

*A project report on*

# **CREATE AN OMNI-CHANNEL POLICY SERVICE APPLICATION**

*Submitted in partial fulfillment for the award of the degree of*

## **Bachelor of Computer Applications**

*by*

**KAVYAA RAO (17BCA0045)**

**Under the guidance of**

**Prof. Kavitha B.R.**

**School of Information Technology and Engineering**

**VIT, Vellore.**



**VIT<sup>®</sup>**

**Vellore Institute of Technology**

(Deemed to be University under section 3 of UGC Act, 1956)

**SCHOOL OF INFORMATION TECHNOLOGY AND  
ENGINEERING**

May, 2020

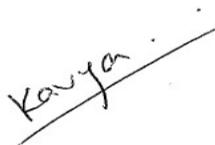
## **DECLARATION**

I hereby declare that the thesis entitled “Create an Omni-Channel Policy Service Application” submitted by me, for the award of the degree of Bachelor of Computer Applications, VIT is a record of bonafide work carried out by me under the supervision of Prof. Kavitha B.R.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

**Place:** Vellore

**Date:** 25-05-2020



**Signature of the Candidate**

## **CERTIFICATE**

This is to certify that the thesis entitled “Create an Omni-Channel Policy Service Application” submitted by KAVYA A RAO (17BCA0045), School of Information Technology and Engineering, VIT, for the award of the degree of Bachelor of Computer Applications is a record of bonafide work carried out by her under my supervision.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfils the requirements and regulations of VIT and in my opinion meets the necessary standards for submission.



**Signature of the Guide**

**Internal Examiner**

**External Examiner**

**AccelTree Software Private Limited.**

604, Supreme HQ, 6th Floor, Mumbai-Banglore Highway, Baner, Pune-411 045.  
Tel.No.: +91 20 67342000 | E-mail: info@acceltree.com | www.acceltree.com  
CIN: U72200PN 2001 PTC 016563



Date: 18/05/2020

### CERTIFICATE BY THE EXTERNAL GUIDE

This is to certify that the project report entitled "**CREATE AN OMNI-CHANNEL POLICY SERVICE APPLICATION**" submitted by **KAVYA A. RAO(17BCA0045)** to Vellore Institute of Technology in partial fulfillment of the requirement for the award of the degree of Bachelors of Computer Applications in Vellore is a record of bonafide work carried out by her under my guidance. The project fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

**Mr. Pramod Chavan**

### EXTERNAL SUPERVISOR

Designation: Tech Module Lead

AccelTree Software Pvt. Ltd., 604, Supreme HQ,  
Bombay Pune Highway, Baner, Pune 411045.

Email : [info@acceltree.com](mailto:info@acceltree.com)

Phone no : +912067342000



## ABSTRACT

When an individual customer purchases life insurance, during the lifetime of the insurance policy the customer might need information on the status of payments, or need to change contact details etc. To do so, the customer would use a policy servicing system. Such a system essentially would have a UI involving customer authentication followed by a series of screens through which data would be fetched from the back-end, or validated or updated.

The objective of this project is to build a complete working application that will cover all aspects of authentication, data retrieval, update and data validation; however, the project is not expected to be a system that offers all services that a customer would need. It would involve creating an application using MEAN stack that can form the basis for development of a larger life insurance policy servicing system in the future. The application should work across mobile and PC browsers of different types and give the same experience to all users. The application should also have components to integrate with multiple backend systems in real time via web services or via messaging queues.

## **ACKNOWLEDGEMENT**

It is my pleasure to express with deep sense of gratitude to **Prof. Kavitha B.R**, Assistant Professor (Senior), School of Information Technology and Engineering (SITE), Vellore Institute of Technology and **Mr. Pramod Chavan**, Tech Module Lead, Acceltree Software Private Limited, for their constant guidance, continual encouragement, understanding; more than all, they taught me patience in my endeavor. My association with them is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Web Application Development.

I would like to express my gratitude to **Dr G. Viswanathan**, Chancellor, **Mr. Sankar Viswanathan**, Vice President, **Dr. Anand A. Samuel**, Vice Chancellor, **Dr. S. Narayanan**, Pro-Vice Chancellor, and **Dr. Balakrushna Tripathy**, Dean, School of Information Technology and Engineering (SITE) for providing an environment to work in and for his inspiration during the tenure of the course.

In a jubilant mood I express ingeniously my whole-hearted thanks to **Dr. Malaserene I**, Head of Department of Smart Computing(HOD), School of Information Technology and Engineering(SITE), all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

**Place: Vellore**

**Date: 18-05-2020**

**Kavya A. Rao**

# **CONTENTS**

<b>CONTENTS</b>	iii
<b>LIST OF ACRONYMS</b>	vi
<b>LIST OF FIGURES</b>	vii
<b>CHAPTER 1</b>	
<b>INTRODUCTION</b>	
1.1 INTRODUCTION ABOUT AREA	1
1.1.1 ADVANTAGES OF APPLICATION	1
1.2 BACKGROUND	1
1.3 PROBLEM STATEMENT	2
1.4 MOTIVATION	2
1.5 OBJECTIVES	2
1.6 SCOPE OF THE PROJECT	3
<b>CHAPTER 2</b>	
<b>LITERATURE SURVEY</b>	
2.1 RELATED WORK	4
2.2 FEASIBILITY STUDY	7
2.3 PROBLEM FORMULATION	7

## **CHAPTER 3**

### **SYSTEM DESIGN**

3.1 SYSTEM MODULES	8
3.2 UML DIAGRAMS	9
3.2.1 CLASS DIAGRAM	9
3.2.2 SEQUENCE DIAGRAM	11
3.2.3 ACTIVITY DIAGRAM	20
3.2.4 USE CASE DIAGRAM	25
3.3 REQUIREMENTS SPECIFICATION	30
3.3.1 SOFTWARE REQUIREMENT	30
3.3.2 HARDWARE REQUIREMENT	30
3.4 TECHNOLOGY USED	30

## **CHAPTER 4**

### **IMPLEMENTATION OF SYSTEM**

4.1 INTRODUCTION	31
4.2 MODULE DESCRIPTION	31
4.2.1 USER MODULE	31
4.2.2 PAGE DESCRIPTION	32
4.3 SAMPLE SOURCE CODE	35
4.3.1 CLAIM.COMPONENT.HTML	35
4.3.2 HEADER.COMPONENT.TS	38

4.3.3 HTML_PDF.JS	41
4.3.4 FREQ.BAT	42
4.3.5CONNECTTODB.JAVA	43
4.4 TEST CASES	45

## **CHAPTER 5**

### **RESULT AND DISCUSSION**

5.1 SNAPSHOTS OF PROJECT (with description)	47
---	----

## **CHAPTER 6**

### **CONCLUSION AND FUTURE WORK**

6.1 CONCLUSION	63
6.2 FUTURE WORK	63

REFERENCES	x
------------	---

BIODATA	xi
---------	----

## **LIST OF ACRONYMS**

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
Pdf	Portable Document Format
JSON	JavaScript Object Notation
Jpg/Jpeg	Joint Photographic Expert Group
Png	Portable Network Graphics

## **LIST OF FIGURES**

<b>Figure No.</b>	<b>Title</b>	<b>Page No.</b>
3.1	System Modules	8
3.2	Class Diagram-1	9
3.3	Class Diagram-2	10
3.4	Sequence Diagram For Policy Servicing Module-1	11
3.5	Sequence Diagram For Policy Servicing Module-2	12
3.6	Sequence Diagram For Customer Search Module	13
3.7	Sequence Diagram For Claim Management Module-1	14
3.8	Sequence Diagram For Claim Management Module-2	15
3.9	Sequence Diagram For Request Management Module-1	16
3.10	Sequence Diagram For Request Management Module -2	17
3.11	Sequence Diagram For Request Management Module-3	18
3.12	Sequence Diagram For Nominee Module	19
3.13	Activity Diagram For Personal Details Module	20
3.14	Activity Diagrams For Customer Search Module	21
3.15	Activity Diagrams For Claim Management Module	22
3.16	Activity Diagram For Request Management Module	23
3.17	Activity Diagram For Nominee Module	24
3.18	Use Case Diagram For Personal Details Module	25
3.19	Use Case Diagram For Customer Search Module	26
3.20	Use Case Diagram For Claim Management Module	27
3.21	Use Case Diagram For Request Management Module	28

<b>Figure No.</b>	<b>Title</b>	<b>Page No.</b>
3.22	Use Case Diagram For Nominee Module	29
5.1	Login Page	47
5.2	Dashboard For Personal Details	47
5.3	Personal Details	48
5.4	Nominee Details	48
5.5	Addition of a Nominee	49
5.6	Policy Details	49
5.7	Payment Receipts	50
5.8	Policy Document	50
5.9	Requests	51
5.10	File Upload	51
5.11	Edit Payment Frequency	52
5.12	Edit Name	52
5.13	Request History	53
5.14	Dashboard For Customer Search	53
5.15	Customer Details	54
5.16	Initialization of Claims	54
5.17	Claims Initialized	55
5.18	Claim Updation and Submition	55
5.19	Dashboard For Claim Management	56
5.20	Claim Management	56
5.21	Claim Form	57
5.22	Accept or Reject Claims	57

<b>Figure No.</b>	<b>Title</b>	<b>Page No.</b>
5.23	Claim Report of Accepted Claims	58
5.24	Claim Report of Rejected Claims	58
5.25	Document View	59
5.26	Document Display	59
5.27	Dashboard For Request Management	60
5.28	Requests	60
5.29	Request Display	61
5.30	Request Display For Name Change	61
5.31	Dashboard For Nominee	62
5.32	Claim Submitions	62

# **Chapter 1**

## **Introduction**

### **1.1 OMNI-CHANNEL POLICY SERVICE APPLICATION**

Omni-channel policy service application is an application which aims in offering the various number of users , who have purchased insurance policies, with services that they can carry out digitally without the use of any manual effort.The application aims at digitalizing as many services possible as well as reducing manual effort on the customers part.

#### **1.1.1 ADVANTAGES OF POLICY SERVICE APPLICATION**

The application digitalizes all the services offered to the customer after they purchase the insurance policy. It reduces any need or requirement of the user to put in any manual effort such as submitting documents required, initiating claims and so on. The application is flexible to connect with any pos system.It is not time consuming.Many applications such as addition and deletion of nominees and so on are automated.Wherever Documents are required to validate a request the customer is allowed to submit via a file upload function which is included instead of physically submitting the documents.The system has image manipulation which will make it easier for both the customer as well as the admin.

### **1.2 BACKGROUND**

Everybody needed to protect themselves against the many risks in life. Farmers wanted crop insurance.People wanted deposit insurance at their banks. Travellers wanted travel insurance. Everybody turned to insurance companies to give them peace of mind. And really, isn't that what insurance is – the paying of a premium to protect against some form of loss. This project is aimed at automating the insurance service.Now the insurance industry was growing to huge proportions. The companies, though competitive, worked together to create productive systems that could be used throughout the industry. They needed to keep up with the requirements of the increasing amount of laws governing insurance.

### **1.3 PROBLEM STATEMENT**

Tracking a policy and tracking the performance of sales force in insurance companies have become methodical and almost flawed in the manual system of tracking and evaluation of sales force performance. It is hard to centralize the core information of policy holders because the manual workload high, thus the chances of error is high. The manual system administration does not ensure complete control over the database thus control over all sorts of fraudulent activities cannot be ensured. The intra-network and coordination between different departments of an insurance business is ineffective and time consuming.

### **1.4 MOTIVATION**

Currently there are many systems which provide their customers with life insurance. Most of these systems provide policy services but most of the applications provided by them require the customer to apply or request it manually. For example if a customer wants to add or delete a nominee/beneficiary the customer will be required to manually go to the company to submit the documents. There are very few systems present which perform all the actions automatically. The proposed system will work like the other systems but include various other features. In this system the customer who has purchased the life Insurance will be offered many services such as they will be able to change their address if required, add a new nominee/beneficiary, delete a new nominee/beneficiary, change the details of an existing nominee, change their payment frequency, or change their name under some circumstances and many more services. But unlike other Systems the customer is not required to do anything manually all services are performed via the webpage they can apply for the service as well as submit important documents. The customer will also be able to view their personal, nominee/beneficiary as well as product details, download receipts and check out their policy documents. At the same time the admin will be able to manage all the requests and claims made. If the customer wants to make a claim all they have to do is call the customer service who manages the full process. The admin will also be able to check all the claims made or the requests made and ask for more documents if required.

### **1.5 OBJECTIVES**

The objective of Policy Servicing is typically to interact with the customers in order to Change Request, resolve support issues, improve credibility, and relationships. It is also to lessen the time consumption as well as to reduce manual labour as much as possible hence making the system more user friendly as well as ideal.

## **1.6 SCOPE OF THE PROJECT**

The application offers customers many services such as viewing personal details, nominee details, policy details, downloading policy documents and payment receipts, upload documents as required, edit payment frequency or name as desired as well as view the history of the requests they have made. On the admin side the application allows the user to initialize claims on the customers request, search for a particular customer, save a claim form, update a claim form as well as submit the form. Further they can accept or reject the form, create a new user, view the accepted and rejected claims as well as view the documents uploaded by the user required by them and approve the claim as they wish. The user also allows the user to view, manage, accept or reject requests as they wish they are also given the opportunity to request for more documents if required.

## Chapter 2

# Literature Survey

### 2.1 RELATED WORK

1. **Journal Name:** Journal of Database Marketing and Customer Strategy Management

**Paper Name:** Customer behavior toward online insurance services in India

**Author/s:** Arpita Khare,Saumya Dixit , Richa Chaudary,Priyanka Kochhar and Shruti Mishra.

**Methodology:** The questionnaire designed to measure customers' perceptions about online insurance services was adapted. This is in line with the growing adoption of technology by insurance companies in India in enabling transaction and improving accessibility. The total items were 23: 2 items in the scale related to awareness and use of online insurance services and the rest of the 21 items were related to various dimensions of online insurance services. The questionnaires were administered in English, as most Indians are fluent in the English language. The responses of the customers were taken on a five-point Likert scale with responses varying on the scale of 1 for strongly agree and 5 for strongly disagree.

**Description:** The use of web-based technologies as a service delivery medium has added new elements to service dissemination. Escalation in self-service technologies has provided customers with multiple choice for using services offered by an organization. The online service attributes of 'convenience' and 'ease of use' are being given priority by companies for designing web-based services. In India, online shopping in general and online insurance services in particular, are yet to gain momentum. A large population still remains skeptical about its relative advantage over the traditional delivery channels. Limited Internet accessibility coupled with low technology literacy makes customers suspicious about online insurance services. The current research was directed toward understanding Indian customers' behavior toward using online insurance services. Correlation, ANOVA and multiple regression tests were run to understand customer behavior. The results suggest that improvement in technological attributes of online insurance Web sites can enhance customers' service usage. 4

**Pitfall:** The research was conducted only in the city of Allahabad, which may not be considered to be an adequate description of the total Indian customers' online insurance purchase behavior. The reason for selecting a non-metropolitan city was to understand non-metropolitan customers' behavior toward online services. In metropolitan cities, the Internet connectivity and communication networks are better; therefore people may be more willing to use online services. In smaller cities, there are frequent communication network disruptions, which make customers skeptical about online services. One of the limitations of the study is over representation of female population in the sample. The findings may differ if there were more number of males in the sample

2. **Journal Name:** Spanish Journal of Marketing - ESIC

**Paper Name:** Omni-channel users and omni-channel customers: a segmentation analysis using distribution services

**Author/s:** Mónica Cortiñas, Raquel Chocarro, Margarita Elorz

**Methodology:** Using a multinomial logit model, the authors perform a customer segmentation based on observed omni-channel behavior, considering the explanatory roles of demographics, distribution service features and customer service policies across the different retail channels.

**Description:** Consumers are increasingly combining distribution channels, thus displaying so-called omni-channel behavior, both to complete a given purchase and between purchases. The authors make a distinction between omni-channel customers, who make use of distribution services in both channels and omni-channel users, who make partial use of the distribution services of one channel to support purchases in another. This paper aims to identify the omni-channel behavior among the customers of a global fast fashion retailer dealing in a wide range of apparel and clothing accessories.

**Pitfall:** Although this three-group customer segmentation can be extended to any retail sector offering a variety of online and offline distribution services, the size of the segments observed in this study may be conditioned by the fact that the fashion industry deals in sensory goods, in relation to which the match between product and consumer is a major issue. Other factors potentially influencing this customer segmentation outcome are the large market share and extensive store network owned by this particular retailer.

This suggests an interesting area for future research extending the analysis to other, less sensory, products, to determine whether the findings regarding the distribution services that play the most decisive role in the segmentation still hold.

3. **Journal Name:** Journal of Insurance, Financial Markets and Consumer Protection No. 26 (4/2017): 67-78

**Paper Name:** Technological innovations in the insurance industry

**Author/s:** Lyubov KLAPKIV, Jurij KLAPKIV1 Maria Curie-Skłodowska University in Lublin, University of Łódź

**Methodology:** SWOT-analysis, questionnaire, correlation-regression model, multiple-regression analysis, overview

**Description:** The main idea of this article is an overview and analysis of the technological innovations in the insurance industry. First of all, the authors have characterised the driving factors of innovative development in the insurance industry. Secondly, the innovation in the financial sector was described as an invention and as an adoption. The main part of the paper was based on an overview of the use of technological innovations (such as software, analytics, sensors, algorithmic) for the effective insurance value chain. At the end, the authors have made a SWOT-analysis of the technological innovations.

**Pitfall:** SWOT-analysis has showed us that InsurTech has a great potential and could provide benefits to the insurance industry (for instance, effective risk underwriting, decrease of insurance frauds, higher effectiveness of the claim process), but also can bring threats (dependence on the quality of technicians, lack of a clear state regulation, risks of cyber-attacks). That is why the risks of radical innovations have to be estimated on each level of this process and have to be regulated by the authority of states for the protection of consumer rights.

## **2.2 FEASIBILITY STUDY**

There are many applications as well as companies which offer the same services to their customers. The existing systems allow the users to submit requests as well as perform other services but the main disadvantage is that the user to access these services have to do some manual effort as well as the company has to manually perform the operations which is hence time consuming. There is no system yet that has digitized all the services and has no requirement of any manual effort. There is also no existing system that has few services being automatically managed and performed. If these are followed through the system will become more efficient as well as more user friendly.

## **2.3 PROBLEM FORMULATION**

There are many existing systems that have been functioning but they are not the ideal applications as they have many limitations. The limitations vary from the application having security issues as well as facing frauds to the application being time consuming as well as requiring a lot of effort on both the user as well as the companies side. There is also the case of tracking any request initiated or claims initiated as well as products and policies purchased.

