Table of Contents

[1. INTRODUCTION 4](#_Toc196337597)

[1.1 Project Overview 4](#_Toc196337598)

[1.2 Purpose 4](#_Toc196337599)

[1.3 Goals and Objectives 4](#_Toc196337600)

[1.4 Scope 5](#_Toc196337601)

[1.4.1 User Roles and Access 5](#_Toc196337602)

[1.4.2 User Management Module 6](#_Toc196337603)

[1.4.3 Policy Management Module 7](#_Toc196337604)

[1.4.4 Quotes Management module 8](#_Toc196337605)

[1.4.5 Payment Module 9](#_Toc196337606)

[1.4.6 Claims Management Module 10](#_Toc196337607)

[1.4.7 Administrator Module 11](#_Toc196337608)

[1.5 Document Overview 12](#_Toc196337609)

[1.6 Definitions and Acronyms 13](#_Toc196337610)

[1.6.1 Definitions 13](#_Toc196337611)

[1.6.2 Acronyms 13](#_Toc196337612)

[2.SYSTEM ARCHITECTURE 14](#_Toc196337613)

[2.1 Decomposition Description 14](#_Toc196337614)

[3.DATA DESIGN 16](#_Toc196337615)

[3.1 Entity Relationship Diagram (ERD) 16](#_Toc196337616)

[3.2 Data Description 17](#_Toc196337617)

[3.3 DATA DICTIONARY 19](#_Toc196337618)

[4.Human Interface Design 25](#_Toc196337619)

[4.1 Home page 25](#_Toc196337620)

[4.2 Administrator interactivity 26](#_Toc196337621)

[4.3 Customer interactivity 26](#_Toc196337622)

[4.4 Underwriter interactivity 27](#_Toc196337623)

LIST OF FIGURES

[Figure 1: Use Case Diagram of the Safe-Nest Insurance Management System 15](#_Toc196337624)

[Figure 2:Figure shows ERD 16](#_Toc196337625)

[Figure 3:Figure shows our homepage for the system. 25](#_Toc196337626)

[Figure 4: Figure shows the Administrator Dashboard 26](#_Toc196337627)

[Figure 5: Figure shows the Customer Dashboard 26](#_Toc196337628)

[Figure 6: Figure shows the underwriter dashboard 27](#_Toc196337629)

LIST OF TABLES

[Table 1: Table shows definitions of certain terms. 14](#_Toc196337675)

[Table 2: Table shows full forms of acronyms used in the document. 14](#_Toc196337676)

[Table 3: Table shows the Data Description 18](#_Toc196337677)

[Table 4: Table shows description of user attributes. 20](#_Toc196337678)

[Table 5: Table shows description of policy attributes. 21](#_Toc196337679)

[Table 6:Table shows description of policy application attributes. 21](#_Toc196337680)

[Table 7: Table shows description of claim application attributes 22](#_Toc196337681)

[Table 8:Table shows description of approved policy attributes 23](#_Toc196337682)

[Table 9: Table shows description of payment attributes 23](#_Toc196337683)

[Table 10: Table shows description of quotes attributes 24](#_Toc196337684)

1. INTRODUCTION

## 1.1 Project Overview

In today's increasingly digital world, the traditional methods used by many insurance companies for managing policies, processing claims, and interacting with customers are often inefficient, time-consuming, and prone to errors. Customers face challenges in accessing policy information, making payments, and filing claims due to a lack of centralized, user-friendly platforms. Meanwhile, underwriters and administrators struggle with managing large volumes of data, maintaining transparency, and ensuring timely service delivery using outdated or fragmented systems.

There is a critical need for a robust, modern, and scalable Insurance Management System that streamlines the entire insurance lifecycle from policy creation and customer registration to claims processing and analytics. The system should provide tailored access to different actors (customers, underwriters, administrators), automate key operations, enhance user experience, and provide real-time insights into business performance.

Customers will require access to their personal profile information, a list of available policies with detailed terms, their active policy records, payment history and transaction records, claim application statuses and ability to submit quotes and claims.

Underwriters will require access to claims submitted by the customers, customer policy applications, approval workflows, tools to create, edit and delete policies

Administrators will require access to logs of all system activities, upload underwriters, view all registered users grouped by role, activate, deactivate or delete user accounts, system analytics and performance reports.

