

# **Software Requirement Specification Document**

## **1. Introduction**

- 1.1 purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms and Abbreviations
- 1.4 References
- 1.5 Overview

## **2. Overall Description**

- 2.1 Product Perspective
  - 2.1.1 System interfaces
  - 2.1.2 User Interfaces
  - 2.1.3 Hardware Interfaces
  - 2.1.4 Software Interfaces
  - 2.1.5 Communication Interfaces
  - 2.1.6 memory constraints
  - 2.1.7 Operations
  - 2.1.8 Site Adaptation requirements
- 2.2 Product Functions
- 2.3 User Characteristics
- 2.4 Constraints'
- 2.5 Assumptions and Dependencies
- 2.6 Apportioning of Requirements

## **3. Specific Requirements**

- 3.1 External Interface Requirements
  - 3.1.1 User Interfaces
  - 3.1.2 Hardware Interfaces
  - 3.1.3 Software Interfaces
  - 3.1.4 Communication Interfaces
- 3.2 Functional Requirements
  - 3.2.1 registration
  - 3.2.2 Login
  - 3.2.3 Policy Management
  - 3.2.4 Claim Processing
  - 3.2.5 Report generation
  - 3.2.6 Payment system
  - 3.2.7 User support and assistance
- 3.3 Performance Requirements
- 3.4 Design Constraints
- 3.5 Software system attributes

3.6 Logical Database Requirements

3.7 Other requirements

# **1. Introduction**

The Insurance Management System is a comprehensive software application designed to streamline the operations of an insurance agency. The primary purpose of this system is to automate the existing manual processes and provide a centralized platform for managing various aspects of insurance management.

The system will provide features for customer management, policy management, claim processing, payment processing, and user support.

## **1.1 Purpose**

The purpose of an Insurance Management System is to provide a robust, efficient, and user-friendly platform for managing insurance policies, claims, accounting, and reporting processes.

## **1.2 Scope**

The key features and objectives of the Insurance Management System include:

1. Customer Database Management: The system allows insurance agencies to store and organize customer data, including personal details, contact information, policy history, and past interactions.
2. Policy Management: The system enables customers to view, update, and renew their insurance policies, while also allowing agents to manage customer policies and recommend changes.
3. Claim Processing: The system facilitates the filing and tracking of insurance claims, enabling customers to submit claims and agents to review, assess, and process the claims.
4. Payment Processing: The system integrates with a secure payment gateway to allow customers to make premium payments for their insurance policies using various payment methods.
5. Reporting and Analytics: The system provides administrators with the ability to generate reports on various aspects of the insurance management system, such as customer demographics, policy data, claim statistics, and premium revenue.

6. User Support and Assistance: The system offers customers and agents access to help resources, the ability to submit support requests, and a mechanism for the support team to respond and resolve issues.

### **1.3 Definitions, Acronyms, and Abbreviations**

- IMS: Insurance Management System
- SRS: Software Requirements Specification
- UI: User Interface
- API: Application Programming Interface
- DB: Database
- DBMS: Database Management System

### **1.4 References**

- Software Engineering by Ruchika Malhotra and Yogesh Singh, New age publishing house 3rd edition.
- IEEE standard for software test Documentation

### **1.5 Overview**

This SRS document is organized into two main sections:

1. Overall Description: This section provides an overview of the product, including the system perspective, product functions, user characteristics, constraints, assumptions, and dependencies.
2. Specific Requirements: This section outlines the detailed requirements for the system, including external interface requirements, functional requirements, performance requirements, design constraints, software system attributes, logical database requirements, and other requirements.

## **2. Overall Description**

The Insurance Management System (IMS) is a comprehensive, reliable, and performance-based software solution designed to streamline insurance operations for insurance service providers. It is a state-of-the-art system that integrates all components into a single application, offering a dynamic tree structure for system management. IMS is tailored to meet the daily operational needs of insurance providers, offering core functionalities that include underwriting, claims management, accounting, reinsurance management, information system management, and application management.

1. Customer Database Management: The system acts as a centralized repository for storing and organizing comprehensive customer data, including personal details, contact information, policy history.
2. Policy Management: The system empowers customers to independently view, update, and renew their insurance policies through a user-friendly interface.
3. Claim Processing: The system facilitates the seamless filing and tracking of insurance claims by customers, providing a convenient and user-friendly experience.
4. Payment Processing: The system integrates with a secure payment gateway, enabling customers to conveniently make premium payments for their insurance policies using various payment methods, such as credit/debit cards, bank transfers, or digital wallets.
5. Reporting and Analytics: The system provides administrators with robust reporting and analytics capabilities, allowing them to generate customized reports on various aspects of the insurance management system.
6. User Support and Assistance: The system offers customers and agents access to a range of help resources, such as FAQs, user guides, and contact information for the support team.

## 2.1 Product Perspective

The Insurance Management System is a standalone software application that will be used by insurance agencies to manage their operations, including customer management, policy management, claim processing, payment processing, and user support.

### 2.1.1 System Interfaces

The IMS will integrate with the following external systems:

- Payment Gateway: For processing premium payments
- Authentication Server: For user authentication and authorization

### 2.1.2 User Interfaces

The IMS will provide a web-based user interface (UI) for the following user roles:

- Customers: To manage their policies, file claims, and make payments
- Agents: To manage customer policies, process claims, and provide customer support
- Administrators: To manage policies, user accounts, system configurations, and generate reports

### 2.1.3 Hardware Interfaces

The IMS will be compatible with the following hardware:

- Desktop computers and laptops running Windows 7 or above, macOS 11 or above, or Linux.
- Tablets and mobile devices running iOS 10 or above or Android 10 or above

#### 2.1.4 Software Interfaces

The IMS will integrate with the following software components:

- Database Management System (DBMS): For storing and managing system data
- Reporting and Analytics Tools: For generating reports and analyzing system data

#### 2.1.5 Communication Interfaces

The IMS will communicate with external systems and components using the following protocols:

- HTTP/HTTPS: For web-based communication
- REST API: For integrating with the Payment Gateway and Authentication Server

#### 2.1.6 Memory Constraints

The IMS should be designed to operate within the following memory constraints:

- Minimum 2GB RAM and 7GB of Hard disk will be required.
- Maximum concurrent user sessions: 1,000

#### 2.1.7 Operations

---

#### 2.1.8 Site Adaptation Requirements

The system should be compatible with a wide range of hardware platforms, including desktop computers, laptops, tablets, and mobile devices running various operating systems (Windows, macOS, Linux, iOS, Android).

## 2.2 Product Functions

The key functions of the Insurance Management System include:

1. Registration: Allowing customers to create an account in the system.
2. Login: Enabling users (customers, agents, and administrators) to securely access the system.
3. Policy Management: Allowing customers to view, update, and renew their insurance policies.
4. Claim Processing: Enabling customers to file and track the status of their insurance claims.
5. Report Generation: Providing administrators with the ability to generate reports on system usage, policy data, and claim statistics.
6. Payment System: Facilitating premium payments for insurance policies.
7. User Support and Assistance: Offering customers and agents access to help resources and the ability to submit support requests.

## 2.3 User Characteristics

The Insurance Management System will have the following user roles:

1. Qualification: At least Matriculation and comfortable with English.
2. Technical Knowledge: Basic knowledge of computers and familiarity with websites.

## 2.4 Constraints

The key constraints for the Insurance Management System include:

1. Regulatory Compliance: The system must comply with relevant insurance industry regulations and data privacy laws.
2. Integration with Existing Systems: The system must be able to integrate with the organization's existing payment gateway and authentication server.
3. Security and Data Privacy: The system must ensure the security and privacy of customer data, including personal information and financial data.
4. Scalability: The system must be designed to handle a growing number of users and policies without significant performance degradation.

## 2.5 Assumptions and Dependencies

The following assumptions and dependencies have been identified for the Insurance Management System:

1. Availability of Payment Gateway: The system assumes the availability of a reliable payment gateway for processing premium payments.

2. Availability of Authentication Server: The system depends on the availability of an authentication server for user authentication and authorization.
3. Stable Internet Connectivity: The system assumes that users will have stable internet connectivity to access the web-based application.
4. User Familiarity with Web Applications: The system assumes that users will have a basic understanding of web-based applications and their navigation.

## 2.6 Apportioning of Requirements

Not required

# 3. Specific Requirements

This section contains software requirements in detail:

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

The Insurance Management System will provide the following user interfaces:

1. Customer UI:
  - Registration and login
  - Policy management (view, update, renew)
  - Claim filing and tracking
  - Premium payment
  - Help and support request submission
2. Agent UI:
  - Login
  - Policy management (view, update)
  - Claim processing
  - Customer support
3. Administrator UI:
  - Login
  - Policies Management(add new policy, delete policy)
  - User account management
  - System configuration
  - Report generation

### 3.1.2 Hardware Interfaces

The IMS will be compatible with the following hardware:

1. Desktop computers and laptops running Windows 7 or above, macOS 11 or above, or Linux.
2. Tablets and mobile devices running iOS 10 or above or Android 10 or above

### 3.1.3 Software Interfaces

The Insurance Management System will integrate with the following software components:

1. Database Management System (DBMS): The system will use a DBMS, such as MySQL or PostgreSQL, to store and manage system data.
2. Reporting and Analytics Tools: The system will integrate with reporting and analytics tools to generate reports and analyze system data.

### 3.1.4 Communication Interfaces

The Insurance Management System will communicate with external systems and components using the following protocols:

1. HTTP/HTTPS: The system will use HTTP/HTTPS for web-based communication with the user interfaces.
2. REST API: The system will use REST API to integrate with the Payment Gateway and Authentication Server.

## 3.2 Functional Requirements

### 3.2.1 Registration

#### A. Use Case Description

##### 1. Introduction

The purpose of the Registration use case is to allow new customers to create an account in the insurance management system and provide their personal and contact information.

##### 2. Actors

- Customer

### **3. Precondition**

The customer does not have an existing account in the insurance management system.

### **4. Postcondition**

The customer has a new account created in the insurance management system.

### **5. Flow of Events**

#### **Basic Flow**

1. The customer navigates to the registration page.
2. The customer enters their personal information (name, address, contact details, etc.).
3. The customer creates a unique username and password for their account.
4. The system validates the input and creates a new customer account.
5. The system sends a confirmation message to the customer.

### **6. Alternative Flow**

- If any required information is missing or invalid during registration:
  1. System prompts the user to correct the errors.
  2. User re-enters the correct information.
  3. Steps 4-5 of the basic flow are followed once the information is validated successfully.

### **7. Special Requirements**

- The system must validate the uniqueness of the username and the format of the personal information.
- The system must securely store the customer's password using hashing and salting techniques.
- The system must send a confirmation email to the customer's provided email address.

### **8. Associated Use Cases**

- Login

#### **3.2.2 Login**

##### **B. Use Case Description**

###### **1. Introduction**

The purpose of the Login use case is to allow authorized users (customers, sales agents, and administrators) to securely access the insurance management system by entering their credentials.

### **1. Actors**

- Customer
- Sales Agent
- Administrator

### **3. Precondition**

The user has a valid account in the insurance management system.

### **4. Postcondition**

The user is logged in and can access the system's functionalities based on their role.

### **5. Flow of Events**

#### **Basic Flow**

1. The user navigates to the login page.
2. The user enters their username and password.
3. The system validates the credentials.
4. The system grants access to the user based on their role (customer, agent, or administrator).