# Chapter 3

## System Design

### 3.1 SYSTEM MODULES

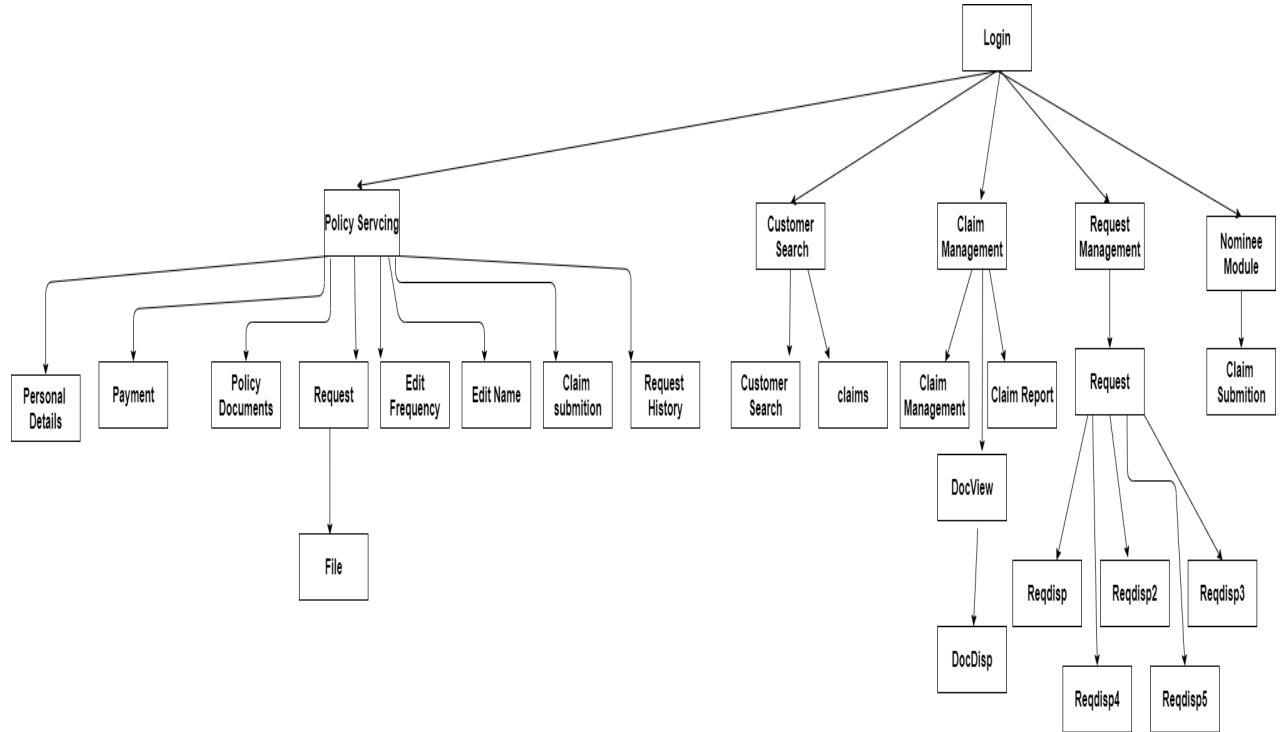


Fig. 3.1 System Modules

## 3.2 UML DIAGRAMS

### 3.2.1 CLASS DIAGRAMS

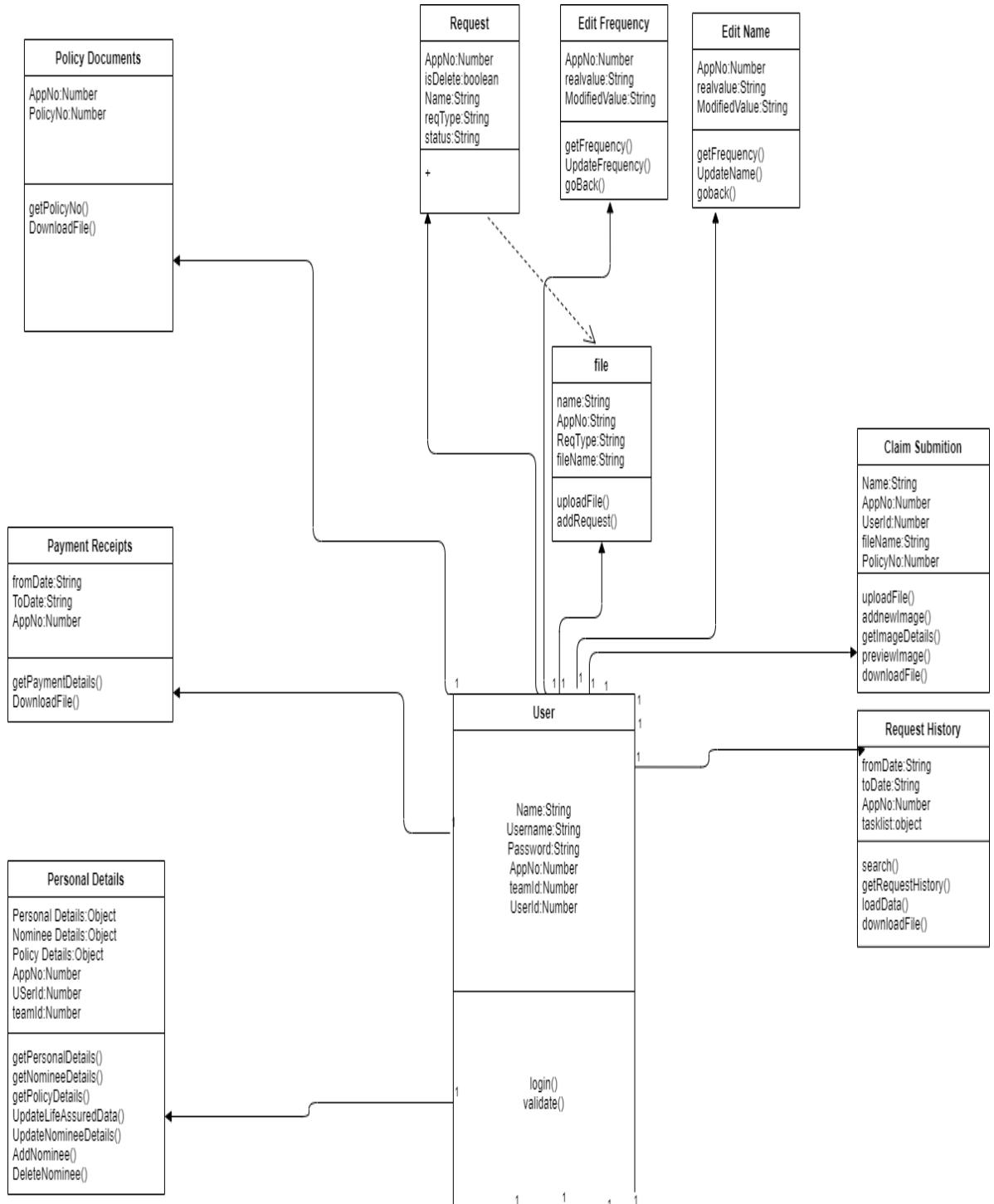


Fig. 3.2 Class Diagram -1

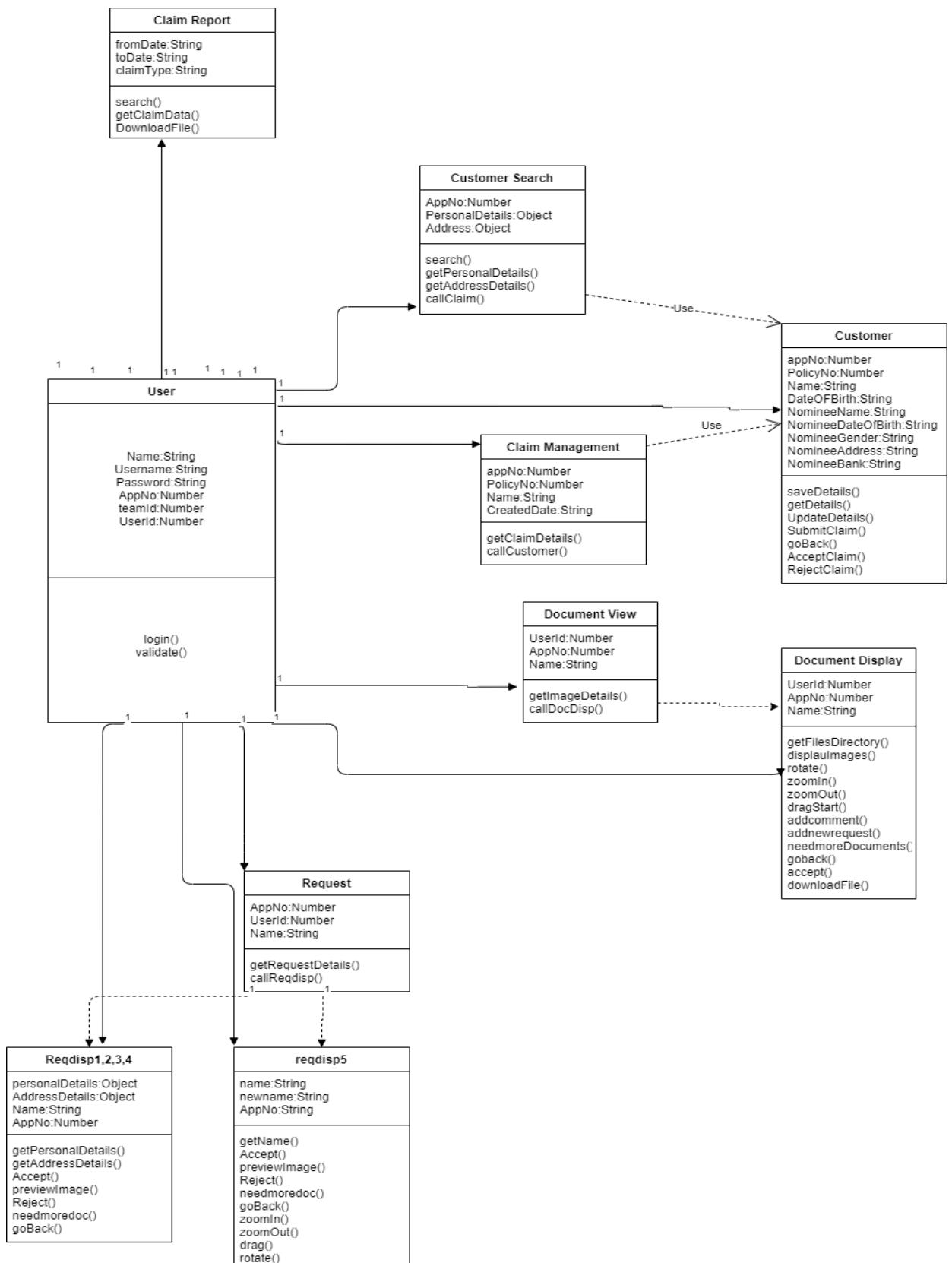


Fig. 3.3 Class Diagram -2

### 3.2.2 SEQUENCE DIAGRAMS

#### SEQUENCE DIAGRAM FOR POLICY SERVICING MODULE

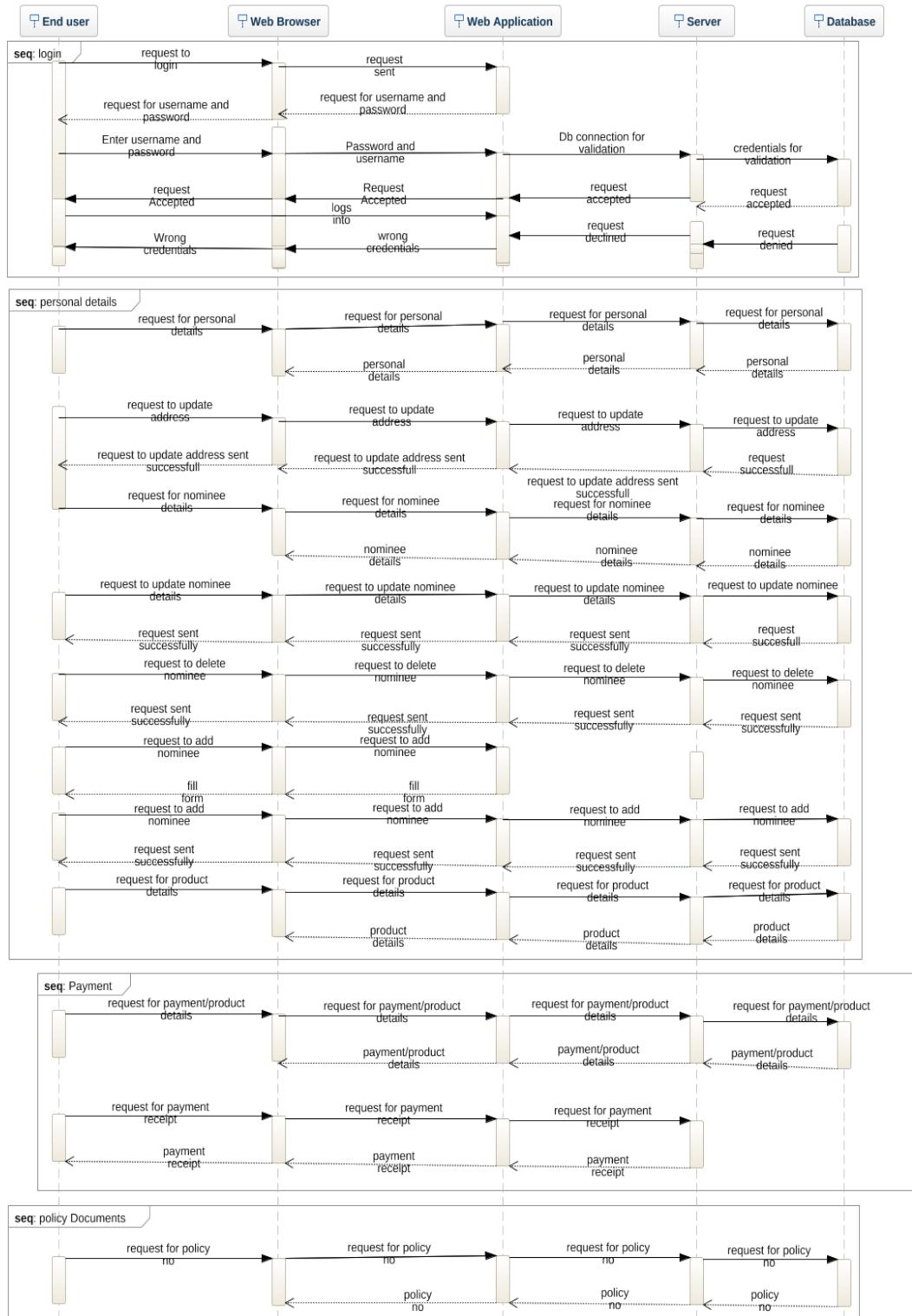


Fig. 3.4 Sequence Diagram for Policy Servicing Module -1

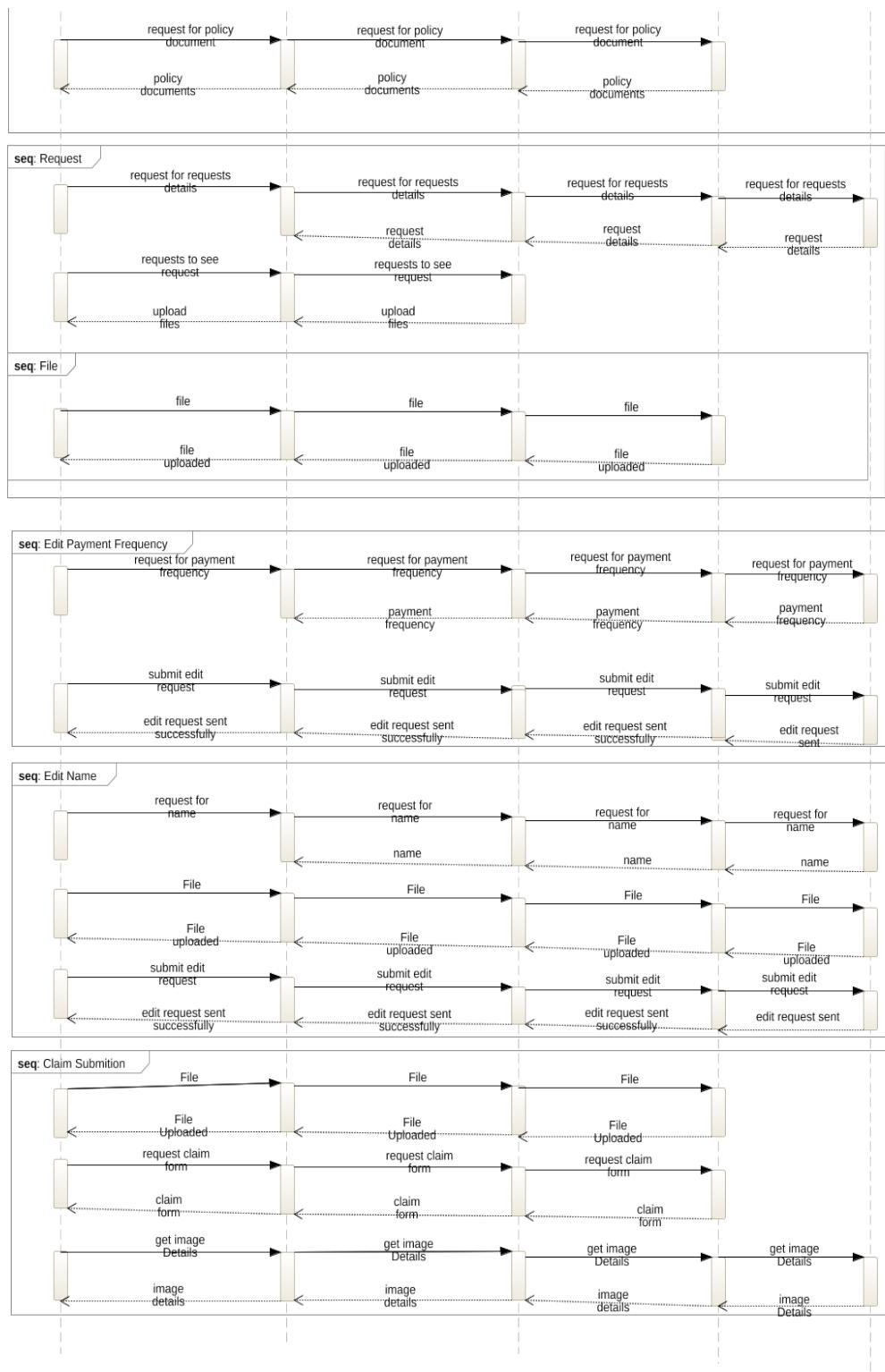


Fig. 3.5 Sequence Diagram for Policy Servicing Module -2

## SEQUENCE DIAGRAM FOR CUSTOMER SERVICE MODULE

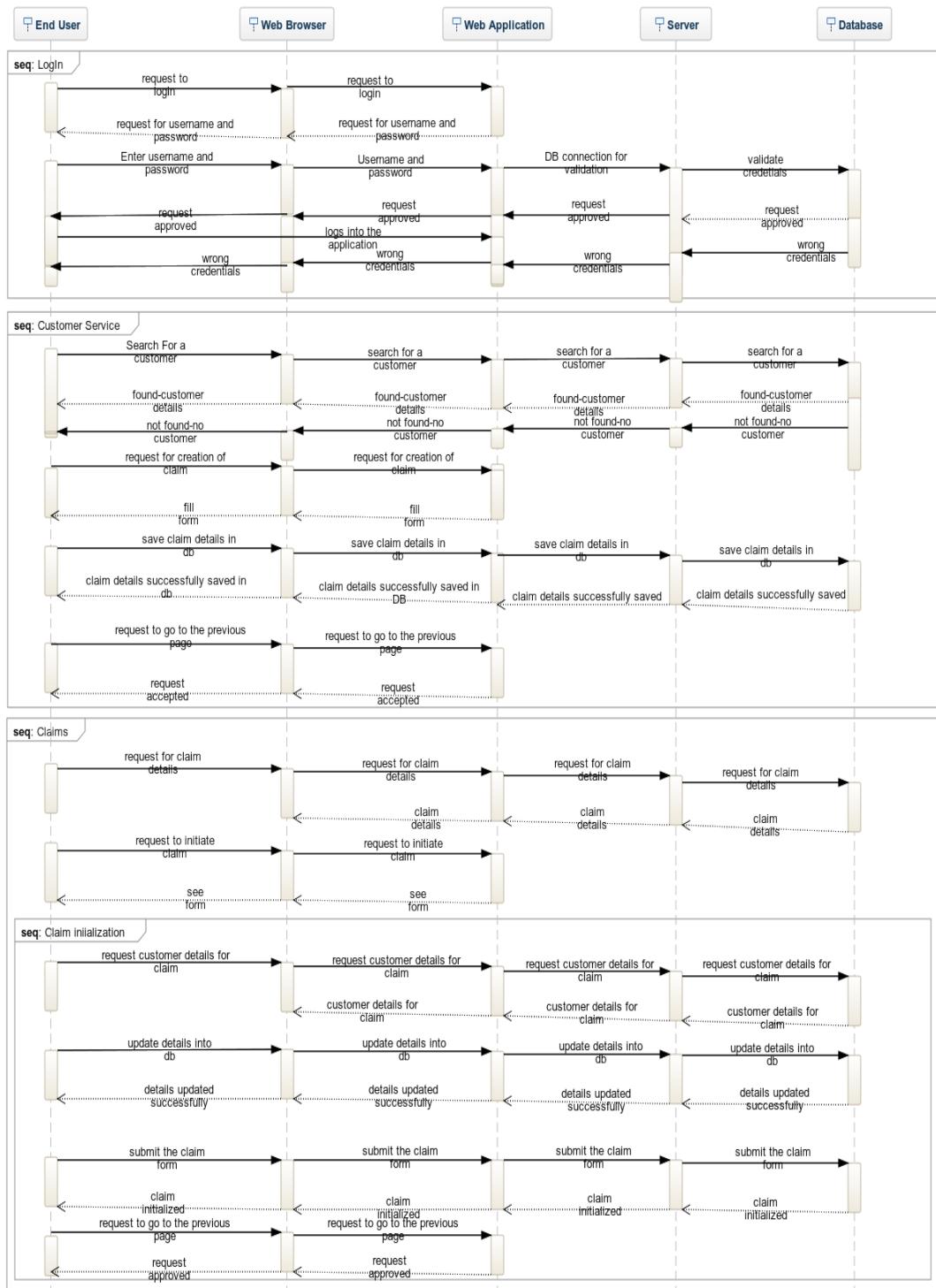


Fig. 3.6 Sequence Diagram for Customer Search Module

## SEQUENCE DIAGRAM FOR CLAIM MANAGEMENT MODULE

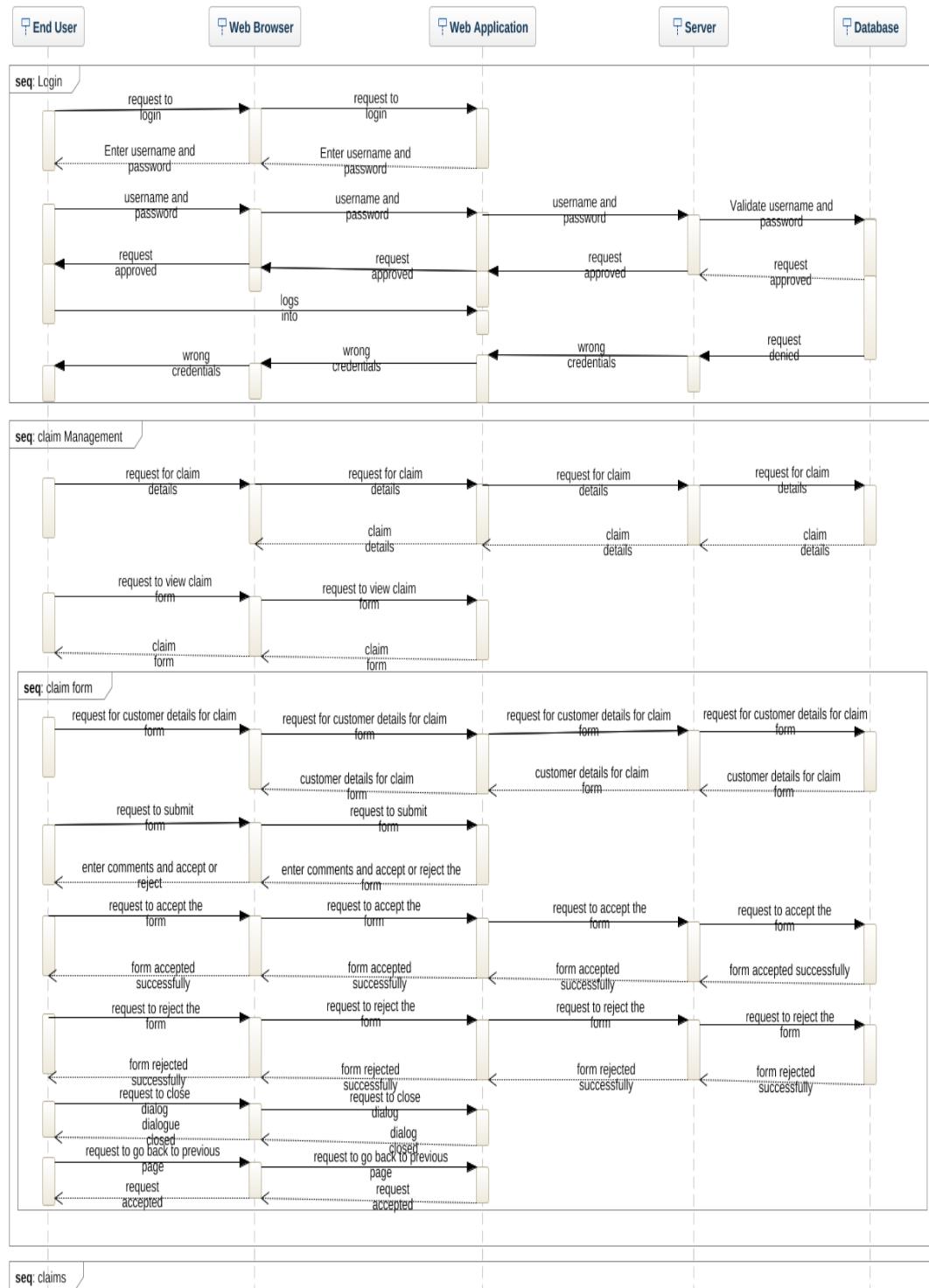


Fig. 3.7 Sequence Diagram for Claim Management Module -1

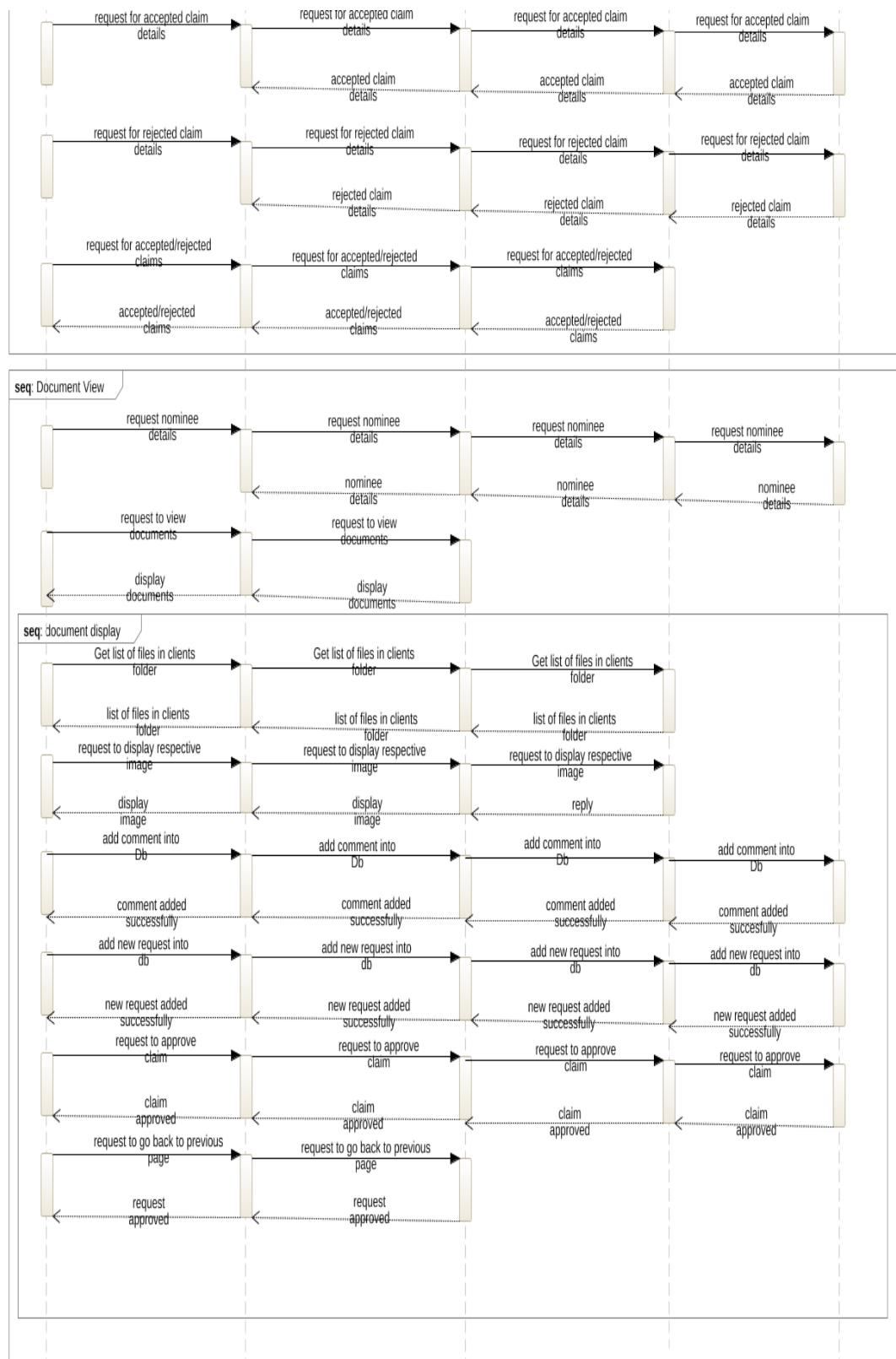


Fig. 3.8 Sequence Diagram for Claim Management Module -2

## SEQUENCE DIAGRAM FOR REQUEST MANAGEMENT

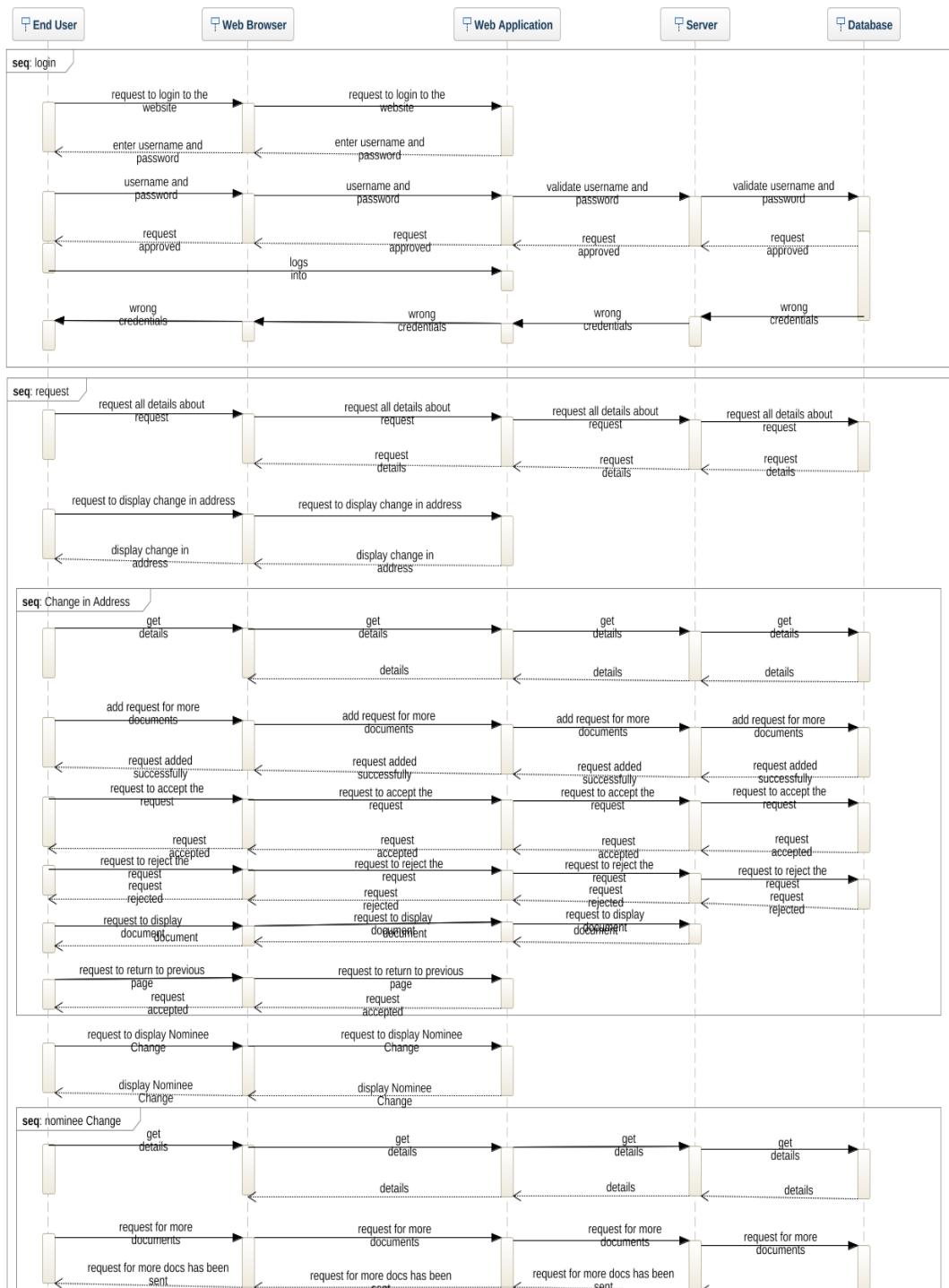


Fig. 3.9 Sequence Diagram for Request Management Module -1

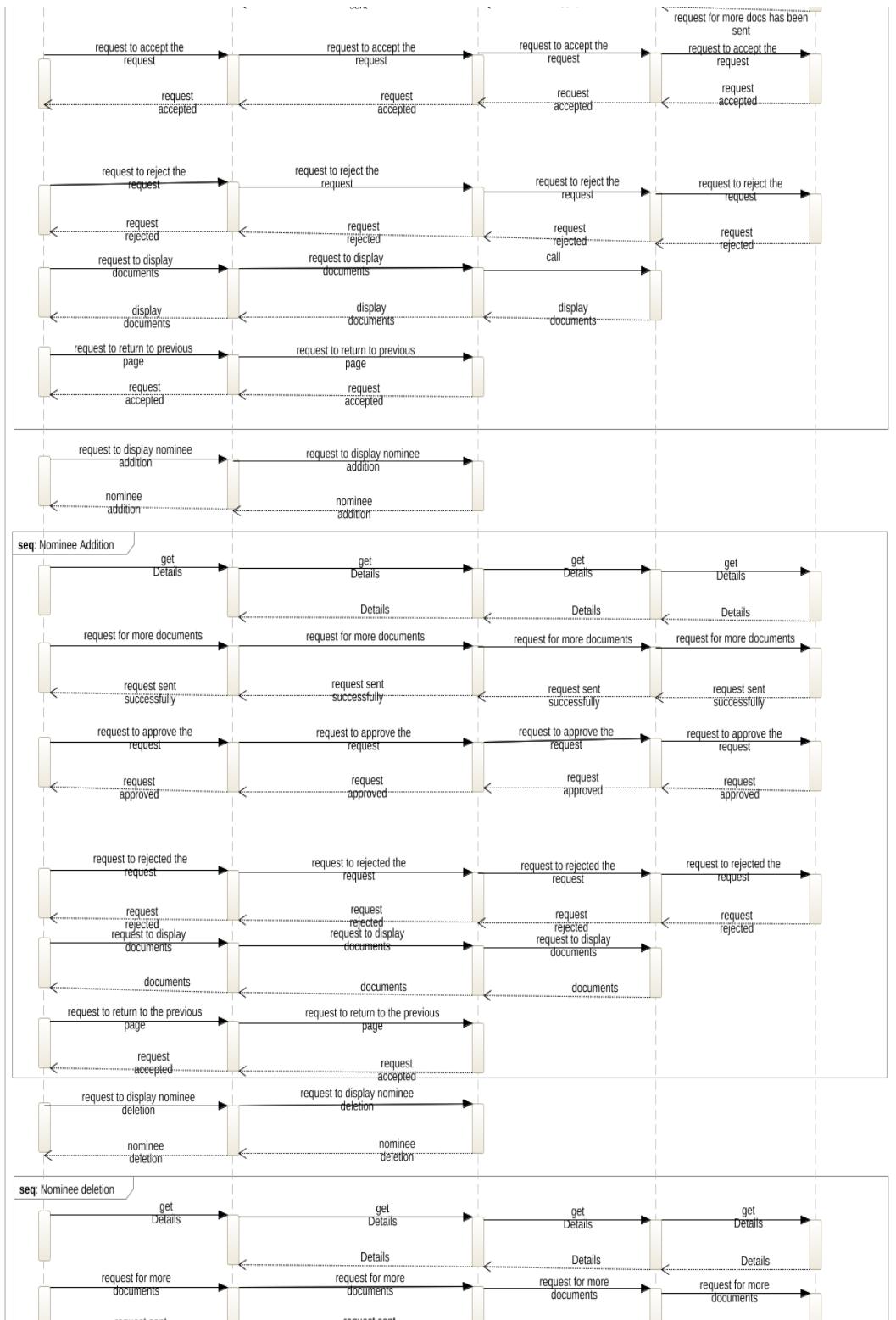


Fig. 3.10 Sequence Diagram for Request Management Module -2

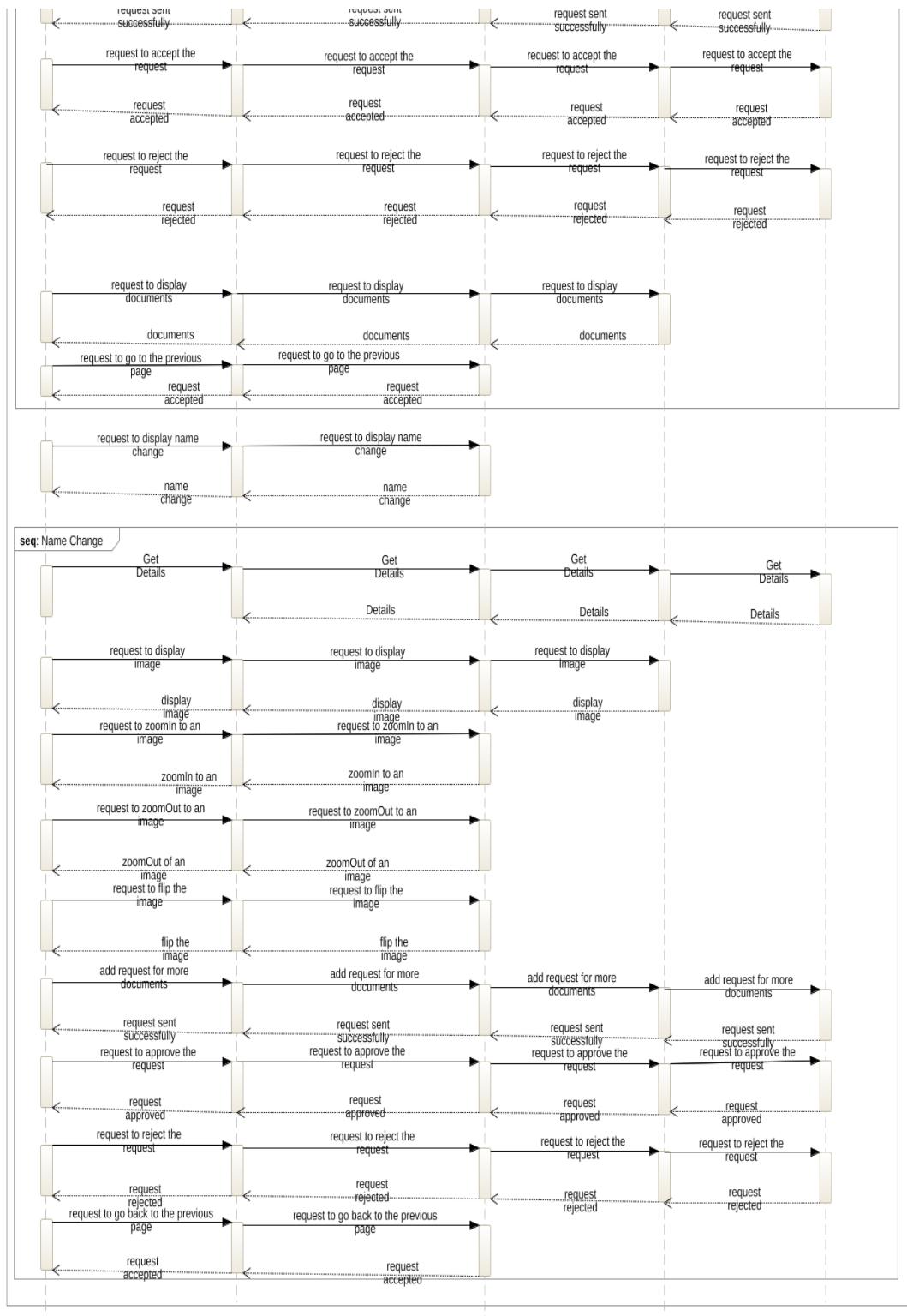


Fig. 3.11 Sequence Diagram for Request Management Module -3

## SEQUENCE DIAGRAM FOR NOMINEE MODULE

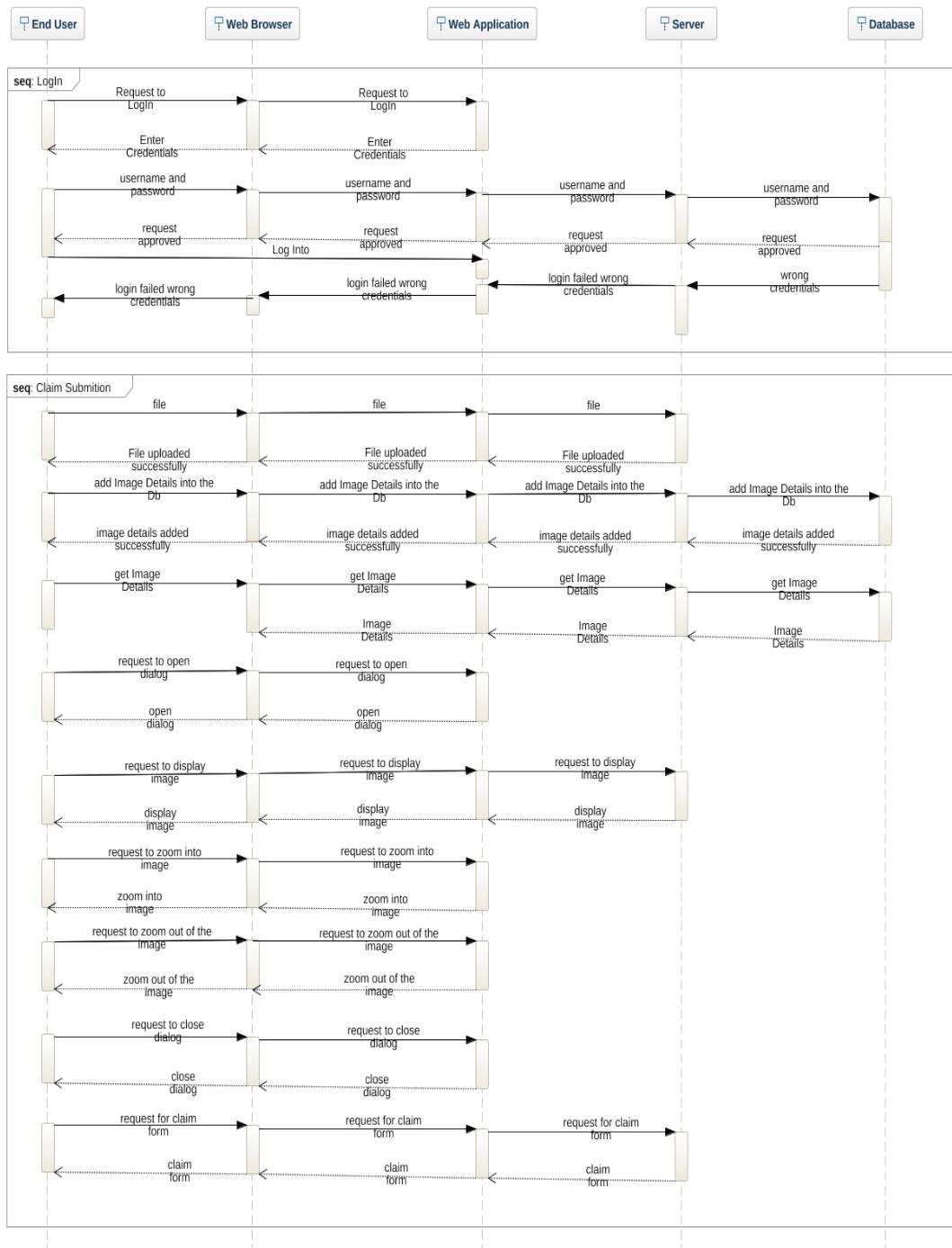


Fig. 3.12 Sequence Diagram for Nominee Module

### 3.2.3 ACTIVITY DIAGRAMS

#### Activity Diagram for Policy Servicing Module

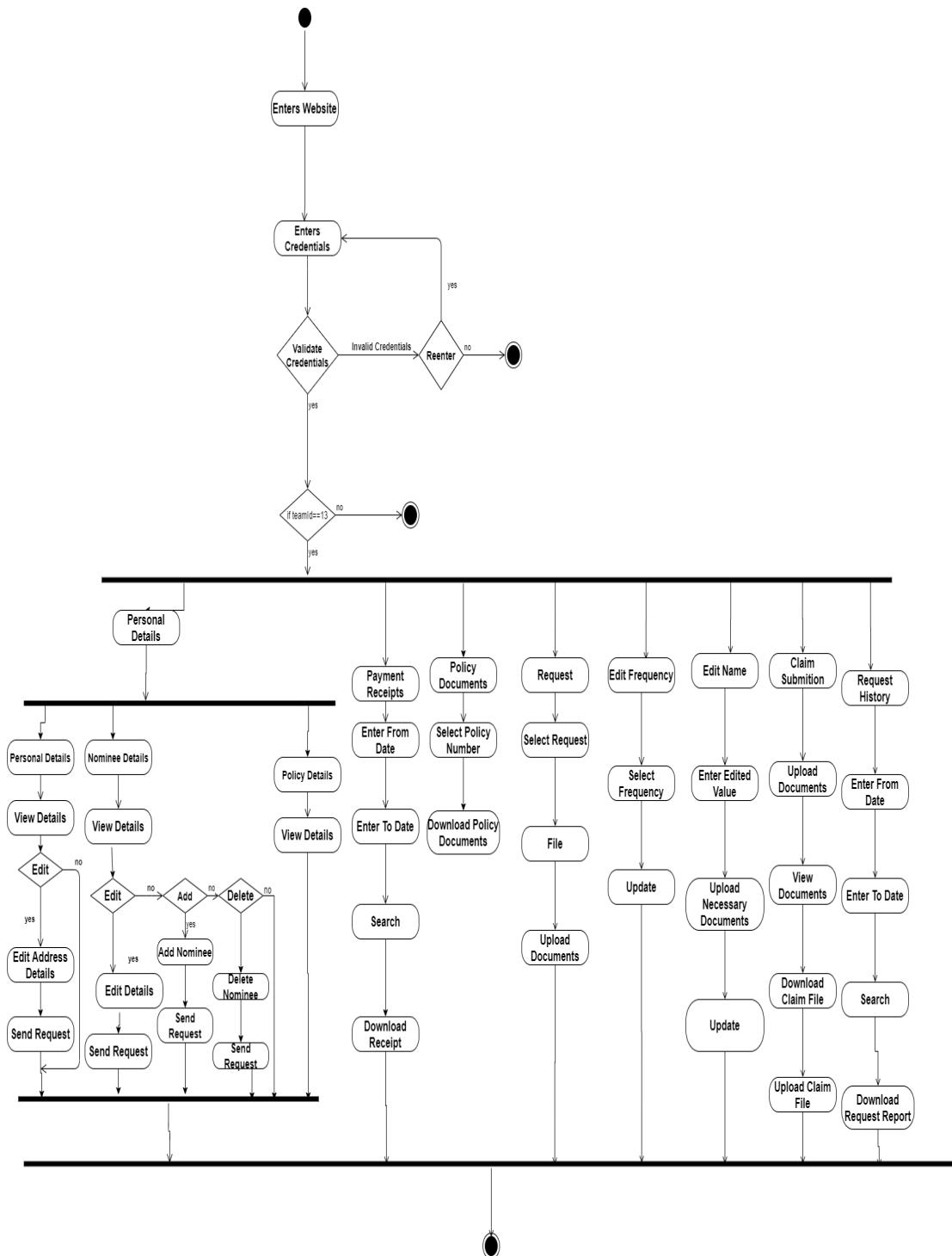


Fig. 3.13 Activity Diagram for Personal Details Module

## Activity Diagram for Customer Search Module

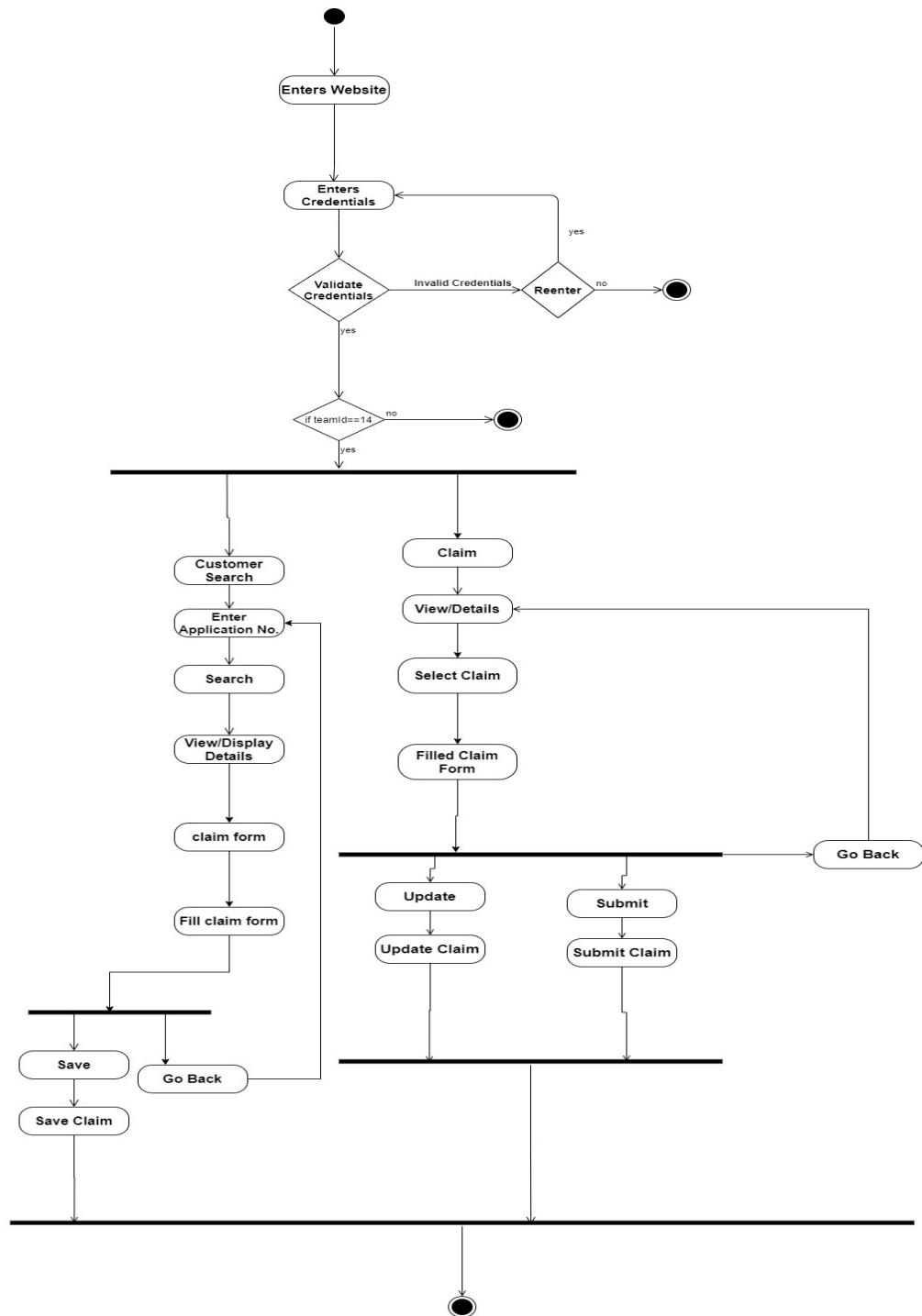


Fig. 3.14 Activity Diagram for Customer Search Module

## Activity Diagram For Claim Management Module

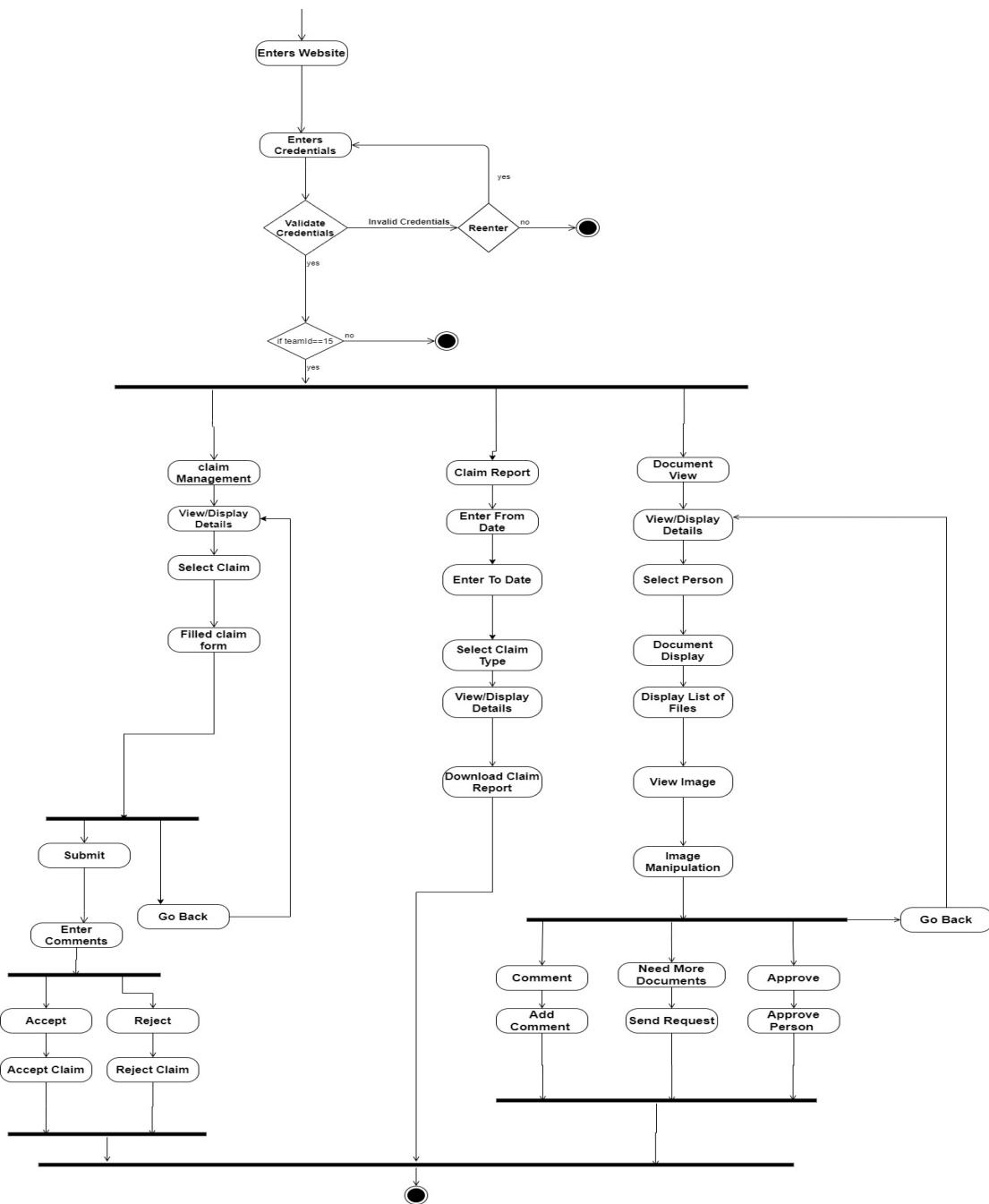


Fig. 3.15 Activity Diagram for Claim Management Module

## Activity Diagram For Request Management Module

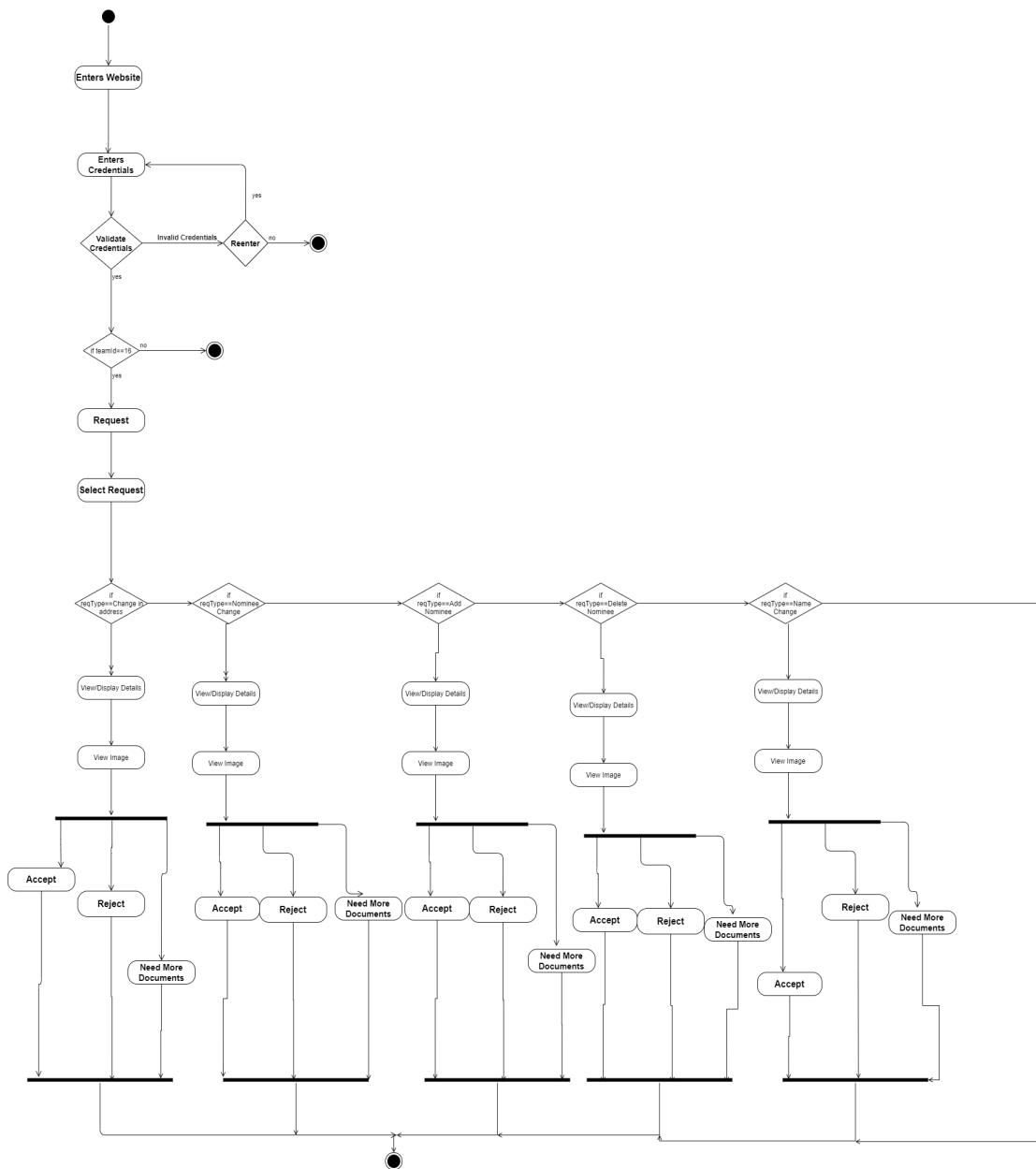


Fig. 3.16 Activity Diagram for Request Management Module

## Activity Diagram for Nominee Module

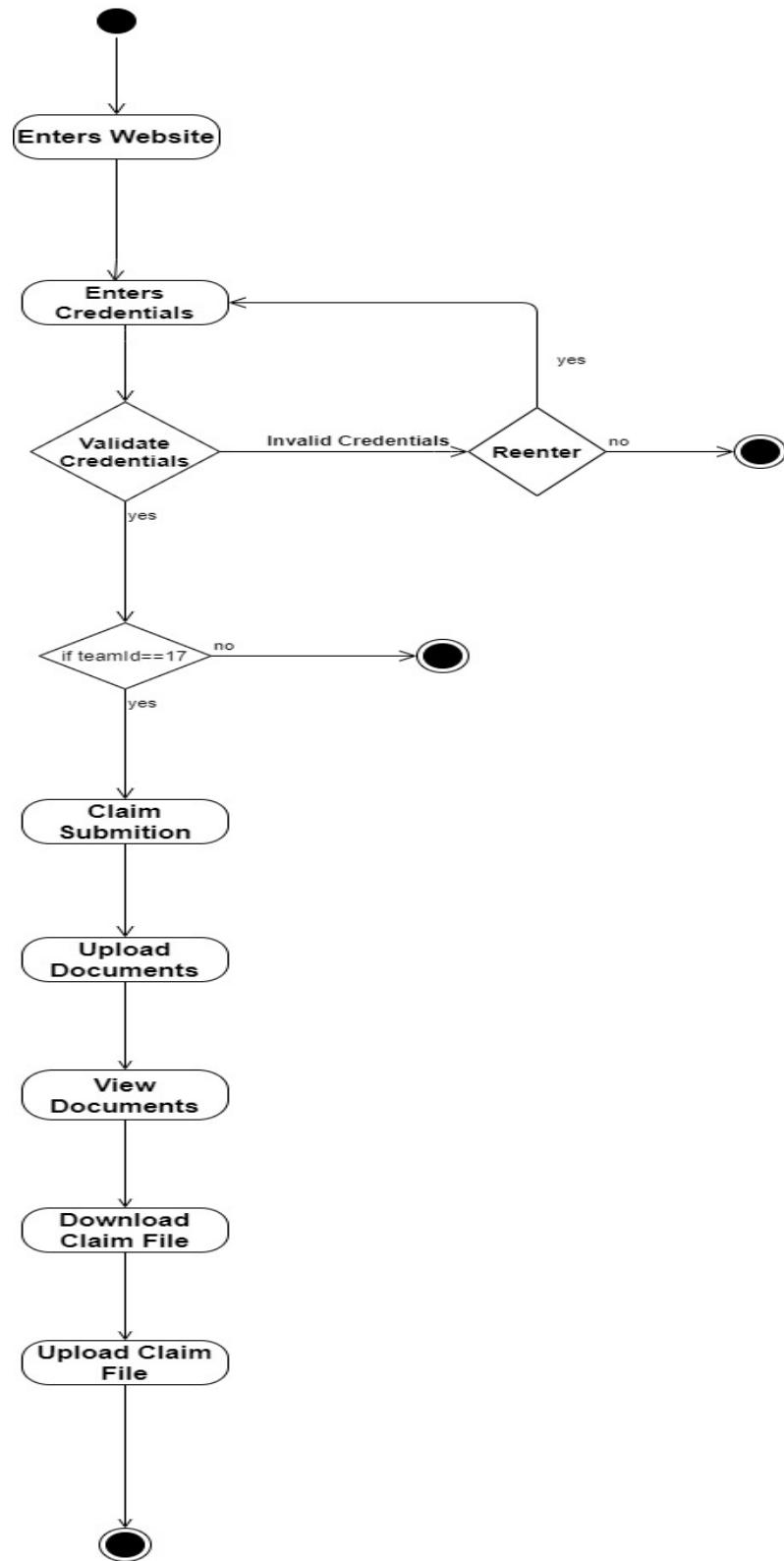


Fig. 3.17 Activity Diagram for Nominee Module

