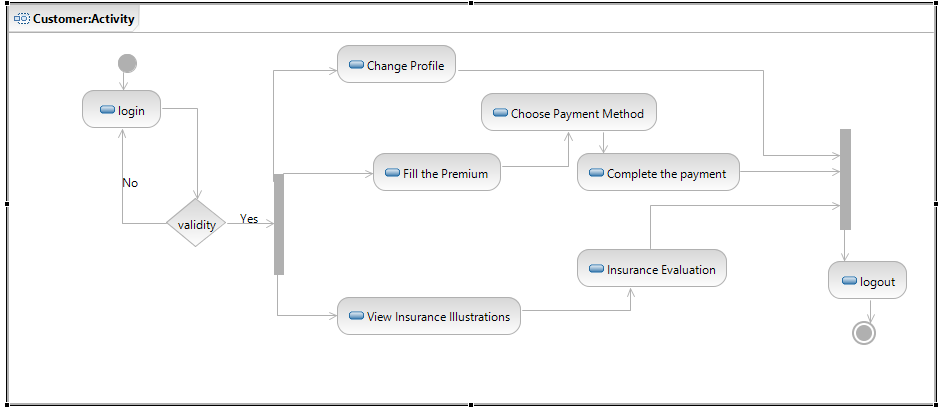
1. **Life apply:** stores the detailed information of life policy clients.

******

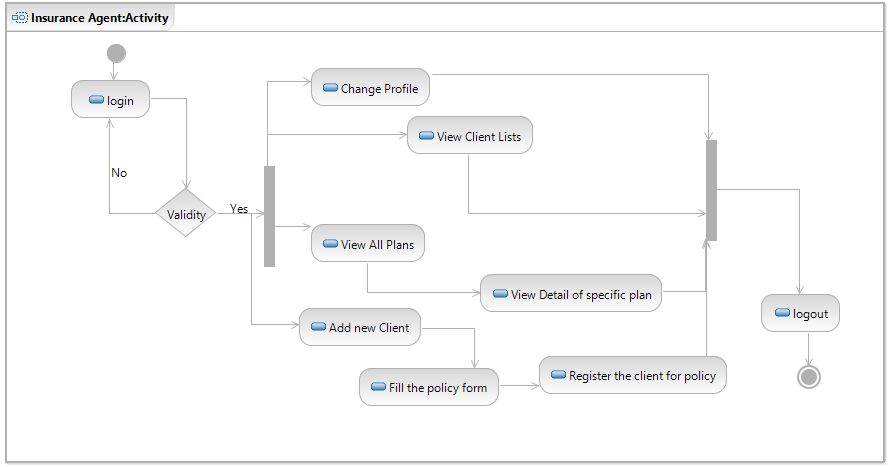
* 1. **Flow Procedure:**

****

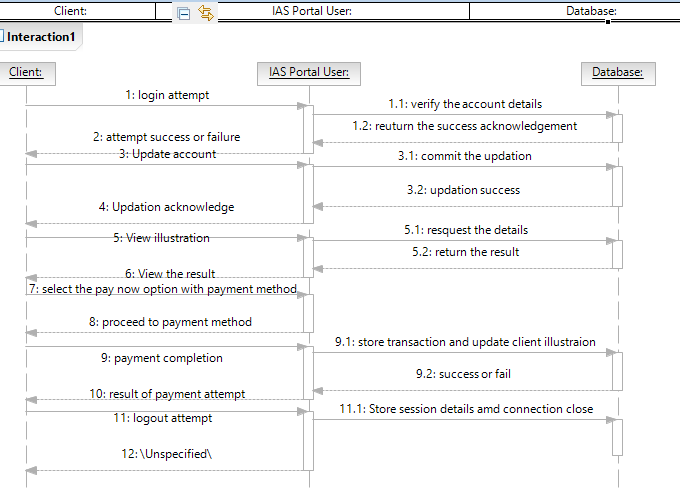
***Fig 2.7: Admin Activity***

****

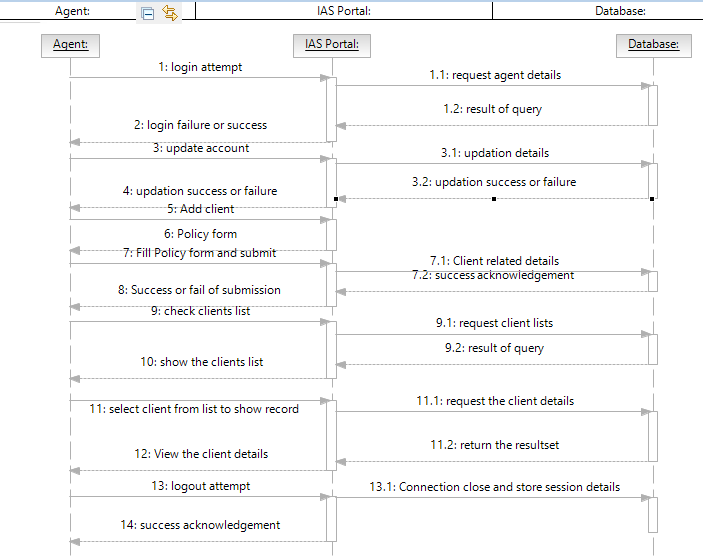
***Fig 2.8: Customer Activity***

****

***Fig 2.9: Agent Activity***

****

***Fig 2.10: Sequence diagram***

****

***Fig 2.11: Agent Sequence Diagram***

**CHAPTER 3: TECHNOLOGIES USED**

* 1. **HTML:**

HTML stands for Hyper Text Markup Language. HTML is not a programming language, it is a markup language.

A markup language is a set of markup tags. HTML uses markup tags to describe web pages. It is extensively used in the web designing for the web applications.

It has a huge collection of tags which can be used to describe the web pages in a wide variety and with a huge functionality.

* 1. **JSP:**

Java Server Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI).

But JSP offer several advantages in comparison with the CGI. Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having a separate CGI files.

JSP are always compiled before it's processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.

Java Server Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including JDBC, JNDI, EJB, JAXP etc. JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

JSP is an integral part of J2EE, a complete platform for enterprise class applications. This means that JSP can play a part in the simplest applications to the most complex and demanding.

* 1. **Microsoft SQL Server:**

Microsoft SQL Server is a [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft). As a [database server](https://en.wikipedia.org/wiki/Database_server), it is a [software product](https://en.wikipedia.org/wiki/Software_product) with the primary function of storing and retrieving data as requested by other [software applications](https://en.wikipedia.org/wiki/Software_application) which may run either on the same computer or on another computer across a network (including the Internet).

SQL Server 2005 includes native support for managing [XML](https://en.wikipedia.org/wiki/XML) data, in addition to [relational data](https://en.wikipedia.org/wiki/Relational_database).

For this purpose, it defined an xml [data type](https://en.wikipedia.org/wiki/Data_type) that could be used either as a data type in database columns or as [literals](https://en.wikipedia.org/wiki/Literal_(computer_science)) in queries.

XML columns can be associated with [XSD](https://en.wikipedia.org/wiki/XSD) schemas; XML data being stored is verified against the schema.

* 1. **Visual Studio 2010 Ultimate:**

Microsoft Visual Studio is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs](https://en.wikipedia.org/wiki/Computer_program) for [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), as well as [web sites](https://en.wikipedia.org/wiki/Web_site), [web applications](https://en.wikipedia.org/wiki/Web_application) and [web services](https://en.wikipedia.org/wiki/Web_service).

Visual Studio uses Microsoft software development platforms such as [Windows API](https://en.wikipedia.org/wiki/Windows_API), [Windows Forms](https://en.wikipedia.org/wiki/Windows_Forms), [Windows Presentation Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation), [Windows Store](https://en.wikipedia.org/wiki/Windows_Store) and [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight). It can produce both [native code](https://en.wikipedia.org/wiki/Native_code) and [managed code](https://en.wikipedia.org/wiki/Managed_code).

Visual Studio 2010 Ultimate comes with [.NET Framework 4](https://en.wikipedia.org/wiki/.NET_Framework_4) and supports developing applications targeting [Windows 7](https://en.wikipedia.org/wiki/Windows_7).

 It supports [IBM DB2](https://en.wikipedia.org/wiki/IBM_DB2) and [Oracle](https://en.wikipedia.org/wiki/Oracle_Database) databases, in addition to [Microsoft SQL Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server). It has integrated support for developing [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight) applications, including an interactive designer.

* 1. **Java Script:**

JavaScript a client side scripting language. Alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS), it is one of the three essential technologies of [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web) content production; the majority of [websites](https://en.wikipedia.org/wiki/Website) employ it and it is supported by all modern [web browsers](https://en.wikipedia.org/wiki/Web_browser) without [plug-ins](https://en.wikipedia.org/wiki/Browser_extension).

JavaScript is [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) with [first-class functions](https://en.wikipedia.org/wiki/First-class_functions), making it a [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm) language,supporting [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming),[imperative](https://en.wikipedia.org/wiki/Imperative_programming), and [functional](https://en.wikipedia.org/wiki/Functional_programming) programming styles.

 It has an [API](https://en.wikipedia.org/wiki/Application_programming_interface) for working with text, [arrays](https://en.wikipedia.org/wiki/Array_data_type), dates and [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), but does not include any [I/O](https://en.wikipedia.org/wiki/Input/output), such as networking, storage or graphics facilities, relying for these upon the host environment in which it is embedded.