### **6. Alternative Flow**

1. The user enters invalid credentials.
2. The system displays an error message and prompts the user to try again.
3. If the user exceeds the maximum number of failed login attempts, the system locks the user's account.

### **7. Special Requirements**

- The system must validate the user's credentials against the stored hashed and salted passwords.
- The system must implement account lockout mechanisms to prevent brute-force attacks.
- The system must provide a password reset functionality for users who have forgotten their credentials.

### **8. Associated Use Cases**

- Registration
- Policy Management
- File Claim
- Claim Processing
- Premium Payment
- User Management

### **3.2.3 Policy Management**

#### **1. Introduction**

The purpose of the Policy Management use case is to allow customers to view, update, and renew their insurance policies.

#### **2. Actors**

- Customer

#### **3. Precondition**

The customer has an active insurance policy.

#### **4. Postcondition**

The customer's policy information is updated or renewed in the system.

#### **5. Flow of Events**

##### **Basic Flow**

1. The customer logs in to the system.
2. The customer navigates to the policy management section.
3. The customer can view the details of their current policy.
4. The customer can update their policy information (e.g., contact details, coverage levels).
5. The customer can renew their policy when it is due for renewal.
6. The system processes the policy updates or renewal and sends a confirmation to the customer.

#### **6. Alternative Flow**

1. The customer attempts to update their policy, but the system detects an invalid change.
2. The system displays an error message and prompts the customer to correct the information.

## 7. Special Requirements

- The system must validate the customer's policy information and prevent invalid updates.
- The system must automatically renew the customer's policy when it is due, unless the customer chooses to make changes.
- The system must provide clear and concise policy details to the customer, including coverage levels, premium amounts, and renewal dates.

## 8. Associated Use Cases

- Login
- Premium Payment

### 3.2.4 File Claim

#### 1. Introduction

The purpose of the File Claim use case is to allow customers to submit insurance claims for coverage.

#### 2. Actors

- Customer

#### 3. Precondition

The customer has an active insurance policy.

#### 4. Postcondition

The customer's claim is submitted and pending review.

#### 5. Flow of Events

##### Basic Flow

1. The customer logs in to the system.
2. The customer navigates to the claims section.
3. The customer provides details about the incident or event that led to the claim.

4. The customer submits the claim, along with any required documentation.
5. The system records the claim and forwards it to the claims processing team.

## 6. Alternative Flow

1. The customer attempts to file a claim, but the system detects missing or invalid information.
2. The system displays an error message and prompts the customer to correct the information.

## 7. Special Requirements

- The system must validate the claim information and ensure that the customer's policy covers the reported incident.
- The system must provide clear instructions and guidance to the customer on the required documentation for the claim.
- The system must securely store the claim details and associated documentation.

## 8. Associated Use Cases

- Login
- Claim Processing

### 3.2.5 Claim Processing

#### 1. Introduction

The purpose of the Claim Processing use case is to allow the claims processing team to review and process the insurance claims submitted by customers.

#### 2. Actors

- Claims Clerk

#### 3. Precondition

A customer has submitted an insurance claim.

#### 4. Postcondition

The customer's claim has been processed, and the decision has been communicated.

#### 5. Flow of Events

### **Basic Flow**

1. The claims clerk receives the customer's claim and associated documentation.
2. The claims clerk reviews the claim for completeness and accuracy.
3. The claims clerk forwards the claim to the claims adjuster for further evaluation.
4. The claims clerk assesses the claim, determines the appropriate coverage and compensation, and makes a decision.
5. The claims clerk updates the claim status and notifies the customer.

### **6. Alternative Flow**

1. The claims clerk determines that the claim is not valid or covered by the customer's policy.
2. The claims clerk updates the claim status and notifies the customer.

### **7. Special Requirements**

- The system must provide a secure and organized way for the claims processing team to access and manage the submitted claims.
- The system must have clear business rules and guidelines for the claims clerk to evaluate the claims and make decisions.
- The system must automatically update the claim status and notify the customer of the decision.

### **8. Associated Use Cases**

- File Claim

## **3.2.6 Payment System**

### **1. Introduction**

The purpose of the Premium Payment use case is to allow customers to pay their insurance premiums, either as a one-time payment or on a recurring basis.

### **2. Actors**

- Customer

### **3. Precondition**

The customer has an active insurance policy.

#### 4. Postcondition

The customer's premium payment is successfully processed and recorded in the system.

#### 5. Flow of Events

##### Basic Flow

1. The customer logs in to the system.
2. The customer navigates to the premium payment section.
3. The customer selects the payment method (e.g., credit card, bank account) and enters the required payment details.
4. The system processes the payment and updates the customer's policy records.
5. The system sends a confirmation to the customer.

##### 6. Alternative Flow

1. The customer's payment is declined due to insufficient funds or other issues.
2. The system displays an error message and prompts the customer to try a different payment method.

#### 7. Special Requirements

- The system must support multiple payment methods and securely handle customer financial information.
- The system must provide clear and transparent information about the premium amounts and due dates.
- The system must automatically generate reminders for upcoming premium payments.

#### 8. Associated Use Cases

- Login
- Policy Management

### 3.2.7 Help & support

#### 1. Introduction

The purpose of the Help use case is to provide customers, sales agents, and administrators with access to support resources and assistance within the insurance management system.

## **2. Actors**

- Customer
- Sales Agent
- Administrator

## **3. Precondition**

The user is logged in to the insurance management system.

## **4. Postcondition**

The user has accessed the desired support resources or assistance.

## **5. Flow of Events**

### **Basic Flow**

1. The user navigates to the help section of the system.
2. The user can access various support resources, such as:
  - Frequently Asked Questions (FAQs)
  - User guides and tutorials
  - Contact information for customer support
3. The user can submit a support request or inquiry, if needed.
4. The system routes the request to the appropriate support team.

### **6. Alternative Flow**

1. The user cannot find the desired information in the support resources.
2. The user submits a support request, and the system provides an estimated response time.

## **7. Special Requirements**

- The system must maintain a comprehensive knowledge base of support resources, covering common user questions and issues.
- The system must provide multiple channels for users to access support, such as online chat, email, and phone.
- The system must have a streamlined process for routing support requests to the correct support team and tracking their resolution.

## **8. Associated Use Cases**

- Login

## 3.2.8 Logout

### 1. Introduction

The purpose of the Logout use case is to allow authorized users to securely exit the insurance management system.

### 2. Actors

- Customer
- Sales Agent
- Administrator

### 3. Precondition

The user is currently logged in to the insurance management system.

### 4. Postcondition

The user's session is terminated, and they are no longer able to access the system's functionalities.

### 5. Flow of Events

#### Basic Flow

1. The user navigates to the logout option or menu.
2. The system terminates the user's active session.
3. The system redirects the user to the login page or the system's homepage.

#### 6. Alternative Flow

- If the user decides not to log out:
  1. User cancels the logout action.
  2. User remains logged in to their account.

### 7. Special Requirements

- The system must ensure that all user data and session information are properly cleared and secured upon logout.
- The system must provide a clear and intuitive logout mechanism for users.

### 8. Associated Use Cases

- Login

### **3.3 Performance Requirements**

1. The system shall have a response time of less than 3 seconds for all user interactions, such as page loads, data retrieval, and form submissions.
2. The system shall be able to handle at least 1,000 concurrent user sessions without significant performance degradation.
3. The system shall have an uptime of at least 99.9% to ensure reliable availability for users.
4. The system shall be able to process and store up to 1 TB of data, including customer records, policy details, and claim information.

### **3.4 Design Constraints**

1. Regulatory Compliance: The system shall comply with relevant insurance industry regulations and data privacy laws, such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA).
2. Integration with Existing Systems: The system shall be designed to integrate with the organization's existing payment gateway and authentication server.
3. Security and Data Privacy: The system shall implement robust security measures, such as encryption, access controls, and audit logging, to protect customer data and ensure compliance with data privacy regulations.
4. Scalability: The system shall be designed to scale horizontally and vertically to accommodate future growth in the number of users and policies.

### **3.5 Software System Attributes**

1. Security: The system shall implement secure authentication mechanisms, data encryption, and access controls to protect against unauthorized access and data breaches.
2. Reliability: The system shall be designed to minimize downtime and ensure consistent availability for users, with mechanisms in place for failover and disaster recovery.
3. Maintainability: The system shall be designed with modular architecture and well-documented code to facilitate easy maintenance, updates, and bug fixes.
4. Usability: The system shall provide intuitive and user-friendly interfaces, with clear navigation, error handling, and help resources to enhance the overall user experience.

### **3.6 Logical Database Requirements**

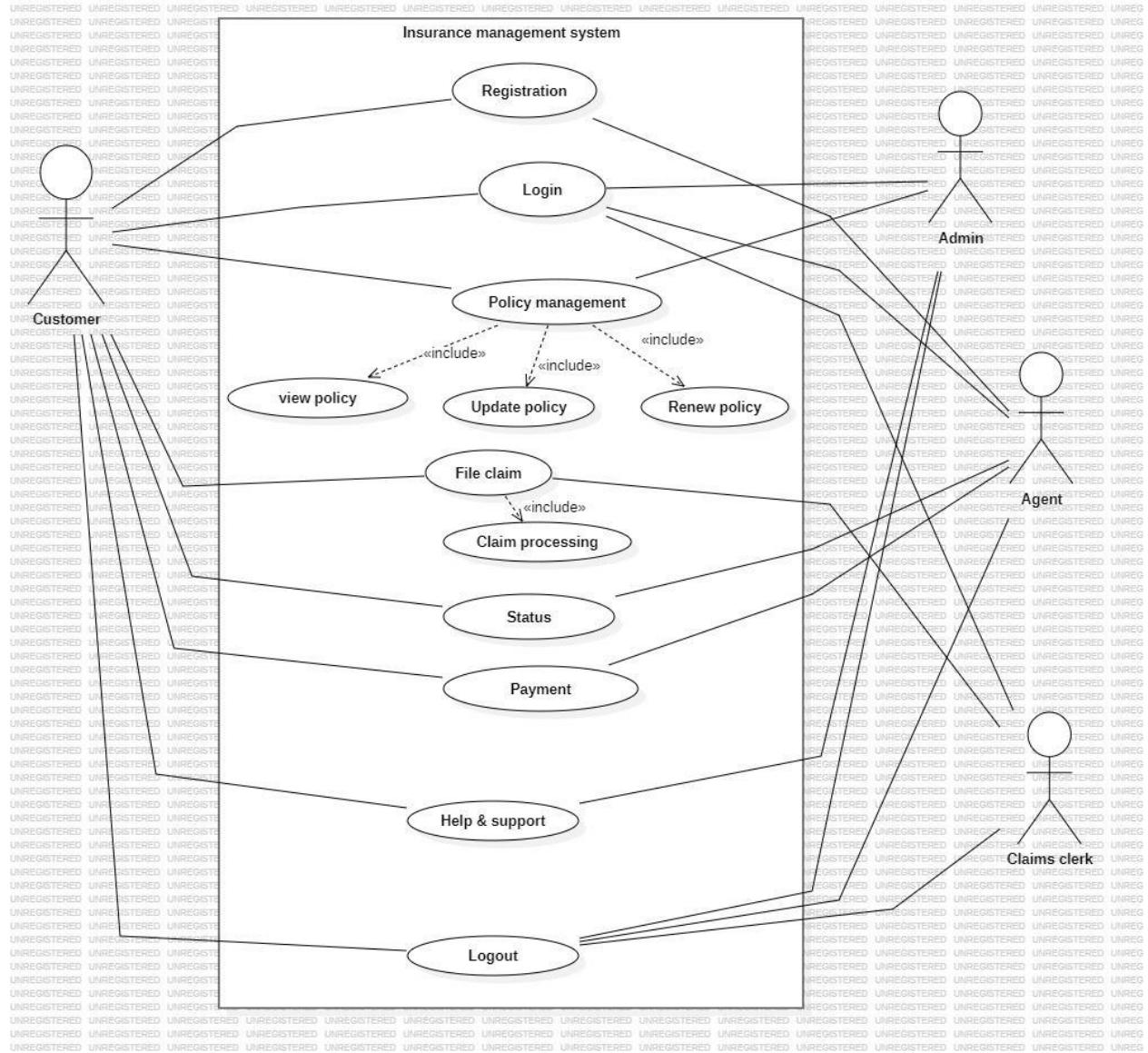
1. The system shall use a relational database management system (RDBMS), such as MySQL or PostgreSQL, to store and manage system data.