This system aims to design and develop a web-based insurance management platform that addresses these challenges by integrating core functionalities into one centralized solution, enhancing operational efficiency, and improving overall service delivery.

## 1.2 Purpose

The Safe-Nest Insurance Management System will be designed to provide a centralized, user friendly and secure platform that facilitates efficient management of insurance operations for customers, underwriters and administrators by streamlining key processes.

## 1.3 Goals and Objectives

The following section outlines the goals and objectives of the Safe-Nest Insurance Management System:

1. To allow customers to browse, apply for and manage insurance policies.
2. To allow customers to make payments and view their transaction history.
3. To allow customers to file claims and quotes
4. To provide real-time status updates on submitted claims and quotes.
5. To enable automated email notifications for payment confirmations, claims submissions and updates.
6. To allow underwriters to create, edit and manage insurance policies.
7. To allow underwriters to approve or reject customer claims applications.
8. To allow an administrator to register underwriters and manage all users.
9. To allow an administrator to view performance reports and analytics.

## 1.4 Scope

### 1.4.1 User Roles and Access

The Safe-Nest Insurance Management System will provide role-based (customer, underwriter, administrator) access to specific functionalities ensuring that each user group can securely interact with the features relevant to their responsibilities as follows:

Customer

1. The customer will have full access to the customer dashboard.
2. The customer will be able to browse and apply for policies.
3. The customer will be able to make payments and view transaction history.
4. The customer will be able to view and manage their profile.
5. The customer will be able to view their active policies.
6. The customer will be able to file claims and quotes with the ability to view their status.

Underwriter

1. The underwriter will have full access to the underwriter dashboard.
2. The underwriter will be able post, edit and delete policies.
3. The underwriter will be able to view and approve customer applications for policies, claims and quotes.
4. The underwriter will be able to view and manage their profile.

Administrator

1. The administrator will have full access to the administrator dashboard.
2. The administrator will be able to register underwriters and manage all users.
3. The administrator will be able to view performance reports and analytics.

The Safe-Nest Insurance Management System will also be divided into modules as follows:

### 1.4.2 User Management Module

Overview

The User Management Module primarily handles access control. It handles user registration, login, authentication, and role-based access. It ensures users can interact with the system securely based on their assigned roles: Customer, Underwriter, or Administrator.

Key Features

1. User Registration (with role assignment)
2. Secure Login/Logout
3. Role-Based Access Control (RBAC)
4. Profile Management
5. Password Reset
6. Account Activation/Deactivation (Admin-controlled)

Core Functionalities

1. **Registration & Login**New users can register with email, password, and role.   
   System verifies credentials during login.  
   System hashes all the passwords.
2. **Role Assignment & Permissions**Customers: Access policies, make claims, payments, can edit profile.  
   Underwriters: Review applications and approves claims, handles quotes generation.  
   Administrator: Upload underwriters, generate reports, activate and deactivate user account and manage system settings.
3. **Profile Management**Clients can update personal information like name, contact, etc.
4. **Password Reset**Clients can reset passwords using emails
5. **Account Control**Admins can suspend, reactivate users.
6. **Secure access**Password hashing for all users.

### 1.4.3 Policy Management Module

Overview

 The policy Management Module handles the lifecycle of all insurance policies offered by the organization. It serves both internal users (Admins / Underwriters) who manage policies, and approve policies and the external users (Customers) who view and apply for those policies.

Key Features

1. Create Policy
2. Update Policy
3. View Policy
4. Apply for policy
5. Policy Approval
6. Terminating a policy

Core Functionalities

**1.Create policy**

Admins and underwriters can create a new policy.

**2.Update policy**

The underwriter can modify the policy details.

Underwriter can Change the status on whether it is pending or approved.

**3.View policy**

Public listing of all the active policies.

Customers can view detailed information about each policy before applying.

**4.Apply for policy**

Customers can apply a selected policy depending on its status.

**5.Policy Approval**

Underwriters review incoming applications.

Underwriters can approve or reject based on the customer profile and the policy rules.

**6.Terminating**

The underwriter can terminate a policy from the existing policies.

### 1.4.4 Quotes Management module

Overview

The Quotes Management Module serves as the customer’s first interaction with the Insurance Management System. It allows potential policyholders to request insurance quotes based on their personal and risk-related information. Underwriters can then review, generate, modify, and approve these quotes before they are finalized and offered to the customer. Once a quote is approved, the customer can proceed to convert it into an official policy application.