### 3.2.4 USE CASE DIAGRAMS

#### USE CASE DIAGRAM FOR POLICY SERVICING MODULE

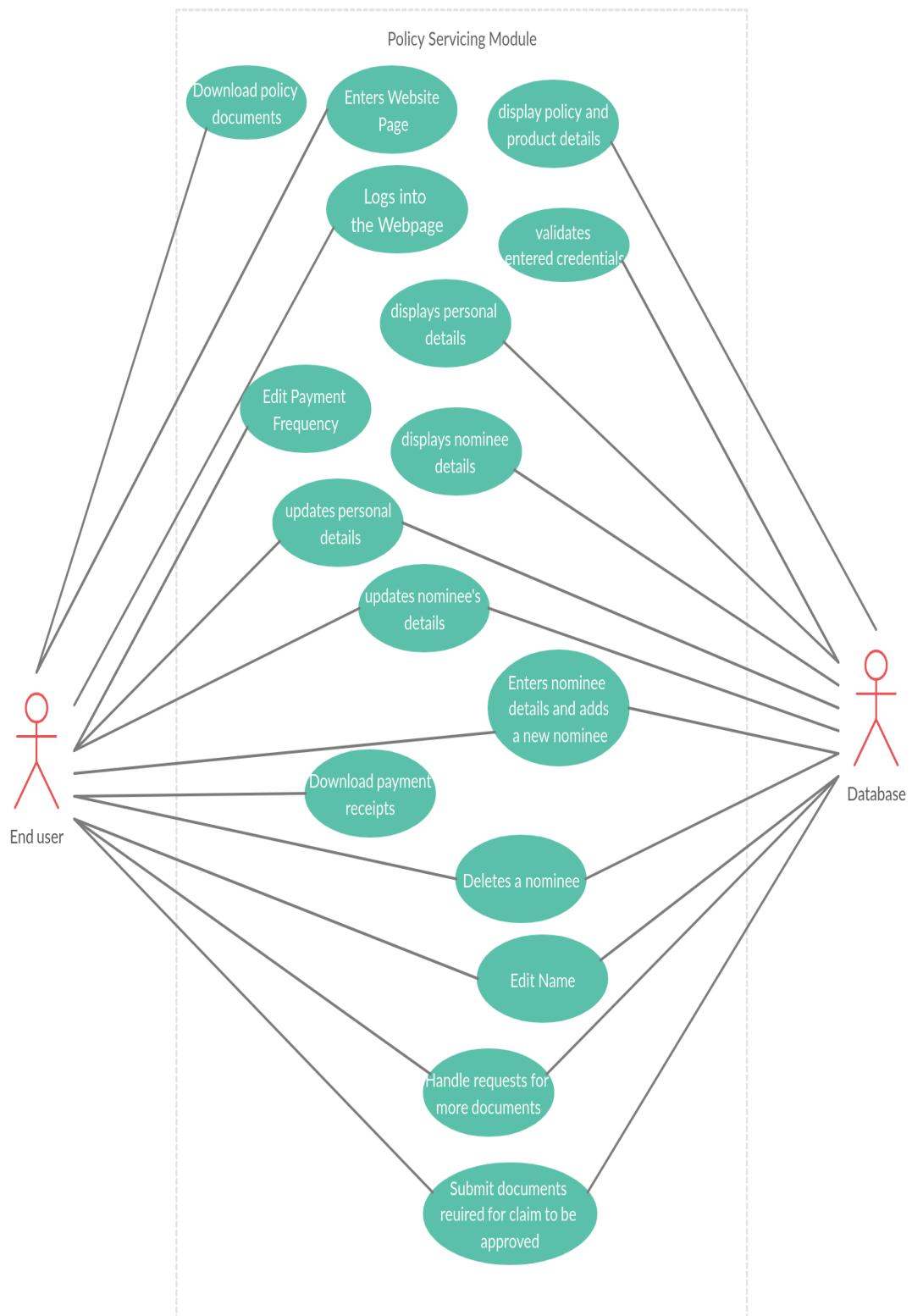


Fig. 3.18 Use Case Diagram for Personal Details Module

## USE CASE DIAGRAM FOR CUSTOMER SERVICE MODULE

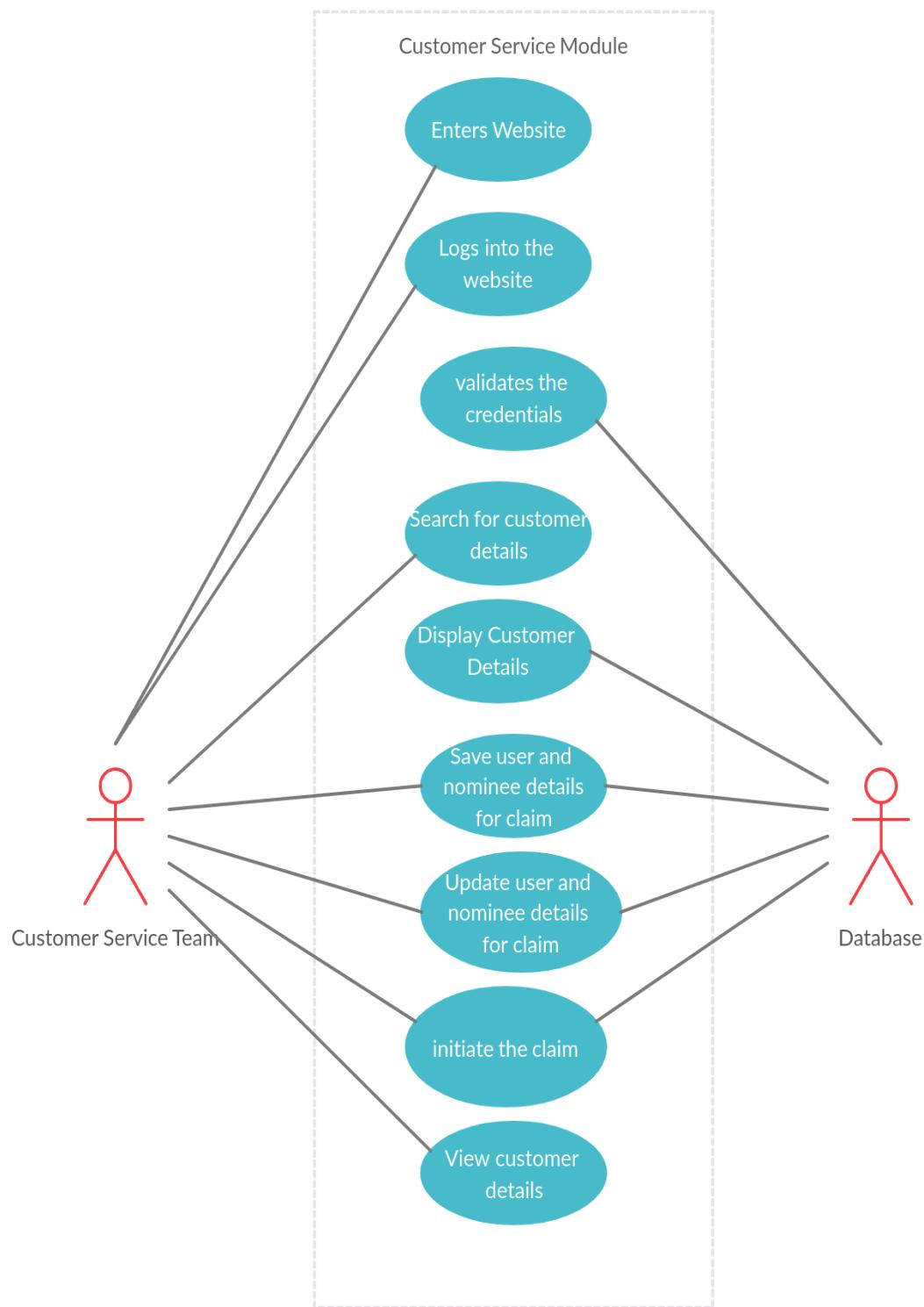


Fig. 3.19 Use Case Diagram for Customer Search Module

## USE CASE DIAGRAM FOR CLAIM MANAGEMENT MODULE

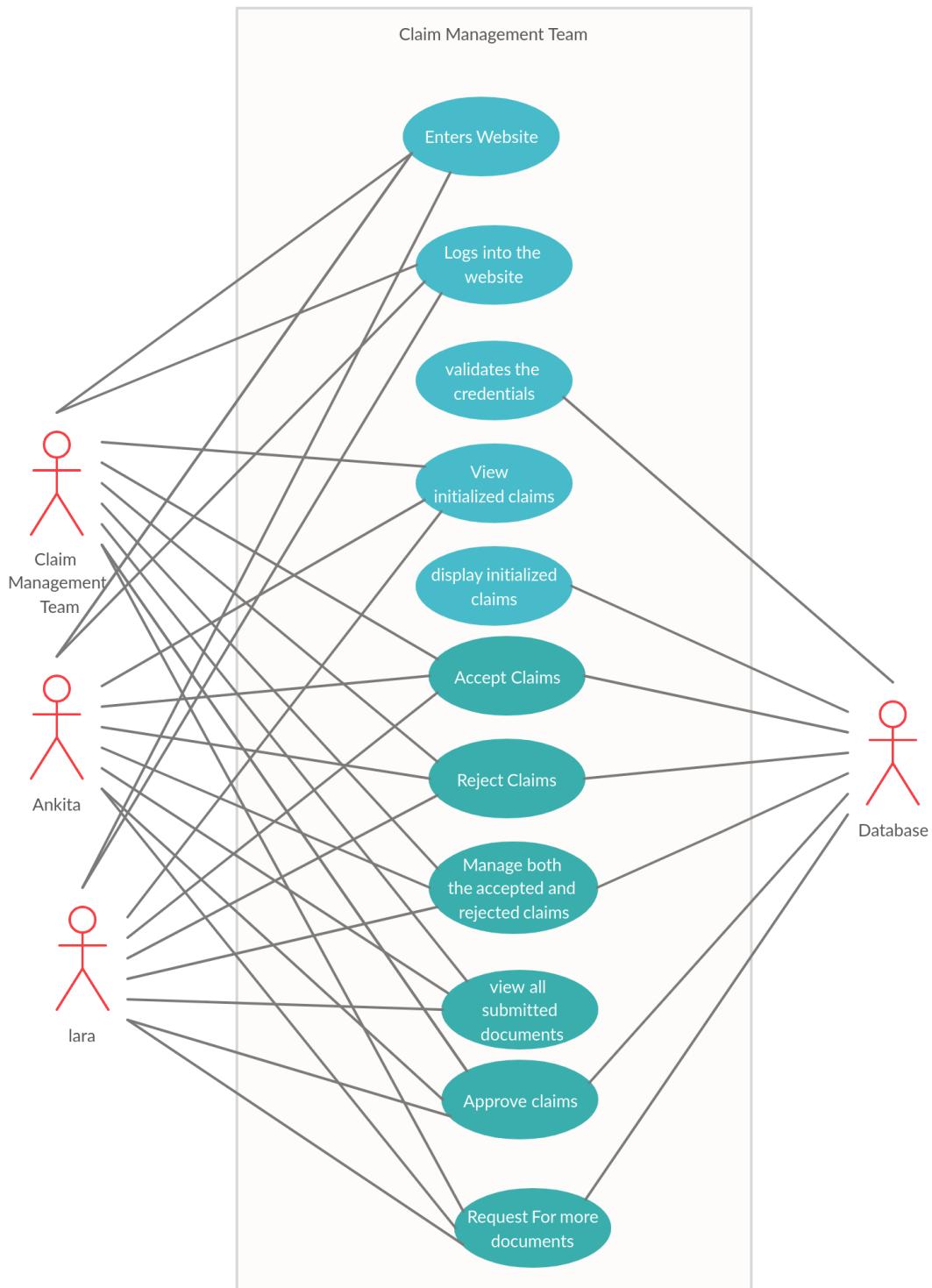


Fig. 3.20 Use Case Diagram for Claim Management Module

## USE CASE DIAGRAM FOR REQUEST MANAGEMENT MODULE

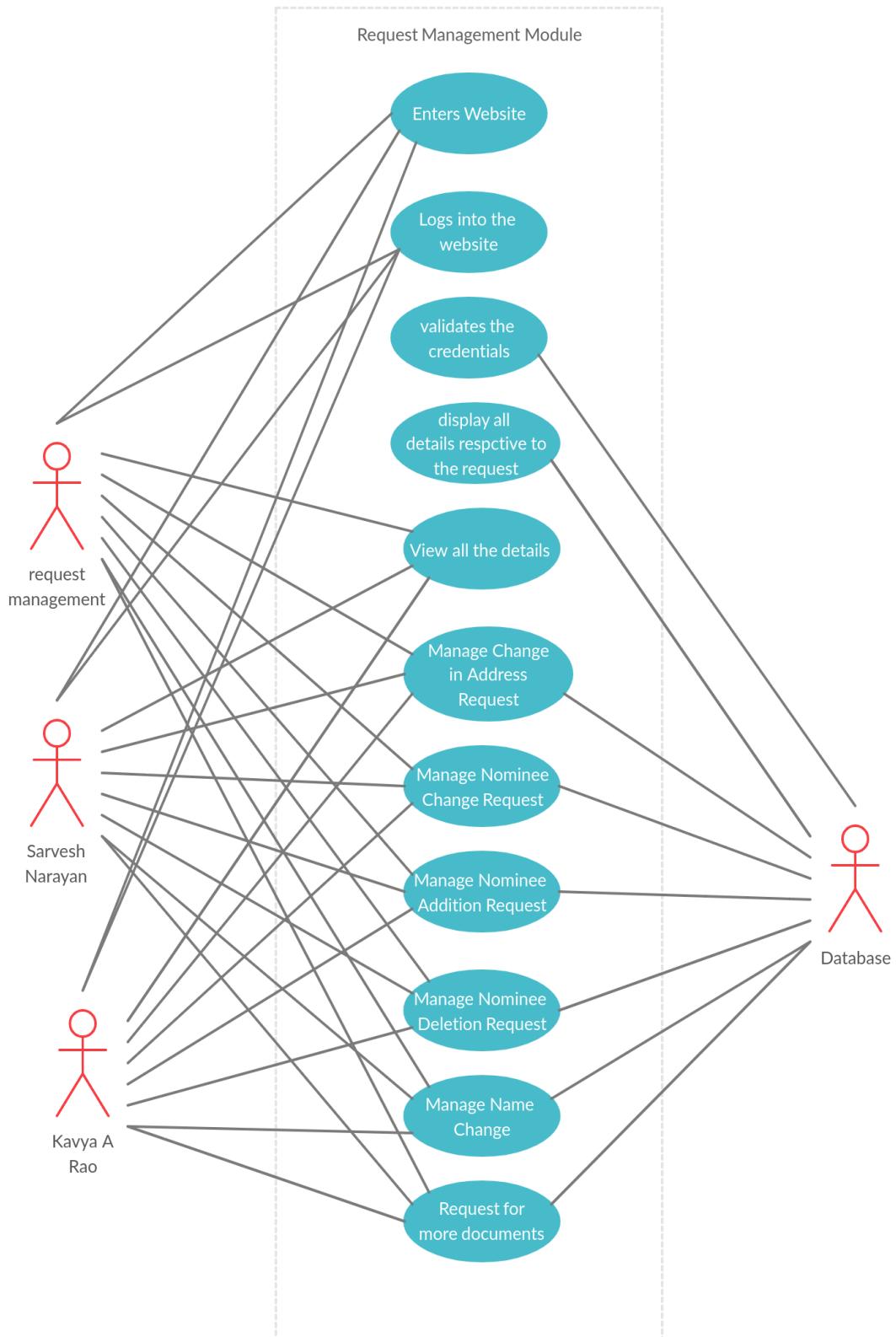


Fig. 3.21 Use Case Diagram for Request Management Module

## USE CASE DIAGRAM FOR NOMINEE

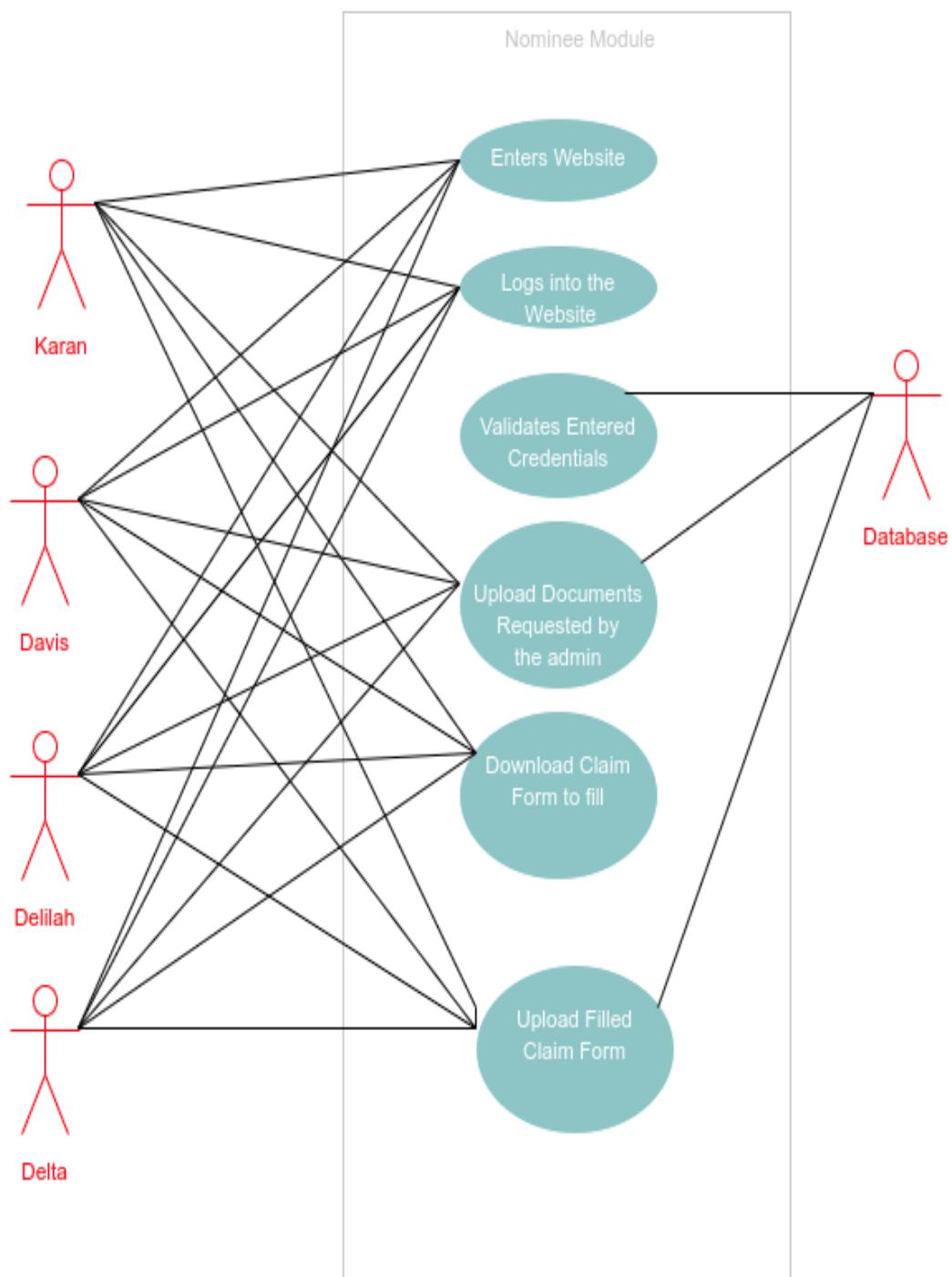


Fig. 3.22 Use Case Diagram for Nominee Module

### **3.3 REQUIREMENT SPECIFICATIONS**

#### **3.3.1 SOFTWARE REQUIREMENTS**

- Imagemagick-v6.9.7-4-Imagemagick software is used in the conversion of pdf files into jpg/png/jpeg files.
- PdfInfo-v0.62.0-pdfinfo software is used in reading pdf files so that they can be converted into images or jpg/png files.
- Sublime Text 3-This software is used as the text editor for the code, markup and prose.
- Operating System-Ubuntu 14 or above
- Robo 3T-is the software used to manage and viewing the records

#### **3.3.2 HARDWARE REQUIREMENTS**

- Hard Disk-capacity 1 TB or more.
- Processor- 8GB RAM Core I3

### **3.4 TECHNOLOGIES USED**

HTML5-HTML is used to design the User interface of the application as desired and make it as user friendly as possible.

CSS-CSS is used to style the page and is used for describing the presentation of a report written in markup language.

Javascript and Typescript-Javascript and typescript is used to perform all the functions in the front end as well as is used to connect the frontend with the backend which is the server.

Mongo Database-v3.6.16-Mongo Database is used to store data of the users such as personal details, login credentials, and many more as well as to retrieve data for particular functions. These functions are used throughout this application.

Express js-Express js is used in the backend to perform many functions important for the application.

