**Claims Management System**

A Project Work  
submitted in partial fulfillment of the   
requirements for the degree of

**Bachelor of Technology**

in

**Computer Science and Engineering**

Submitted By :-

**Shriyans Nidhish**768818

Under the Supervision of

**Ms. Shilpa mahajani and Mr. Sabbir Poonawala**



**School of Computing Science and Engineering**

**Greater Noida, Uttar Pradesh**

**2019**

**DECLARATION**

Project Title: **CLAIMS MANAGEMENT SYSTEM**

Degree for which the project work is submitted: **Bachelor in Technology in Computer Science and Engineering**

I declare that the presented project represents largely my own ideas and work in my own words. Where others ideas or words have been included, I have adequately cited and listed in the reference materials. The report has been prepared without resorting to plagiarism. I have adhered to all principles of academic honesty and integrity. No falsified or fabricated data have been presented in the report. I understand that any violation of the above will cause for disciplinary action by the Institute, including revoking the conferred degree, if conferred, and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken.

Shriyans Nidhish

(768818)

Date:

CERTIFICATE

It is certified that the work contained in this project entitled **“Claims Management System”** submitted by, Shriyans Nidhish (768818), for the degree of Bachelor in Technology in Computer Science and Engineering is absolutely based on his/her own work carried out under my supervision and this project work has not been submitted elsewhere for any degree.

Mr Sabbir Poonawala

Cognizant Technology Solutions

Pune, Maharashtra, India

Date:

## Countersigned by

**(Mrs. Shilpa Mahajani)**

Cognizant Technology Solutions,

Pune, Maharashtra, India

**Table Of Contents**

**1. Abstract**

**2. Introduction**

**2.1 Overall description**

**2.2 Purpose**

**3. Proposed Model**

**3.1 Process Architecture**

**3.2 System Flow**

**3.2.1 Admin Process Flow**

**3.2.2 Member Process Flow**

**3.3 Business Requirements**

**3.4 Business Case**

**3.4.1 Technologies Recommended**

**3.4.2 Hardware and Software Requirements**

**4. Implementation**

**5. Results and Discussions**

**6. Conclusions and Future Works**

**7.References**

**Chapter 1**

**Abstract**

Claims Management System is developed to provide an online platform for the staff workers to manage the claim activities. By changing the current manual claim method to computerised system, the effectiveness and efficiency of claim process have been improved, for instances this system can quicken the approval process as well as allow applicants to check the status of claim application, minimizes human errors in calculation and provides several type of claim report. By putting the system online, there is no more problems of time and distance. Several frameworks such as Spring and hibernate along with, Model-View-Controller (MVC) architecture is also used for developing this project. Data dictionary is produced for better understanding on system database. Several user interfaces are created to visualise the actual system environment. Last but not least, Hypertext Mark-up Language(HTML), Cascading Style Sheets(CSS), JavaScript, Java Server Page(JSP), Spring, Hibernate and Oracle are used during system development. As a conclusion, Claims Management system can replace the existing manual claim management as well as bring benefits to Organization in which it is used.

**Chapter 2**

**Introduction**

**2.1 Overall Description**

**Claims Management System project** is implemented in java platform using oracle as backend application. Main aim of this project is to develop an online application for insurance company to atomize work procedure, The Claims Management System will be a platform independent solution developed that will have all the features and functionalities which shall let the insurance company allow its customer to submit a claim requests, view the real-time status of the submitted claims and also submit additional documents as required. The solution will also address the admins needs to view the submitted claims, view the member eligibility as per the policy plan and approve the claims.

In existing system manual procedure is followed where records are used to maintain data which is a time taking process and require more man power and calculating claim amounts...etc. are done manually.

In present system there is no need of human interference in calculating any details. Total work is done using management system which will save time and less paper work and even human resource.

This application is implemented in two modules, Admin and Customer.

**Admin Module:** This module is the main module in which admin, super user, will look after the claims that are to be processed. He is the person responsible for the final processing of all the claims and segregating fake or pseudo claim requests.

**Customer Module:**Using this module customer can register and login after which he/she can raise a claim request which will be further processed by the admin. If the admin approves then customer will be notified and requested amount will be sanctioned.

**2.2 Purpose**

Claims Management System (CMS) application. CMS is basically a system that helps members to do member registration for claims, profile update, Claim Request, Search Claims and View Status, Claim Process. This system helps in online processing of claims by the admin after it’s verification so that no pseudo claim requests are granted approval.

Hence, this system also helps in overcoming the old procedure that was followed where manual calculations were done which used to consume lots of time and resources.

These are the few important modules in the system.

* Member registration & credential authentication.
* Member profile update.
* Member claim request.
* Search Claim and View Status
* Claim processing.

**Chapter 3**

**Proposed Model**

**3.1 Process Architecture**

Below is the overall functional flow of the project including the components of interaction

* Create Member Registration page & Authentication.
* Create Admin Registration page & Authentication.
* Get and manage Member details.
* Get the Claim details and save them.
* Process the claims.
* Update the claim status
* List the claims request for each member.
* Display the details of the claims.
* Disconnect on log off.