Key features

1. Review a quote
2. Generate a quote
3. Approve a quote
4. View a quote
5. Request for a quote

Core Functionalities

**1.Review a quote**

Underwriters access all the quotes requests.

The underwriter can view the quotes of all the customers

**2. Generate a quote**

The underwriter generates the quote

**3.Approve a quote**

 Underwriters finalize and approve the quote.

**4.View a quote**

Customers can view all the quotes and the status of each quote.

The customer can view the

**5.Request for a quote**

Customers can request a quote

### 1.4.5 Payment Module

Overview

The Payment module is responsible for managing all premium-related financial transactions within the Insurance Management System. It ensures secure payment processing, tracks payment history and supports integration with third-party payment APIs. Through this module, policyholders can pay premiums, view their transaction history, and receive confirmations. Administrators and finance personnel can manage payments, generate reports, and reconcile transactions. The module ensures financial transparency, accuracy, and ease of payment of users.

Key features

1. Payment of premiums
2. Payment history tracking
3. Integration with third-party Payment APIs
4. Transaction Status tracking
5. Automated reminders and notifications
6. Receipt generation
7. Reports and Analytics
8. Role Based Access

Core Functionalities

**1.Payment of Premiums**

* The system calculates premium amounts based on selected policy terms.
* Policyholders pay via card, mobile money, or bank transfer.
* Administrators can manually record offline payments.
* The system updates status upon successful payment.

**2.Payment history tracking**

* Policyholders and administrators view the date, amount, method, and status of all payments.
* Administrators and the finance team download payment history for record keeping.

**3.Integration with third-party Payment APIs**

* Administrators integrate the system with gateways like Visa, Mobile Money.

**4.Transaction Status tracking**

* Policyholders and administrators view records statuses such as pending, successful, failed, or refunded.

**5. Automated reminders and notifications**

* Policyholders receive notifications of upcoming due dates or failed payments.
* System sends notifications to policyholders.

**6. Receipt generation**

* Policyholders view and download PDF receipts generated immediately after successful payment.
* Policyholders receive emails and are available in the user dashboard.

**7. Reports and Analytics**

* Administrators view overdue payments and user payment behaviors.

**8. Role Based Access**

* Policyholders make their own payments and view their receipts.
* Administrators view/manage all payments, process refunds, generate reports.

### 1.4.6 Claims Management Module

Overview

The Claims Management Module handles all operations related to insurance claims. Customers can file claims for insured events, attach evidence, and monitor claim status. Underwriters review and decide on these claims, while admins oversee and audit claim activities.

Key Features

1. Claim Submission by Customers
2. Document Upload Support
3. Claim Status Tracking
4. Claim Review & Decision by Underwriters
5. Audit Trail for Admin Oversight
6. Notifications for Updates

Core Functionalities

1. **Claim Filing**Customers fill out a claim form tied to a specific policy.  
   Option to upload photos, receipts, or any proof documents.
2. **Claim Review Process**Underwriters see incoming claims.  
   Can approve, reject, or request more info.
3. **Status Updates**Claims go through stages: Pending → Under Review → Approved/Rejected.  
   Customers can see real-time updates.
4. **Administrative Oversight**Admins view all claims.  
   Can monitor frequency, status breakdowns and perform audits.
5. **Automated Alerts**Email notifications when a claim is filed, reviewed, or updated.

1.4.7 Administrator Module  
Overview

The Admin Dashboard serves as the central control panel for administrators to monitor, manage and configure all core aspects of the Safe-Nest Insurance Management system. It provides real-time access to user data, policy statistics, financial records, Logs and Audit trails.

Core Features

1. User Management:
2. Reports and Analytics.
3. Logs and Audit trails.

Core Functionalities

1**. User Management**

* View all registered users grouped by role, e.g. customers and underwriters.
* Add new users manually, for example, underwriters.
* Activate, deactivate or delete user accounts.
* Set login credentials of the underwriters.
* Audit user activities and access logs.

2. **Reports and Analytics**

* View real-time charts and graphs of key performance indicators:
* Number of active policies
* Total collected premiums
* Number of claims submitted and approved.
* Compare performance metrics over time, for example, monthly

3. **Logs and Audit Trails**

* Full activity logs for all admin and user operations.
* Searchable by user ID, action type, or date/time.