Angular- v 6.1.10 and Angular CLI of v 7.3.9-Angular is the main framework where the application is coded and run on and is crucial for the project to run.

Node js-v10.16.2-Node js is the main language used for the running of the server or the backend. This is where the mongodb gets connected,file gets uploaded,data from the database is retrieved and inserted and pdf gets generated.

Java-v1.8.0\_201-Java is used in the batch processing programs used in the application. It is used in the updation of some requests automatically.

## **Chapter 4**

# **Implementation of System**

### **4.1 INTRODUCTION**

Life insurance is a contract between an insurance policy holder and an insurer or assurer, where the insurer promises to pay a designated beneficiary a sum of money in exchange for a premium, upon the death of an insured person. Policy Servicing is the applications that a customer procures when purchasing such life insurance. The services provided depends on the insurer or assurer and what they reckon is suitable for their product.

Policy Servicing is the process by which a customer is allowed to change or update a policy's coverage and details that might affect the coverage and premium of the policy. The various services that can be provided by an insurer ranges from "Transfer of ownership and rights of the policy to a natural person or corporation. Assignments will override nominations/trust." to "The time when an insurance company will pay a sum of maturity value to policyholders, if the policy has not been previously invoked to cover contingencies like death." and many more.

### **4.2 MODULE DESCRIPTION**

#### **4.2.1 USER MODULE**

##### **Module 1: Policy Services**

This module allows the customer to perform various different activities such as Changing their address, changing the details of the nominees, adding nominees, deleting nominees, viewing personal details, viewing nominee details, viewing product details, downloading Receipts, downloading policy documents, changing payment frequency and changing their names. The customer will also be able to upload more documents if requested. The customer will also be able see their requests and claim statuses as well as request history and be able to download the reports.