**3.2 System Flow**

### 3.2.1 Admin Process flow

### 

### 3.2.1 Member Process flow

### 

**3.3 Business Requirements**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Business Requirement ID** | **Short Description** | **Description in detail** | **Interacting Business**  **Processes** |
| 1 | Req\_1 | Initial Selection | Ability of the system to allow to user to choose  Admin/User Login/Registration |  |
| 2 | Req\_2 | Member Registration | Ability of the system to procure the fundamental details of the Member |  |
| 3 | Req\_3 | Member Authentication | Ability of the system to authenticate the Member credentials of the registered Member |  |
| 4 | Req\_4 | Admin Registration | Ability of the system to procure the fundamental details of the Admin |  |
| 5 | Req\_5 | Admin Authentication | Ability of the system to authenticate the credentials of the registered Admin |  |
| 6 | Req\_6 | Claim request | Ability of the system to procure the Claim details |  |
| 7 | Req\_7 | Claim process | Ability of the system to  process the claim based on member eligibility |  |
| 8 | Req\_8 | Display claim details at member  level | Ability of the system to display the claim details at  member level |  |
| 9 | Req\_9 | Member / Admin logoff | Ability of the system to enable Member / Admin to  logoff |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Req. # | Rationale Categorization | Business Requirement | Req. Type  \* | Priority  \*\* | Originator  \*\*\* | BR Traced to Business Requirement / Use case ID | Impacted Stakeholders |
| Req\_ | Initial selection | Screen should | UI |  | NA | Req\_1 |  |
| 1.1 |  | display the option |  |  |  |  |
|  |  | for Admin login / |  |  |  |  |
|  |  | Registration and |  |  |  |  |
|  |  | Member login / |  |  |  |  |
|  |  | Registration |  | Critical |  |  |
| Req\_ | Member | When the Member | UI |  | NA | Req\_2 |  |
| 2.1 | Registration | clicks on the |  |  |  |  |
|  |  | registration link, it |  |  |  |  |
|  |  | should re-direct to |  |  |  |  |
|  |  | registration form. |  | Critical |  |  |
| Req\_ | Member | Member needs to | UI |  | NA | Req\_2 | Please refer to |
| 2.2 | Registration | fill some of the |  |  |  |  | Table 1.0 under |
|  |  | basic |  |  |  |  | References |
|  |  | attributes/fields as |  |  |  |  |  |
|  |  | mentioned below in |  |  |  |  |  |
|  |  | requirement: First |  |  |  |  |  |
|  |  | Name, Last Name, |  |  |  |  |  |
|  |  | Age, Gender, |  |  |  |  |  |
|  |  | Contact Number, |  |  |  |  |  |
|  |  | Member Id, |  |  |  |  |  |
|  |  | Password |  | Critical |  |  |  |
| Req\_ | Member | Clicking ‘Submit’ | F |  | NA | Req\_2 |  |
| 2.3 | Registration | should validate the |  |  |  |  |
|  |  | datatype |  |  |  |  |
|  |  | constraints for each |  |  |  |  |
|  |  | field |  | Critical |  |  |
| Req\_ | Member | Member failing to | E |  | NA | Req\_2 |  |
| 2.4 | Registration | provide information |  |  |  |  |
|  |  | on the mandatory |  |  |  |  |
|  |  | fields be provided |  |  |  |  |
|  |  | with an alert |  |  |  |  |
|  |  | message – ‘Please |  |  |  |  |
|  |  | update the |  |  |  |  |
|  |  | highlighted |  |  |  |  |
|  |  | mandatory field(s).’ |  |  |  |  |
|  |  | Also, highlight the |  |  |  |  |
|  |  | missed out field in |  |  |  |  |
|  |  | red |  | Medium |  |  |
| Req\_ | Member | Post-successful | F |  | NA | Req\_2 |  |
| 2.5 | Registration | field level |  |  |  |  |
|  |  | validation, save the |  |  |  |  |
|  |  | information in the |  |  |  |  |
|  |  | database |  | Critical |  |  |
| Req\_ | Member | Upon saving the | E |  | NA | Req\_2 |  |
| 2.6 | Registration | information in the |  |  |  |  |
|  |  | database, display |  |  |  |  |
|  |  | the message ‘Your |  |  |  |  |
|  |  | details are |  |  |  |  |
|  |  | submitted |  |  |  |  |
|  |  | successfully’. |  | Medium |  |  |
| Req\_ | Member | Admin should be | F |  | NA | Req\_2 |  |
| 2.7 | Registration | able to view the |  |  |  |  |
|  |  | New Members for |  |  |  |  |
|  |  | registration |  | Critical |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Req. #** | **Rationale Categorization** | **Business Requirement** | **Req. Type**  **\*** | **Priority**  **\*\*** | **Originator**  **\*\*\*** | **BR Traced to Business Requirement / Use case ID** | **Impacted Stakeholders** |
|  |  |  |  |  |  |  |  |
| Req\_ 2.9 | Member Registration | If rejected, the Member should not be allowed to login with the registered  credentials | F | Critical | NA | Req\_2 |  |
| Req\_ 2.10 | Member Registration | Member should get  SMS on Approval / Rejection | E | Medium | NA | Req\_2 |  |
| Req\_ 3.1 | Credential Authentication | A registered Member – is able click ‘Login’ link, after keying in ‘Member ID’ &  ‘Password’ field and get his credentials authenticated with the existing  database entry. | F | Critical | NA | Req\_3 |  |
| Req\_ 4.1 | Admin Registration | When the Admin clicks on the registration link, it should re-direct to  registration form. | UI | Critical | NA | Req\_4 |  |
| Req\_ 4.2 | Admin Registration | Admin needs to fill some of the basic attributes/fields as mentioned below in requirement: First Name, Last Name, Age, Gender, Contact Number, Vendor Id,  Password | UI | Critical | NA | Req\_4 | Please refer to Table 2.0 under References |
| Req\_ 4.3 | Admin Registration | Clicking ‘Submit’ should validate the datatype constraints for each  field | F | Critical | NA | Req\_4 |  |
| Req\_ 4.4 | Admin Registration | Admin failing to provide information on the mandatory fields be provided with an alert message – ‘Please update the highlighted  mandatory field(s).’ Also, highlight the  missed out field in red | E | Medium | NA | Req\_4 |  |
| Req\_ 4.5 | Admin Registration | Post-successful field level validation, save the information in the database | F | Critical | NA | Req\_4 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Req. #** | **Rationale Categorization** | **Business Requirement** | **Req. Type**  **\*** | **Priority**  **\*\*** | **Originator**  **\*\*\*** | **BR Traced to Business Requirement / Use case ID** | **Impacted Stakeholders** |
| Req\_ 4.6 | Admin Registration | Upon saving the information in the database, display the message ‘Your details are submitted  successfully’. | E | Medium | NA | Req\_4 |  |
| Req\_ 5.1 | Credential Authentication | A registered user – is able click ‘Login’ link, after keying in ‘Admin ID’ &  ‘Password’ field and get his credentials authenticated with the existing  database entry. | F | Critical | NA | Req\_5 |  |
| Req\_ 6.1 | Claim request | On clicking “Raise claim request”, it should redirect to the Claim request  Page | UI | Critical | NA | Req\_6 |  |
| Req\_ 6.2 | Claim request | Member should be able to update the mandatory fields of  the claim | F | Critical | NA | Req\_6 | Please refer to Table 3.0 under References |
| Req\_ 6.3 | Claim request | Clicking ‘Submit’ should validate the datatype constraints for each  field | F | Critical | NA | Req\_6 |  |
| Req\_ 6.4 | Claim request | Member failing to provide information on the mandatory fields be provided with an alert message – ‘Please update the highlighted  mandatory field(s).’ Also, highlight the  missed out field in red | F | Critical | NA | Req\_6 |  |
| Req\_ 6.5 | Claim request | Post-successful field level  validation, save the information in the database | F | Critical | NA | Req\_6 |  |
| Req\_ 6.5 | Claim request | Upon saving the information in the database, display the message ‘Your details are  submitted successfully’. | F | Critical | NA | Req\_6 |  |
| Req\_ 7.1 | Claim Process | When Admin clicks “Claims to be processed”, it should list all the  claims available for Processing | UI | Critical | NA | Req\_7 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Req. #** | **Rationale Categorization** | **Business Requirement** | **Req. Type**  **\*** | **Priority**  **\*\*** | **Originator**  **\*\*\*** | **BR Traced to Business Requirement / Use case ID** | **Impacted Stakeholders** |
| Req\_ 7.2 | Claim Process | When Admin clicks the Claim ID, the claim details should  be displayed. | F | Critical | NA | Req\_7 |  |
| Req  \_7.3 | Claim Process | Admin should validate the Claim, check member eligibility by  checking the Member details and  plan | F | Critical | NA | Req\_7 |  |
| Req  \_7.4 | Claim Process | Upon processing, update the claim status as Accepted  / Denied.  If accepted, Claim amount should be  updated | F | Critical | NA | Req\_7 |  |
| Req  \_7.5 | Claim Process | Clicking ‘Submit’ should validate the datatype constraints for each  field | F | Critical | NA | Req\_7 |  |
| Req  \_7.6 | Claim Process | Admin failing to provide information on the mandatory fields be provided with an alert message – ‘Please update the highlighted  mandatory field(s).’ Also, highlight the  missed out field in red | F | Critical | NA | Req\_7 |  |
| Req  \_7.7 | Claim Process | Post-successful field level  validation, save the information in the database | F | Critical | NA | Req\_7 |  |
| Req  \_7.8 | Claim Process | Upon saving the information in the database, display the message ‘Your details are  submitted successfully’. | F | Critical | NA | Req\_7 |  |
| Req  \_7.9 | Claim Process | When Admin clicks “Processed Claims”, it should list all the  Processed Claims | UI | Critical | NA | Req\_7 |  |
| Req  \_7.1 0 | Claim Process | On clicking the Claim ID, it should display the details of the claims | F | Critical | NA | Req\_7 |  |
| Req\_ 7.11 | Claim Process | When Admin clicks “Member Details”, it should ask for the  Member ID | F | Critical | NA | Req\_7 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Req. #** | **Rationale Categorization** | **Business Requirement** | **Req. Type**  **\*** | **Priority**  **\*\*** | **Originator**  **\*\*\*** | **BR Traced to Business Requirement / Use case ID** | **Impacted Stakeholders** |
| Req\_ | Claim Process | On entering the | UI |  | NA | Req\_7 |  |
| 7.12 |  | Member ID, Admin |  |  |  |  |
|  |  | should be able to |  |  |  |  |
|  |  | view the Member |  |  |  |  |
|  |  | details. |  | Critical |  |  |
| Req\_ 7.12 | Claim Process | Admin on clicking “Plan codes” it should redirect to the page where Admin can add/  update plan code details | UI | Critical | NA | Req\_7 | Please refer to Table 4.0 under References |
| Req\_ 8.1 | Display claim details at member level | When Member clicks on “Claim  Requests”, it should display all | F |  | NA | Req\_8 |  |
|  |  | the claim IDs raised |  |  |  |  |
|  |  | by the member |  | Critical |  |  |
| Req\_ 8.2 | Display claim details at member level | When Member clicks on Claim ID, it should Display  the claim details | F |  | NA | Req\_8 |  |
|  |  | with status |  | Critical |  |  |
| Req\_ 9.1 | User Log off | Option to log off from the system. | UI | Critical | NA | Req\_9 |  |

## 

## (Member Table)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Member ID | Text(10) | Alphabetic | Yes | System Generated |
| First Name | Text(50) | Alphabetic | Yes |  |
| Last Name | Text(50) | Alphabetic | Yes |  |
| Age | Numeric(2) | Numeric | Yes |  |
| Gender | Drop Down | NA |  | Male, Female |
| DoB | Text(10) | Alphanumeric | Yes |  |
| Contact Number | Text(10) | Numeric | Yes |  |
| Alt Contact Number | Text(10) | Numeric | No |  |
| Email ID | Text(50) | Alphanumeric | Yes |  |
| Password | Text(15) | Alphanumeric | Yes |  |
| Plan Code | Text(10) | Alphanumeric | Yes |  |
| Coverage Start Date | Text(10) | Alphanumeric | Yes |  |
| Coverage End Date | Text(10) | Alphanumeric | Yes |  |
| Address Line 1 | Text(100) | Alphanumeric | Yes |  |
| Address Line 1 | Text(100) | Alphabetic | Yes |  |
| City | Text(50) | Alphabetic | Yes |  |
| State | Text(50) | Alphabetic | Yes |  |
| Zip Code | Text(10) | Numeric | Yes |  |

(Admin Table)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Admin ID | Text(10) | Alphabetic | Yes | System Generated |
| First Name | Text(50) | Alphabetic | Yes |  |
| Last Name | Text(50) | Alphabetic | Yes |  |
| Age | Numeric(2) | Numeric | Yes |  |
| Gender | Drop Down | NA |  | Male, Female |
| DoB | Text(10) | Alphanumeric | Yes |  |
| Contact Number | Text(10) | Numeric | Yes |  |
| Alt Contact Number | Text(10) | Numeric | No |  |
| Email ID | Text(50) | Alphanumeric | Yes |  |

(Claim Table)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Claim ID | Auto- generated(5) | Numeric | Yes | Non- editable system  generated |
| Member ID | Text(10) | Alphabetic | Yes | System Generated |
| Claim Service Date | Text(10) | Alphanumeric | Yes |  |
| Claim Submission  Date | Text(10) | Alphanumeric | Yes |  |
| Claim Processing Date | Text(10) | Alphanumeric | Yes |  |
| Claim status | Text (10) Drop down | Alphabetic | Yes | Submitted  Approved Denied |
| Claim Amount | Numeric (10) | Numeric | Yes | Default 0 |
| Approved Amount | Numeric (10) | Numeric | Yes | Default 0 |

(Plan code Table)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Plan Code | Text(10) | Alphanumeric | Yes |  |
| Plan Description | Text(50) | Alphabetic | Yes |  |
| Coverage1 | Text(50) | Alphabetic | Yes |  |
| Coverage2 | Text(50) | Alphabetic | Yes |  |
| Coverage3 | Text(50) | Alphabetic | Yes |  |
| Coverage4 | Text(50) | Alphabetic | Yes |  |
| Coverage5 | Text(50) | Alphabetic | Yes |  |

**3.4 Business Case**

# **3.4.1 Technologies Recommended**

|  |  |
| --- | --- |
| Front End | Java (HTML5, CSS3, JavaScript, Angular) |
| Middleware | Java (Spring, Spring MVC, Hibernate MVC, WEB API2) |
| Backend | Oracle/SQL Server |