* 1. **CSS:**

**C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

* 1. **Apache Tomcat Server:**

Apache tomcat often refer to as tomcat is an open source web server developed by apache software foundations. Tomcat implements several java EE specificationas follows:

* Java servlet
* JSP
* Java EL
* Web Socket

And provides a pure java http web server environment for java code to run in.

* 1. **Eclipse IDE:**

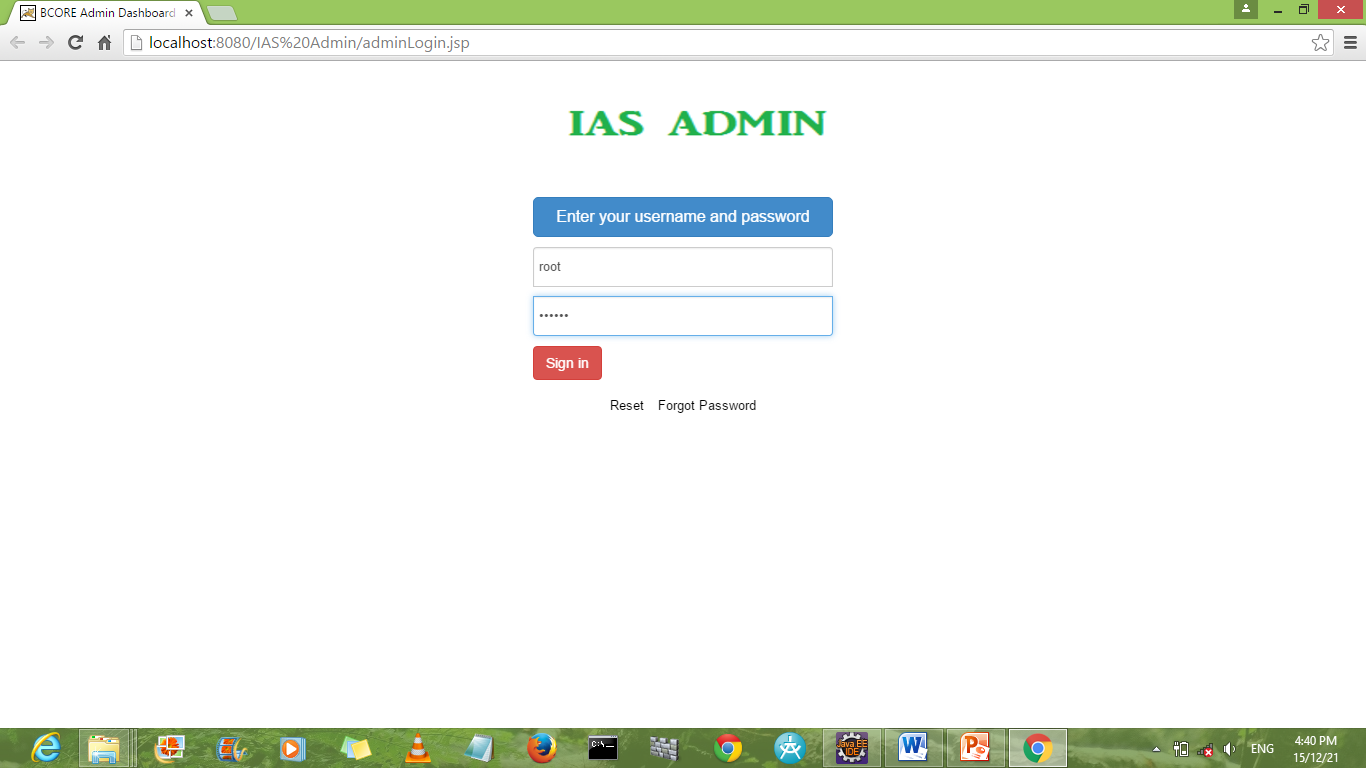
Eclipse is an IDE which contains a base workspace and an extensible plugin system for customizing the environment.

Eclipse is written mostly in java and its primary use is for developing java application but it may also be used to develop application in other programing language through the use of plugin including Ada, C, C++ , FORTRAN, Java Script, Perl, Python etc.

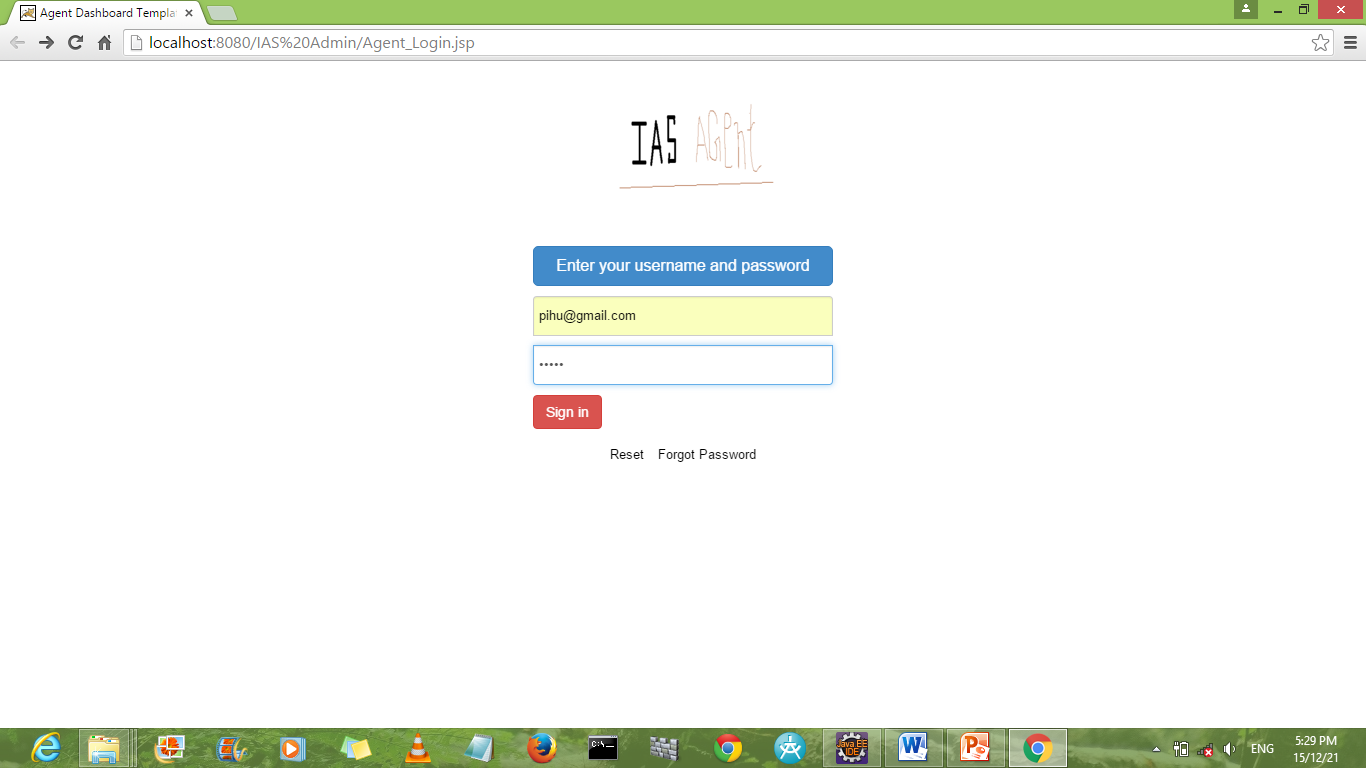
It can also be used to develop package for the software mathematica.Development environments nlude the eclipse java development tool for java and scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP among others.