2. The database schema shall include tables for storing customer information, policy details, claim records, payment transactions, and user accounts.
3. The database shall implement appropriate data types, constraints, and indexing to ensure data integrity, performance, and efficient querying.
4. The database shall provide mechanisms for backup, restoration, and disaster recovery to protect against data loss.

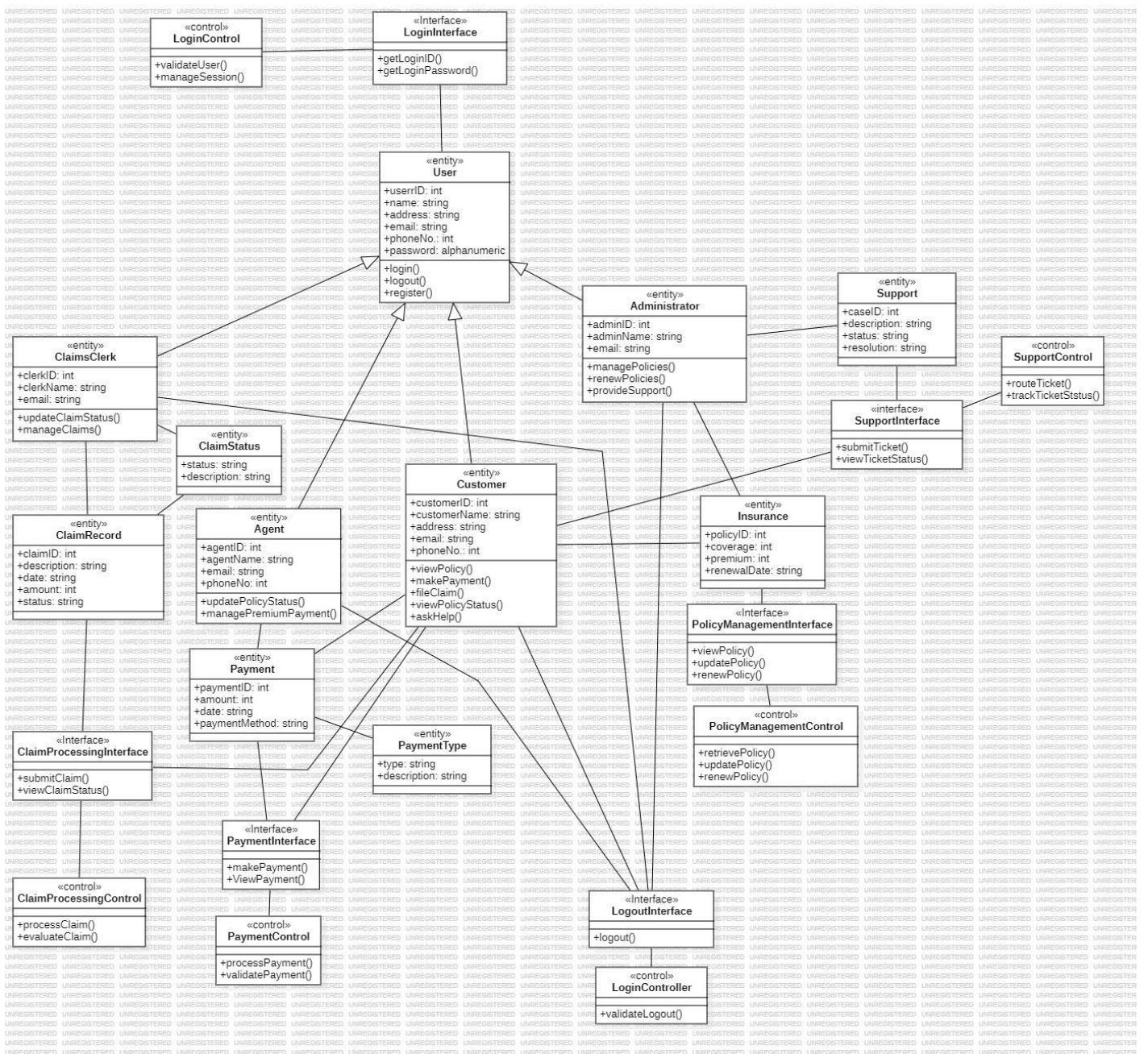
### **3.7 Other Requirements**

1. Internationalization and Localization: The system shall be designed to support multiple languages and regional settings to accommodate users from different geographical locations.
2. Accessibility: The system shall comply with Web Content Accessibility Guidelines (WCAG) to ensure that the user interface is accessible to users with disabilities.
3. Mobile Responsiveness: The system's user interfaces shall be designed to be mobile-responsive, ensuring a seamless experience for users accessing the system from various devices.