# **3.4.2 Hardware and Software Requirements**

|  |  |  |
| --- | --- | --- |
| Technology | Hardware | Software |
| Java | Desktop PC with 8GB RAM | 1. Node.js 10.15.1   1. Angular 5.0 2. Visual Studio Code 1.30 3. Eclipse IDE for Java EE Developers (Oxygen) 4. Maven 3.6.0 5. Tomcat 9 6. MySQL Community Server 8.0 7. MySQL Workbench 8.0.14 8. Putty 9. WinSCP 5.9.4 10. Oracle 11g express version |

**Chapter 4**

**Implementation**

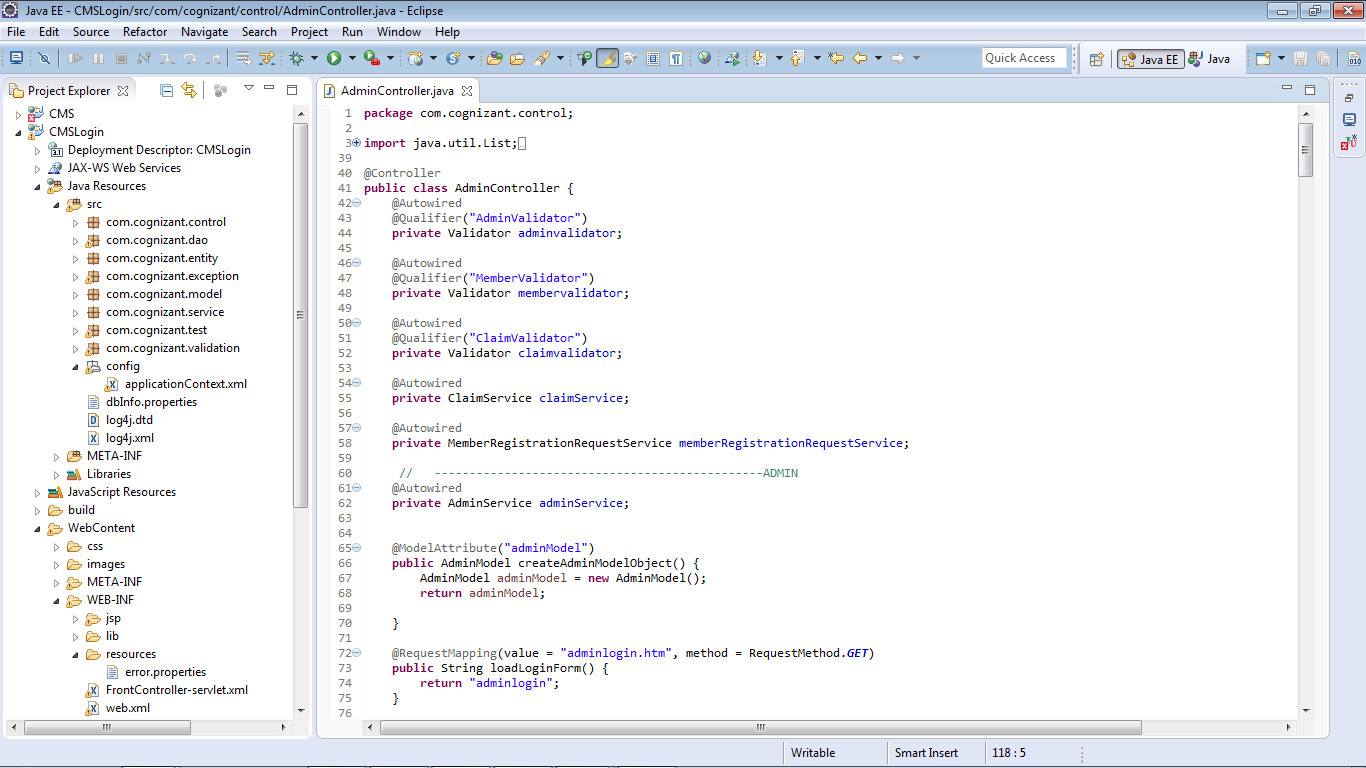


Fig: 4.1 – Project Hierarchy

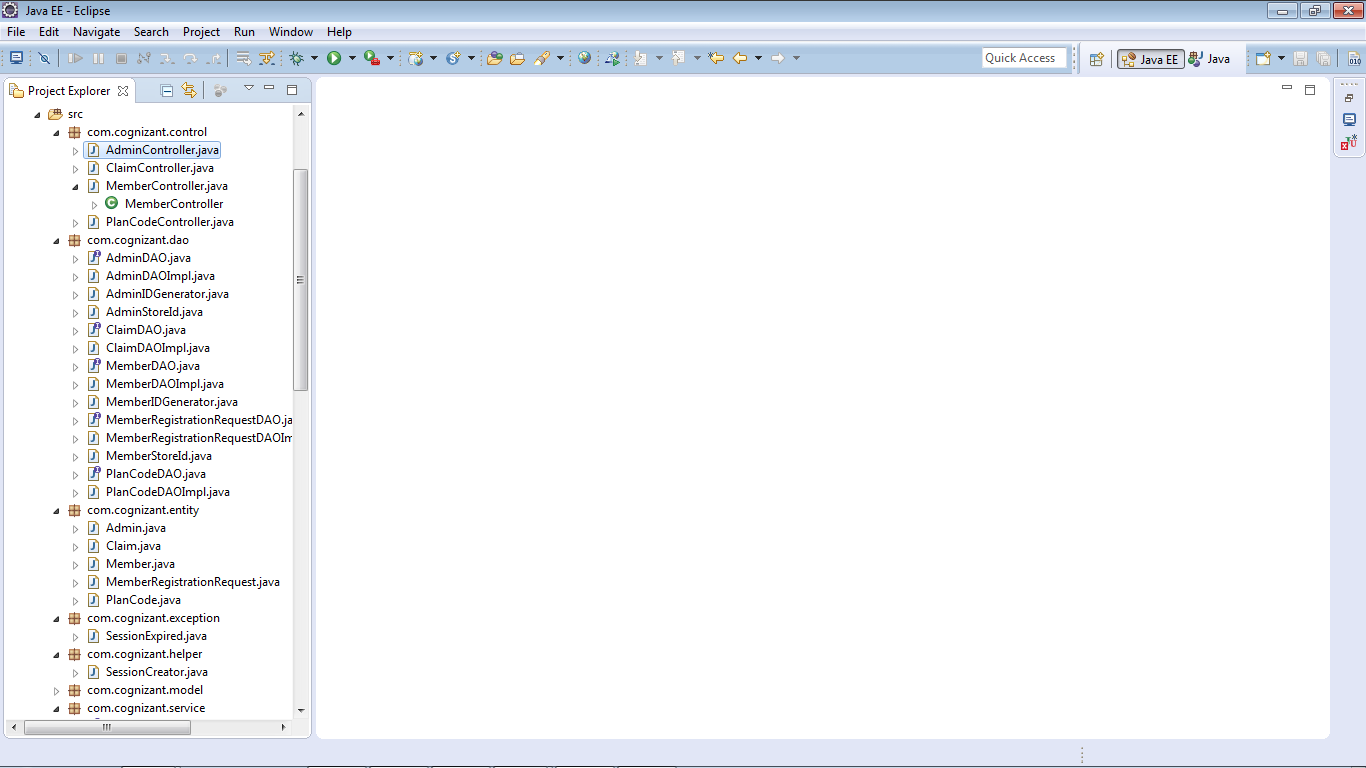


Fig: 4.2 – Classes in packages

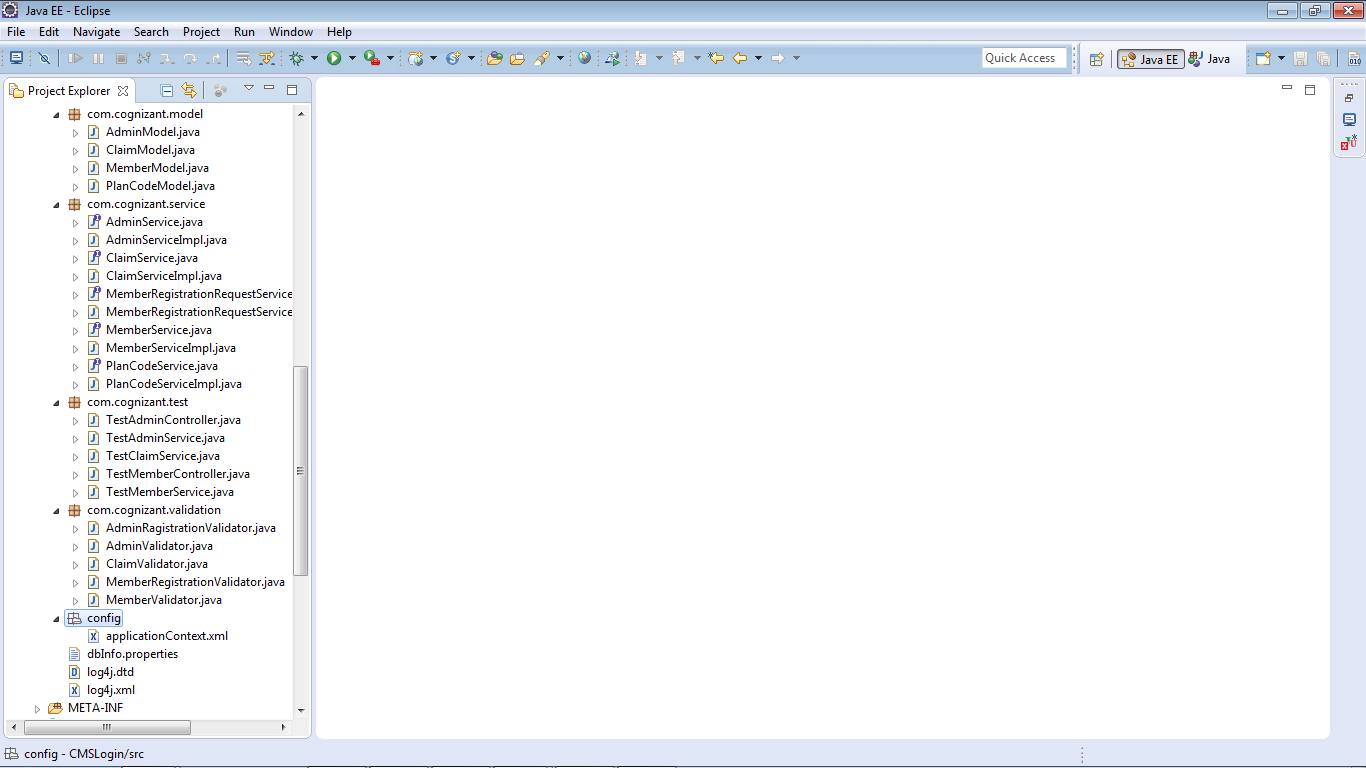


Fig: 4.3 – Classes in packages

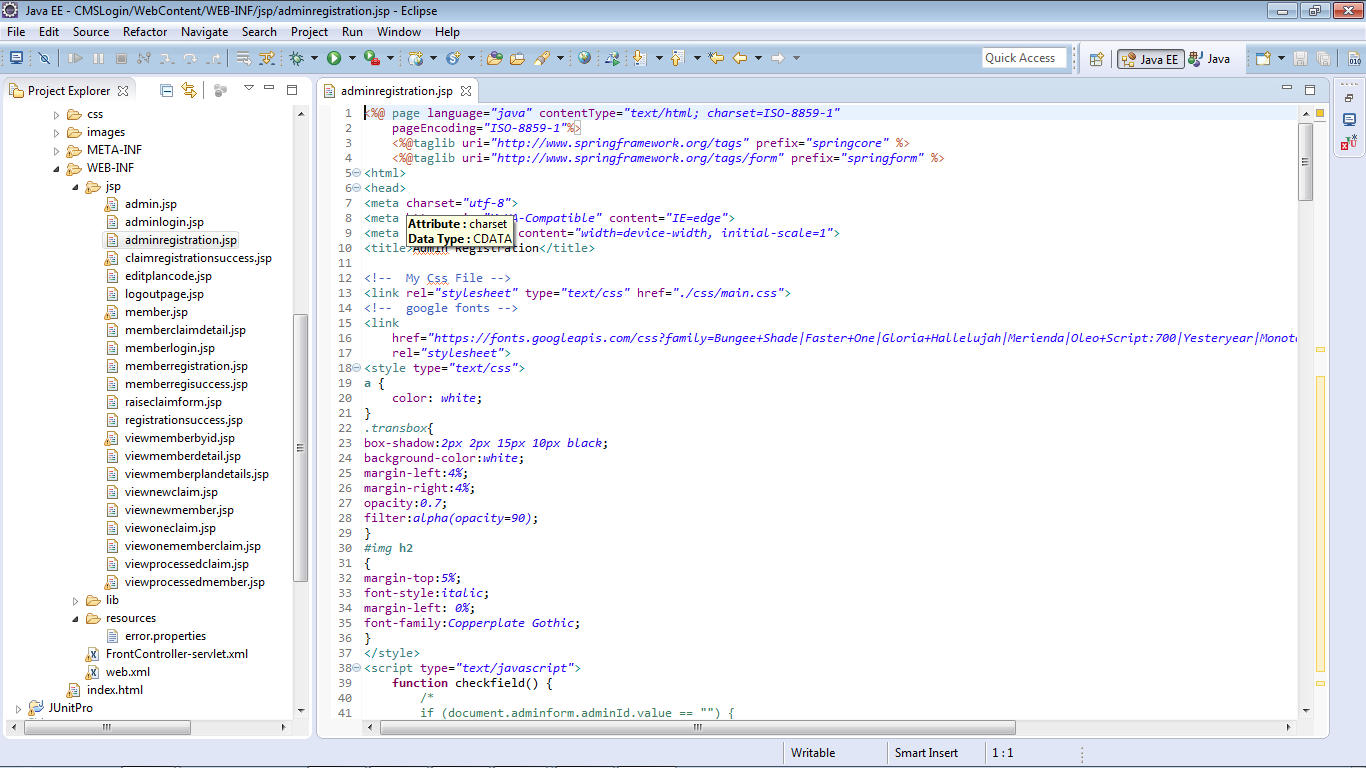


Fig: 4.4 – JSP Pages