**CHAPTER 4 – USER INTERFACE**

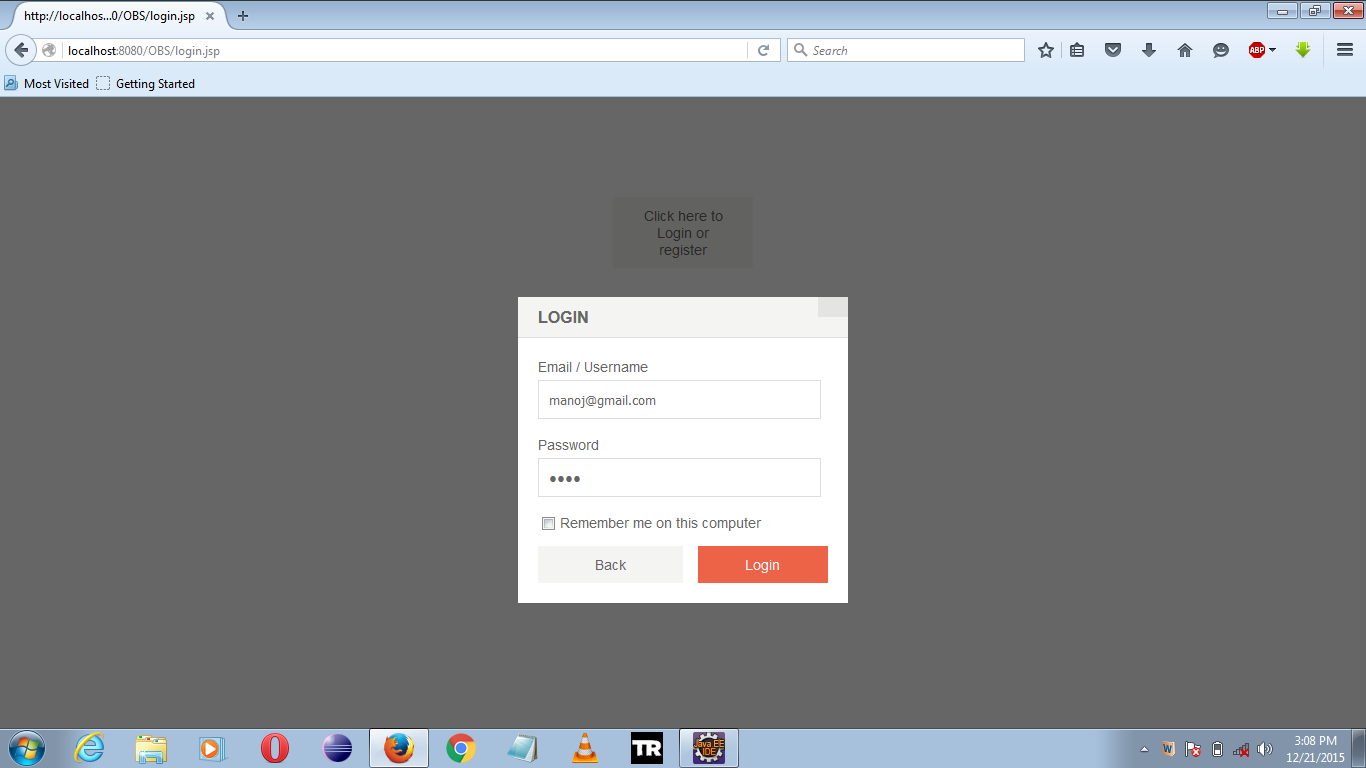
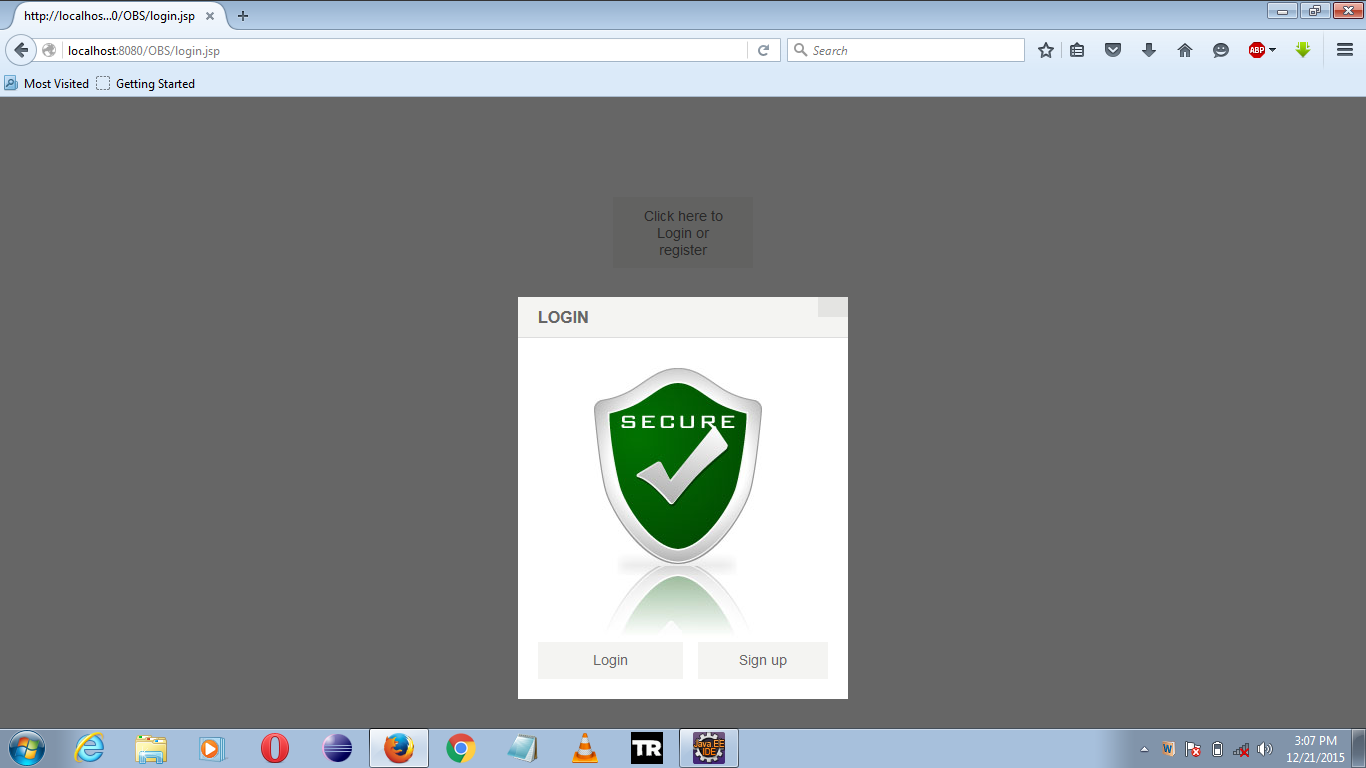
* 1. **User Login:**



