Software Requirements Specification

Introduction

Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the requirements for the development of an **Insurance Management System**. The system aims to streamline the management of insurance policies for end users, ensuring that they have easy access to their insurance information, can receive timely notifications regarding policy expirations, and are provided with secure, reliable services.

This document outlines the functional and non-functional requirements, including user requirements, system constraints, and operational environment, to guide the design and development of the system. The goal is to create an intuitive, secure, and efficient platform for users to manage their insurance policies and associated tasks, such as policy renewal, claims processing, and product recommendations.

Key objectives of the system include:

- **Improved User Experience**: The system will display insurance information in a concise, clear, and intuitive manner to facilitate easy navigation and quick access to key details.
- **Automatic Categorization**: Insurance policies will be automatically categorized to enhance organization and user accessibility.
- **Proactive Notifications**: The system will notify users about policy expiration dates, reducing the likelihood of coverage lapses.
- **Personalized Recommendations**: Users will receive tailored insurance product recommendations based on their past history and current needs.
- **Data Security**: Strong security protocols will be implemented to protect sensitive user information, ensuring trust and privacy.

Success Criteria

The success of the system will be measured by:

- **User Satisfaction**: User feedback and surveys will gauge ease of use and satisfaction with the system's functionality.
- **Improvements in Business Operations Efficiency**: Increased efficiency in processing insurance policies, renewals, and claims. Tracking key performance indicators (KPIs) such as reduced policy renewal time, fewer errors in claims processing, and reduced administrative overhead will reflect the impact on business operations.

Scope

The scope of this Software Requirements Specification (SRS) document is to define the functional and non-functional requirements for the development of the Insurance Management System. This system is designed to assist end users in managing their insurance policies effectively and securely. The system will provide a platform for displaying insurance information, managing policy renewals, tracking claims, and ensuring data security. It will also offer personalized product recommendations and proactive notifications, all while maintaining a user-friendly interface.

Product Perspective

System Interfaces

The Insurance Management System will integrate with various external and internal systems to ensure smooth operation, data synchronization, and communication across different components. The following section outlines the system interfaces, including communication protocols, data formats, and integration points.

Internal System Interfaces

1. **Backend-Frontend Interface**
    - Description: The frontend web and mobile interfaces will communicate with the backend system (Java Spring Boot or Python Django/Flask) through HTTP requests.
    - Protocol: RESTful API over HTTPS.
    - Data Format: JSON for all data exchanges between the frontend and backend.
    - Operations: The system will use standard HTTP methods such as GET, POST, PUT, and DELETE to fetch, update, and manage user insurance information, claims, notifications, etc.
2. **Backend-Database Interface**
    - Description: The backend application will interact with the database (MySQL or PostgreSQL) to store, retrieve, and manage insurance data, user profiles, claims, and notifications.
    - Protocol: SQL queries through a Database Management System (DBMS) protocol.
    - Data Format: Structured data in SQL format.
    - Operations: CRUD operations (Create, Read, Update, Delete) for managing insurance policies, claims, user data, and notifications.
3. **Backend-Security Interface**
    - Description: The backend system will communicate with security services (OAuth 2.0, JWT, and encryption modules) to ensure secure user authentication and authorization, as well as data protection.
    - Protocol: HTTPS with SSL/TLS encryption for secure communication.
    - Operations:
        - User authentication via OAuth 2.0 or JWT tokens.
        - Encryption of sensitive data using AES-256 for storage and transmission.

External System Interfaces

1. **Third-Party Notification Service Interface**
    - Description: The system will integrate with a third-party notification service (e.g., email, SMS, push notifications) to send reminders for policy expiration and other notifications to users.
    - Protocol: HTTP/HTTPS for API communication.
    - Data Format: JSON or XML for data exchange, depending on the third-party service requirements.
    - Operations:
        - Send policy expiration reminders, product recommendations, and other user notifications.
        - Provide feedback on successful or failed notifications.
2. **Payment Gateway Interface (Optional, for future integration)**
    - Description: The system will integrate with external payment gateways for processing payments related to insurance renewals, claims, or other services in future releases.
    - Protocol: HTTPS with secure payment protocols (e.g., PCI-DSS for card payments).
    - Data Format: JSON for data exchanges such as transaction details and payment statuses.
    - Operations:
        - Initiate and process payments.
        - Retrieve payment confirmations and error messages.

Requirements

Functions

1. **Insurance Information Display**
   Priority: P0
    - Description: The system must display insurance information in a concise, clear, and intuitive manner.
    - Functional Requirements:
        - The system will provide a user-friendly interface that presents insurance details such as policy types, expiration dates, coverage levels, and other key information in an easily accessible format.
        - Information will be grouped and displayed in logical categories, allowing users to find relevant details quickly.
        - The system must minimize the number of steps required to access insurance data, ensuring that users can view critical information without unnecessary navigation through multiple pages.
2. **Insurance Expiration Date Notifications**
   Priority: P1
    - Description: The system should proactively notify users one week before their insurance policy expiration date.

- Functional Requirements:
  - The system will monitor expiration dates for all active insurance policies.
  - A notification will be sent to users via email, SMS, or within the system at least seven days prior to the expiration of their policies.
  - Notifications will contain relevant details, such as the insurance policy name, expiration date, and renewal instructions.
  - The system will track whether the notification has been acknowledged by the user and remind them periodically until action is taken.

3. **Data Security and Privacy Protection**
   Priority: P0
   - Description: The system must have strong security measures in place, particularly regarding the protection of personal information.
   - Functional Requirements:
     - All user data, including personal details and insurance records, must be encrypted both during transmission (via SSL/TLS) and at rest (using AES-256 encryption).
     - The system will implement role-based access control (RBAC) to restrict data access based on user roles (e.g., end users, administrators).
     - User authentication must be handled securely using OAuth 2.0 or JWT for authorization, ensuring that only authorized individuals can access sensitive information.
     - The system will log all access attempts and data changes to detect and respond to any potential security breaches.
     - Further security measures will include preventing SQL injection, cross-site scripting attacks, and ensuring compliance with GDPR and HIPAA.

Performance Requirements

1. **Response Time**
   The system must respond to user requests within a reasonable time frame. The maximum response time for the following critical operations should be:
   - **Login/Authentication**: 2 seconds
   - **Policy Information Retrieval**: 3 seconds
   - **Claims Status Update**: 3 seconds
   - **Search Results Display**: 2 seconds
   - **Notification Alerts**: 5 seconds
   - **Dashboard Load Time**: The home page/dashboard must load fully within 5 seconds for a typical user session. It should display relevant insurance policy details and notifications promptly.
   - **Real-time Notifications**: The system must push notifications (e.g., for insurance expiration or claims status updates) to the user within 10 seconds of triggering events.
2. **Load Testing and Scalability**

- The system must support at least 5,000 concurrent users accessing the system simultaneously without a significant degradation in performance.
- Load testing should ensure the system performs reliably under peak user load, particularly during high-traffic periods such as policy renewal deadlines.