##### **Module 2: Initialization of Claim**

In this module a claim can be made by the customer with the help of a customer service agent. The details of the customer are also displayed to the agent who will be able to create and submit a claim for further reviewing.

## **Module 3: Claim Management**

In this module all the claims that have been submitted by the agent will be handled that is another agent will be able to either accept or reject the claim while providing comments as well as view a report of the rejected/accepted claims

## **Module 4: Request Management**

In this module all Requests such as Changing address, Changing name, Changing nominee details, Nominee deletion and Nominee addition can be handled and carried out as well as respective and required documents and details can be viewed.

## **Module 5: Nominee**

In this module all Nominees once their claim has been approved will be allowed to upload necessary documents as well as download the claim form, fill it and re-upload it.

### **4.2.2 PAGE DESCRIPTION**

**Login Page:** In this page the user enters their credentials, which are validated and if the credentials are valid then the user is navigated to their respective modules or else if credentials are Invalid then the message “wrong credentials” is displayed to the user.

**Dashboard:** This page is common to all the modules. It helps the user to navigate to the pages that they desire to visit.

### **Module 1:**

**Personal Details:** This page displays personal details, nominee details as well as product details to the user. This page also allows the user to send requests to modify the details of the nominee , modify address of the user as well as add a new nominee or delete a nominee.

**Payment Receipts:** This page displays all the payments made between the date given by the user as well as allows the user to download the receipt for the particular date that they want.

**Policy Documents:** This page displays all the policy Numbers of the user and allows the user to download the document depending on the policy Number that they select.

**Request:** This page displays all the requests made by the admin. The page also allows the user to perform the task required.