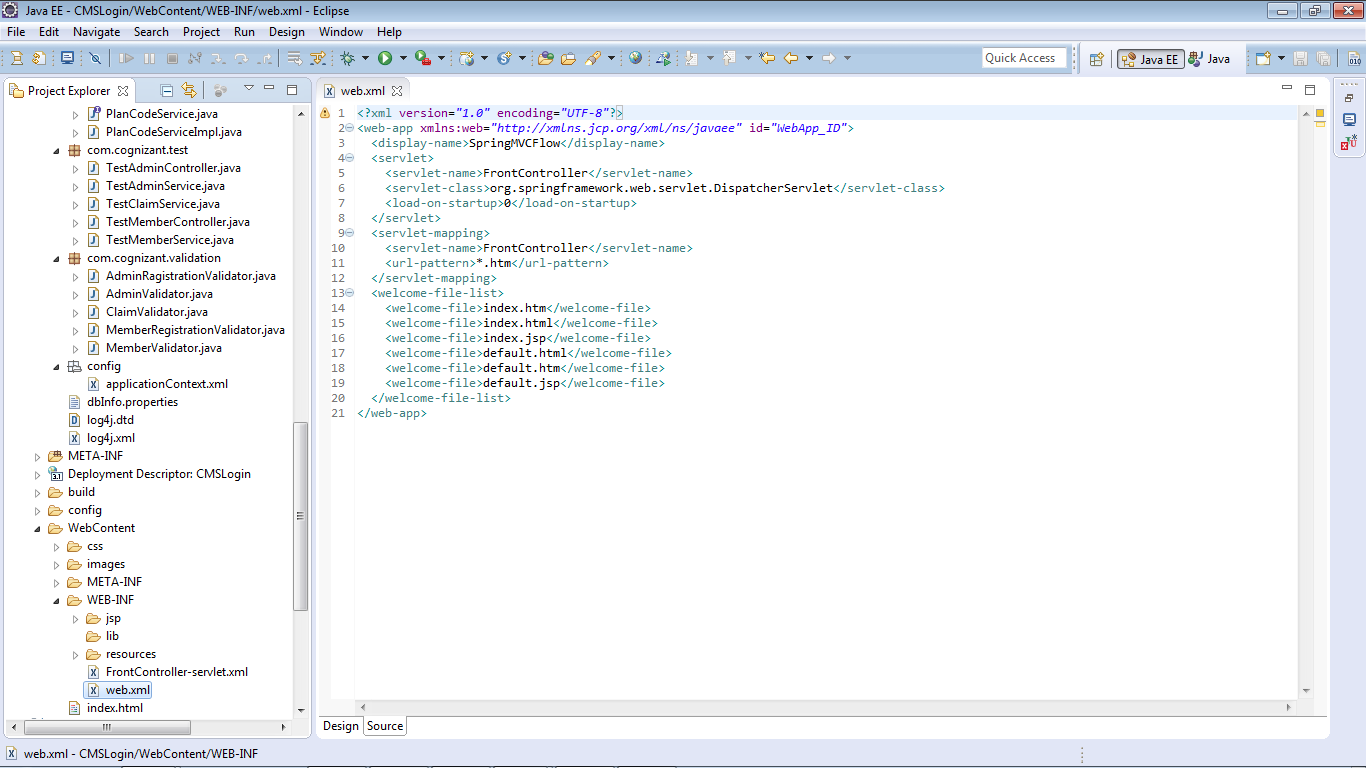


Fig: 4.5 – web.xml

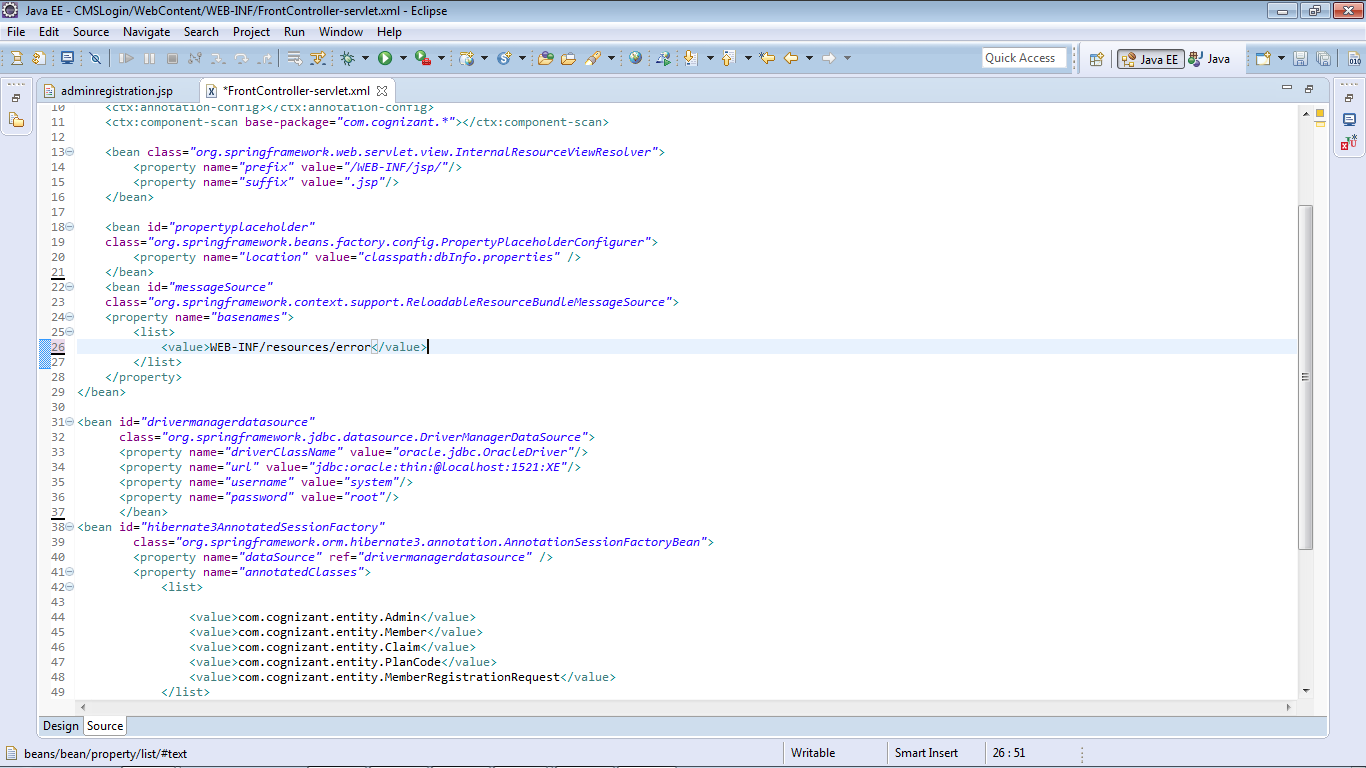


Fig: 4.6 – Front Controller-servlet.xml

**Chapter 5**

**Results and Discussions**

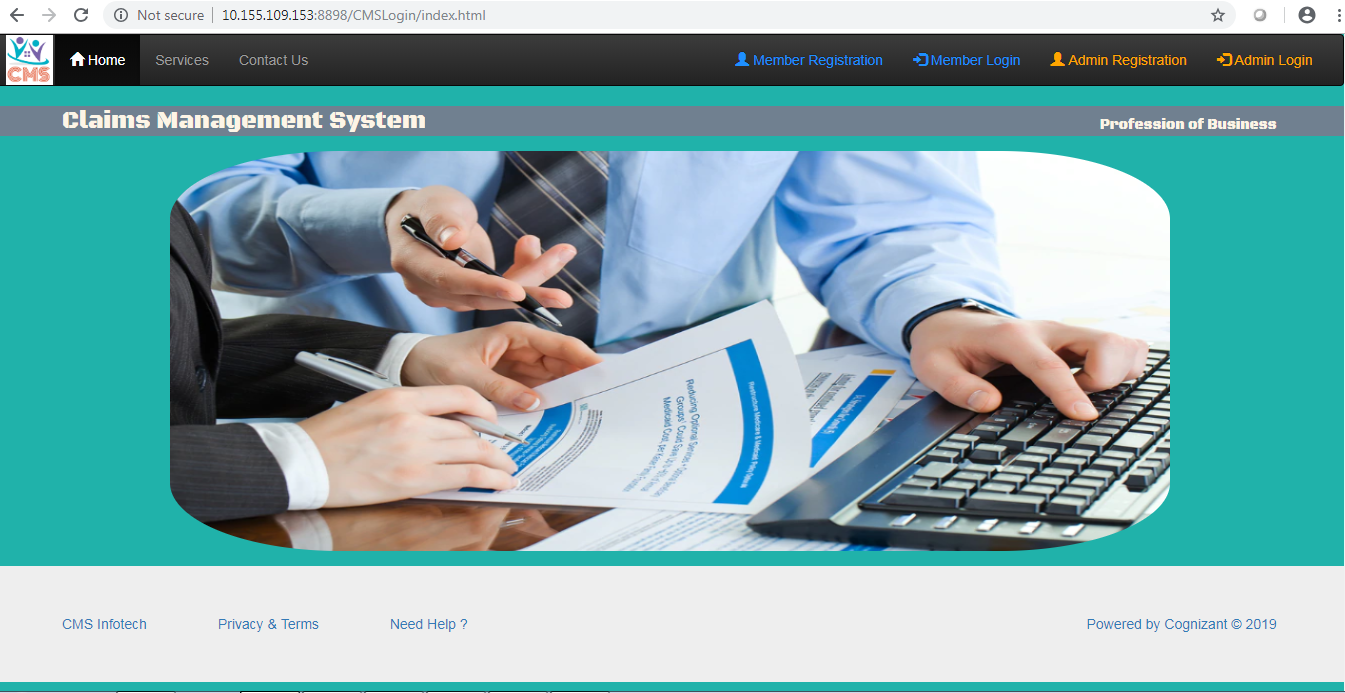
****

Fig: 5.1 – Homepage

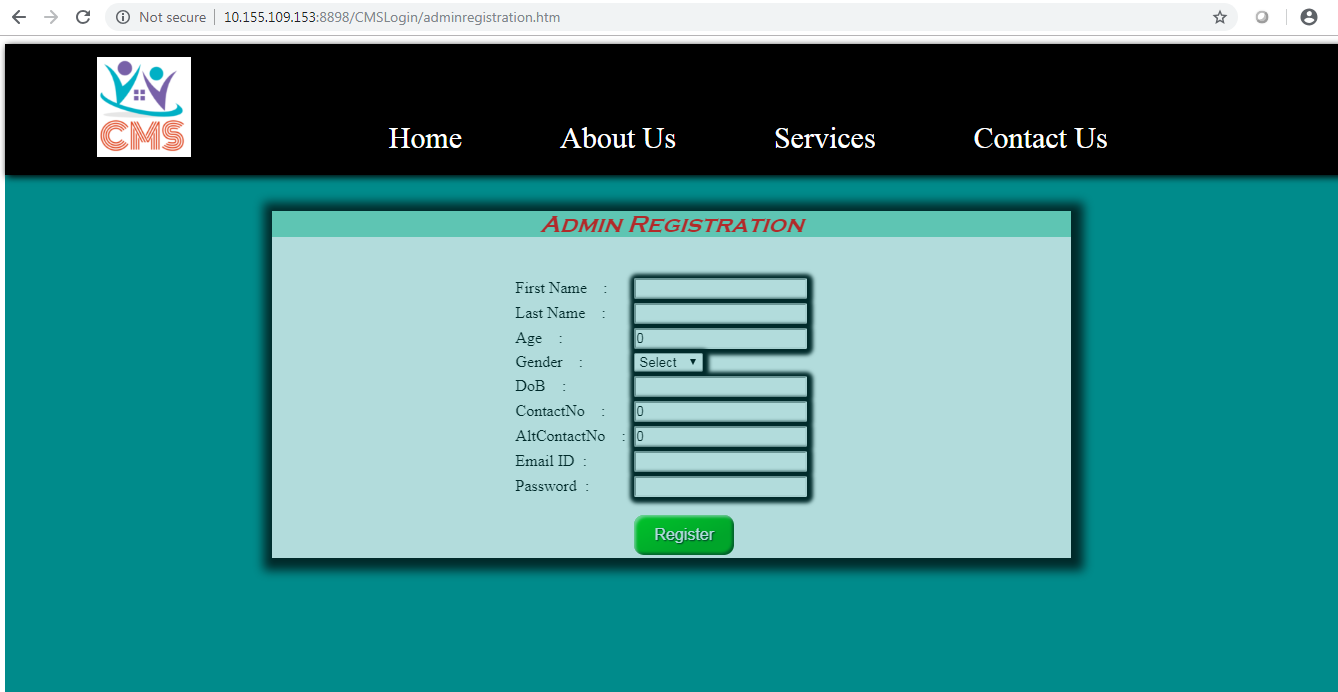
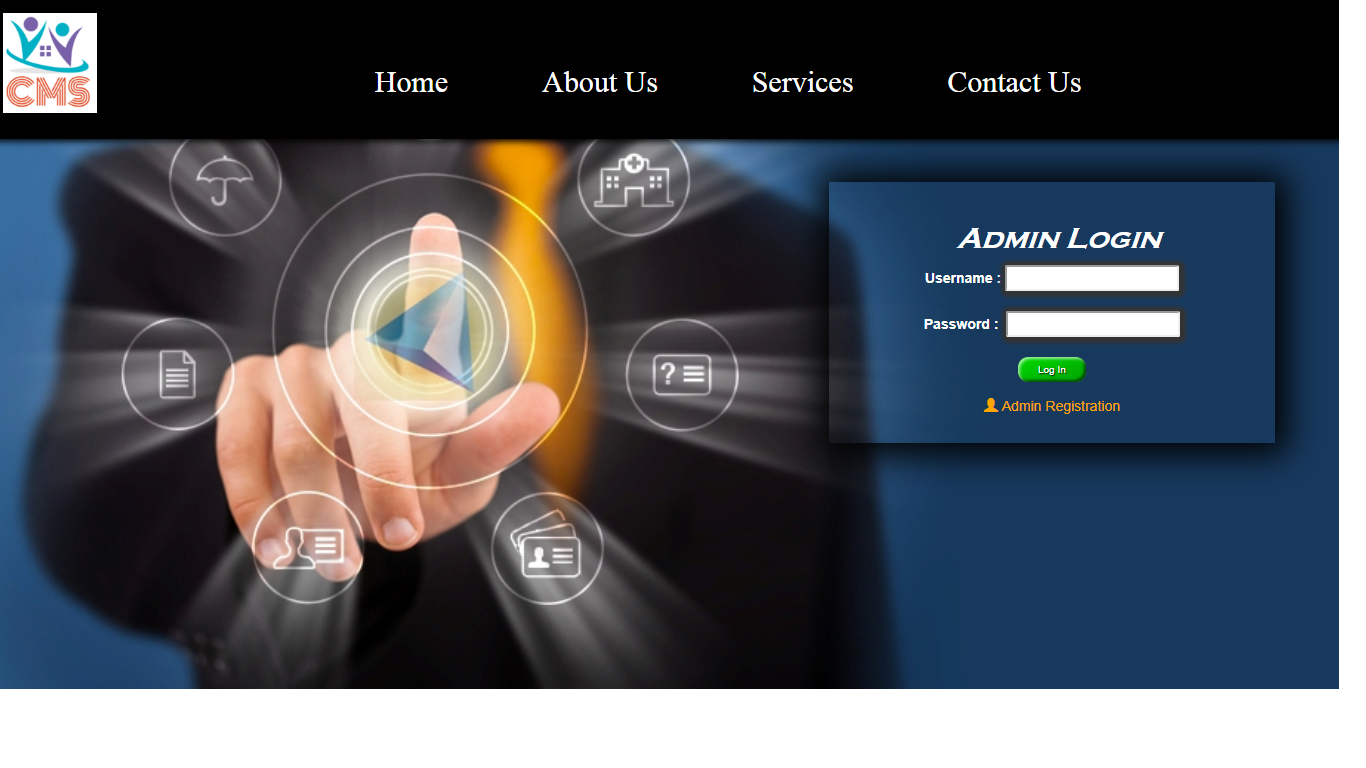
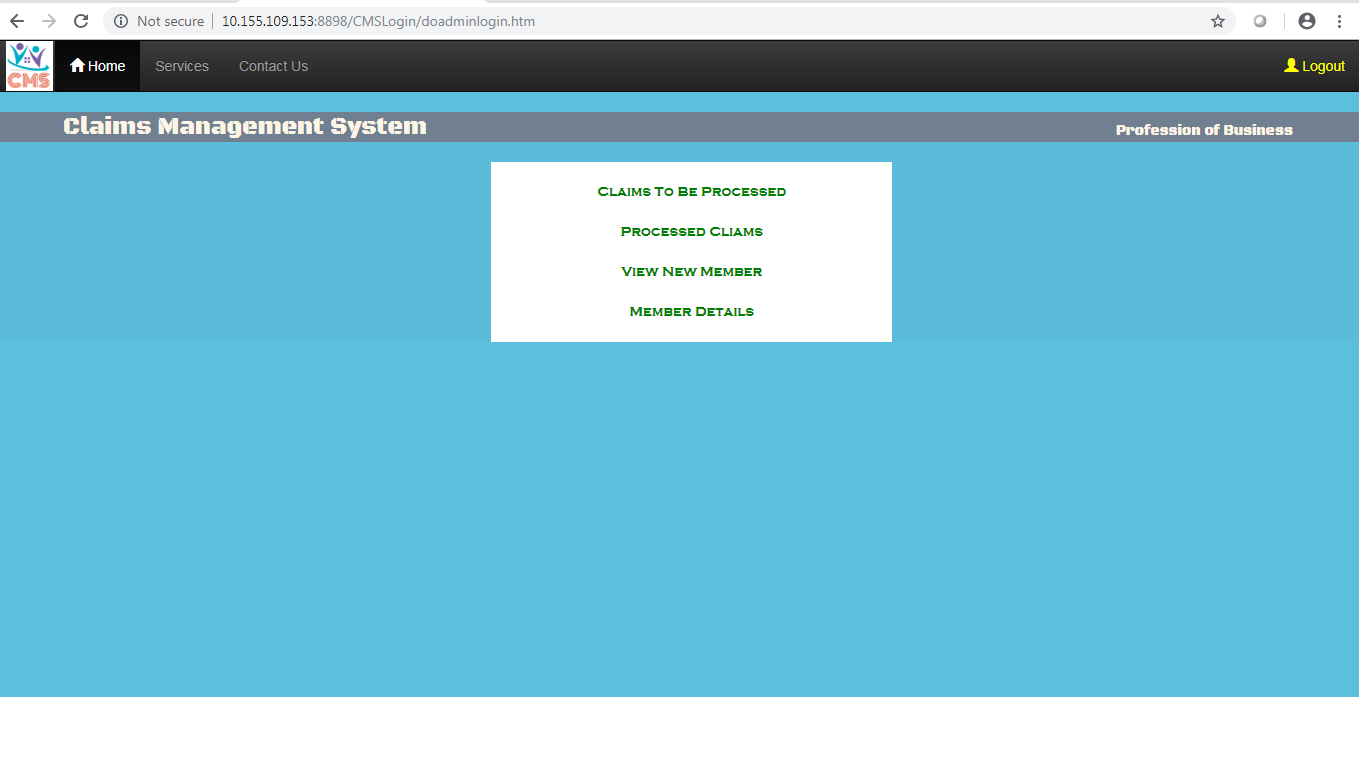