***Snapshot 1: Login Page:Admin***



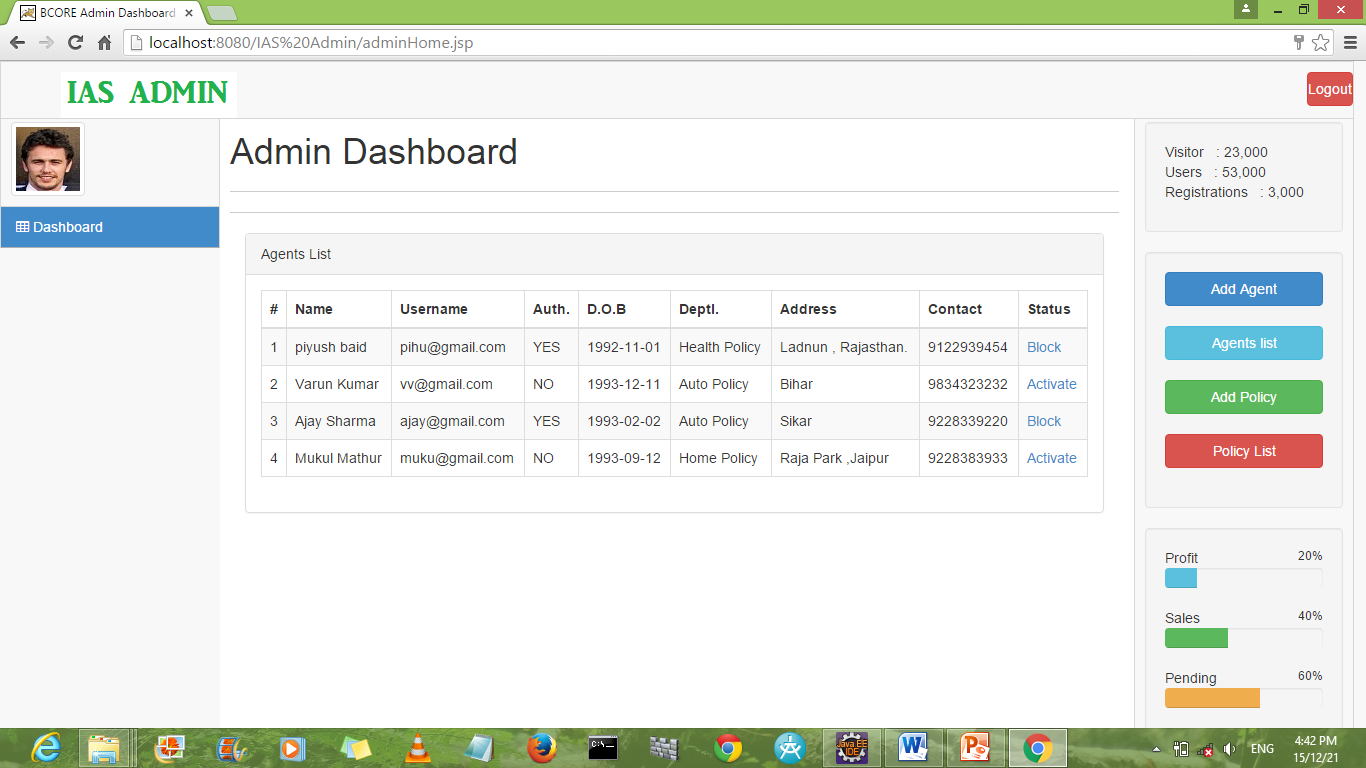
***Snapshot 2: Login Page:Agent***

******

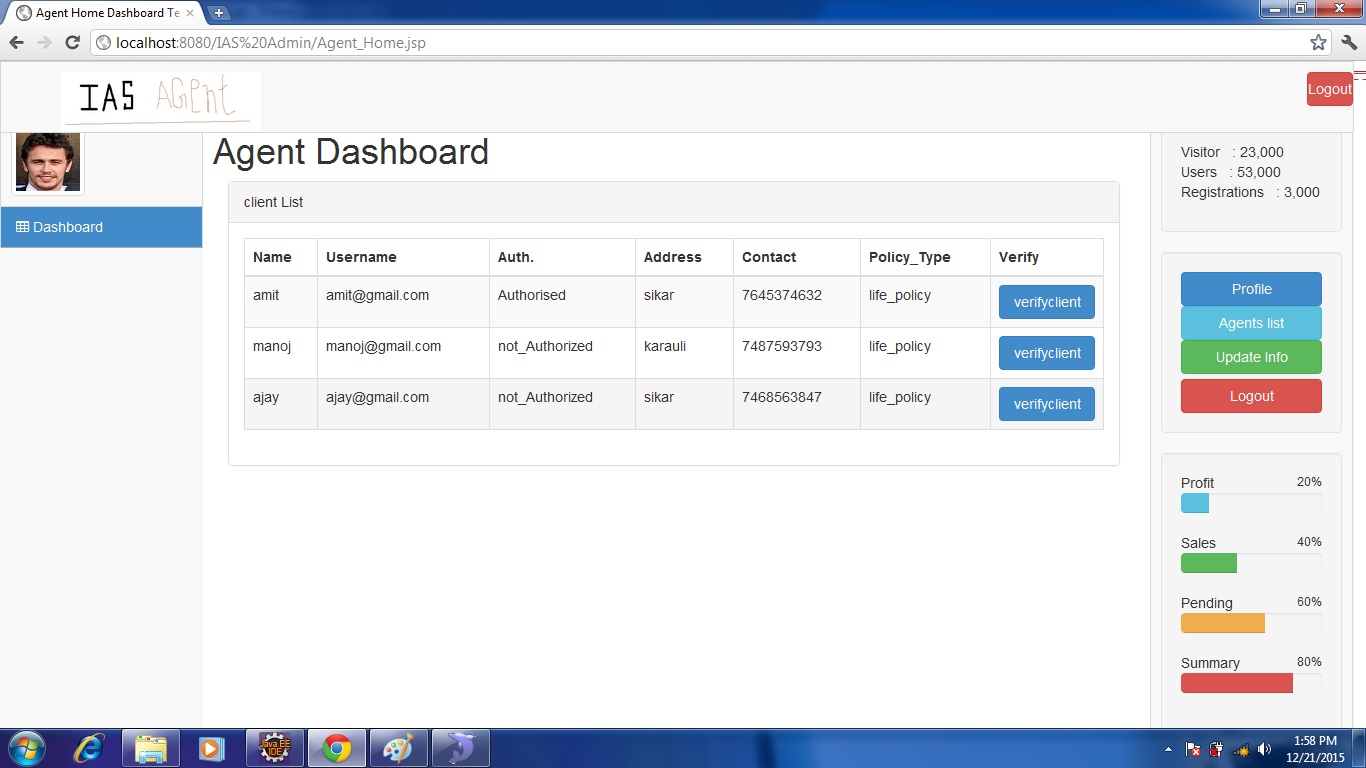
***Snapshot 3: Login Page:Client***

Login page is the first page appears when a use login to the system. User has to enter his correct user name and password to work on the system. Our system provides secure log in for the system.