# Use Case Diagram



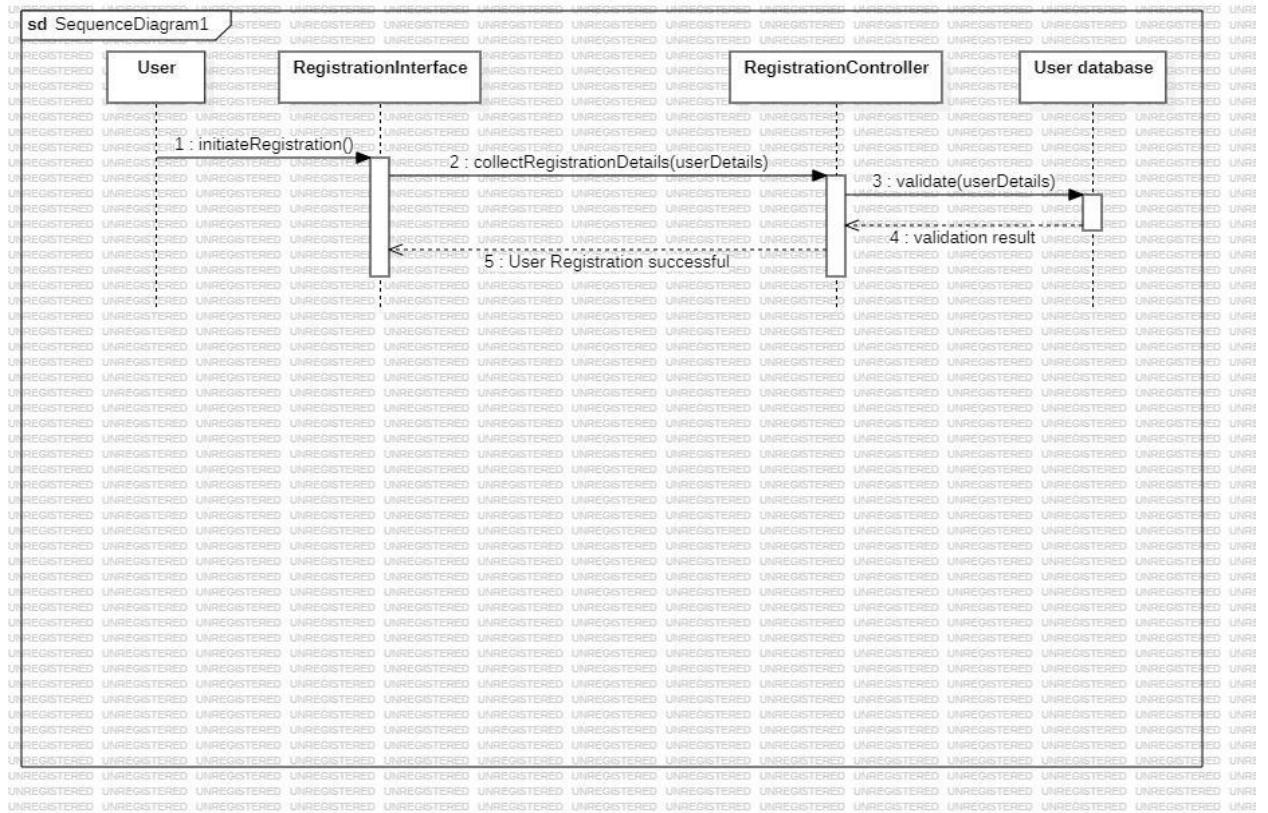
# Class Diagram



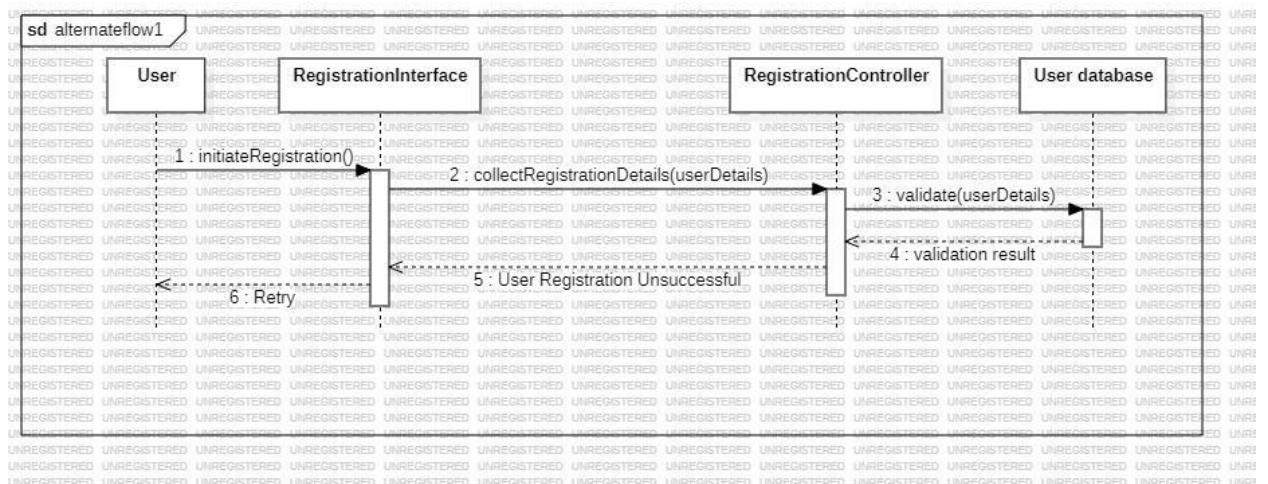
# Sequence Diagram

## 1. Registration

### 1.1 Basic Flow

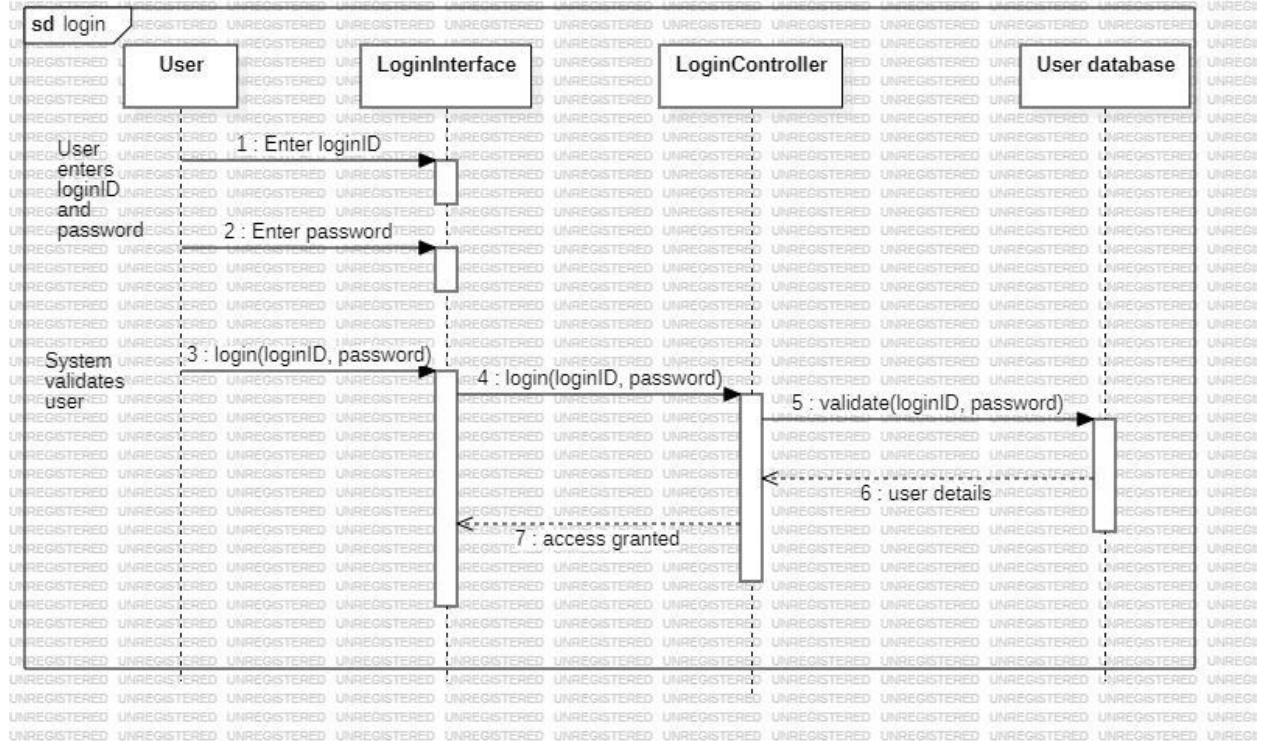


### 1.2 Alternative flow

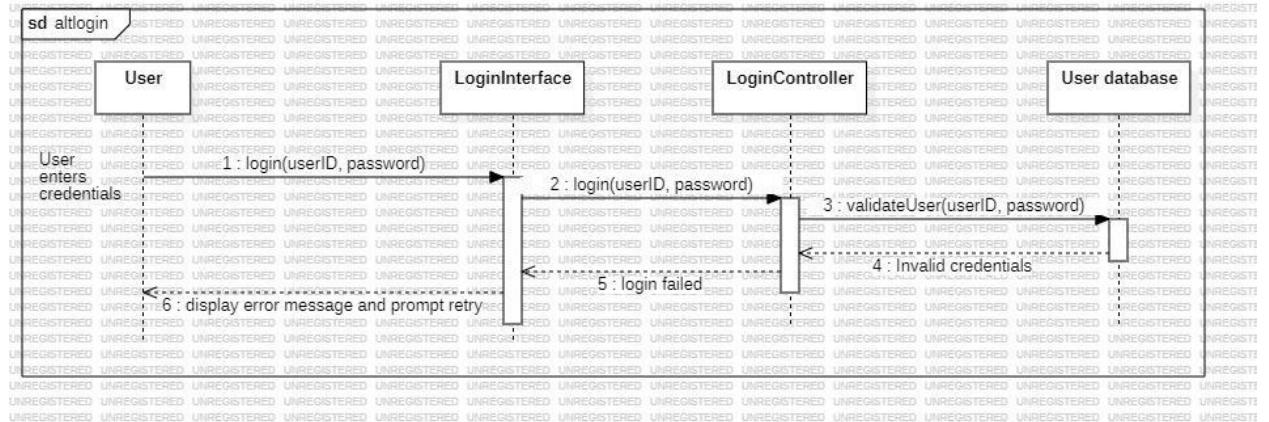


## 2. Login

### 2.1 Basic Flow



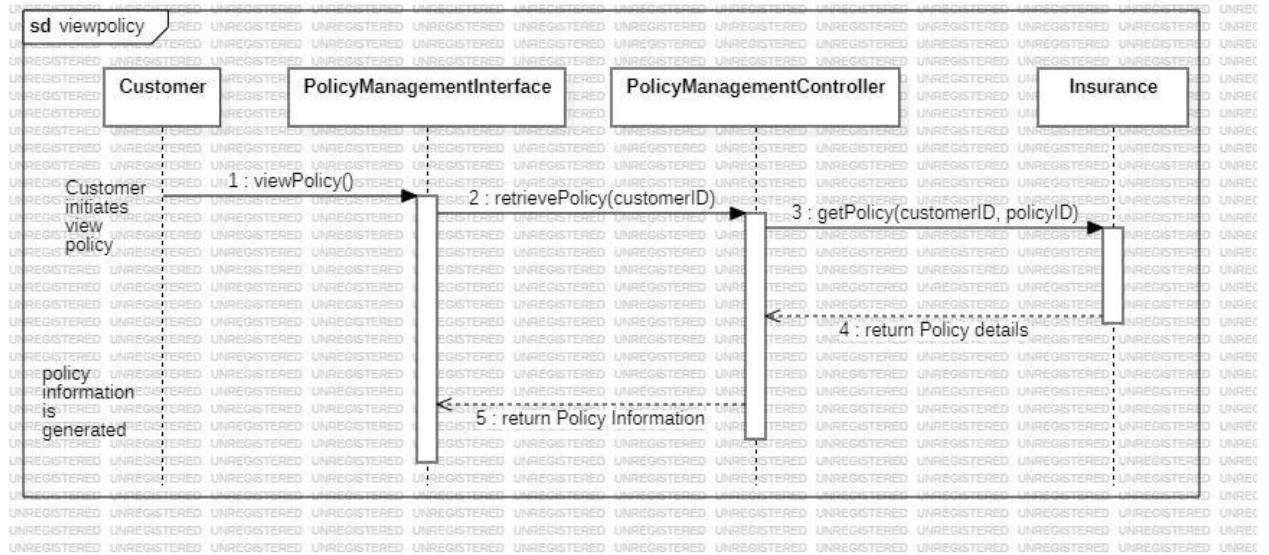
### 2.2 Alternate flow



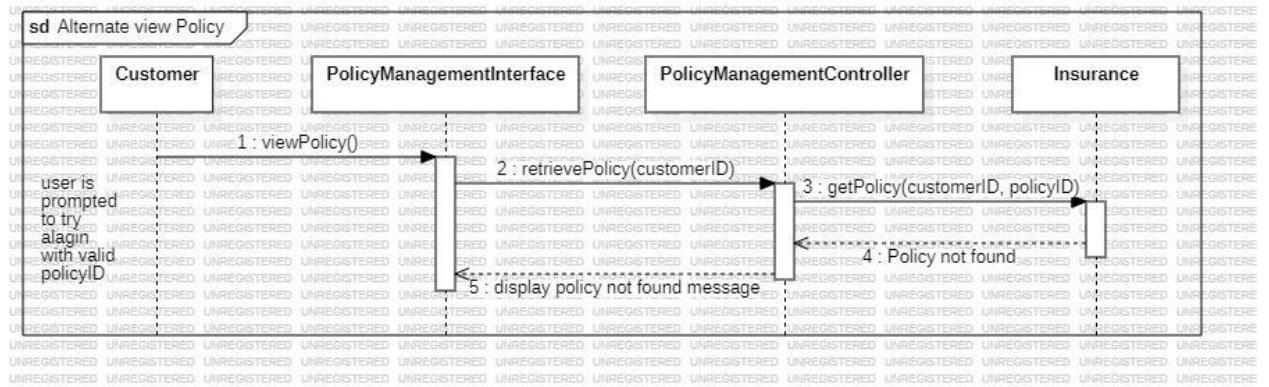
### 3. Policy Management

### 3.1 View Policy

### 3.1.1 Basic Flow

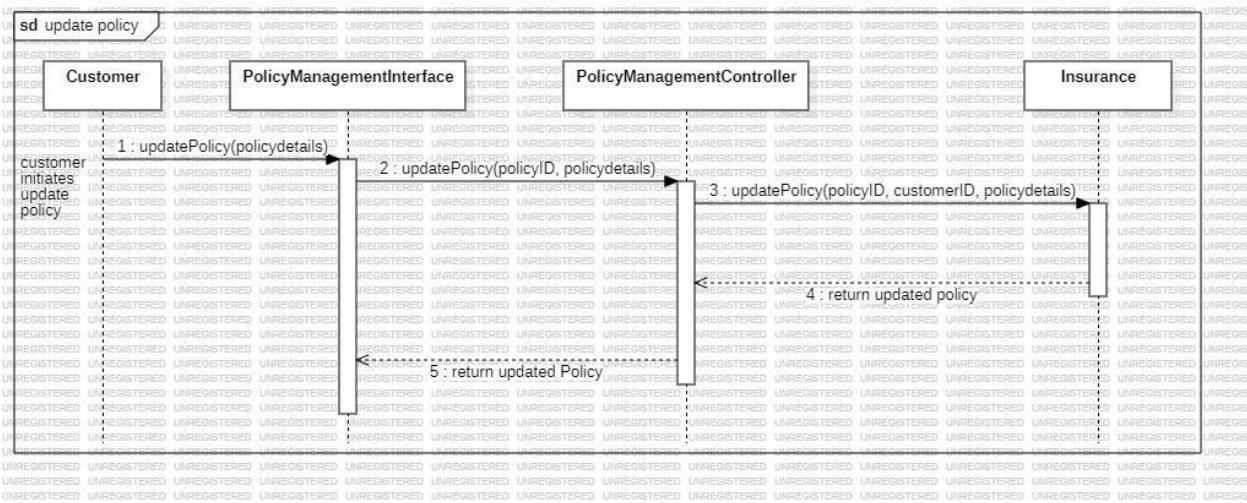


### 3.1.2 Alternate Flow

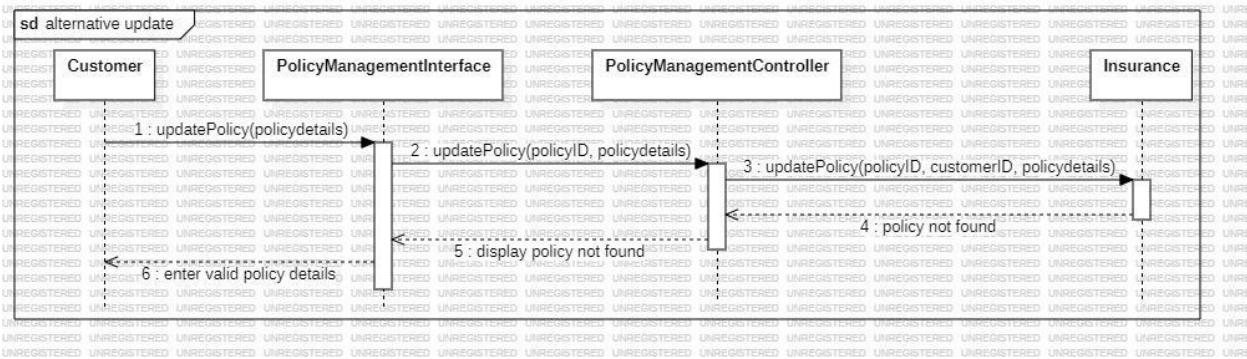


## 3.2 Update Policy

### 3.2.1 Basic flow

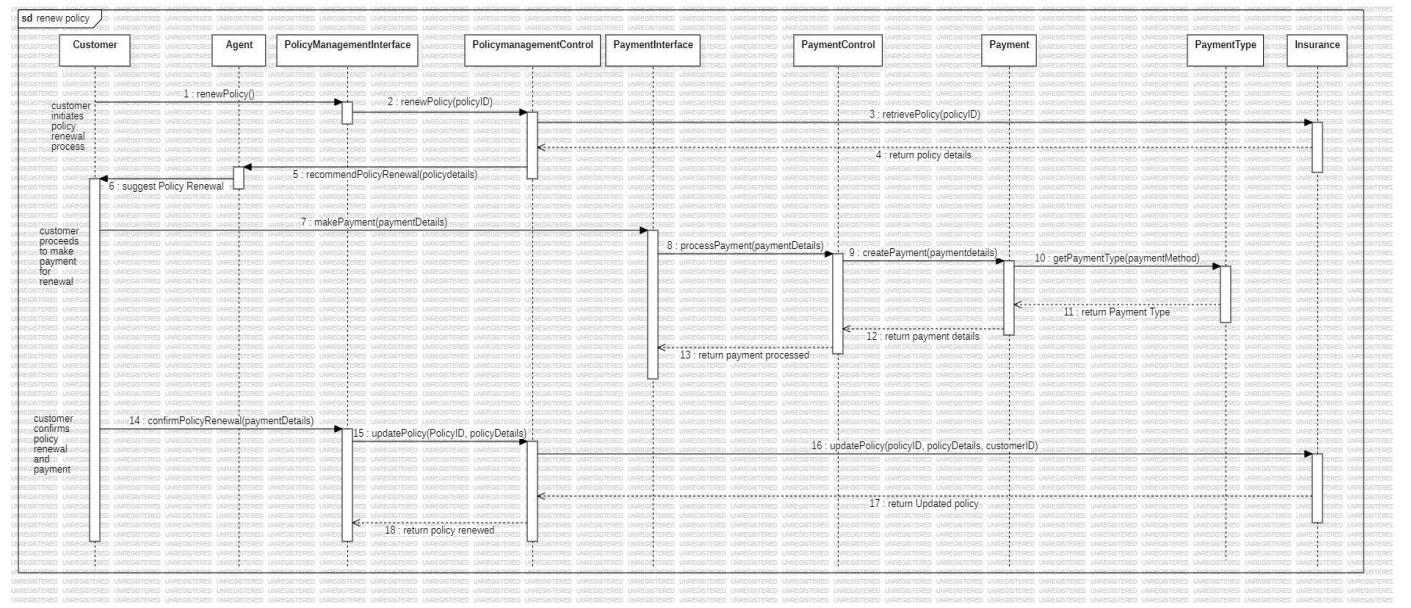


### 3.2.2 Alternate Flow

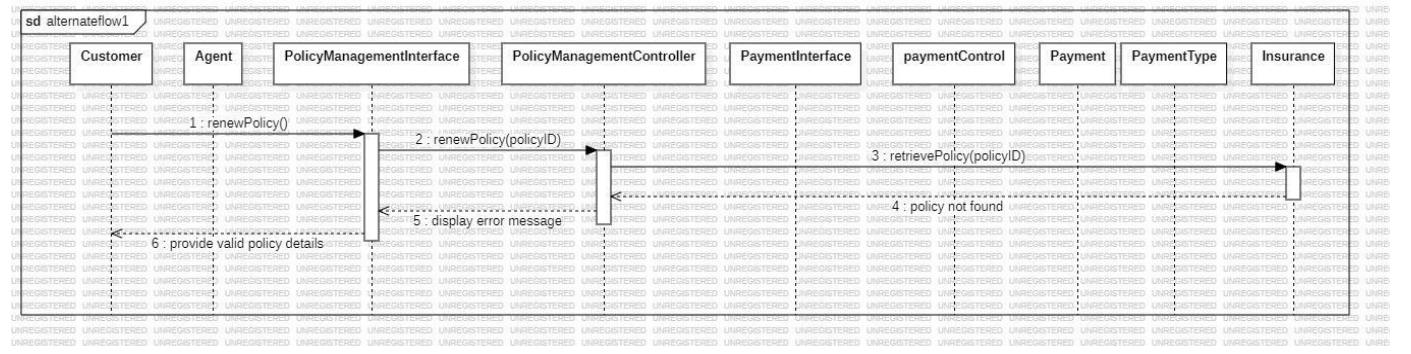


### 3.3 Renew Policy

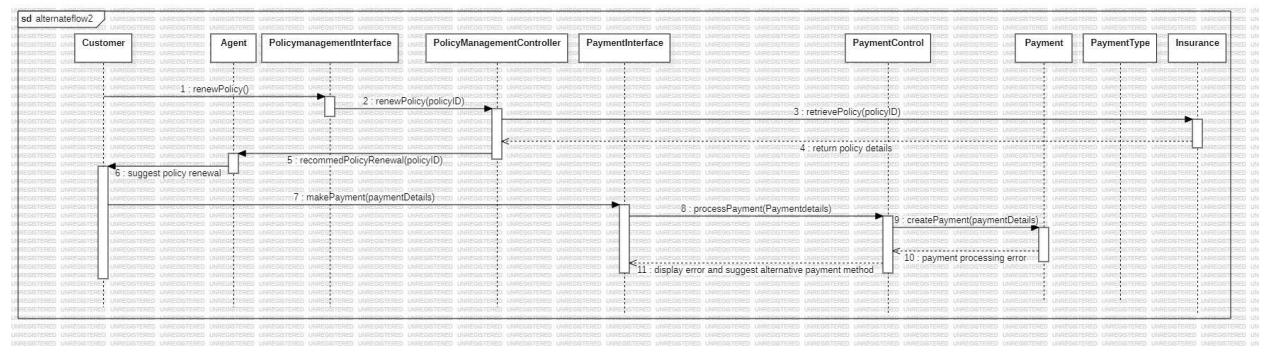
#### 3.3.1 Basic Flow



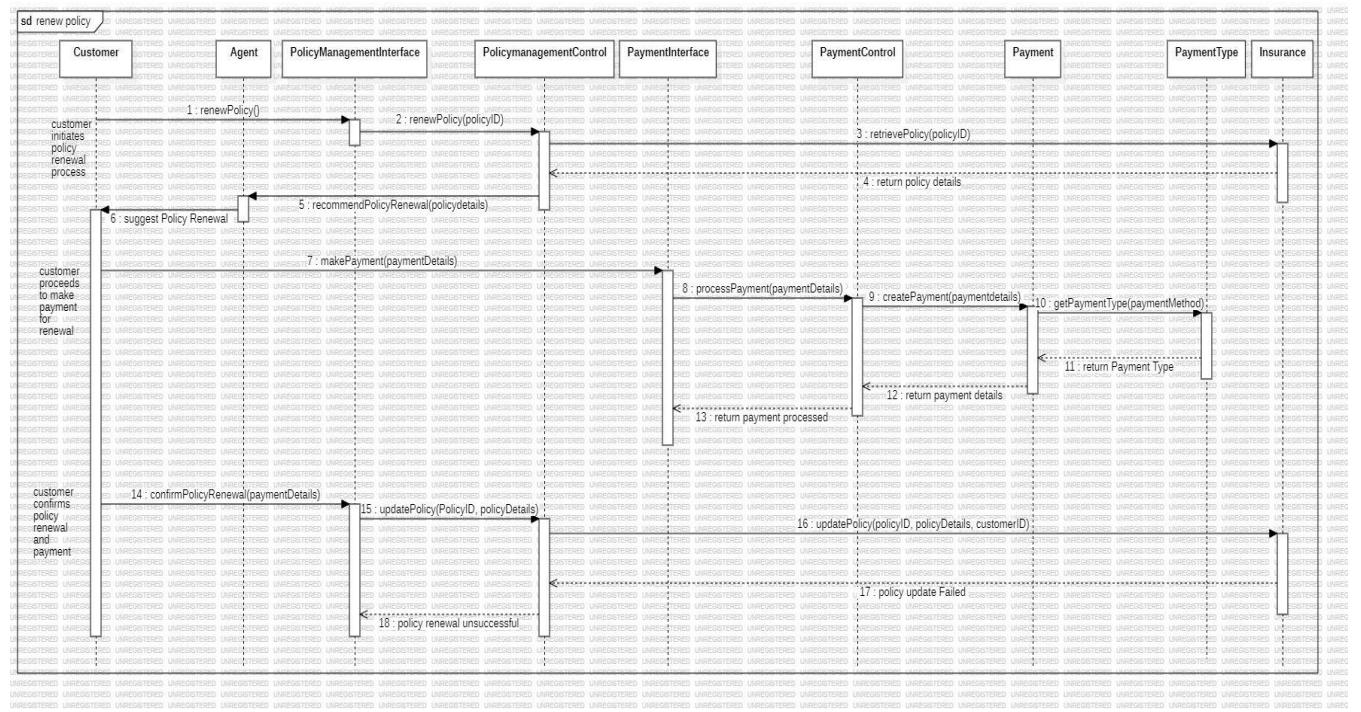
#### 3.3.2 Alternate Flow 1



#### 3.3.3 Alternate Flow 2

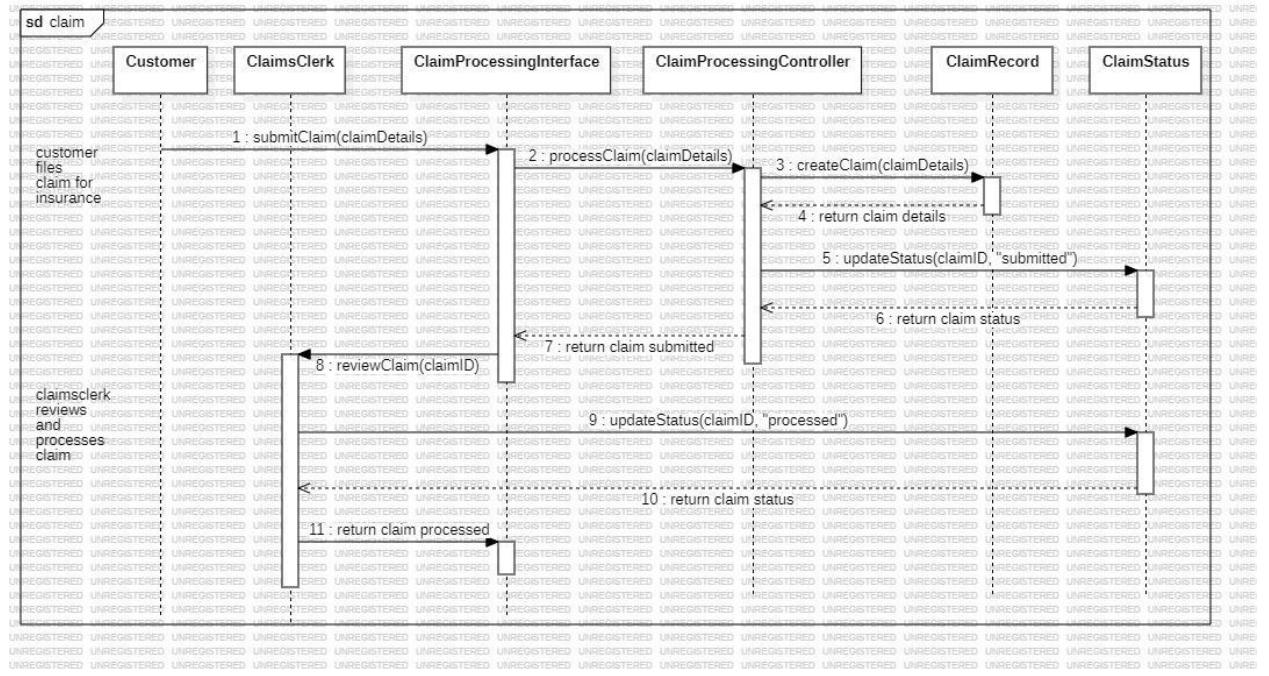


### 3.3.4 Alternate flow 3

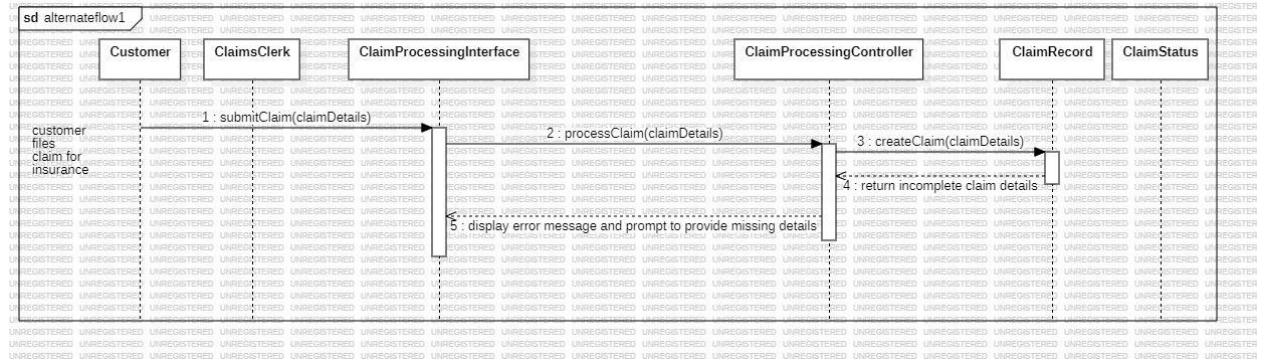


## 4. Claim Management

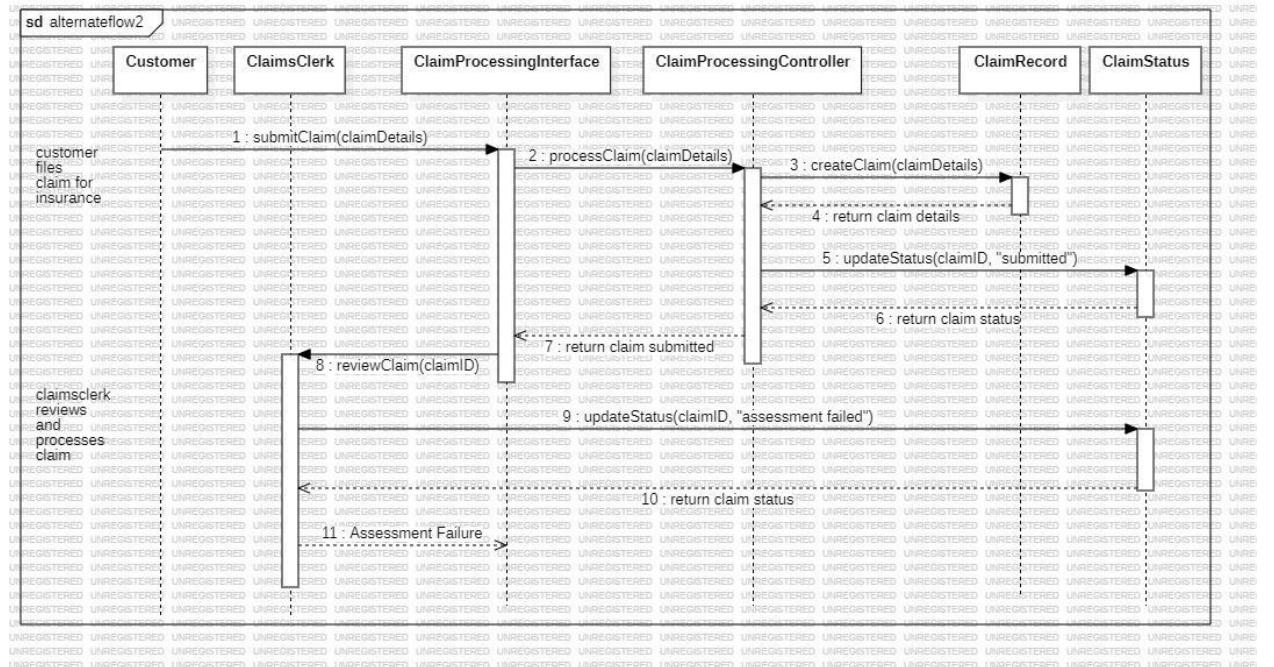
### 4.1 Basic Flow



### 4.2 alternate flow 1

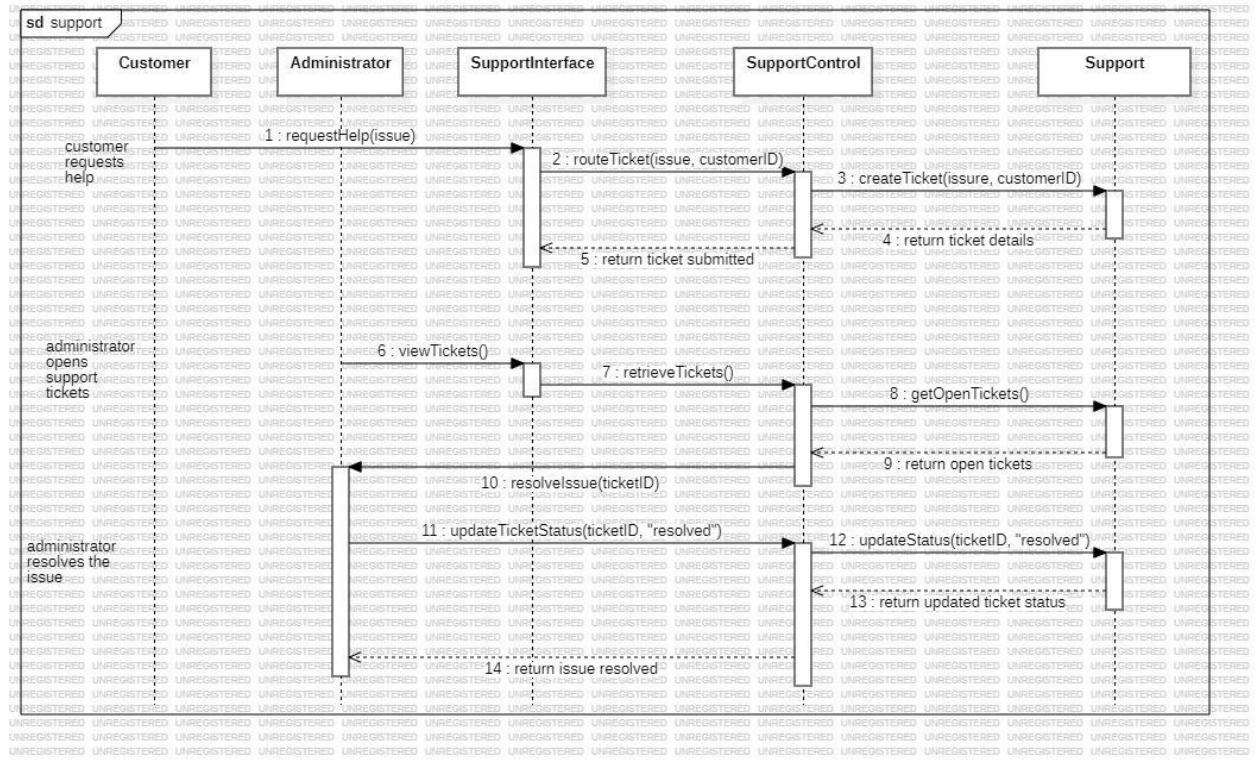