## 1.5 Document Overview

This document has been arranged in chapters that describe the entire system requirements.

Chapter 1: Introduction

This chapter identifies the project overview, purpose, goals and objectives plus scope of our SDD.

Chapter 2: System Architecture

This chapter has the Decomposition Description that has use cases for our SafeNest Insurance Management System.

Chapter 3: Data Design

This chapter has the entity relationship diagram that highlights the relevant classes in our system. It also has the Data Description that explains how the information domain of the system is transformed into data structures.

Chapter 4: Human Interface Design

This chapter has mockups of our User Interface.

## 1.6 Definitions and Acronyms

### 1.6.1 Definitions

Table 1: Table shows definitions of certain terms.

|  |  |
| --- | --- |
| Term | Definition |
| Entity Relationship Diagram | A graphical representation that illustrates the relationship between entities in the database. |
| Use Case Diagram | A visual representation of interactions between the system and actors within the system. |

### 1.6.2 Acronyms

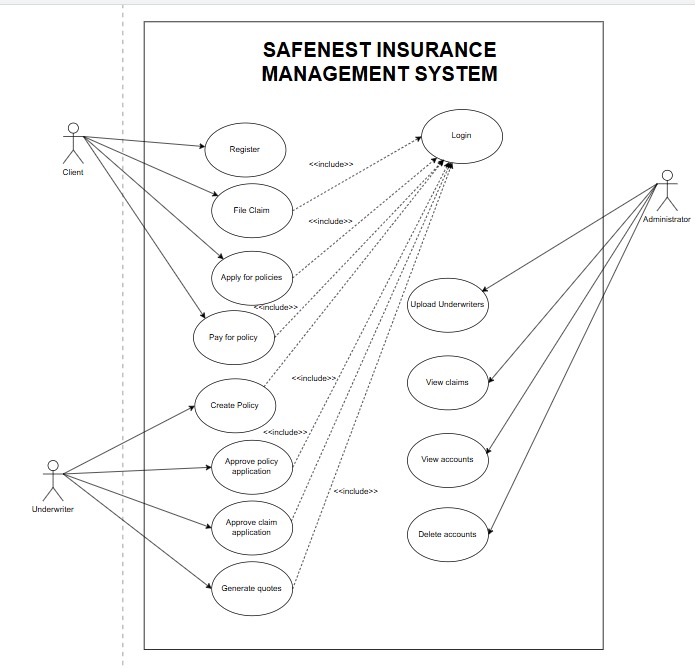
Table 2: Table shows full forms of acronyms used in the document.

|  |  |
| --- | --- |
| Abbreviation | Meaning |
| SDD | Software Design Document |

# 2.SYSTEM ARCHITECTURE

## 2.1 Decomposition Description

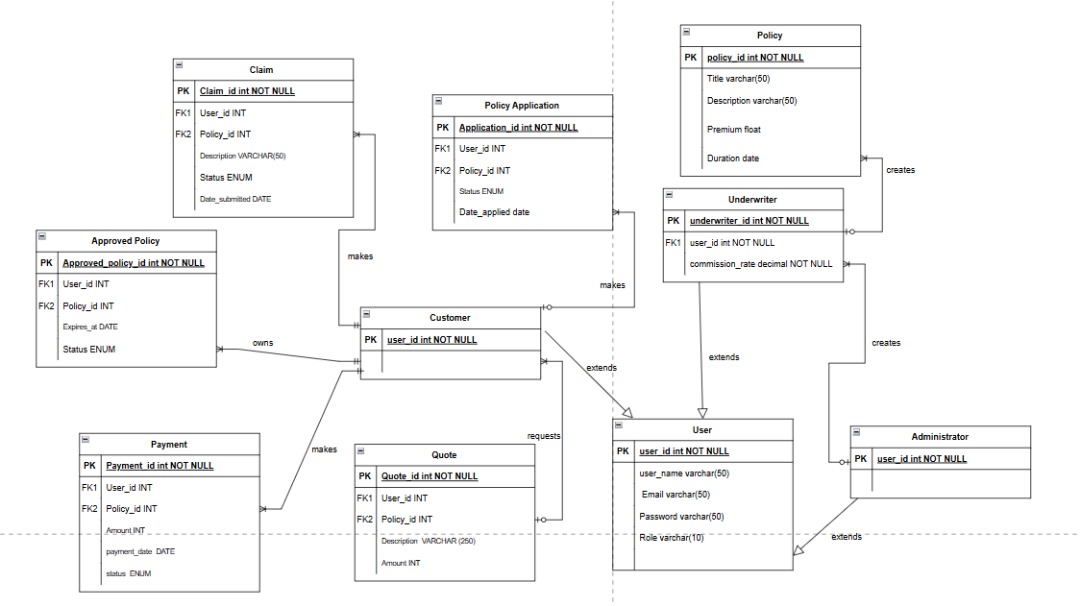
Figure 1: Use Case Diagram of the Safe-Nest Insurance Management System