Fig: 5.2 – Admin Registration

Fig: 5.3 – Admin Login

Fig: 5.4 – Admin Dashboard

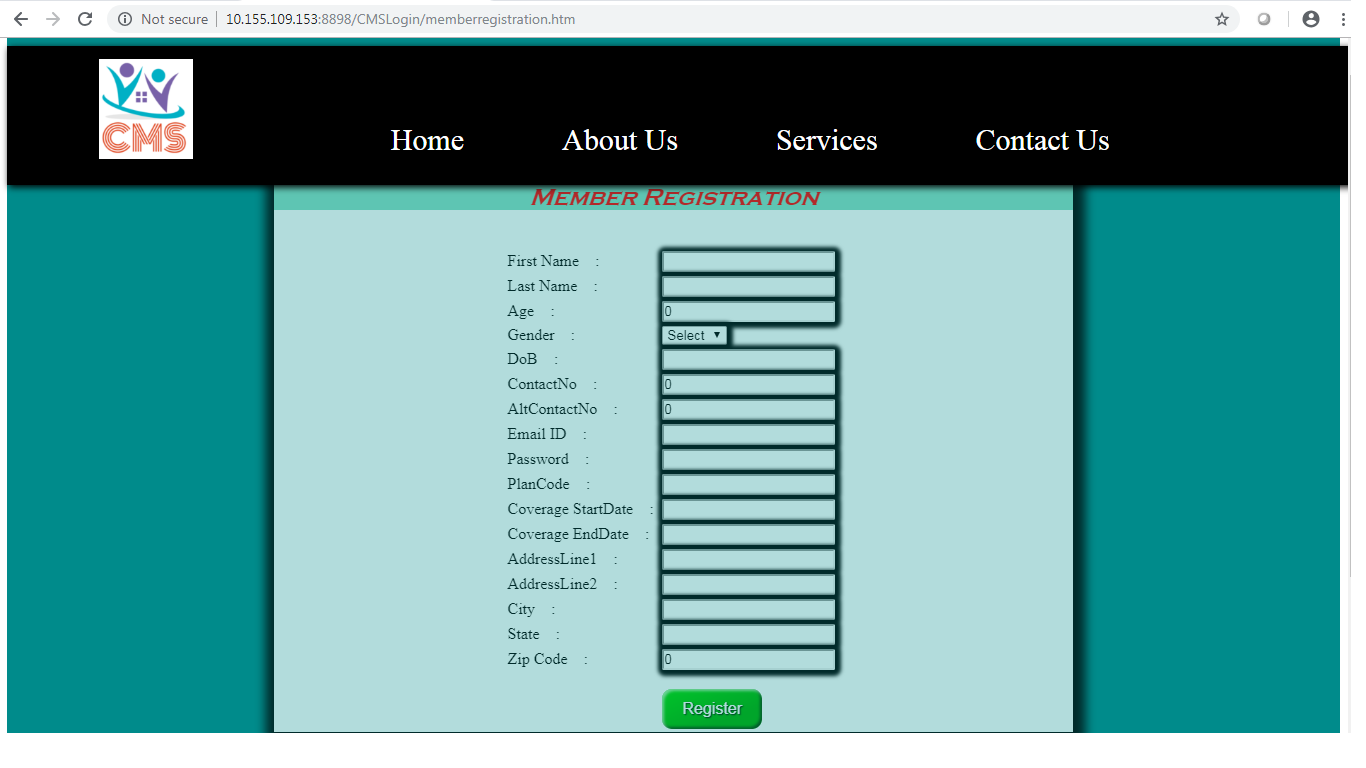


Fig: 5.5 – Member Registration

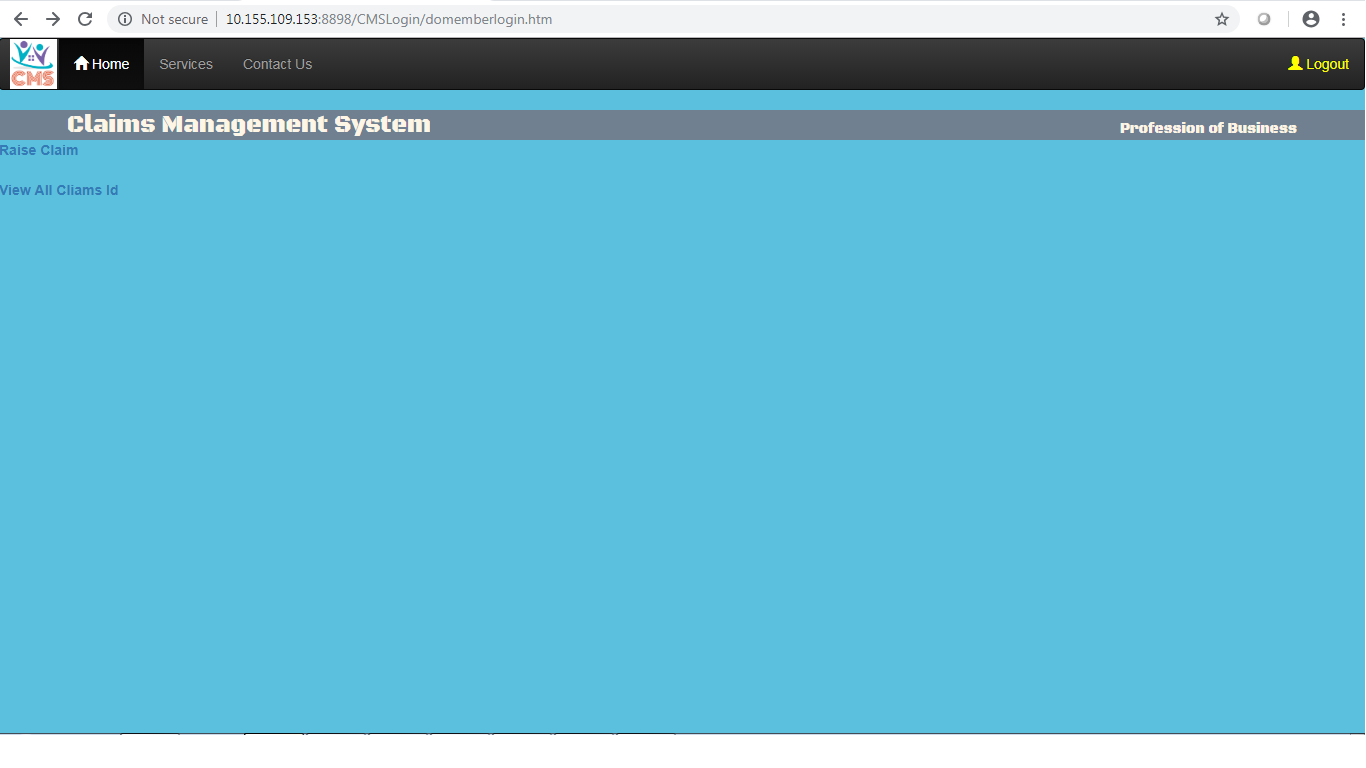
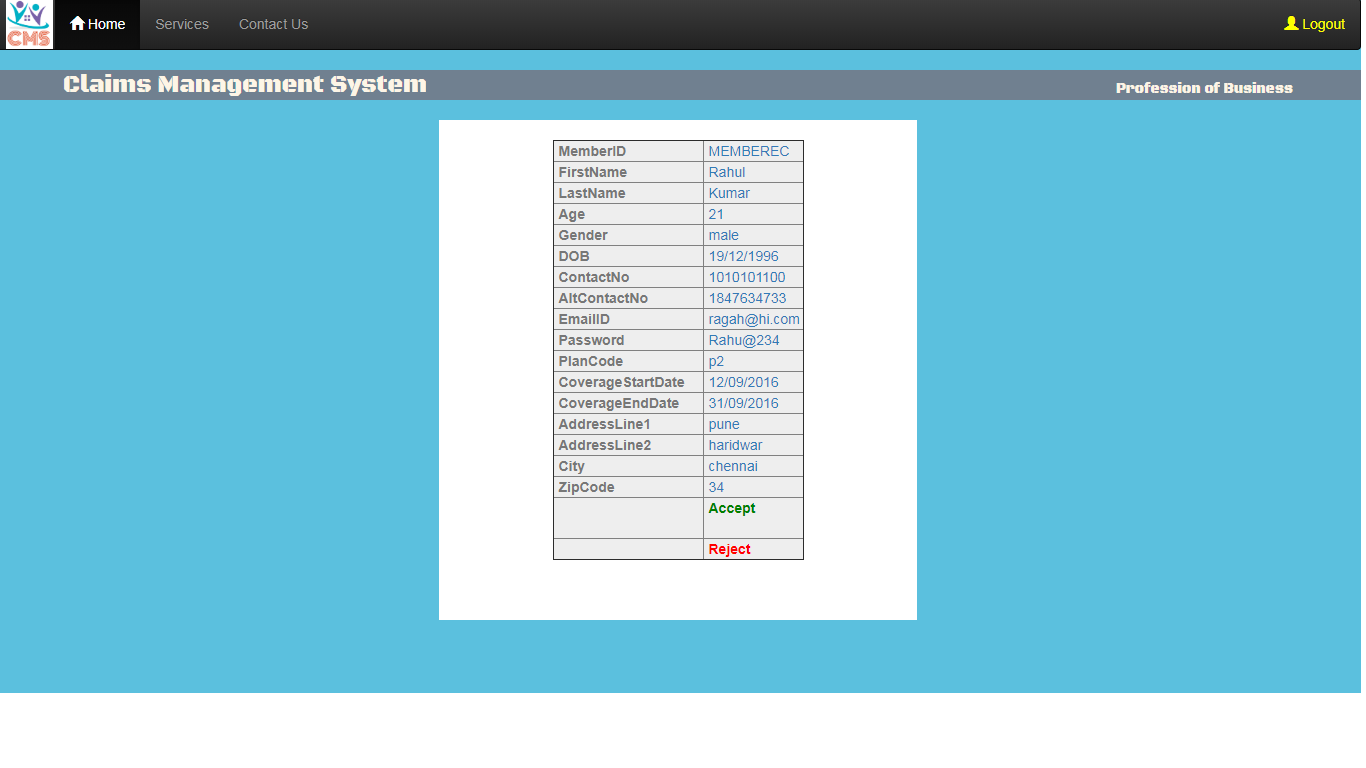