**File:** This page allows the user to upload documents as requested by the admin.

**Edit Frequency:** This page gives the user an opportunity to update the payment frequency if they want.

**Edit Name:** This page allows the user to generate a request to update their as well as upload and view files required for the request to be accepted.

**Claim Submission:** This page allows the user to submit all the documents that are requested after the claim has been accepted as well as view documents uploaded and download the claim form which has to be filled and reuploaded.

**Request History:** This page allows the user to view the request history between specified dates as well as download the report

## **Module 2:**

**Customer Search:** This page allows the user to look for a particular user with the help of their Application Number which then displays their personal details as well as their address details. After viewing their details the user can invoke the claim form which needs to be filled to initialize a claim.

**Claim:** This page allows the user to view all the claims initialized and make any changes if required by invoking the form for that particular customer.

**Claim Form:** This page is the claim form that is to be filled and either saved, updated or submitted.

## **Module 3:**

**Claim Management:** This page allows the user to view all the claims that have been submitted for further review.

**Claim Form:** This page is the filled claim form that is to be reviewed and a decision is to be made whether to accept or reject the claim. If accepted a new user for the nominee is created and an email is sent to them with their login credentials.

**Claim Report:** This page is a report of all the claims that have been accepted or rejected and also allows the user to download the reports.

**Document View:** This page displays all the users who have to upload documents as their claim has been accepted. This page allows you to navigate to the page that displays each user's documents.

**Document Display:** This page displays a list of all the documents that the user has uploaded and allows the user to view each image and manipulate the image in whichever form they like. The user is further allowed to ask for more documents, add comments or approve of the claim and download the document required.

## **Module 4:**

**Request:** This page displays all the requests made and assigned to that particular user. And allows the user to navigate to the particular page that holds the details for the particular request.

**Request display 1:** This page displays all the details required for the request “Change in Address” based on which the user can view documents required, Ask for more documents, reject the request or accept the request.

**Request Display 2:** This page displays all the details required for the request “Nominee Change” based on which the user can view documents required, Ask for more documents, reject the request or accept the request.

**Request Display 3:** This page displays all the details required for the request “Addition of Nominee” based on which the user can view documents required, Ask for more documents, reject the request or accept the request.

**Request Display 4:** This page displays all the details required for the request “Deletion of Nominee” based on which the user can view documents required, Ask for more documents, reject the request or accept the request.

**Request Display 5:** This page displays all the details required for the request “Change in Address” based on which the user can view documents required, Manipulate the Image as they wish (ZoomIn, ZoomOut, rotate, dragIn, dragOut), Ask for more documents, reject the request or accept the request.

## Module 5:

**Claim Submission:** This page allows the user to submit all the documents that are requested after the claim has been accepted as well as view documents uploaded and download the claim form which has to be filled and reuploaded.

### 4.3 SAMPLE SOURCE CODE

#### 4.3.1 claim.component.html

```
<div class="container-fluid h-100" ><!-- start of container -->
<form class="max-height col-12 p-0" role="form" [formGroup]="Details1"
(submit)=>addDetails()' method="POST" #myForm="ngForm" autocomplete="off"
novalidate>
<div class="row m-0 h-90"><!-- start of row -->
<div class="col-sm-2 col-md-2 col-lg-2 col-xl-2 col-2 col-lg-offset-2 center-block
h-100 ">
<br><br>
<mat-form-field class="example-full-width">
<input matInput [matDatepicker]="picker" placeholder="From Date"
formControlName="fromdate" name="fromdate" disabled readonly="readonly"
style="width: 5em">
<mat-datepicker-toggle matSuffix [for]="picker"></mat-datepicker-toggle>
<mat-datepicker #picker disabled="false" touchUi></mat-datepicker>
</mat-form-field>
<br><br><br><br><br><br>
```

```

<mat-form-field class="example-full-width">
  <input matInput [matDatepicker]="picker2" placeholder="To Date"
    formControlName="todate" name="todate" disabled readonly="readonly"
    style="width: 5em">
    <mat-datepicker-toggle matSuffix [for]="picker2"></mat-datepicker-toggle>

  <mat-datepicker #picker2 touchUi color="secondary"
    disabled="false"></mat-datepicker>
</mat-form-field>
<br><br><br><br><br><br>
<div class="input-group form-group input-group-sm mb-3 h-100">
  <div class="input-group-prepend h-100" >
    <span class="input-group-text" id="inputGroup-sizing-sm"
      [ngStyle]="{{'background':'white','height':'3.5em','width':'4.5em'}}">
      <mat-card
        [ngStyle]="{{'height':'2.5em','width':'3em','padding':'0.5em'}}">Type</mat-card>
    </span>
  </div>
  <mat-form-field class="b">
    <div class="h-100">
      <mat-select #matSelect [panelClass]="a" (selectionChange)="display()"
        formControlName="claim">
        <mat-option value="Accepted Claim">Accepted Claim</mat-option>
        <mat-option value="Rejected Claim">Rejected Claim</mat-option>
      </mat-select>
    </div>
  </mat-form-field>
</div>
</div></div>
<div class="col-sm-1 col-md-1 col-lg-1 col-xl-1 col-1 col-lg-offset-1 h-100 ">
  <mat-divider [vertical]="true"></mat-divider>
</div>
<div class="col-sm-9 col-md-9 col-lg-9 col-xl-9 col-9 h-100" *ngIf="A">
  <div class="mat-elevation-z8 h-100 table-striped ">
    <div id="contentToConvert" class="scroll h-100">
      <h3>{{Title}}</h3>
      <table mat-table [dataSource]="dataSource" class="example-container" matSort>
        <!-- ID Column -->

```

```

<ng-container matColumnDef="id" >
  <th mat-header-cell *matHeaderCellDef mat-sort-header> No. </th>
  <td mat-cell *matCellDef="let row; let i = index;"> {{i+1}} </td>
</ng-container>

<ng-container matColumnDef="AcceptedDate" >
  <th mat-header-cell *matHeaderCellDef mat-sort-header> Accepted Date </th>
  <td mat-cell *matCellDef="let row"> {{row.AcceptedDate}} </td>
</ng-container>

<!-- Progress Column -->
<ng-container matColumnDef="appNo">
  <th mat-header-cell *matHeaderCellDef mat-sort-header> App No </th>
  <td mat-cell *matCellDef="let row"> {{row.appNo}} </td>
</ng-container>

<!-- Name Column -->
<ng-container matColumnDef="PolicyNo">
  <th mat-header-cell *matHeaderCellDef mat-sort-header> Policy No </th>
  <td mat-cell *matCellDef="let row"> {{row.PolicyNo}} </td>
</ng-container>

<ng-container matColumnDef="name">
  <th mat-header-cell *matHeaderCellDef mat-sort-header> Person's Name</th>

  <td mat-cell *matCellDef="let row" > {{row.name}} </td>
</ng-container>
<!-- Color Column -->
<ng-container matColumnDef="ApprovedAmt">
  <th mat-header-cell *matHeaderCellDef mat-sort-header> Approved Amt.</th>
  <td mat-cell *matCellDef="let row" > {{row.AcceptedSum}} </td>
</ng-container>

  <tr mat-header-row *matHeaderRowDef="displayedColumns; sticky: true"></tr>
  <tr mat-row *matRowDef="let row; columns: displayedColumns; let i = index; " class="sc">
    </tr>
  </table>
</div>

```

```

<mat-paginator [pageSizeOptions]=[5, 10, 25, 100]></mat-paginator>
</div>
</div>
</div><!-- end of row -->
</form>
<div class="row m-0 h-10" *ngIf="A" >
  <div class="col-sm-12 col-md-12 col-lg-12 col-xl-12 col-12 p-0 h-100">
    <button type="button" id="float" (click)="convertHTMLToPDF()" class="float-right" style="position: absolute; right: 2em; bottom: 3em;"><i class="fa fa-download"></i></button>
  </div>
</div><!-- end of container -->

```

### 4.3.2 Header.component.ts

```

@Component({
  selector: 'app-header',
  templateUrl: './header.component.html',
  styleUrls: ['./header.component.scss']
})
export class HeaderComponent implements OnInit {
  public newTaskCount:string="0";
  public workInProgress:string="0";
  public rescanCase:string="0";
  public policyService:string="0";
  public personalDetails:String="0";
  public CustomerService:String="0";
  public pol:boolean=false;
  public per:boolean=false;
  public cus:boolean=false;
  userName="Welcome";
  teamName:string="";
  userId: string;
  teamId: string;
  admin:boolean=true;
  user:boolean=false;
  pushRightClass: string = 'push-left';
  constructor(private translate: TranslateService, public router: Router, private dataService:DataService) {
    this.translate.addLangs(['en', 'fr', 'ur', 'es', 'it', 'fa', 'de', 'zh-CHS','mr']);
  }
}

```

```

this.translate.setDefaultLang('en');
    const browserLang = this.translate.getBrowserLang();
        this.translate.use(browserLang.match(/en|fr|ur|es|it|fa|de|zh-CHS|mr/) ? browserLang : 'en');
    this.router.events.subscribe(val => {
        });
}
ngOnInit() {
var txt = JSON.parse(localStorage.getItem('usersinfo'));
console.log("username--->" + txt[0]["username"]);
for (let i = 0; i < localStorage.length; i++) {
    let key = localStorage.key(i);
    let value = localStorage.getItem(key);
    console.log(key, value);
}
this.userName = this.userName + " " + txt[0]["name"] + ",";
this.userId = txt[0]["_id"];
this.teamId = (txt[0]["teamId"]);
this.teamName = txt[1]["teamName"];
console.log("teamName--->" + this.teamName);
if (this.teamId === "11") {
}
else if ((this.teamId === "13") || (this.teamId === "17")) {
    this.per = true;
    this.user = true;
    this.admin = false;
}
else if (this.teamId === "14") {
    this.cus = true;
}
this.dataService.setTaskCount$.subscribe((data) => {
    this.newTaskCount = data["pendingCounter"];
    this.workInProgress = data["inprogressCounter"];
    this.rescanCase = data["rescanCaseCounter"];
    this.policyService = data["policyServiceCounter"];
    this.personalDetails = data["detailsCounter"];
}

```

```

        this.CustomerService=data["CustomerServiceCounter"]
        console.log("responseObj----->"+data["pendingCounter"]);
    }
)
}
isToggled(): boolean {
    const dom: Element = document.querySelector('body');
    return dom.classList.contains(this.pushRightClass);
}
toggleSidebar() {
    const dom: any = document.querySelector('body');

    dom.classList.toggle(this.pushRightClass);
    if(dom.classList.contains(this.pushRightClass)){
        var el = document.getElementById("sideBarIsActive");
        console.log("el----->"+el)
        if(el!=null){
            if (el.classList.contains('r-0px')) {
                // document.getElementById("sideBarIsActive").classList.remove('l-0px');
                document.getElementById("sideBarIsActive").classList.remove('r-0px');
            }
            // document.getElementById("sideBarIsActive").classList.add('l-235px');
            document.getElementById("sideBarIsActive").classList.add('r-235px');
            document.getElementById("mask").style.display = 'block';
            // document.getElementsByTagName("app-sidebar")[0].setAttribute("style",
            "width:18.5% !important");
            // document.getElementsByTagName("section")[0].setAttribute("style",
            "width:81.5% !important");
        }
    }else{
        var el = document.getElementById("sideBarIsActive");
        if(el!=null){
            if (el.classList.contains("r-235px")) {
                document.getElementById("sideBarIsActive").classList.remove('r-235px');
                document.getElementById("mask").style.display = 'none';
            }
            document.getElementById("sideBarIsActive").classList.add('r-0px');
        }
    }
}

```

```

        }
    }
    rltAndLtr() {
        const dom: any = document.querySelector('body');
        dom.classList.toggle('rtl');
    }
    onLoggedout() {
        localStorage.removeItem('isLoggedIn');
    }
    changeLang(language: string) {
        this.translate.use(language);
    }
    hideSidebar(){
        let el = document.getElementById("sideBarIsActive");
        if(el!=null){
            if (el.classList.contains("r-235px")){
                document.getElementById("sideBarIsActive").classList.remove('r-235px');
                document.getElementById("sideBarIsActive").classList.add('r-0px');
                document.getElementById("mask").style.display = 'none';

```

```

        }
    }
}
```

//class closing

#### **4.3.3 Html\_pdf.js**

```

router.post('/createRequesthistorypdf', async (req,res) =>
{
    console.log("directory path "+req);
    dirpath=path.join(__dirname,'../../uploads/pdf_output/request/'+req.body inputValue['name']+ '/');
    console.log("directory path "+dirpath);
    function mkdirpath(dirPath)
    {
        console.log("in mkdir function dir path"+dirPath);
        if(!fs.existsSync(dirPath))
        {
            console.log("in if.. "+dirPath)

```

```

try
{
    console.log("in try.. "+dirPath)
    fs.mkdirSync(dirPath);
}
catch(e)
{
    mkdirpath(path.dirname(dirPath));
    mkdirpath(dirPath);
    console.log("in catch.. "+dirPath)
}
}

mkdirpath(dirpath);
(async function()
{
    var imgArray = [];
    try
    {
        var fn="request";
        var filename=fn.concat(req.body inputValue['count']);
        filename=filename.concat(".pdf");
        console.log(filename);
        const browser = await puppeteer.launch();
        const page = await browser.newPage();
        await page.setContent(`<html>`+req.body inputValue['htmlContent']+`</html>`);
        await page.emulateMedia('screen');

filepath=path.join(__dirname,'../../uploads/pdf_output/request/'+req.body inputValue['name']) + '/' + filename;
        console.log("file name with path "+filepath);
        await page.pdf({
            printBackground: true,
            path: filepath
        });
        console.log("done");
        await browser.close();
    }
})

```

#### 4.3.4 Freq.bat

```

@ECHO OFF
java -jar "C:\Users\Admin\Documents\work\editfreq.jar"

```

#### 4.3.5 ConnectToDb.java

```
package trial;

import com.mongodb.client.FindIterable;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.model.Filters;
import com.mongodb.client.model.Updates;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import org.bson.Document;
import org.json.JSONException;
import org.json.JSONObject;
import com.mongodb.BasicDBObject;
import com.mongodb.MongoClient;
import com.mongodb.MongoCredential;
public class ConnecttoDB implements Runnable {

    public static void main( String args[] ) throws JSONException { }

    @Override
    public void run() {

        Logger mongoLogger = Logger.getLogger( "org.mongodb.driver" );
        mongoLogger.setLevel(Level.SEVERE);
        MongoClient mongo = new MongoClient( "localhost" , 27017 );
        MongoCredential credential;
        credential = MongoCredential.createCredential(" ", "PolicyServicing",
            "".toCharArray());
        System.out.println("Connected to the database successfully");
        MongoDB database = mongo.getDatabase("PolicyServicing");
        MongoCollection<Document> collection = database.getCollection("product");
        System.out.println("Collection sampleCollection selected successfully");
        BasicDBObject andQuery = new BasicDBObject();
        List<BasicDBObject> obj = new ArrayList<BasicDBObject>();
        obj.add(new BasicDBObject("isUpdate", " "));
        obj.add(new BasicDBObject("Edited Value",new BasicDBObject("$ne", " ")));
        andQuery.put("$and", obj);
        Object b = new ArrayList();
        System.out.println(andQuery.toString());
        FindIterable<Document> iterDoc = collection.find(andQuery);
        int i = 0;
```

```

Iterator it = iterDoc.iterator();
while (it.hasNext()) {
    JSONObject x;
    try {
        x = new JSONObject(((Document) it.next()).toJson());
        System.out.println("x----->"+x.get("Term"));
        String Edit=x.getString("Edited Value");
        String pay=x.getString("PaymentFrequency");
        System.out.println(pay);
        System.out.println("Edited Value"+Edit);
        collection.updateOne(Filters.eq("isUpdate",""),
                             "),
Updates.set("PaymentFrequency", Edit));
        collection.updateOne(Filters.eq("isUpdate"," "), Updates.set("Edited Value",
pay));
        collection.updateOne(Filters.eq("isUpdate"," "), Updates.set("isUpdate",
"Done"));
        System.out.println("Document updated successfully...");
    } catch (JSONException e) {
        e.printStackTrace();
    }
    i++;
}

}
System.out.println("Length of doc---->" + i);
}
}

var filePath=path.resolve(__dirname,'../../uploads/pdf_output/request/'+req.body.inputValue['name']+ '/' + filename);
response.data=filename;
res.json(response);

} catch (e){
    console.log("error ..", e);
    res.send(e.message);
}
})();

```

## **4.4 TEST CASES**

- 1 Validate User Entered Credentials with Database (If wrong appropriate message should be displayed)
- 2 Ensure that the user is navigated to the correct page depending on the team id.
- 3 Ensure that the dashboard and all data is displayed correctly
- 4 Check that in the dashboard all buttons work and are related with the specific module.

### **Module 1**

- 5 Make sure that all required details are being displayed as well as that all data entered is inserted into the database
- 6 Ensure that the user is able to download payment receipts of desired date as well as download policy documents as they wish.
- 7 Ensure that the user is able to upload necessary files
- 8 Check that all tables holds necessary data
- 9 Make sure that all requests are submitted successfully
- 10 Check if navigation is taking place smoothly

### **Module 2**

- 11 Verify customers depending on application number
- 12 Ensure that navigation takes place smoothly
- 13 Validate all the data entered by users and that the data gets inserted into the database
- 14 Verify that buttons are working properly
- 15 Ensure that required data are displayed in the tables

### Module 3

- 16 Ensure that the user is able to download documents as required
- 17 Make sure that the user is able to view images and approve it if required
- 18 Test to make sure that the user is able to submit the form
- 19 Ensure all required data is displayed in the tables
- 20 Ensure that comments are inserted into the database
- 21 Make sure that image manipulation works

### Module 4

- 22 Ensure that the user is able to view all the details
- 23 Ensure that the user is able to navigate to view whichever request they desire
- 24 Make sure that the user is able to accept,reject the request or request for more documents
- 25 Make sure that the user is able to view image as well as manipulate it
- 26 Make sure that the request displayed is the requests that has been assigned to that particular person

### Module 5

- 27 Test to make sure that the user is able to upload files as required
- 28 Ensure that all data are displayed correctly
- 29 Ensure that the user is able to view the image
- 30 Make sure that the user is able to download the claim form to fill

## Chapter 5

# Result and Discussion

## 5.1 SNAPSHTOS

### LOGIN PAGE

Login page is the page which connects all the modules the user has to enter the username and password which is then validated if it is correct then the user depending on the users team Id gets navigated to the particular module. If it is wrong then the message “Wrong Credentials ” is displayed.

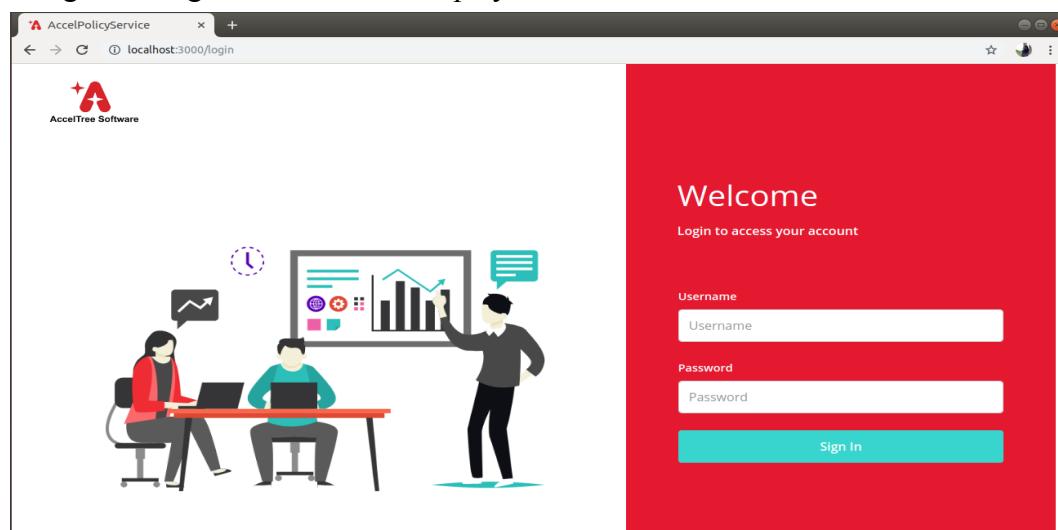


Fig. 5.1 Login Page

### Policy Service Module

#### Dashboard

The Dashboard page is common to all the module the only difference is the buttons which navigates the user to the different pages

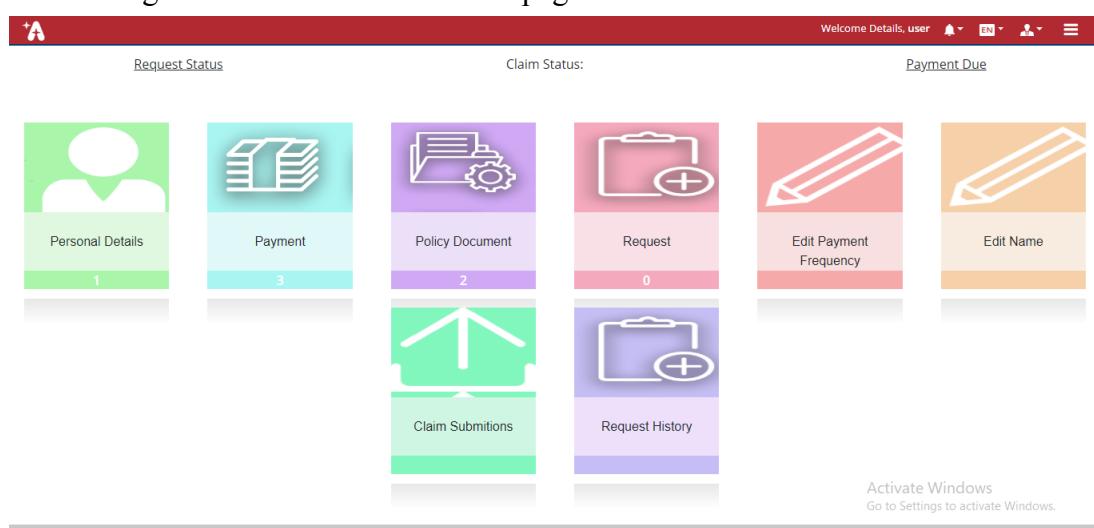


Fig. 5.2 Dashboard for Personal Details

## Personal Details

This page allows the user to view their personal details as well as submit a request to change their address

The screenshot shows a web browser window with the URL `localhost:3000/dashboard`. The title bar says "AccelPolicyService". The main content area has a red header bar with tabs: "Personal Details", "Nominee Details", and "Policy Details". Below the header, it says "Personal Details of the customer". The form fields include:

name Kavya R	Date Of Birth 12/01/2000	Gender <input type="radio"/> Male <input checked="" type="radio"/> Female <input type="radio"/> Transgender	Marital status single
Mobile No 9876543210	Email kavya@gamil.com	Occupation coder	Smoking status not a smoker
Address line 1 chaitanya army Quarters	Address line 2 khandivaaj,opposite railway station	city pune	state maharashtra
country india	PinCode 555000	[Edit icon]	

Fig. 5.3 Personal Details

## Nominee Details

This page allows the user to view their nominee details as well as submit a request to change their nominee details, add nominee as well as delete a nominee.