### 4.3 Alternate Flow 2

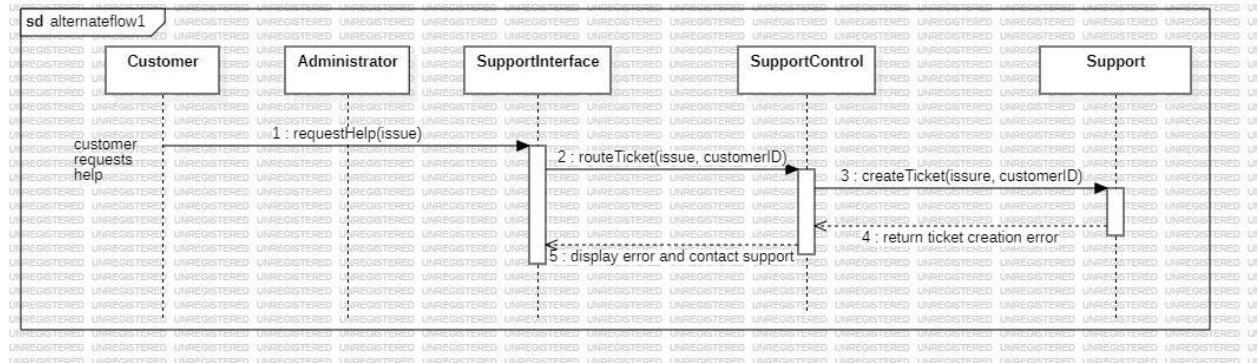


## 5. Help & support

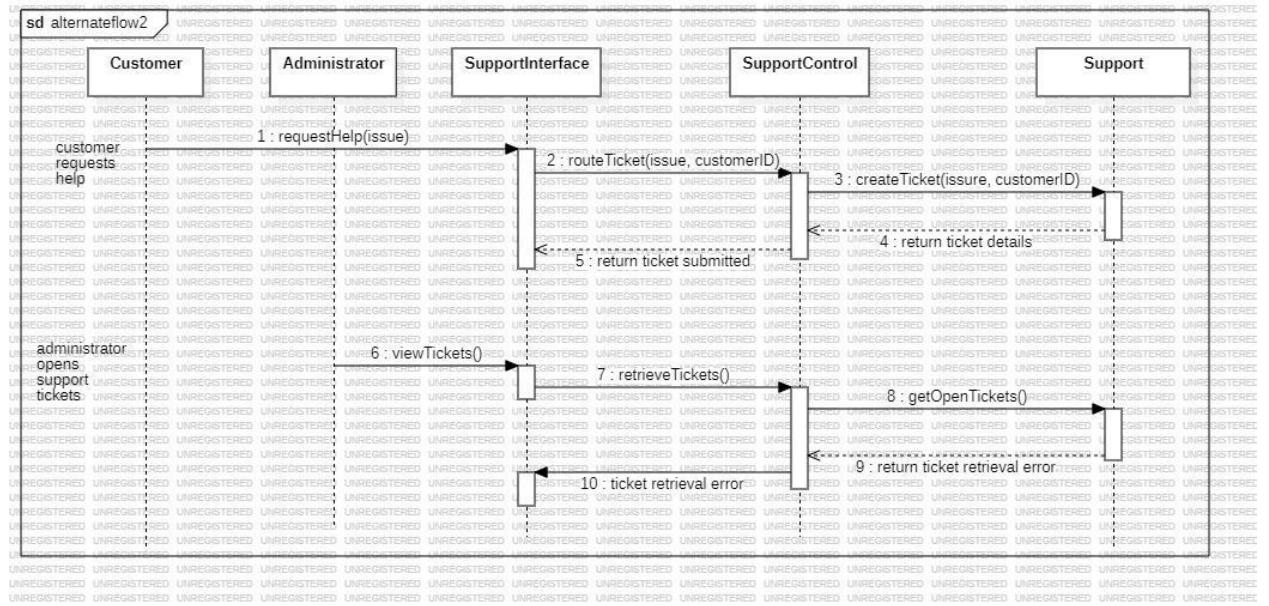
### 5.1 Basic Flow



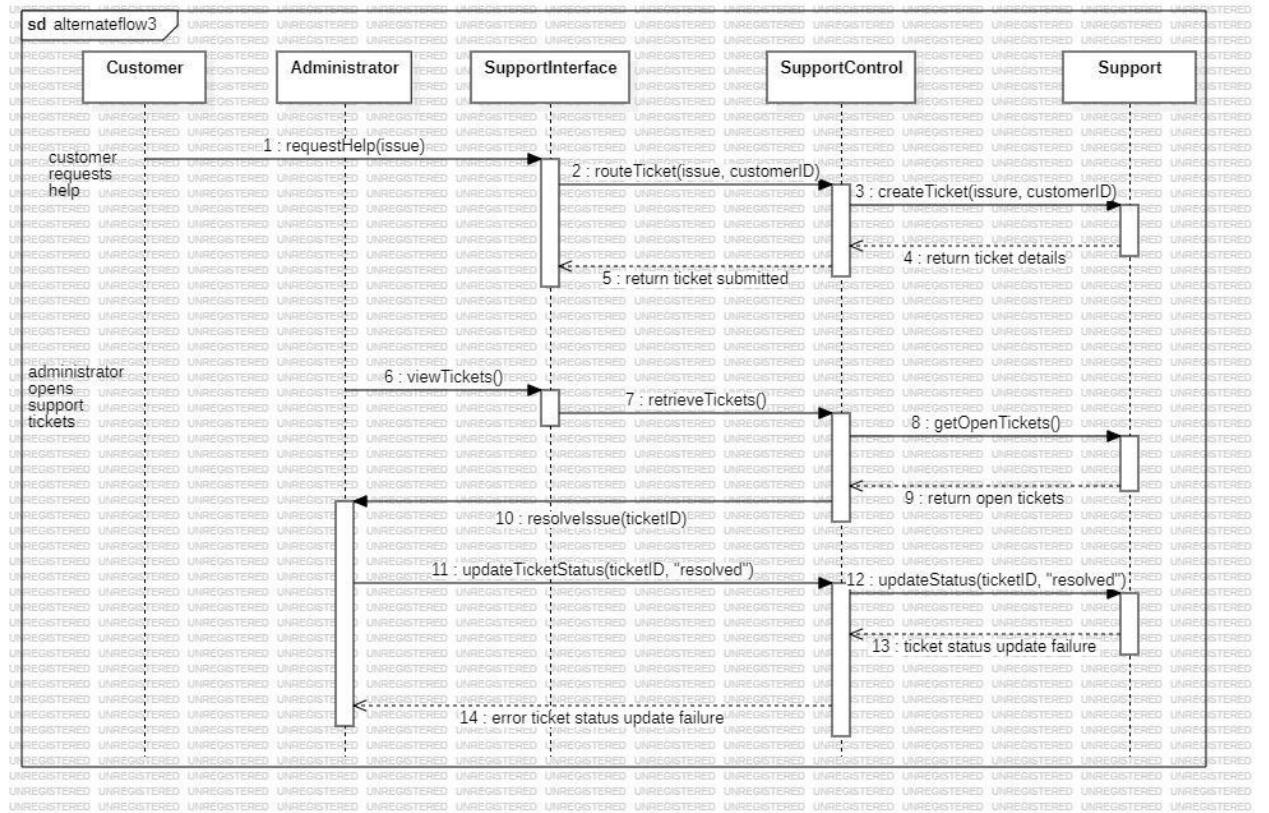
### 5.2 Alternate Flow 1



### 5.3 Alternate Flow 2

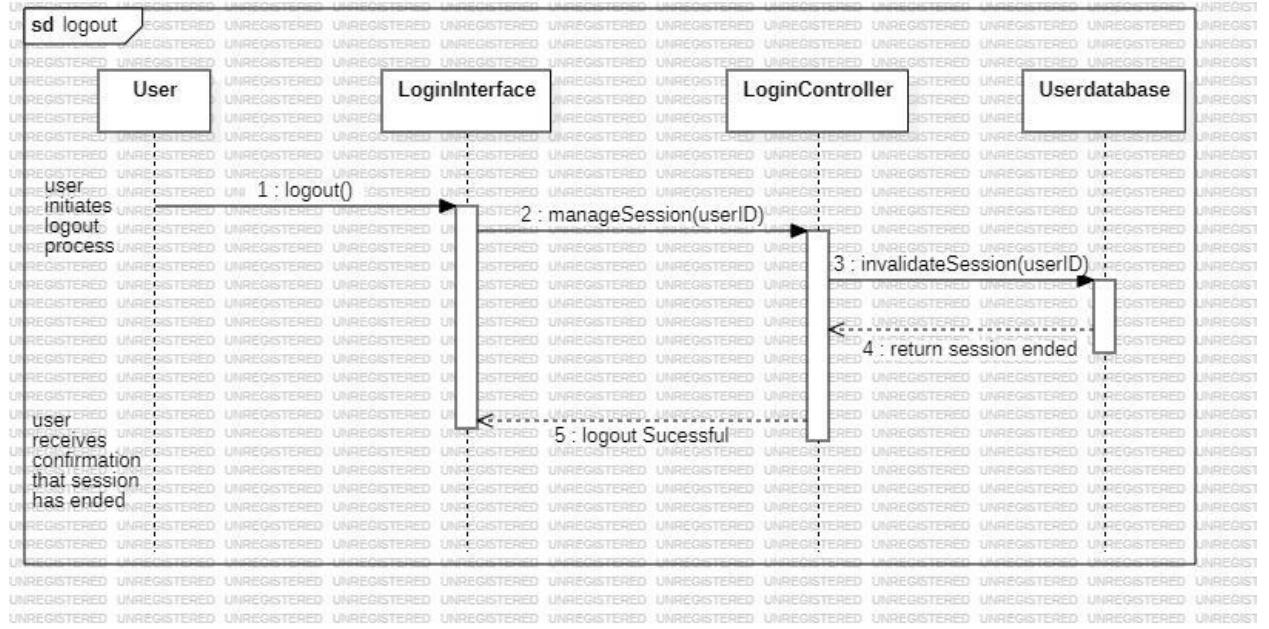


### 5.4 Alternate Flow 3

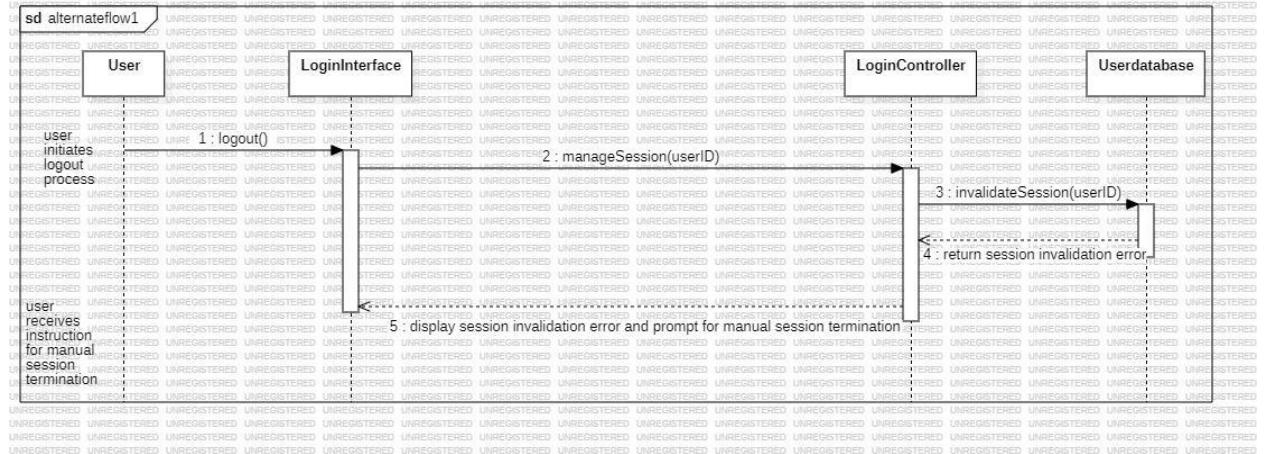


## 6. Logout

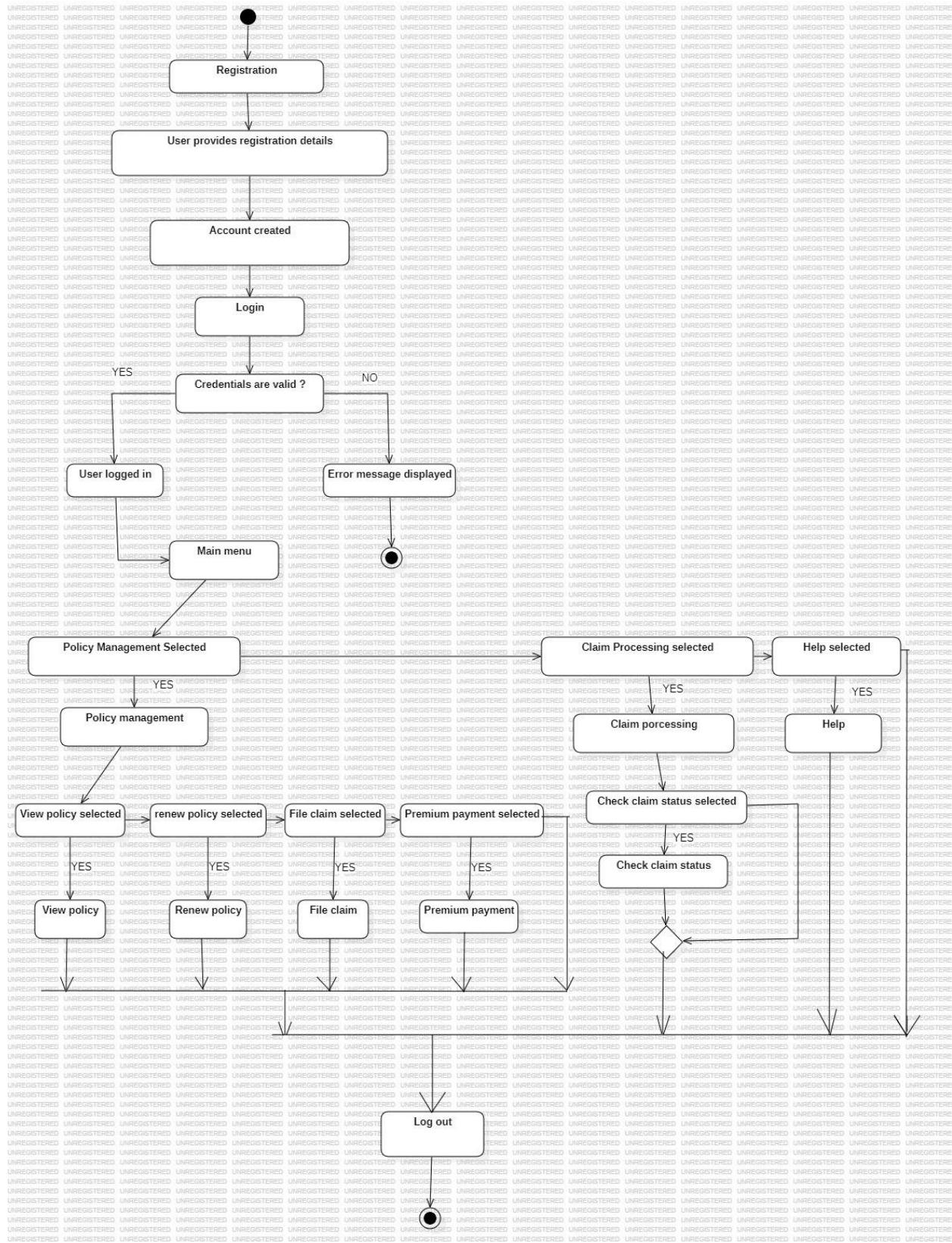
### 6.1 Basic Flow



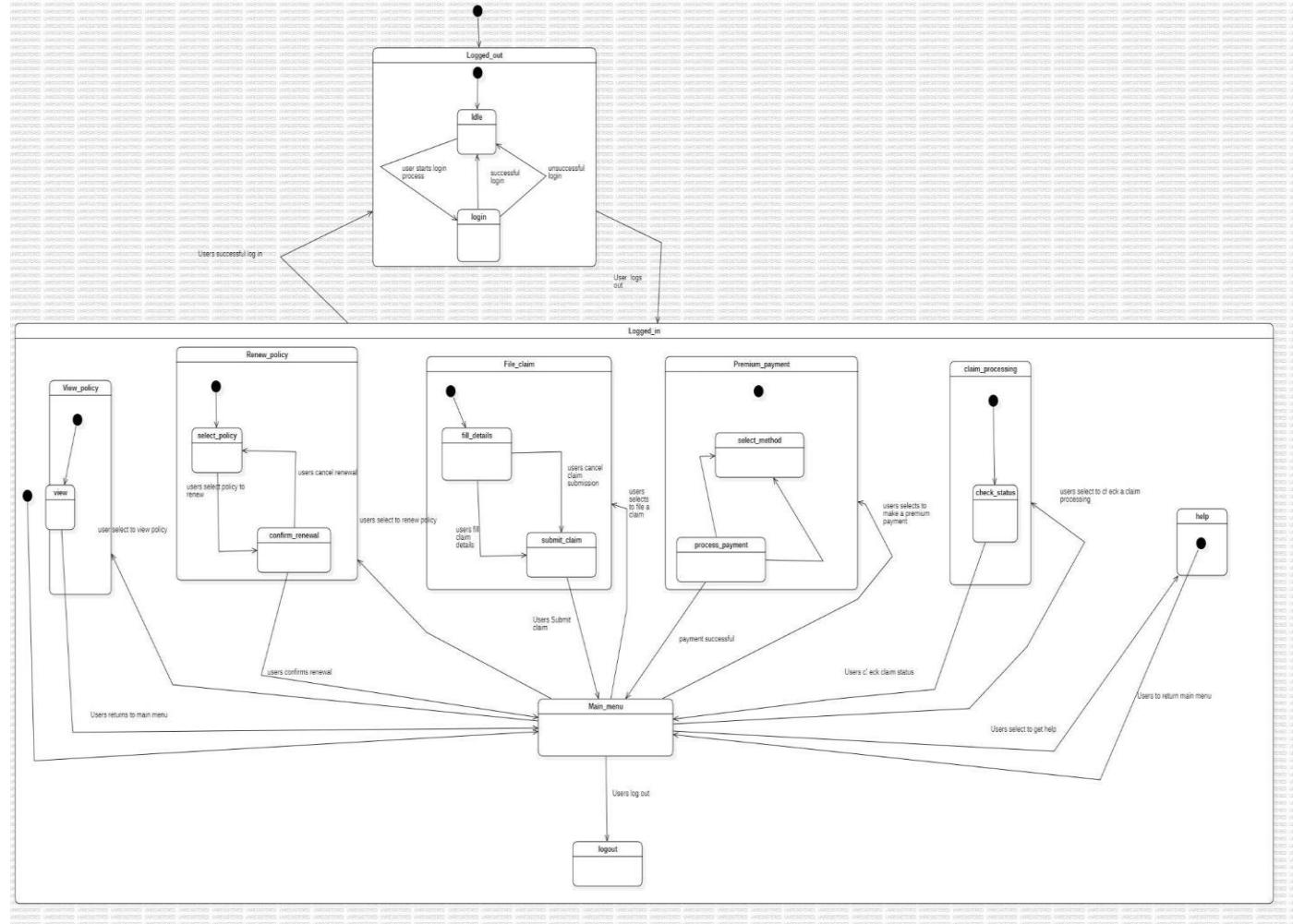
### 6.2 Alternate Flow



# Activity Diagram



# State Chart Diagram



## Test Case Matrix

### 1. Registration

Test Case ID	Scenario name and description	Input 1 User Name	Input 2 User dob	Input 3 User Phone number	Input 4 User Password	Expected Output	Remarks
TC_1	Scenario 1 - Registration	Valid	Valid	Valid	Valid	User is registered into the system	-
TC_2	Scenario 2 - registration alternative flow: missing values	n/a	valid	valid	valid	Missing information	User name is a required field.
TC_3		valid	n/a	valid	valid	Missing information	User date of birth is a required field
TC_4		valid	valid	n/a	valid	Missing information	Phone number is a required field
TC_5		valid	valid	valid	n/a	Missing Information	Password is a required field
TC_5	Scenario 3 - registration alternative flow: Duplicate User	valid	valid	valid	valid	User already exists	User already registered