Fig: 5.6 – Member Dashboard

Fig: 5.7 – Member Request Approval

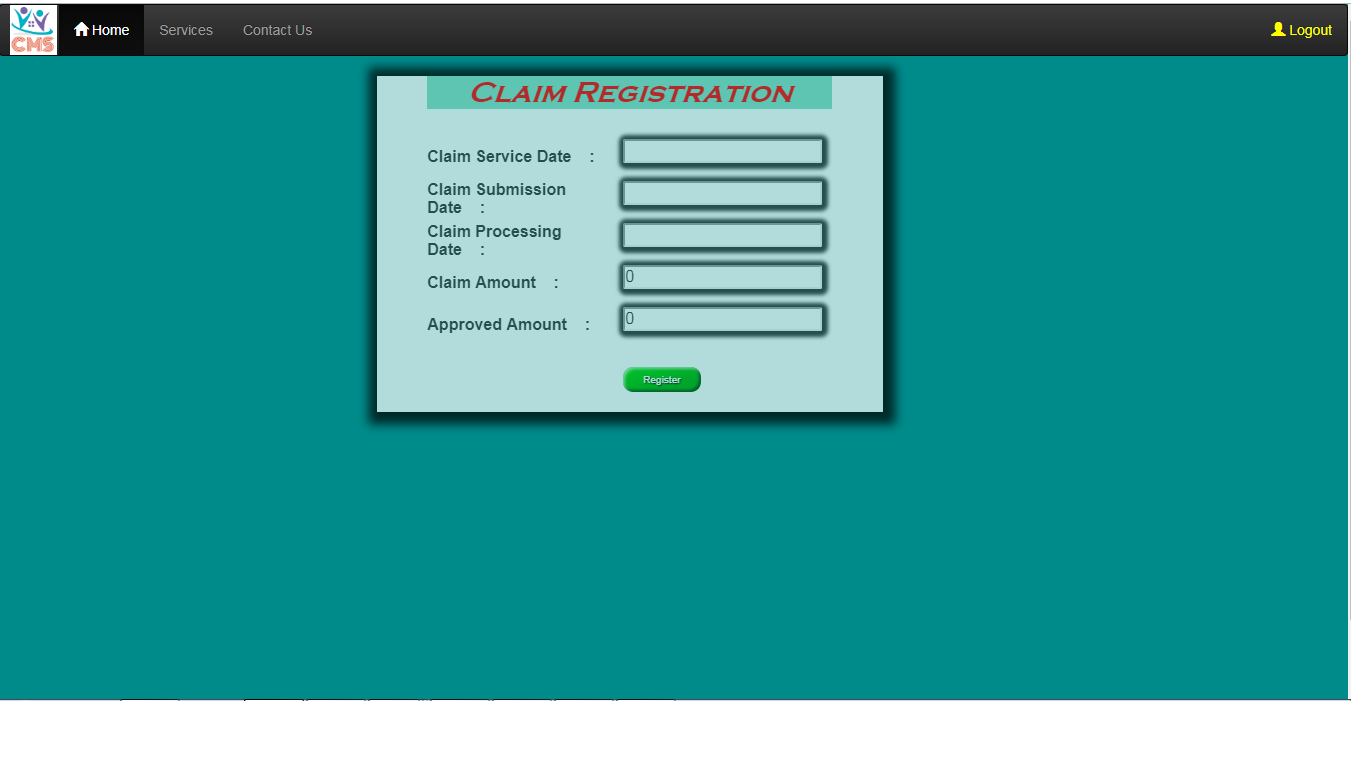


Fig: 5.8– Claim Registration

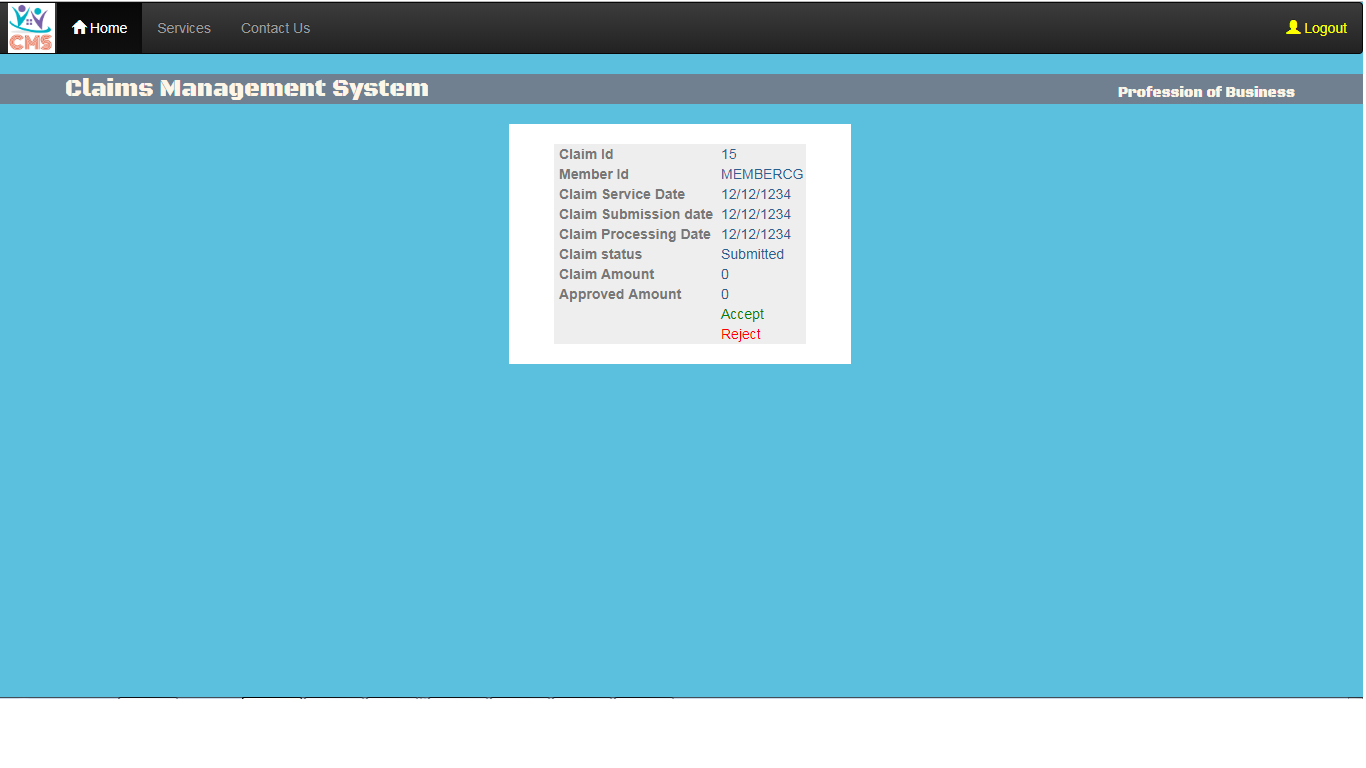
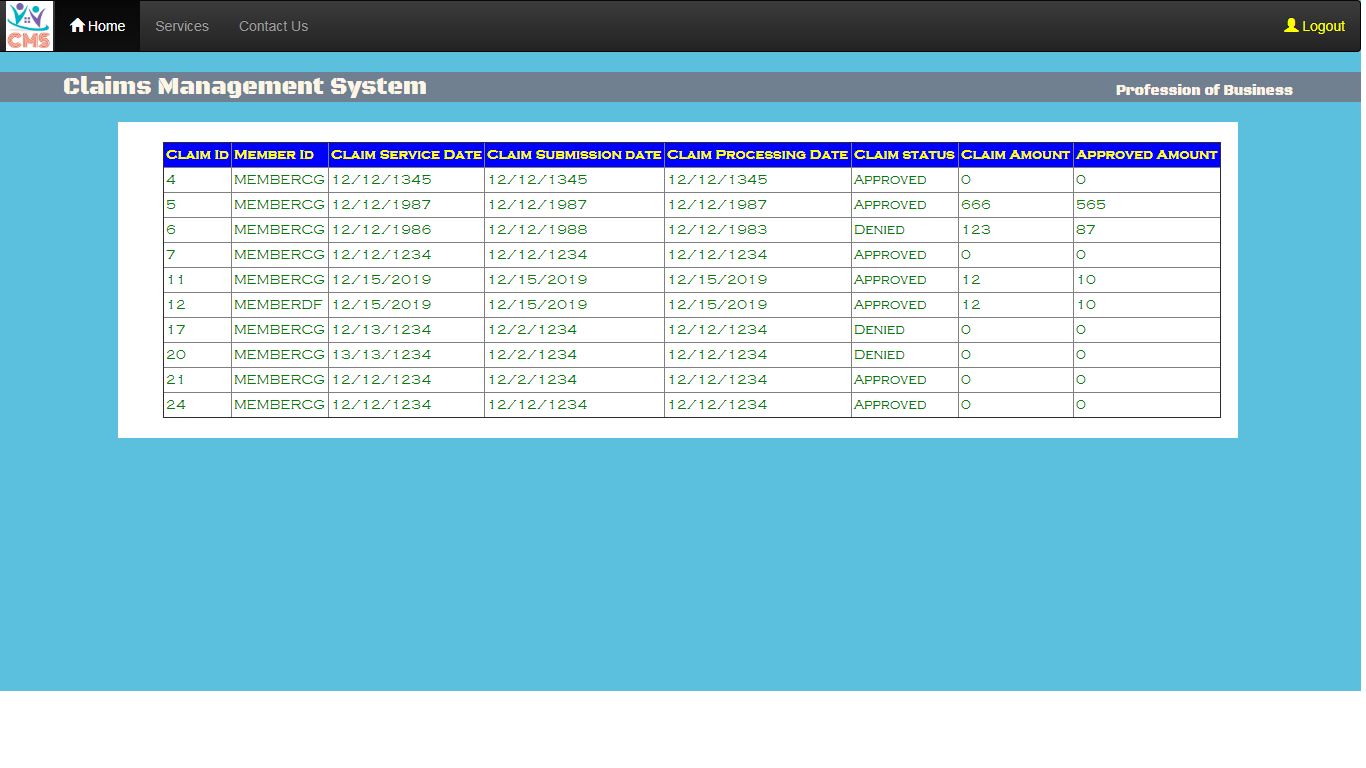


Fig: 5.9– Claim Approval

Fig: 5.10– Claim Details

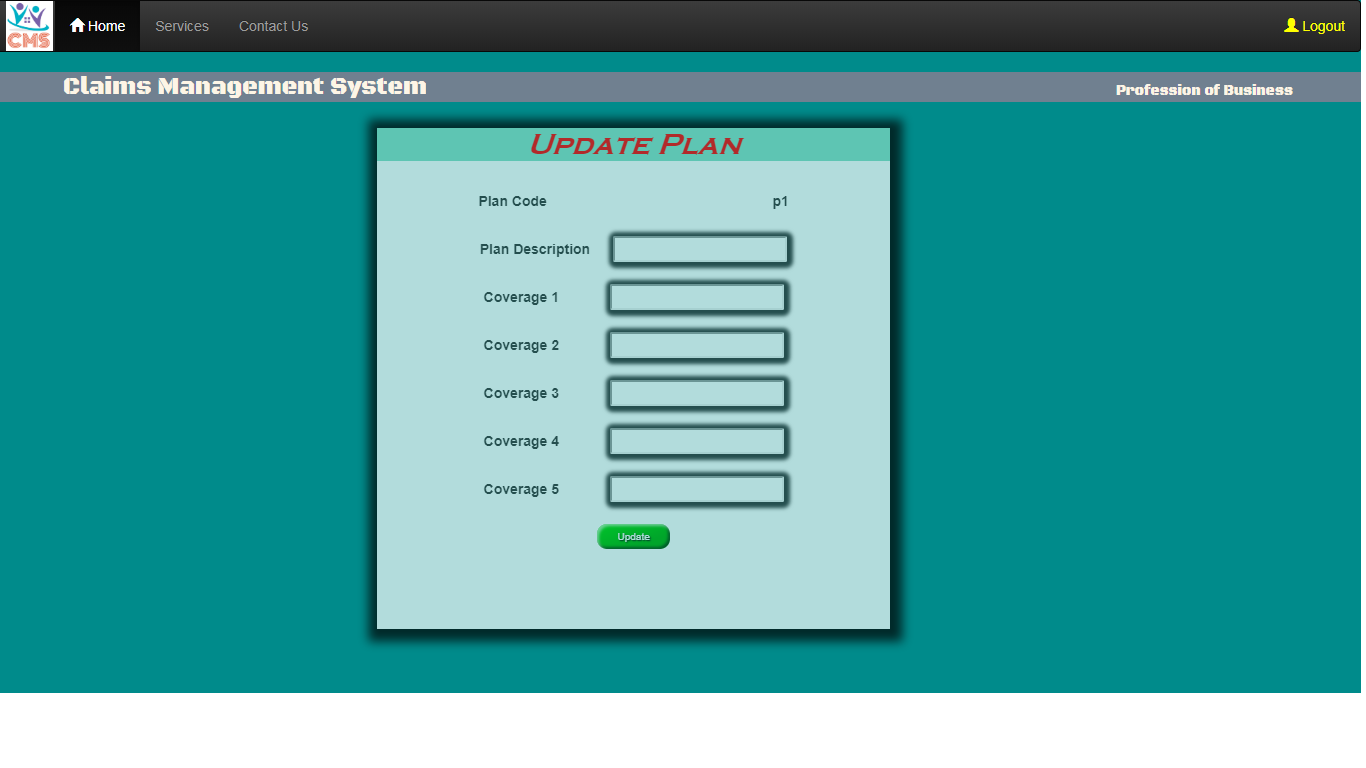


Fig: 5.11– Plan Code Update

**Chapter 6**

**Conclusions and Future Works**

**6.1 Conclusions**

This project of Managing Claims is successful in determining between pseudo and legal claims and hence is successful in implementing what was asked in the requirement specification. Claims submitted by the member are approved by the admin who is solely responsible for processing of claims as he only decides which claim request is pseudo/legal. Further member is notified if the claim is approved and the claim amount is credited into his account. Thus in this manner asked requirements are successfully implemented.

**6.2 Future Works**

* Development of mobile application for managing claims.
* Focussing more on ways of efficient processing of claims.
* Adding more security features.
* Making it more user friendly.

**Chapter 7**

**References**

[1]. Javatpoint (<https://www.javatpoint.com/>

[2]. w3schools (<https://www.w3schools.com/>)

[3]. tutorialspoint (<https://www.tutorialspoint.com/>)

[4]. geeksforgeeks (<https://www.geeksforgeeks.org/>)

[5]. stackoverflow (<https://stackoverflow.com/>)