The screenshot shows a web browser window with the URL `localhost:3000/dashboard`. The title bar says "AccelPolicyService". The main content area has a red header bar with tabs: "Personal Details", "Nominee Details", and "Policy Details". Below the header, it says "Nominee Details of the customer". The form fields include:

name delilah	Date Of Birth 13/01/2003	Gender <input type="radio"/> Male <input checked="" type="radio"/> Female <input type="radio"/> Transgender	Address line 1 43.Ashok Tapasvi, kottur gardens
Address line 2 Kotturpuram, Opposite to blossoms	city chennai	state tamil Nadu	country india
PinCode 600011	Mobile No 9878973288	Email delilah@gmail.com	Account Number 123456789
IFSC Code ghigh9873899000	Bank name ghigh	Branch sholingnur	Account Type savings

Buttons at the bottom: "Add Nominee" and "Delete Nominee".

Fig. 5.4 Nominee Details

## Addition of nominee

This is the page which is to be filled if the user wants to add a new nominee and on pressing save the request is submitted.

NOMINEE'S DETAILS

Nominee Name	Nominee's Date OF Bi...	Nominee's Gender <input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Transgender	Address line 1
Address line 2	Nominee's city	Nominee's state	Nominee's country
Nominee's PinCode	Nominee's Mobile No	Nominee's Email	Nominee's Account No
IFSC Code	Nominee's Bank name	Nominee's Branch	Nominee's Account Type
percent... <input type="button" value="%"/>	Save		

Fig. 5.5 Addition of a Nominee

## Policy Details

This is the page that displays all the policy details of the user.

Personal Details		Nominee Details		Policy Details			
		Policy Details					
_id	AppNo	ProductName	Term	SumAssured	Premium	PaymentFrequency	DateCreated
1	11127	Term Plan	60	60000	15000	quarterly	2019-12-31
2	11127	Health	60	25000	5000		
3	11127	ValueMed	60	30000	5000		
4	11127	Saver	60	25000	4000		

Fig. 5.6 Policy Details

## Payment Receipts

This page allows the user to view and download receipts of payments made by them between two dates.

The screenshot shows a web application interface for viewing payment receipts. At the top, there is a red header bar with a logo, the text "Welcome Details, user", and some user icons. Below the header is a title "Payment Receipts". The main content area contains a table with the following data:

No.	Application No	Policy No	Receipt No	Receipt Date	Premium Date	Premium Amount	Receipt Download
1	11127	1010	1	2020-01-03	2050-01-03	30000	
2	11127	1010	2	2020-01-06	2050-01-06	60000	
3	11127	101011	3	2020-01-07	2050-01-07	36000	

At the bottom of the page, there are navigation links for "Items per page: 5", "1 - 3 of 3", and some other icons. A copyright notice "© 2020 AccelTree Software. All rights reserved." is also present.

Fig. 5.7 Payment Receipts

## Policy Document

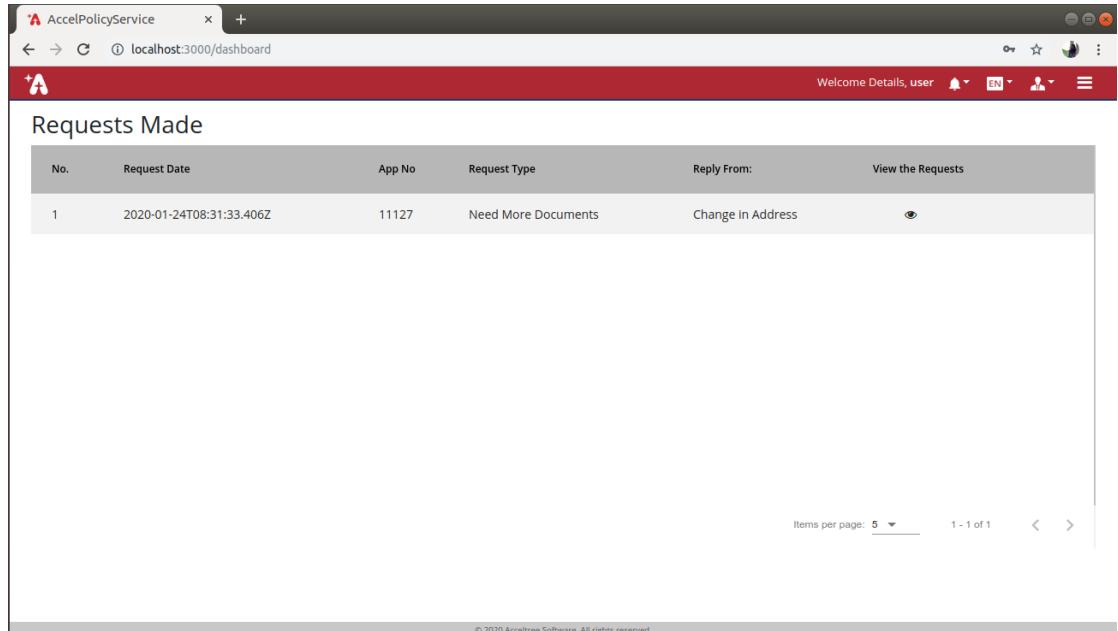
This page allows the user to download policy documents for the particular policy number that they would like.

The screenshot shows a web application interface for selecting a policy number to download its document. At the top, there is a browser-like header with a tab labeled "AccelPolicyService", a URL "localhost:3000/dashboard", and some icons. Below the header is a red header bar with a logo, the text "Welcome Details, user", and some user icons. The main content area contains a message box with the text "Select Policy Number to download policy documents". Below this is a form field labeled "PolicyNo:" with a dropdown arrow. At the bottom of the page, there is a copyright notice "© 2020 AccelTree Software. All rights reserved.".

Fig. 5.8 Policy Document

## Uploading Additional Documents

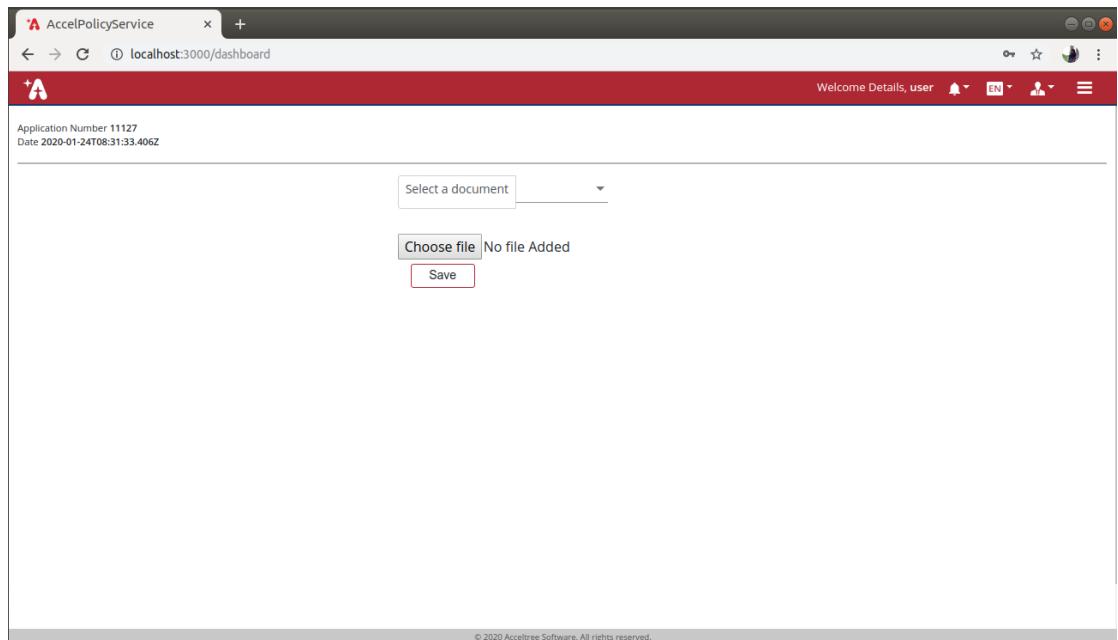
If the admin requests for more documents the user can view the request using the page that is displayed in Fig. 5.9 as well as They can use Fig. 5.10 to upload and view the image that they have uploaded and submit a request.



A screenshot of a web browser window titled "AccelPolicyService" showing the URL "localhost:3000/dashboard". The main content area is titled "Requests Made" and displays a table with one row of data. The table has columns: No., Request Date, App No, Request Type, Reply From, and View the Requests. The data row shows: No. 1, Request Date 2020-01-24T08:31:33.406Z, App No 11127, Request Type Need More Documents, Reply From Change in Address, and a View button. At the bottom of the table, there are pagination controls: "Items per page: 5", "1 - 1 of 1", and navigation arrows.

No.	Request Date	App No	Request Type	Reply From:	View the Requests
1	2020-01-24T08:31:33.406Z	11127	Need More Documents	Change in Address	

Fig. 5.9 Requests



A screenshot of a web browser window titled "AccelPolicyService" showing the URL "localhost:3000/dashboard". The main content area displays application details: Application Number 11127 and Date 2020-01-24T08:31:33.406Z. Below this, there is a form with a "Select a document" input field containing the placeholder "Choose file" and the message "No file Added". A "Save" button is located below the input field. At the bottom of the page, there is a copyright notice: "© 2020 Acceltree Software. All rights reserved."

Application Number 11127  
Date 2020-01-24T08:31:33.406Z

Select a document

Choose file No file Added

Save

Fig. 5.10 File Upload

### Edit Payment Frequency

The user is allowed to edit their payment frequency if they wish by entering the new payment frequency and clicking update.

Editing Payment Frequency

Old Payment Frequency: quarterly

new Payment Frequency: Half Yearly

Update

Fig. 5.11 Edit Payment Frequency

### Edit Name

If by chance the user wants to update their name then the user can use the above page to enter the new name , upload the necessary documents and submit the requests.

Editing Name

Old name: Kavya R

New name: Mrs.

Upload Marriage Certificate  
Choose file No file Added

Update

Fig. 5.12 Edit Name

## Request History

This page gives the user access to their request history between two dates as well as download the report.

The screenshot shows a 'Request History' page with a red header bar. On the left, there are date filters: 'From Date' set to '01/02/2020' and 'To Date' set to '30/04/2020'. A search icon is located below these filters. The main area is titled 'Request History' and contains a table with the following data:

No.	Requested Date	App No	Person's Name	Status	Request Type	Original Request Type
1	2020-02-28	11127	Kavya Rao	Request Accepted	NameChange	-
2	2020-03-13	11127	Kavya R	Documents Uploaded	NameChange	need more Documents

At the bottom right, there are pagination controls: 'Items per page: 5', '1 - 2 of 2', and navigation arrows. A watermark for 'Activate Windows' is visible on the right side.

Fig. 5.13 Request History

## Customer Search Module

### Dashboard

The Dashboard page is common to all the module the only difference is the buttons which navigates the user to the different pages

The screenshot shows a dashboard for the Customer Search module. At the top, there is a red header bar with the text 'Welcome Customer Service, Customer Service team'. Below the header, there are two large buttons:

- Customer Search**: Represented by a green icon of a person and a green background section.
- Claims**: Represented by a blue icon of a document and a blue background section. To the right of this button is the number '0'.

A watermark for 'Activate Windows' is visible at the bottom right.

Fig. 5.14 Dashboard for Customer Search

## Viewing of Customer Details

This page allows the user to search for a customer by their application number as well as initiating a claim by calling the claim form.

The screenshot shows a web browser window titled "AccelPolicyService" with the URL "localhost:3000/dashboard". The main content area displays two tables of customer data.

**Personal Details**

_id	appNo	name	DateOfBirth	Gender	creationDate	MobileNo	Email	personType
2	11134	arun	1969-09-29	Male	2019-12-27	9871085049	arun@gmail.com	life Assured
3	11134	karthik	2000-08-04	Male	2019-12-27	9876543210	karthik@gmail.com	nominee

**Address Details**

_id	appNo	buildingname	HouseNo	street	Locality	landmark	city	state	country	PinCode
12	11134	chaitanya grahakalyam	#21	pondy bazaar	tngar	naidu hall	pondi cherry	pondicherry	india	660001

Fig. 5.15 Customer Details

## Initialization Of CLaims

This is the page which is used to initiate the claim, accept or reject the claim or submit the claim after the form is filled.

The screenshot shows a web browser window titled "AccelPolicyService" with the URL "localhost:3000/claim". The main content area displays a form for "CLAIM FORM" and "NOMINEE'S DETAILS".

**CLAIM FORM**

Application No.	Policy No.	Name(Life Assured's)	Life Assured Date Of Birth
-----------------	------------	----------------------	----------------------------

**NOMINEE'S DETAILS**

Nominee Name	Nominee's Date OF Birth	Nominee's Gender <input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Transgender	Address line 1
Address line 2	Nominee's city	Nominee's state	Nominee's country
Nominee's PinCode	Nominee's Mobile No	Nominee's Email	Nominee's Account No
IFSC Code	Nominee's Bank name	Nominee's Branch	Nominee's Account Type
Type Of claim:	<input type="button" value="Save"/> <input type="button" value="Go Back"/>		

Fig. 5.16 Initialization of Claims

## Claims initialized

This page displays all the claims initialized to the user and allows them to view or edit whichever they want.

The screenshot shows a web browser window titled 'AccelPolicyService' with the URL 'localhost:3000/dashboard'. The page has a red header bar with the text 'Welcome Customer Service, Customer Service' and a dropdown menu. Below the header is a table titled 'Claims Made' with the following data:

No.	Claim Date	App No	Policy No	Date Of Birth(life Assured)	life Assured's Name	View/Edit
1	2020-01-21	11167	12321221	1943-12-16	mangal	

At the bottom of the table, there are links for 'Items per page: 5' and '1 - 1 of 1'.

Fig. 5.17 Claims initialized

## Claim Detail Updation and Submission

This is the page which is used to initiate the claim, accept or reject the claim or submit the claim after the form is filled

The screenshot shows a web browser window with a red header bar and a red footer bar. The main content area is titled 'CLAIM FORM' and contains the following fields:

Application No. 11113	Policy No. 1234143	Name(Life Assured's) Shanth Mannige	Life Assured Date Of Birth 14/08/1953
NOMINEE'S DETAILS			
Nominee Name Chetan Mannige	Nominee's Date OF Birth 30/01/1982	Nominee's Gender <input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Transgender	Address line 1 #204, Guruprasad
Address line 2 Priyanka Park	Nominee's city Pune	Nominee's state Maharashtra	Nominee's country India
Nominee's PinCode 4100013	Nominee's Mobile No 9874392010	Nominee's Email chetan.mannige@gmail.com	Nominee's Account No 934609285200024098
IFSC Code ICICI248091249012	Nominee's Bank name ICICI	Nominee's Branch Koregaon Park	Nominee's Account Type Savings
Type Of claim: <input type="button" value="Death Claim"/>	<input type="button" value="Submit"/> <input type="button" value="Go Back"/>		

Fig. 5.18 Claim Updation and Submition

## Claim Management Module Dashboard

The Dashboard page is common to all the module the only difference is the buttons which navigates the user to the different pages

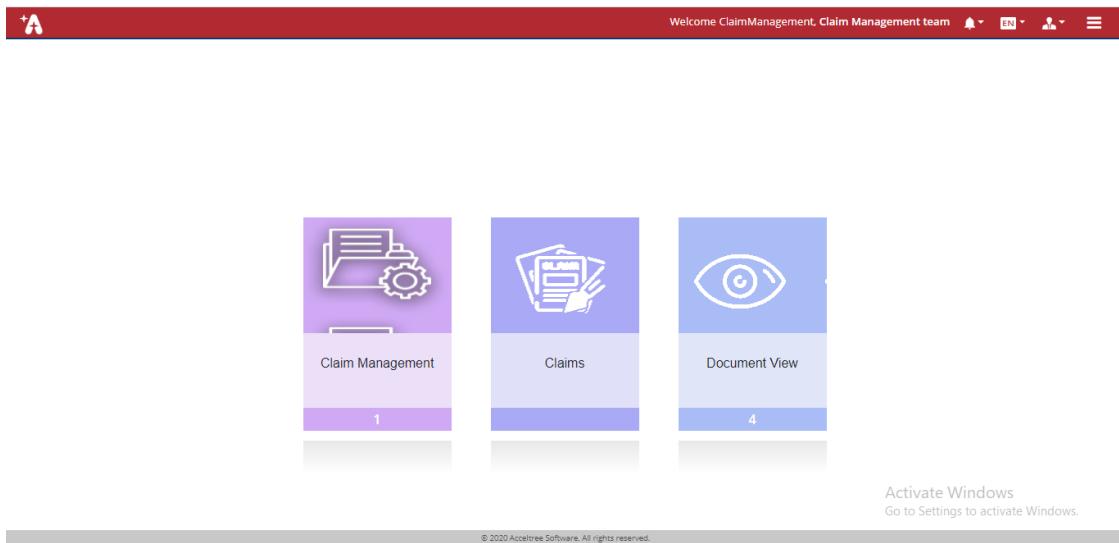


Fig. 5.19 Dashboard for Claim Management

### Claim Management

This page displays all the submitted claims to the user and allows them to view or edit whichever they want.

No.	Claim Date	App No	Policy No	Date Of Birth(life Assured)	life Assured's Name	View/Edit
1	2020-01-02	11127	101101	1978-08-28	nethini	
2	2020-01-03	11134	1213	1980-04-12	arun	
3	2020-01-03	11150	12345	1975-12-05	pranithi	
4	2020-01-03	11151	11345	1975-12-05	pranith	
5	2020-01-20	11159	122445	1978-05-21	Arjun	

Fig. 5.20 Claim Management

## Accept or Reject of Claims

This is the page which is used to initiate the claim, accept or reject the claim or submit the claim after the form is filled.

The screenshot shows a web-based claim form. At the top, there's a header bar with a logo and navigation links. Below it, the main form area has several sections:

- CLAIM FORM**: Includes fields for Application No. (11127), Policy No. (101011), Name(Life Assured's) (Kavya Rao), and Life Assured Date Of Birth (12/08/1989).
- NOMINEE'S DETAILS**: A large section containing multiple rows of input fields. Some examples include:
  - Nominee Name: shiv
  - Nominee's Date Of Birth: 18/02/2011
  - Nominee's Gender: Male (radio button selected)
  - Address Line 1: 43 Ashok Tapasyi, kottur gardens
  - Nominee's building name: Opposite to blossoms hair studio
  - Nominee's city: chennai
  - Nominee's state: tamil Nadu
  - Nominee's PinCode: 600011
  - Nominee's Mobile No: 9789954332
  - Nominee's Email: shiv@gmail.com
  - Nominee's Branch: kottapuram
  - Nominee's Account No: 73562998374898
  - Nominee's Account Type: savings
  - Nominee's IFSC Code: FFI18381273621
  - Nominee's Landmark: Tangerine
  - Nominee's House No: 204
  - Nominee's PinCode: 532014
  - Nominee's IFSC Code: ICSE000001
- A dropdown menu for "Type Of claim" with options like "Non Death Claim".
- Buttons at the bottom right: "Go Back" and "Submit".

Fig. 5.21 Claim Form

The screenshot shows a web-based interface for accepting or rejecting claims. It features a dark-themed dashboard with a central modal dialog box. The dialog box contains the text "Enter Comments" and two buttons: "Accept" and "Reject". The background of the page shows various input fields for nominee details, similar to Fig. 5.21, such as Nominee Name (nidhin), Nominee's Date Of Birth (12/03/2000), Nominee's city (bangalore), and Nominee's Branch (Koramangala). There are also buttons for "Go Back" and "Submit".

Fig. 5.22 Accept or Reject Claims

## Report of claims

This page gives the user access to view the accepted or rejected claim reports between dates as well as download the claim report.

The screenshot shows a web-based application for claim management. At the top, there is a red header bar with the text "Welcome Claim Management, Claim Management team" and various system icons (bell, EN, user, etc.). Below the header, the title "Accepted Claims" is displayed. On the left side, there are two date input fields: "From Date" set to "01/02/2020" and "To Date" set to "29/02/2020". Below these dates is a dropdown menu labeled "Type" with "Accepted Claim" selected. To the right of the date fields is a table header row with columns: No., Accepted Date, App No, Policy No, Person's Name, and Approved Amt. Below the table header, there are pagination controls showing "Items per page: 5" and "0 of 0". At the bottom of the page, a copyright notice reads "© 2020 AccelTree Software. All rights reserved."