# 3.DATA DESIGN

## 3.1 Entity Relationship Diagram (ERD)

Figure 2:Figure shows ERD



## 3.2 Data Description

Table 3: Table shows the Data Description

|  |  |
| --- | --- |
| **Entities** | **Description** |
| user | User\_ID (PK)  Name  Email  Password  remember token  Role |
| policy | Policy\_ID(PK)  Title  Description  Premium  Date |
| policy-application | Application\_ID (PK)  User\_ID (FK)  Policy\_ID (FK)  Status (pending, rejected, accepted)  Requirements\_Path  Notes  Date |
| Claim Application | Claim\_ID (PK)  User\_ID (FK)  Policy\_ID (FK)  Description  Status  Date\_submitted  attachment |
| Approved policy | Approved\_policy\_ID (PK)  User\_ID (FK)  Policy\_ ID (FK)  expires\_at  Status |
| Payment | Payment\_ID (PK)  User\_ID (FK)  Policy\_ID(FK)  Amount  due\_date  payment\_date  status (pending, overdue, paid) |
| Quote | Quote\_ID (PK)  User\_ID (FK)  Policy\_ID (FK)  Description  Amount |

## 3.3 DATA DICTIONARY

User table

Table 4: Table shows description of user attributes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Datatype | Size | Key | Description |
| User\_ID | INT |  | PK | A unique identifier for the user. |
| Name | VARCHAR | 50 |  | Name of the user. |
| Email | VARCHAR | 50 |  | Personal email of the user. |
| Password | VARCHAR | 50 |  | This password must be correct to allow the administrator to access the system. |
| Role | VARCHAR | 50 |  | Every user has a specific role they are assigned i.e., admin, policyholder, underwriter. |
| remember\_token | VARCHAR | 50 |  | This is used to provide permission to the user to change the password |

Policy table

Table 5: Table shows description of policy attributes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Datatype | Size | Key | Description |
| Policy\_ID (PK) | INT |  | PK | A unique identifier for the policy. |
| Title | VARCHAR | 50 |  | Name of the policy. |
| Description | VARCHAR | 50 |  | Detailed overview of the policy. |
| Premium | VARCHAR | 50 |  | Amount of money paid at a regular interval for a policy. |
| Duration | int |  |  | Time within which the policy is still valid in years. |

Policy application table

Table 6:Table shows description of policy application attributes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Datatype | Size | Key | Description |
| Application\_ID | INT |  | PK | A unique identifier for the policy application. |
| User\_ID | INT |  | FK | A unique identifier for the user. |
| Policy\_ID | INT |  | FK | A unique identifier for the policy. |
| Status | ENUM (Approved, Pending, Rejected) | 3 |  | Describes the state of a policy application. |
| Date\_Applied | DATE |  |  | Date when application when is made. |
| Requirements\_path | VARCHAR | 100 |  | This is the location of the proof files submitted. |
| Notes | VARCHAR | 100 |  | This is used to store the reason of rejection of an application |

Claim application table

Table 7: Table shows description of claim application attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Datatype | Size | Key | Description |
| Claim\_ID | INT |  | PK | A unique identifier for the claim application. |
| User\_ID | INT |  | FK | A unique identifier for the user. |
| Policy\_ID | INT |  | FK | A unique identifier for the policy. |
| Description | VARCHAR | 50 |  | A detailed text of the claim made. |
| Status | ENUM (Approved, Pending, Rejected) | 3 |  | Describes the state of the claim application. |
| Date\_submitted | DATE |  |  | Date when the claim application is submitted. |