* 1. **Dashboard:**

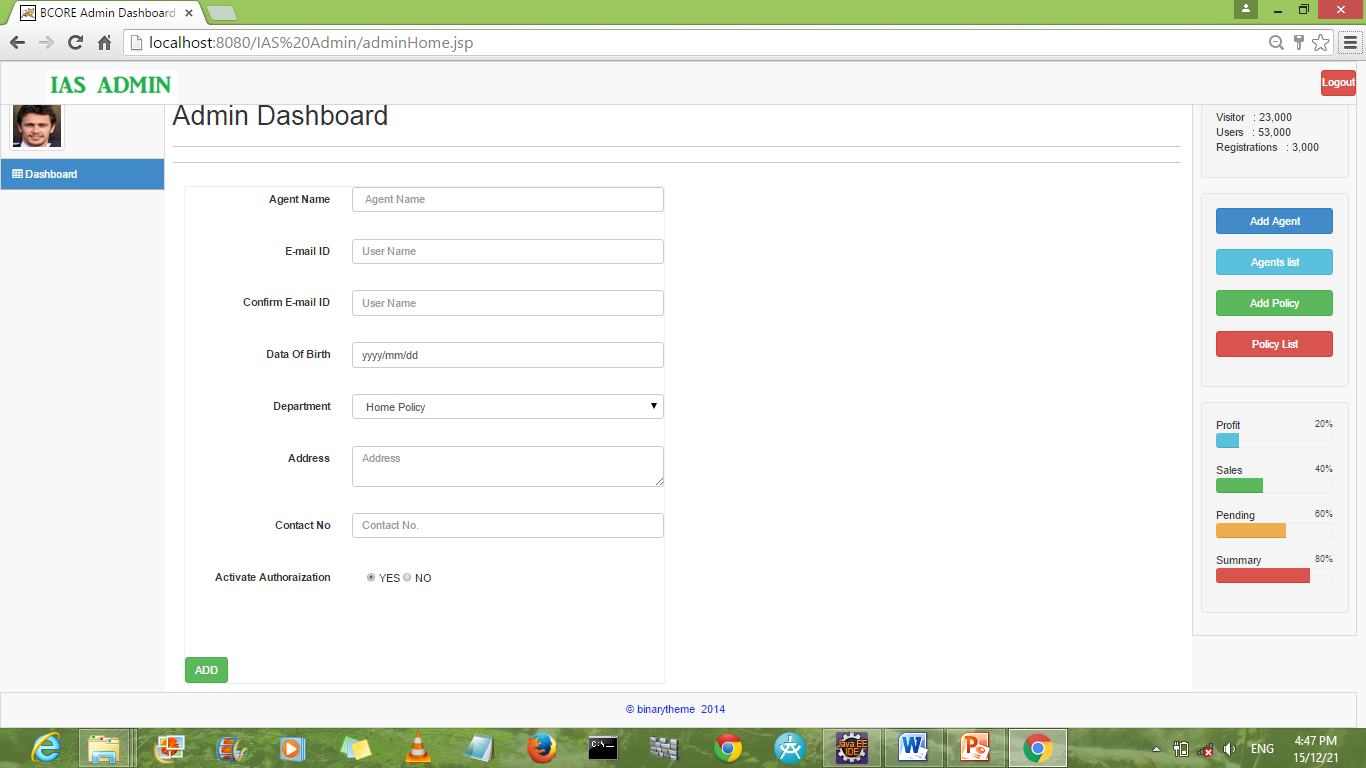


***Snapshot 4: Admin Dashboard***

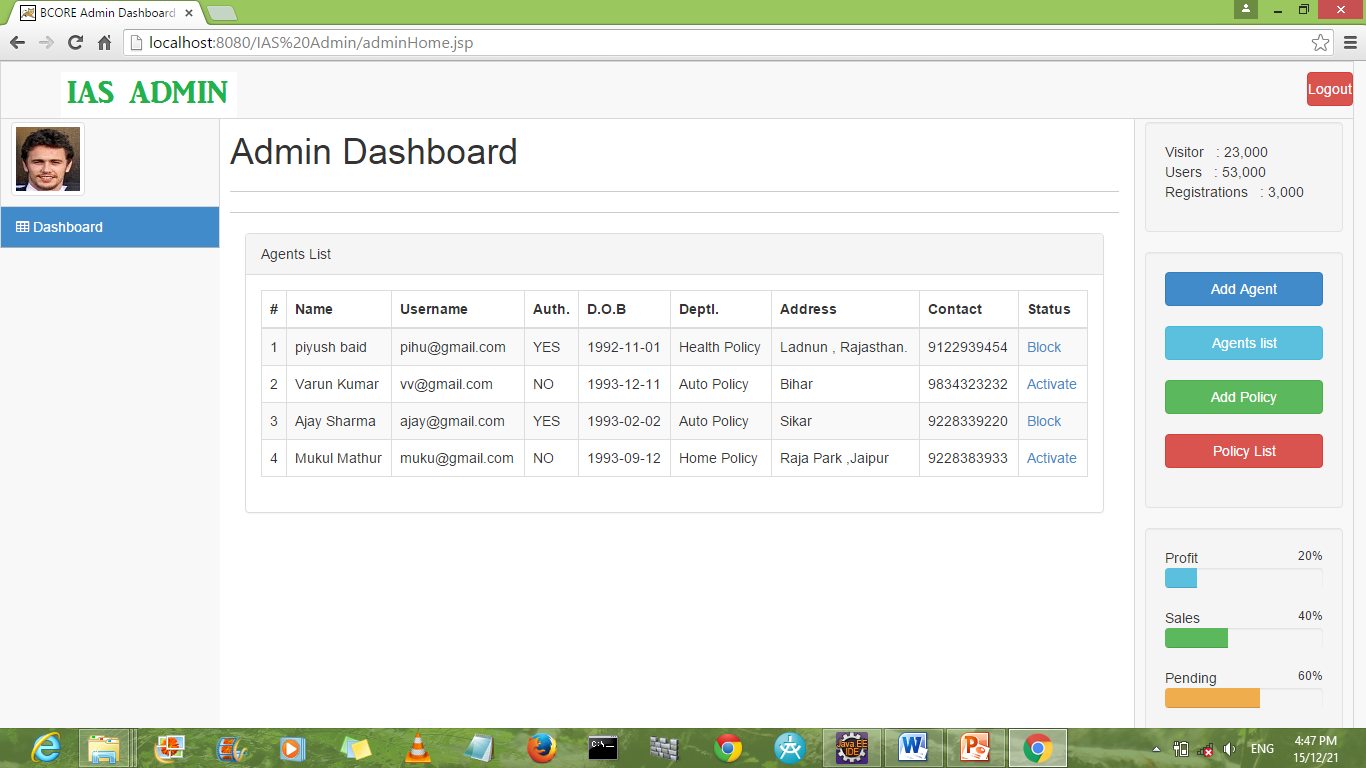
****

***Snapshot 5: Agent Dashboard***

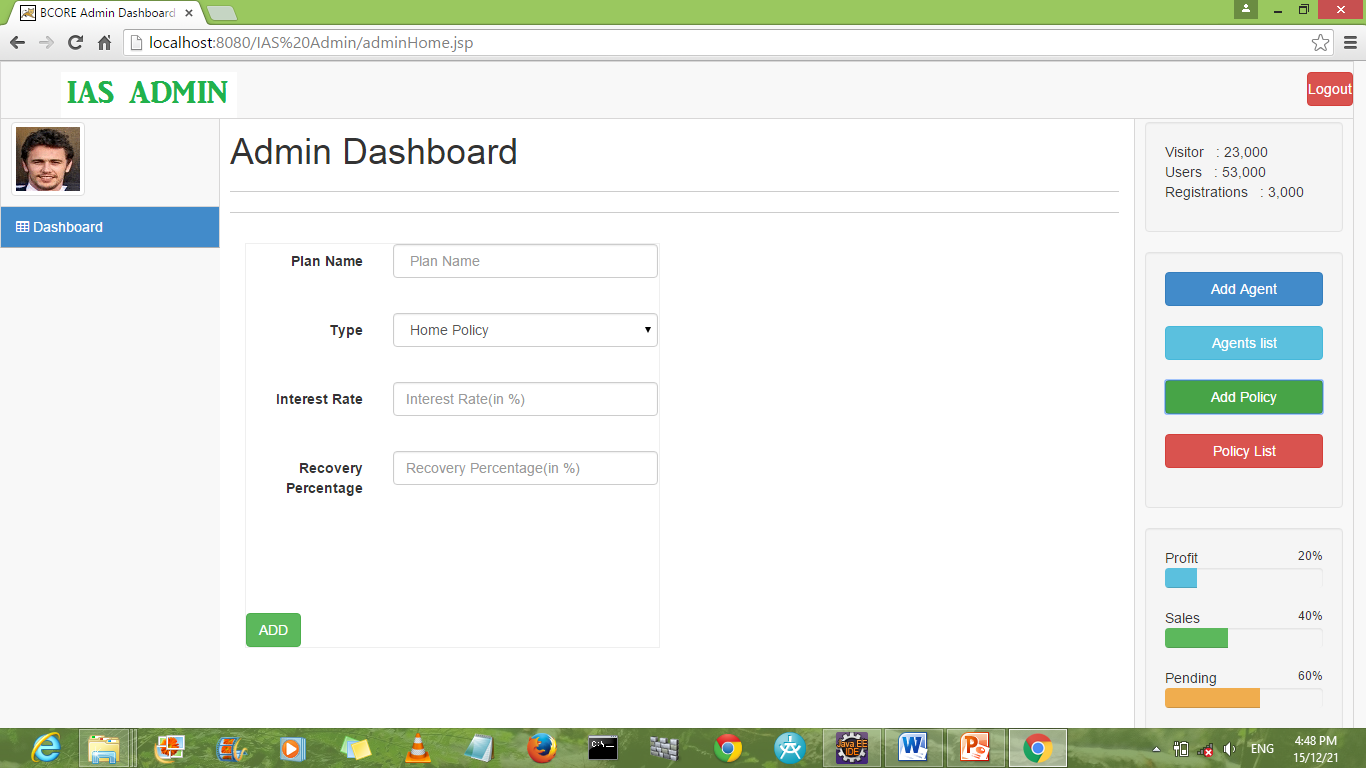
* 1. **Administrator:**



***Snapshot 6: Add agent***

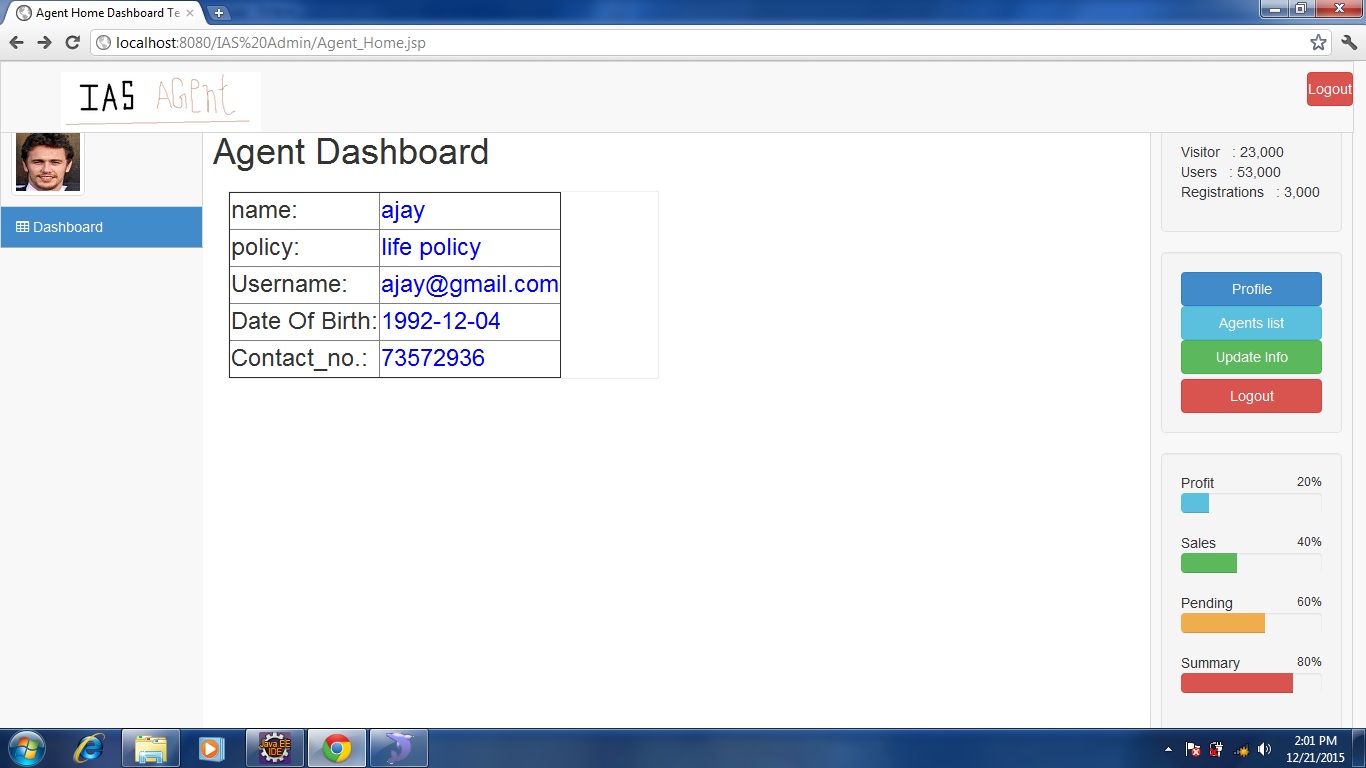


***Snapshot 7: Agent List***

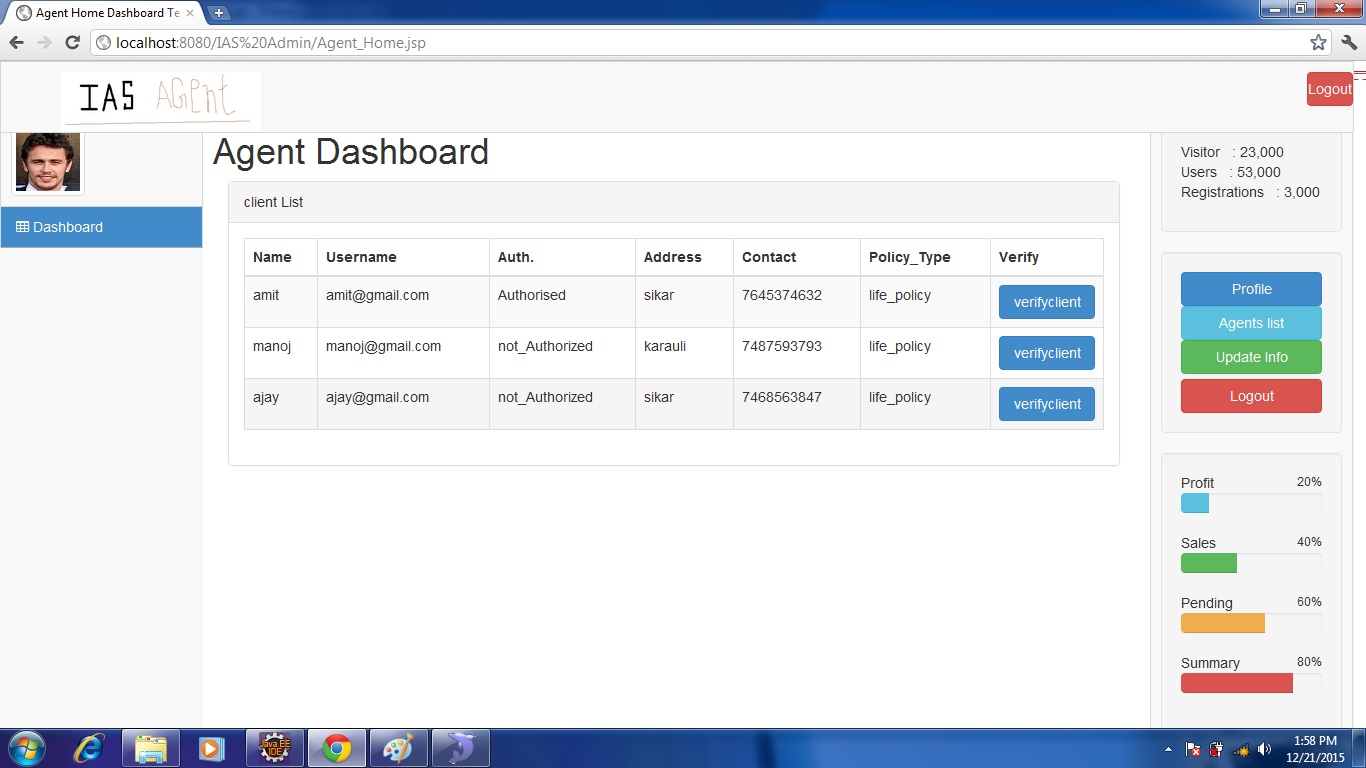


***Snapshot 8: Add policy***

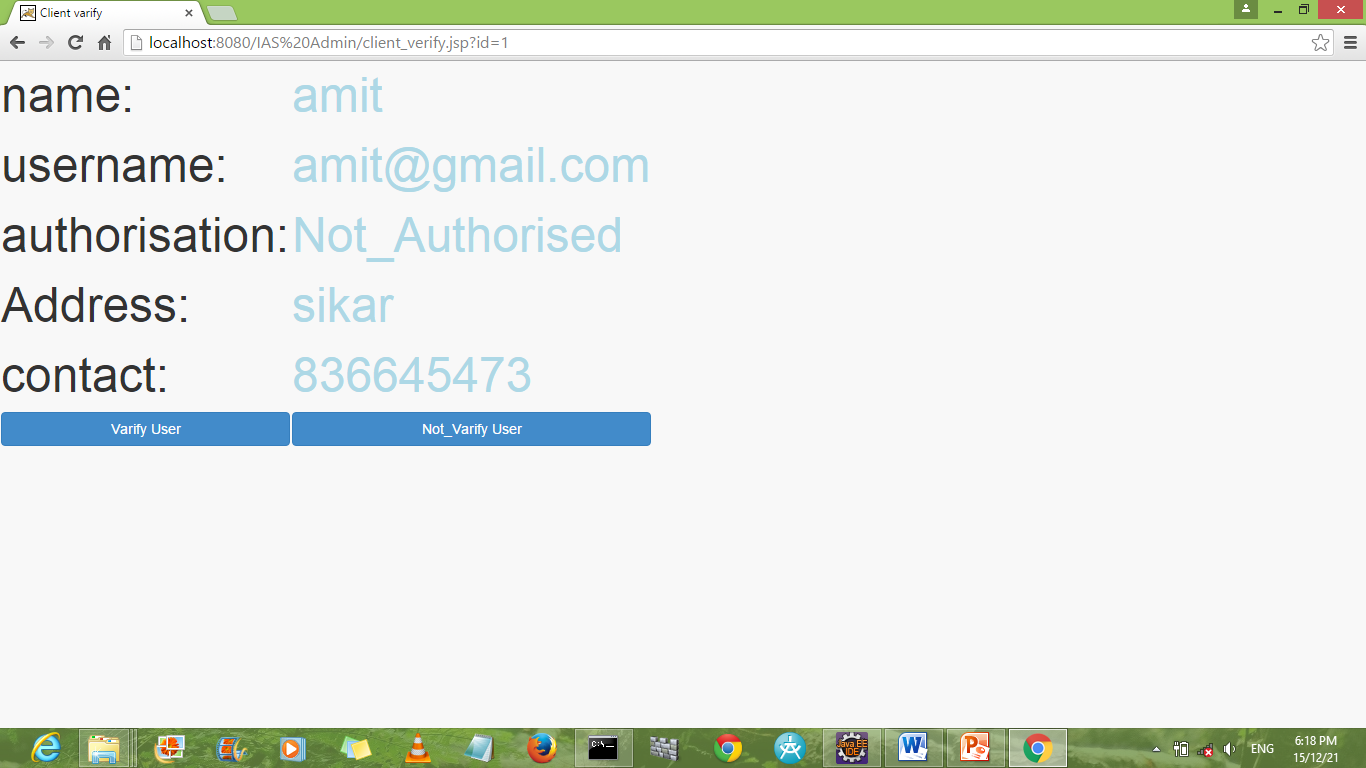
* 1. **Agent:**

******

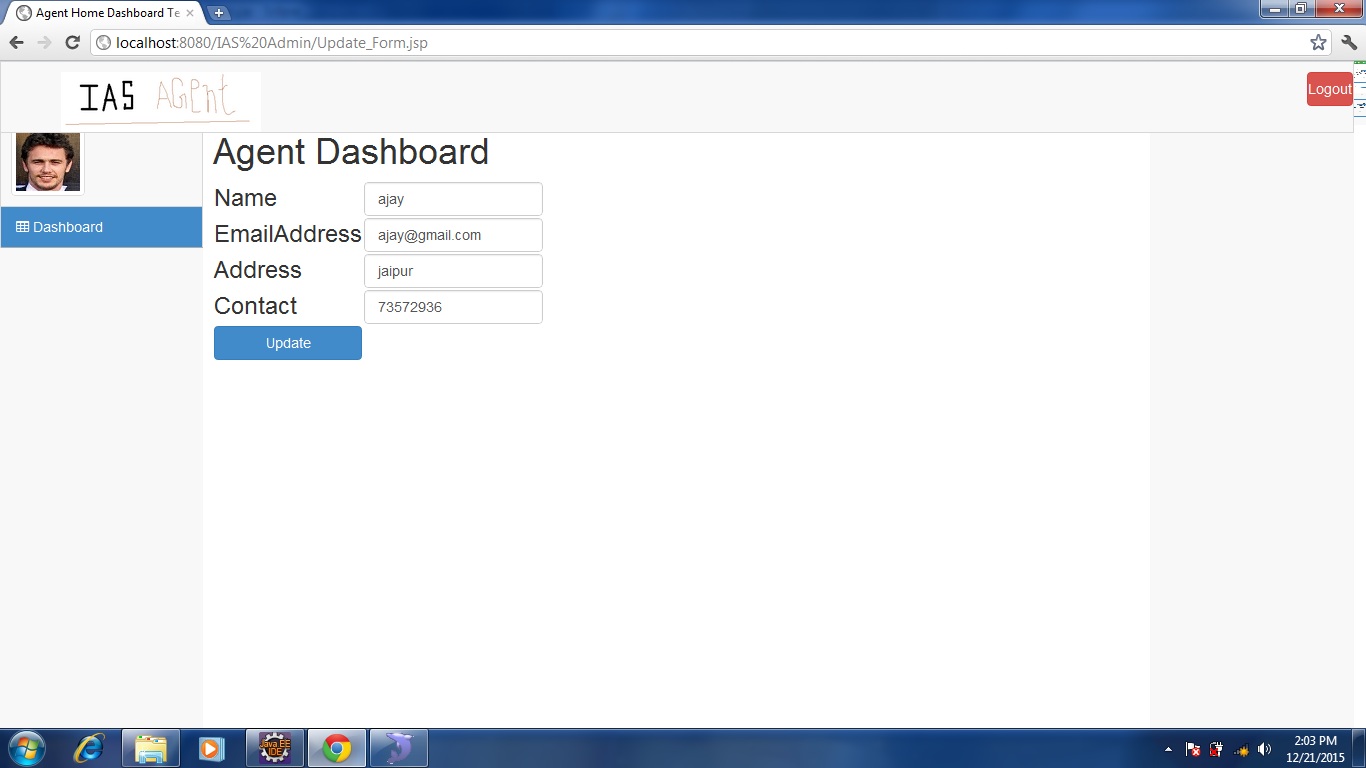
***Snapshot 9: Agent profile***

****

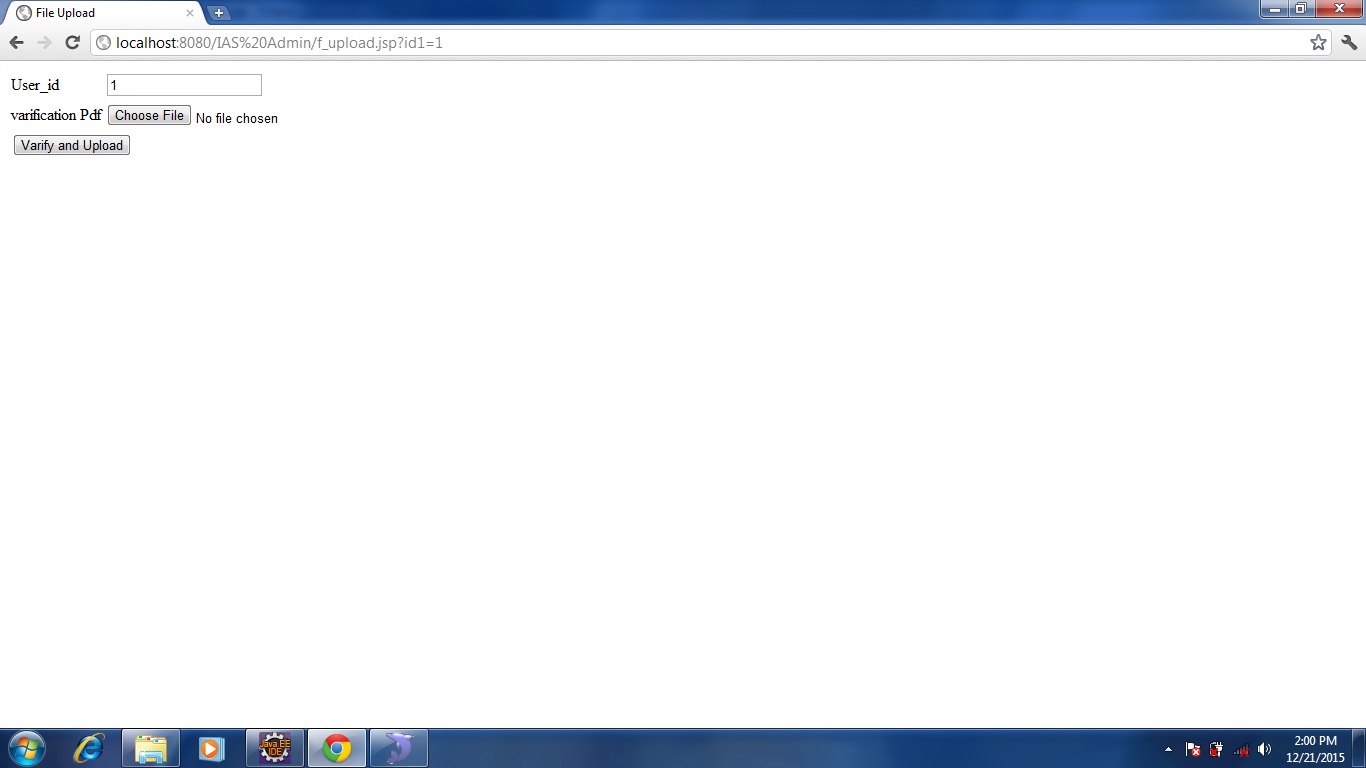
***Snapshot 10: Agent Home***



***Snapshot 11 – Client Verification***

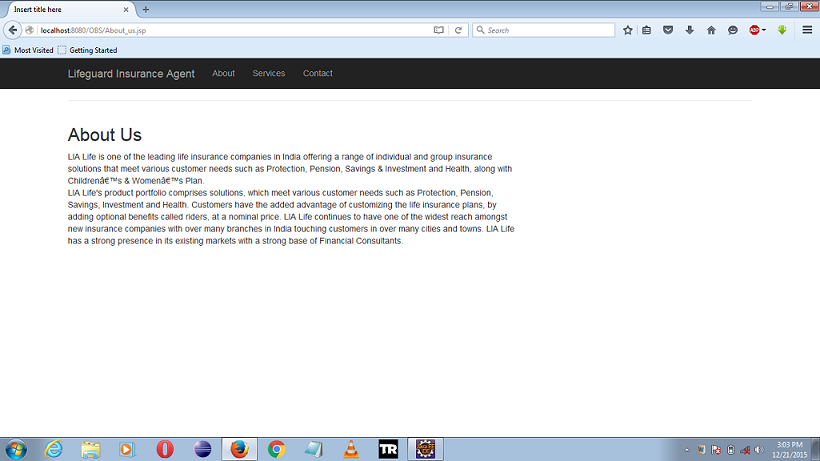
****

***Snapshot 12 – Update Agent Info***

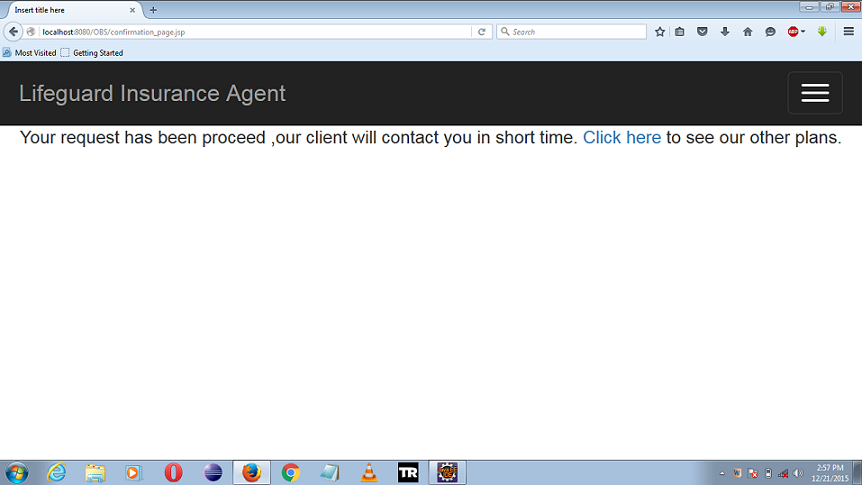