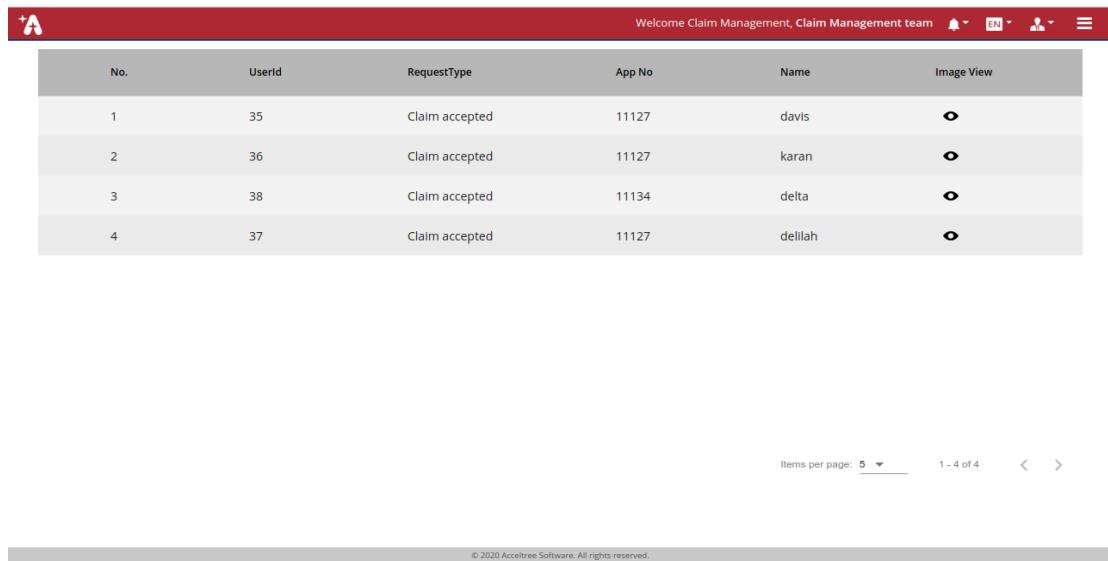
Fig. 5.23 Claim Report of Accepted Claims

The screenshot shows the same web-based application for claim management. The title "Rejected Claims" is displayed at the top. The left side features the same date input fields ("From Date: 01/02/2020" and "To Date: 29/02/2020") and a dropdown menu labeled "Type" with "Rejected Claim" selected. To the right is a table header row with columns: No., Accepted Date, App No, Policy No, Person's Name, and Approved Amt. Below the table header, there are pagination controls showing "Items per page: 5" and "0 of 0". At the bottom of the page, a copyright notice reads "© 2020 AccelTree Software. All rights reserved."

Fig. 5.24 Claim Report of Rejected Claims

## Document View

This page displays to the user all the different customers who have uploaded their documents after their claim has been accepted.



The screenshot shows a table titled "Welcome Claim Management, Claim Management team". The table has columns: No., Userid, RequestType, App No, Name, and Image View. There are four rows of data:

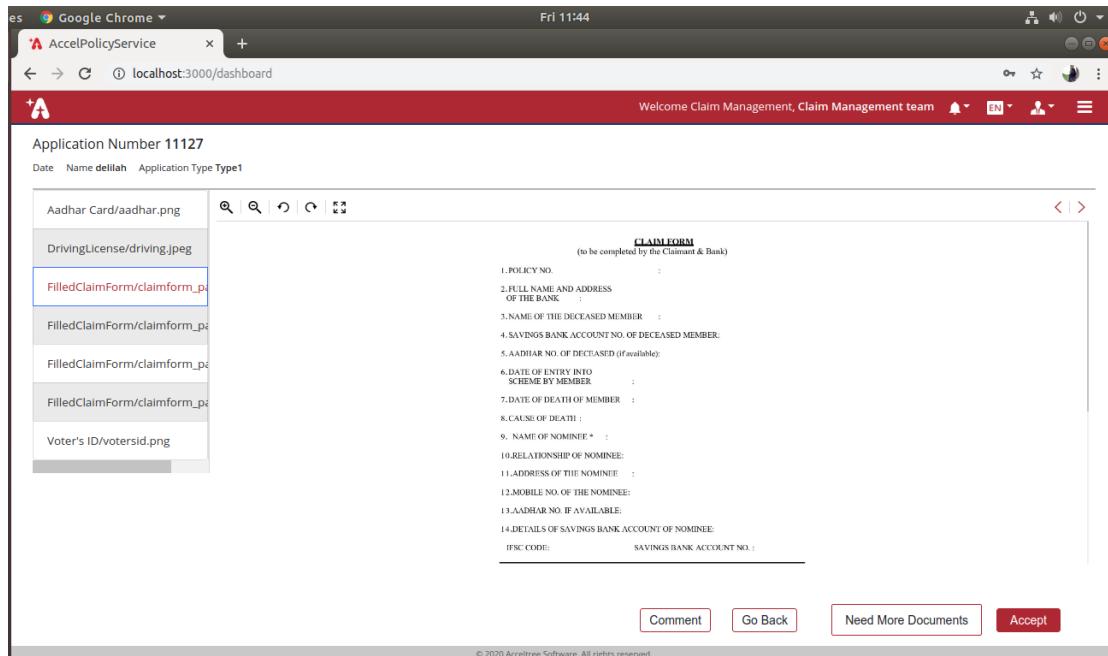
No.	Userid	RequestType	App No	Name	Image View
1	35	Claim accepted	11127	davis	
2	36	Claim accepted	11127	karan	
3	38	Claim accepted	11134	delta	
4	37	Claim accepted	11127	delilah	

At the bottom, there are pagination controls: "Items per page: 5" and "1 - 4 of 4".

Fig. 5.25 Document View

## Document Display

This page allows the user to view all the documents uploaded as well as manipulate the image. It gives them the power to add comments, request for more documents or approve the claim.



The screenshot shows a web browser window with the URL "localhost:3000/dashboard". The page title is "Application Number 11127". It displays a list of uploaded documents on the left and a "CLAIM FORM" on the right.

Documents listed on the left:

- Aadhar Card/aadhar.png
- DrivingLicense/driving.jpeg
- FilledClaimForm/claimform.pdf
- FilledClaimForm/claimform.pdf
- FilledClaimForm/claimform.pdf
- Voter's ID/votersid.png

The "CLAIM FORM" section contains fields for personal information and banking details. At the bottom, there are buttons: "Comment", "Go Back", "Need More Documents", and "Accept".

Fig. 5.26 Document Display

## Request Management Module

### Dashboard

The Dashboard page is common to all the module the only difference is the buttons which navigates the user to the different pages

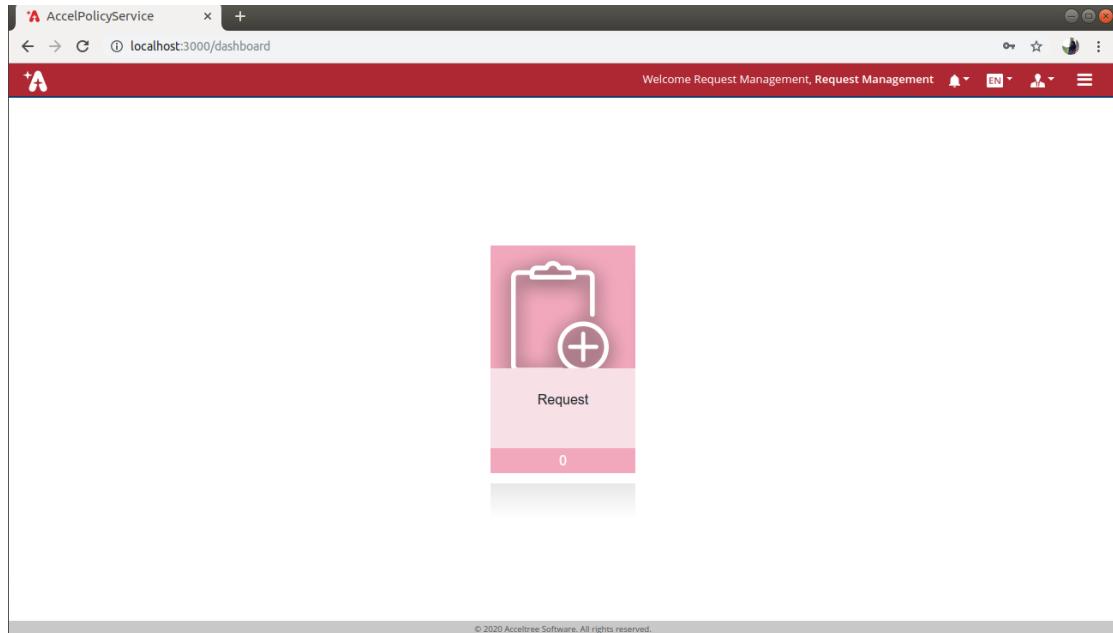


Fig. 5.27 Dashboard for Request Management

### Requests

This page displays all the requests made by the customers to the specific admins and allows them to view the request they wish to.

A screenshot of a web browser window titled "AccelPolicyService" showing the "Requests Made" page. The URL is "localhost:3000/dashboard". The interface has a red header bar with the text "Welcome Request Management, Request Management" and icons for notifications, language (EN), user profile, and a menu. The main content area shows a table titled "Requests Made" with columns: No., Request Date, App No, Request Type, Reply From:, and View the Requests. There are three rows of data. At the bottom, there are pagination controls for "Items per page: 5" and "1 - 3 of 3".

No.	Request Date	App No	Request Type	Reply From:	View the Requests
1	2020-01-08	11143	Change in Address	-	
2	2020-01-08	11127	Change in Address	need more Documents	
3	2020-01-09	11127	nominee Change	need more Documents	

Fig. 5.28 Requests

## Request Display

This page is common for the four requests Change in Address,Nominee Change,Addition of a Nominee, Deletion of a Nominee. This page also allows the user to view documents uploaded, accept or reject the request or request for more documents.

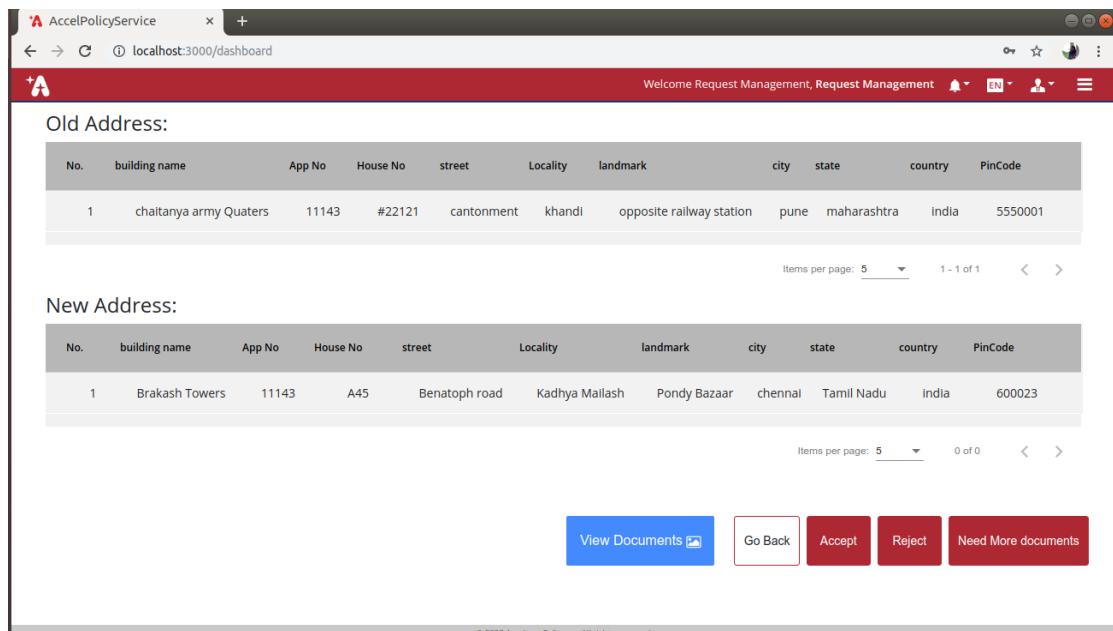


Fig. 5.29 Request Display

Fig. 5.30 depicts a page that allows the user to view documents uploaded,Manipulate the image, accept or reject the request or request for more . This page is for the request Name Change

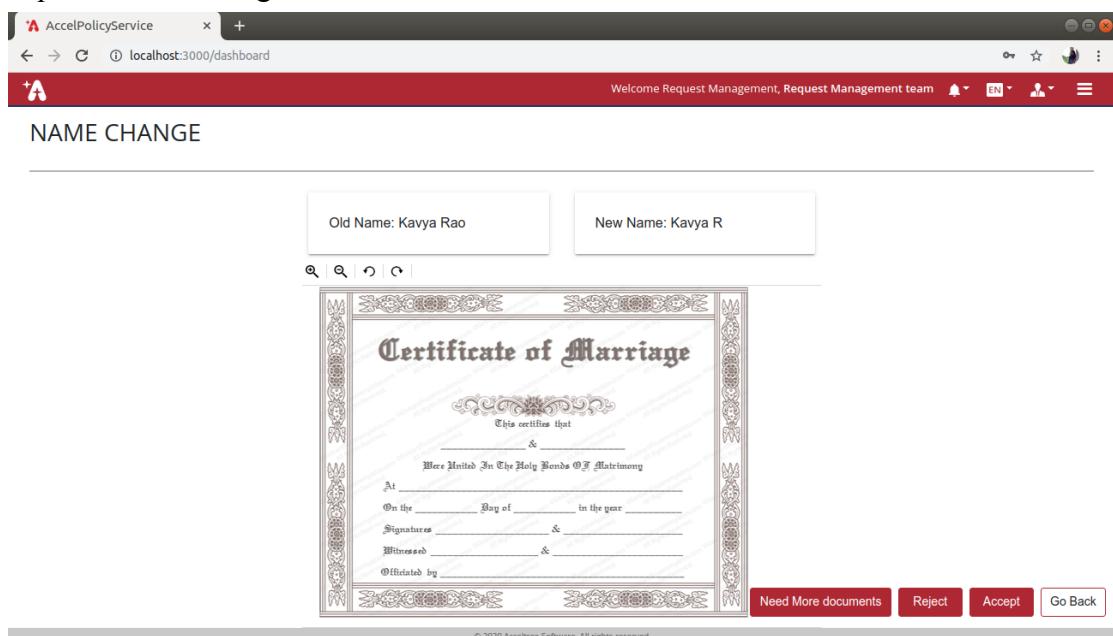


Fig. 5.30 Request Display for Name Change

## Nominee Module

### Dashboard

The Dashboard page is common to all the module the only difference is the buttons which navigates the user to the different pages

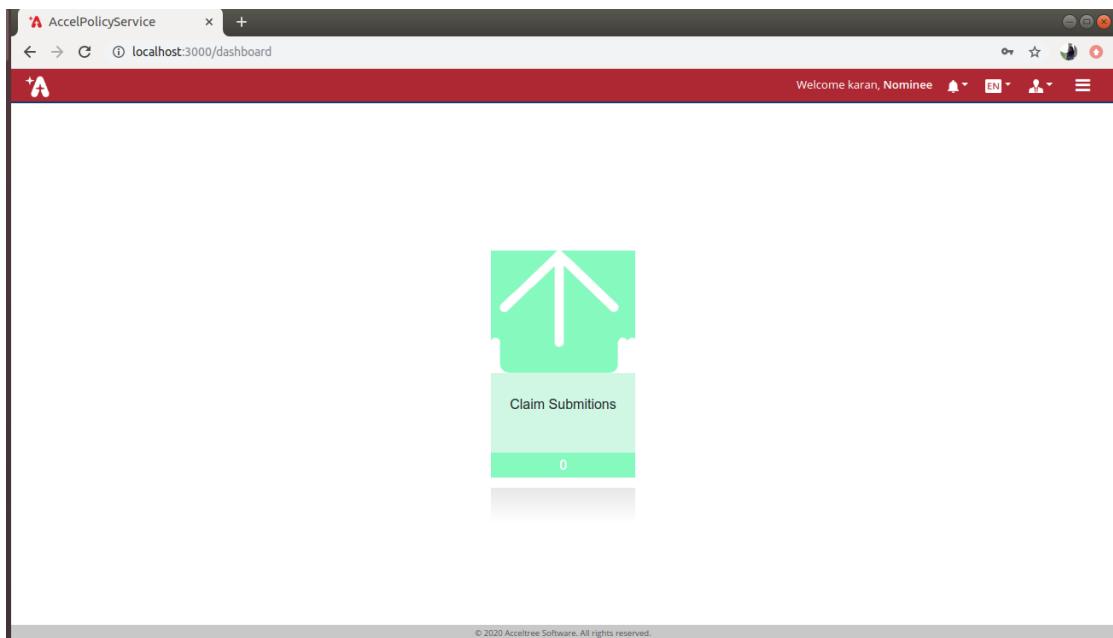


Fig. 5.31 Dashboard for Nominee

### Claim Submitions

This page allows the user to upload files, downloading the claim form as well as view the images. This is mainly for the users whose claims have been accepted.

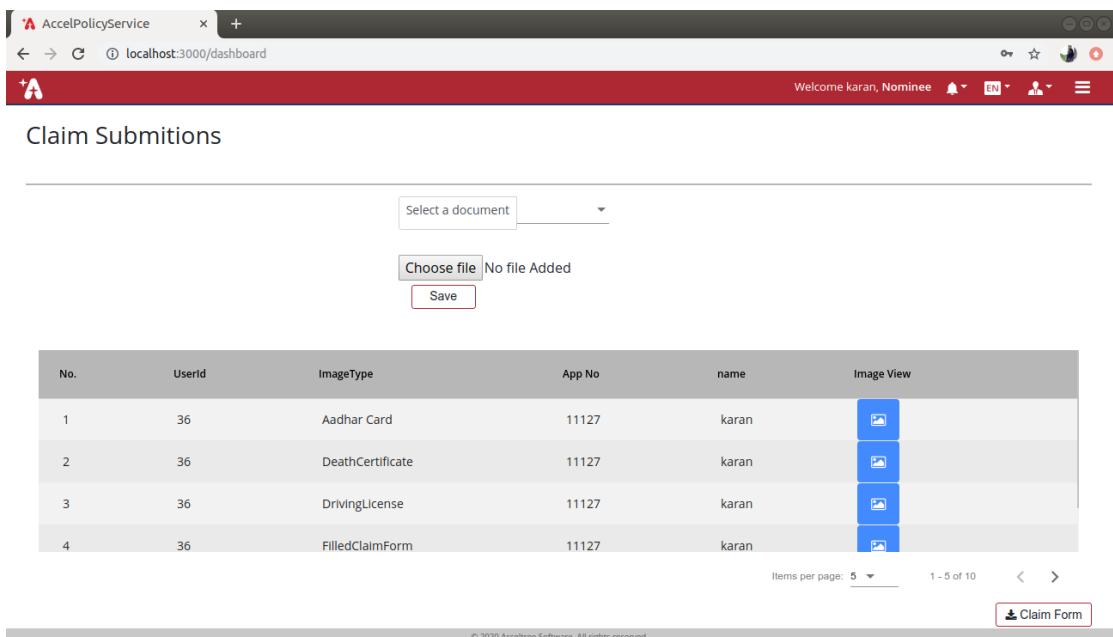


Fig. 5.32 Claim Submitions

## **Chapter 6**

# **Conclusion and Future Work**

### **6.1 CONCLUSION**

The main aim of the project was to limit time consumption and manual effort. The project revolves around the topic digitalization of insurance policy services. There are many companies at the present which offer all of these services but the companies have digitized only few of the services and out of those few there are few cases where manual labour or effort is necessary. The project works on that to make sure that all the services can be offered via the application without the need of the user putting in any form of manual labour. The project also works in such a manner that each module belongs to a different team completely. Each team is given a certain number of operations that have to be carried out in an efficient manner hence making the application more efficient and less time consuming. The project helped in understanding how important it is in implementing a hierarchy so that the project runs more efficiently as well as the operations are more clear and easy to understand.

### **6.2 FUTURE WORK**

The project can be further improved by including an operation where the user doesn't need to call the customer service to initiate a claim or for any queries but will be able to chat with them. There can also be an increase in automatic management of requests.

## REFERENCES

- [1] Cortiñas, M., Chocarro, R., & Elorz, M. (2019). Omni-channel users and omni-channel customers: a segmentation analysis using distribution services. *Spanish Journal of Marketing - ESIC*, 23(3), 415–436. doi: 10.1108/sjme-06-2019-0031
- [2] HDFC Life Insurance Company Limited (Formerly HDFC Standard Life Insurance Company Limited). (n.d.). Retrieved from <https://www.hdfcliffe.com/>
- [3] Khare, A., Dixit, S., Chaudhary, R., Kochhar, P., & Mishra, S. (2012). Customer behavior toward online insurance services in India. *Journal of Database Marketing & Customer Strategy Management*, 19(2), 120–133. doi: 10.1057/dbm.2012.14
- [4] KLAPKIV, L., KLAPKIV, J., & Curie-Skłodowska , M. (3n.d.). Technological innovations in the insurance industry. *Journal of Insurance, Financial Markets and Consumer Protection*, 26(67-78). Retrieved from <https://rf.gov.pl/join/wp-content/uploads/2018/01/RU26-5.pdf>