## 2. Login

Test Case ID	Scenario Name and Description	Input 1 Login ID	Input 2 Password	Expected Output	Remarks
TC_1	Scenario 1 - Login	Valid	Valid	User is logged into the system	-
TC_2	Scenario 2 - Login alternative flow: Invalid credentials	Invalid	valid	LoginID is invalid	LoginID is not in the specified format
TC_3		Valid	Valid	LoginID is invalid	LoginID is not registered
TC_4		Valid	Invalid	LoginID or password invalid	Password doesn't match with database
TC_5		Invalid	Invalid	LoginID or password invalid	LoginID and password do not exist in the database.

### 3. Policy management

Test Case ID	Scenario Name and Description	Input 1 Policy ID	Input 2 Customer ID	Update Confirmed	Payment	Expected Output	Remarks
TC_1	Scenario 1: view policy	Valid	Valid	n/a	n/a	Policy details displayed correctly	-
TC_2	Scenario 2: View policy alternative flow: Policy not found	Invalid	valid/Invalid	n/a	n/a	Policy not found	Given Policy do not exist in the database
TC_3		Valid/Invalid	Invalid	n/a	n/a	Policy not found	Given Policy do not exist in the database
TC_4	Scenario 3: Update Policy	valid	valid	Yes	n/a	Policy is updated Successfully	-