****

***Snapshot 13: Upload verification doc***

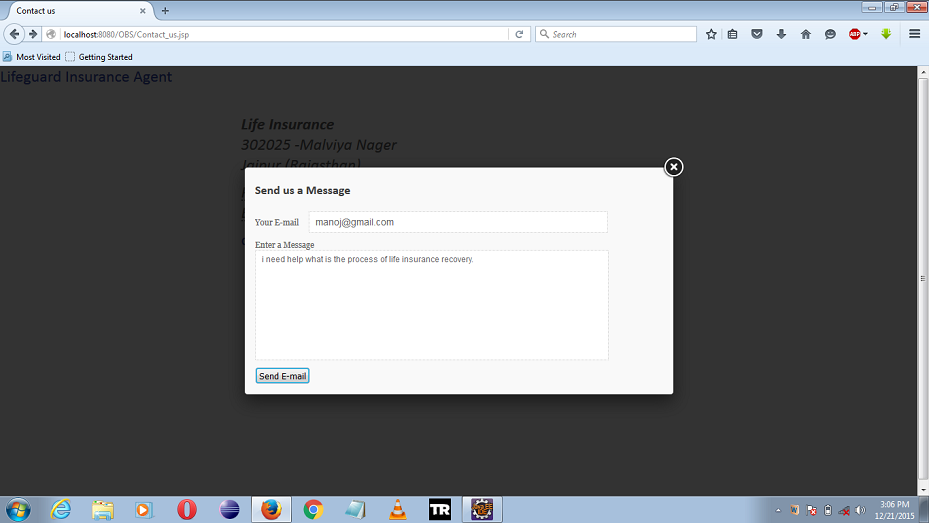
* 1. **Client Interface:**

****

***Snapshot 14: About Us***

****

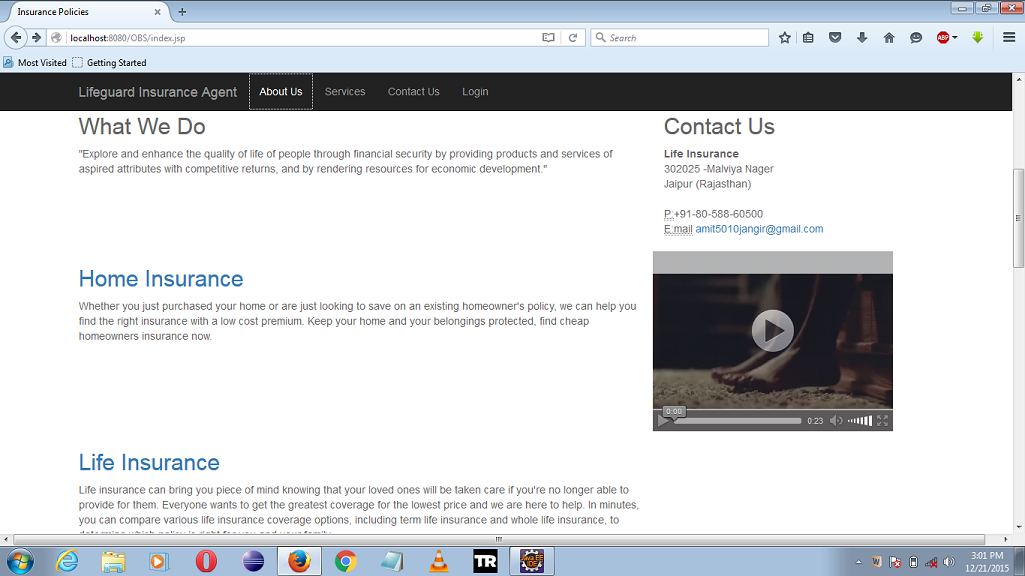
***Snapshot 15: Acknowledgement***

****

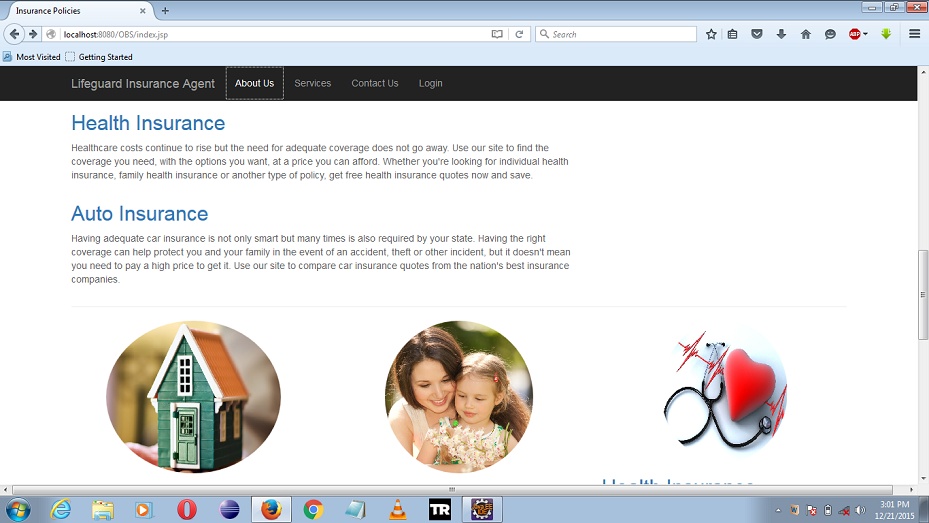
***Snapshot 16: Send Mail***

****

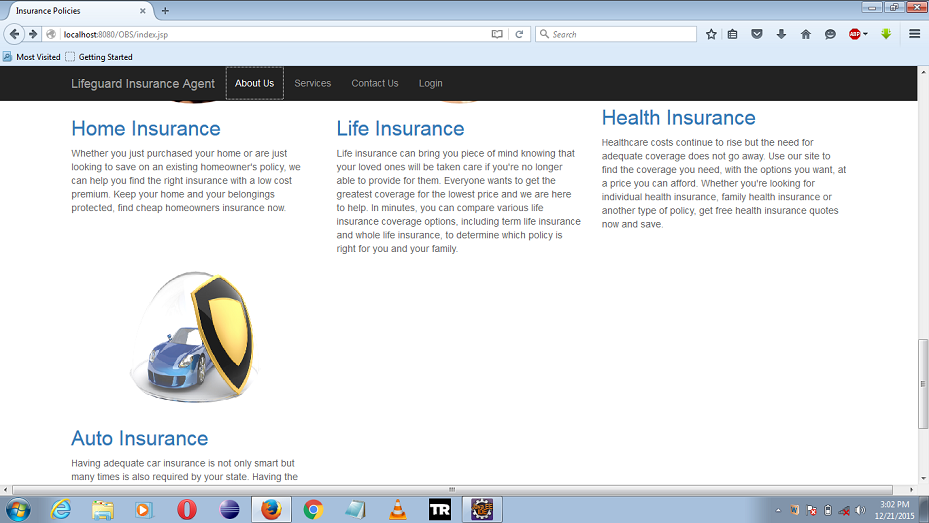
***Snapshot 17: Client Home1***

****

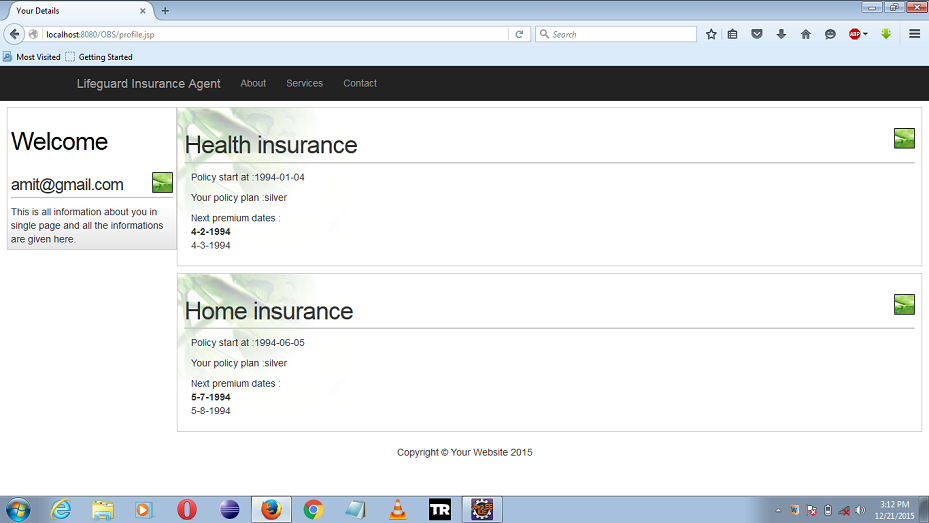
***Snapshot 18: Client Home2***

****

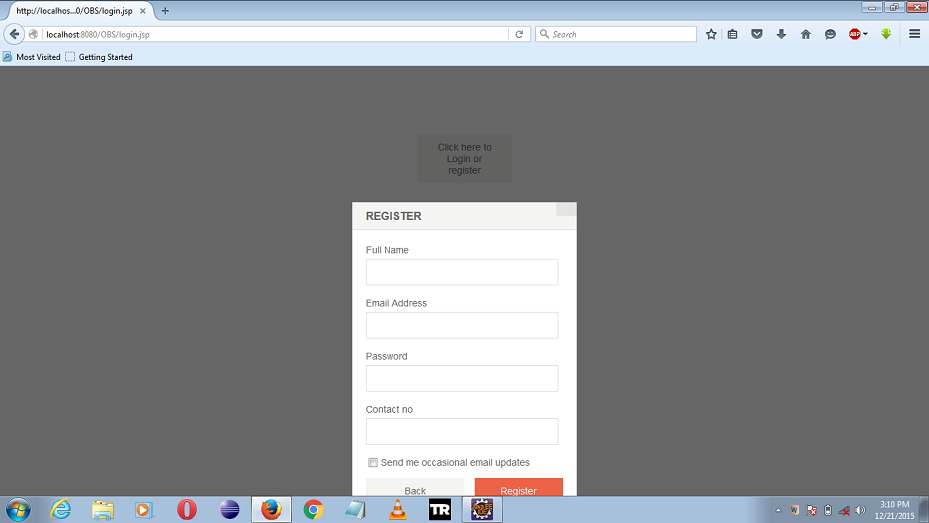
***Snapshot 19: Client Home3***

****

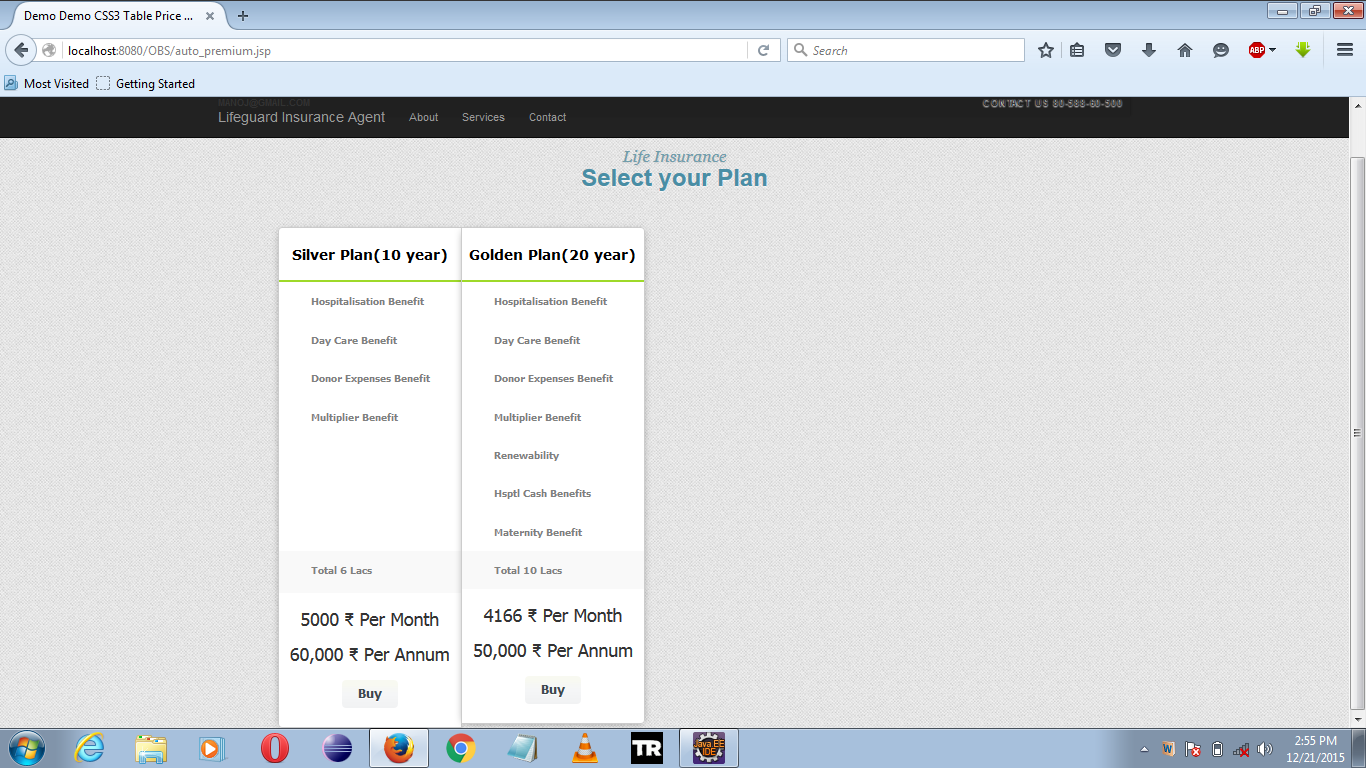
***Snapshot 20: Client Home4***

****

***Snapshot 21: Record client Illustration***

****

***Snapshot 22: Client Registration***



***Snapshot 23: Select Your plan***

**CHAPTER 5: CONCLUSION & FUTURE SCOPE**

**5.1 Conclusion:**

The Insurance Agent Samurai System is a better way to store the information about customer, his/her policy illustration and ease to process his/her policy activations contract while taking a policy. The system also include some components like adding agent, policy, listing agents, see illustration ,register etc. to make it features app. We can conclude that:

* The system brings transparency in handling the customer request.
* Friendly & interactive environment for users.
* Since it’s a web application it can be used by different users simultaneously on different systems.
* Policy illustration and agent handling is a special feature of this web application.
* It reduces a lot of paper work and time.

**5.2 Future Scope of the System:**

* In future an Android, iOS and Windows application can be develop for a customer to track status of his policy without going or calling to the agency.
* It’s also possible to convert the system into an management system as different managements are already present in the system.
* Multithreading can be used to increase the performance of the system.
* Predefined format of report can be added in the system.
* Make system so efficient and light that it can open on mobile or tablet easily.
* Security of the system can be improved by applying different encryption and authentication techniques.
* More tabs can be added to perform more functionality.
* User notifications can be added to the system.