Approved policy table

Table 8:Table shows description of approved policy attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Datatype | Size | Key | Description |
| Approved\_policy\_ID | INT |  | PK | A unique identifier for the approved policy. |
| User\_ID | INT |  | FK | A unique identifier for the user. |
| Policy\_ID | INT |  | FK | A unique identifier for the policy |
| Duration | DATE |  |  | Time within which the policy is approved. |
| Status | ENUM (Expired, Active) |  |  | Describes the current state of your policy. |

Payment table

Table 9: Table shows description of payment attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Datatype | Size | Key | Description |
| Payment\_ID | INT |  | PK | A unique identifier for payment. |
| User\_ID | INT |  | FK | A unique identifier for the user. |
| Policy\_ID | INT |  | FK | A unique identifier for the policy. |
| Amount | INT |  |  | Amount paid for a policy. |
| Date\_paid | DATE |  |  | Date when the amount was paid. |
| Status (paid, pending, overdue) | Enum | 3 |  | This is the status of payment |
| Due\_date | DATE |  |  | This is the date when the payment is due. |

Quotes table

Table 10: Table shows description of quotes attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Datatype | Size | Key | Description |
| Qoutes\_ID | INT |  | PK | A unique identifier for the quoting. |
| User\_ID | INT |  | FK | A unique identifier for the user. |
| Policy\_ID | INT |  | FK | A unique identifier for the policy |
| Description | VARCHAR | 50 |  | A detailed text showing the quoting. |
| Amount | INT |  |  | Amount in the quoting. |

# 4.Human Interface Design

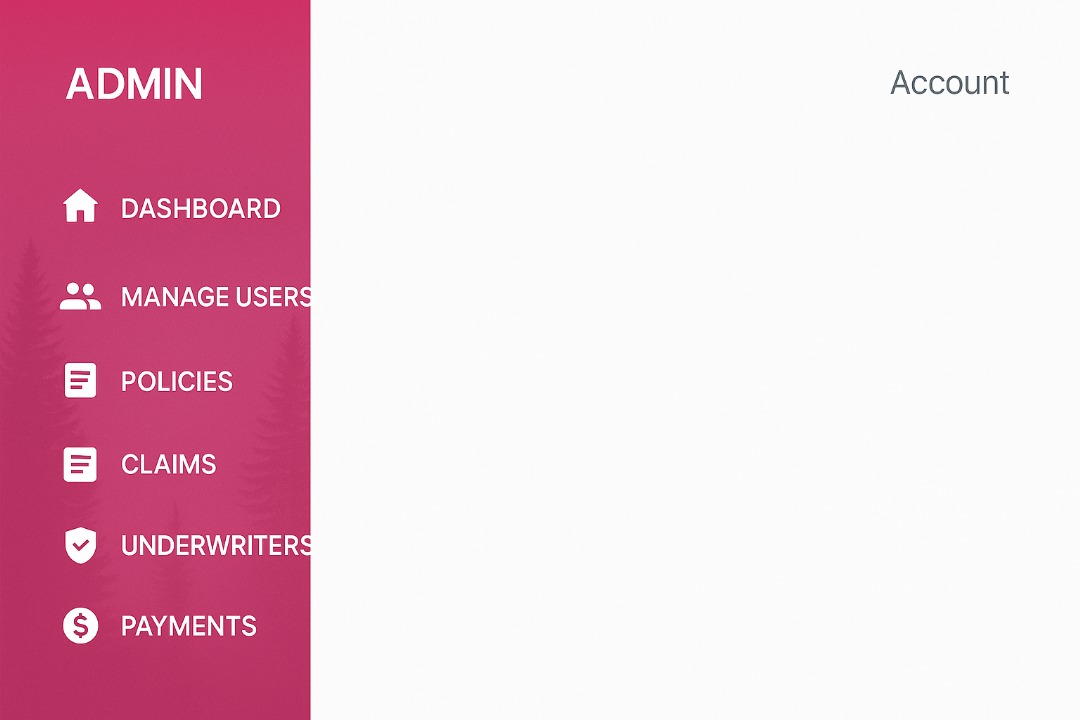
## 4.1 Home page

Figure 3:Figure shows our homepage for the system.



## 4.2 Administrator interactivity

Figure 4: Figure shows the Administrator Dashboard



## 4.3 Customer interactivity

Figure 5: Figure shows the Customer Dashboard



## 4.4 Underwriter interactivity

Figure 6: Figure shows the underwriter dashboard