TC_5	Scenario 4: Update policy alternative flow: Invalid data	Invalid	valid/Invalid	n/a	n/a	Invalid Policy number	Policy number not in specified format
TC_6		Valid/Invalid	Invalid	n/a	n/a	Invalid CustomerID	CustomerID do not exist in database
TC_7	Scenario 5: Update policy alternative flow: Update canceled	valid	valid	No	n/a	Update Canceled and Policy details are displayed	-
TC_8	Scenario 6: Update policy alternative flow: Policy not found	valid	valid/Invalid	n/a	n/a	Policy not found	-
TC_9	Scenario 7: Renew Policy	valid	valid	n/a	valid	Policy is Renewed	-
TC_10	Scenario 8: Renew policy alternative flow: Invalid	valid	valid	n/a	Invalid	Renew canceled and expired policy	Payment not received

	payment					details are displayed	
TC_11	Scenario 9: Renew Policy alternative flow: Policy not found	valid	n/a	n/a	n/a	Policy not found	Policy with specified policy number not found

#### 4. File claim

Test Case ID	Scenario description and	Input 1 PolicyNumber	Input 2 Claim details	Expected result	Remarks
TC_1	Scenario 1 - File Claim successful	valid	valid	Claim submitted successfully, claim number provided	-
TC_2	Scenario 2 - File claim alternative flow: Policy not found	valid	n/a	Policy not found	Policy with specified policy number do not exist.
TC_3	Scenario 3 - File claim alternative flow: Incomplete details	valid	invalid	Invalid claim details provided	-

#### 5. Status

Test Case ID	Scenario Name and description	Input 1 Claim number	Expected result	Remarks
TC_1	Scenario 1 - Status: Claim status	valid	Claim status is displayed	-
TC_2	Scenario 2 - Status alternative flow: Non existent claim	valid	Claim not found	Claim with specified Claim number is not found
		Invalid	Claim not found	Claim with specified Claim number not found.

## 1. Payment

Test CaseID	Scenario name and description	Input 1 Policy Number	Input 2 Customer ID	Expected Output	Remarks
TC_1	Scenario 1 - Payment: successful	valid	valid	Payment processed successfully	-
TC_2	Scenario 2 - Payment: Unsuccessful	valid	valid	Error message displayed	Error occurred due to payment failure
TC_3	Scenario 3 - policy not found	invalid	valid/invalid	Policy not found	Invalid Policy number

## 2. Help and Support

Test CaseID	Scenario name and description	Input 1 CustomerID	Input 2 Help request details	Expected Output	Remarks
TC_1	Scenario 1 - User help request: submitted	valid	valid	Help request submitted successfully, ticket reference number provided	-
TC_2	Scenario 2 - User help request: Incomplete details	valid	invalid	Error message displayed	Prompting user to provide missing information