Functional Requirements Document (FRD)

Project Name: E-Insurance Application
Prepared For: Development Team, QA, Project Managers
Prepared By: Rishav

1. Introduction

The E-Insurance Application is a digital application that allows users to buy, administer, and claim insurance policies. Its aim is to automate the insurance process by giving a user-friendly and secure platform for policyholders and administrators. It allows user registration, policy administration, payment, claim, and administration and is facilitated in the application as well as supports regulatory compliance.

2. System Overview

The system follows a multi-tier architecture with the following layers:

1. **Presentation Layer**: User interfaces (web/mobile) for customers and admin.
2. **Application Layer**: Handles business logic, policy management, claims, payments.
3. **Data Layer**: Stores user, policy, claim, and payment information in structured databases.
4. **Integration Layer**: Communicates with external APIs (payment gateway, compliance)

3. Business Requirements

BR ID	Description
BR1	Allow users to register and log in their accounts.
BR2	Allow users to view and buy insurance policies.
BR3	Provide users with payment functionality for policy premiums.
BR4	Allow users to submit and track insurance claims.
BR5	Provide admin access to manage users and policies.
BR6	Enable admin to generate reports and tracks system usage
BR7	Integrate with third-party payment and compliance APIs.
BR8	Ensure secure handling of personal and financial data.

4. Functional Requirements

FR ID	Description
FR1	The system shall allow users to register with email, mobile phone number and password.
FR2	The system should authenticate users by checking their credentials.
FR3	The system should also allow users to access available policy details.
FR4	The system should allow users to purchase a policy and create a digital receipt.
FR5	The system should allow users to make payments through payment gateways
FR6	The system should enable users to submit a claim with relevant details.
FR7	The system should allow users to check the status of submitted claims.
FR8	The system should allow admins to add/edit/delete policies.
FR9	The system should allow admins to view user data and activity.
FR10	The system should allow admins to generate reports (for ex., claims, payments).

5. User Roles and Permissions

Role	Permissions
User	Register, Login, View Policies, Buy Policy, Make Payment, Submit Claim, Track Claim
Admin	Login, Manage Users, Manage Policies, View Claims, Generate Reports
System/Server	Collecting data from the external APIs, Store/Fetch data from DB

6. Use Case Example

Use Case: Submit Insurance Claim

- **Actor**: Registered User
- **Preconditions**: The User has an active policy
- **Main Flow**:
    1. User logs into the system.

2. Navigates to the Policies Section.
3. Selects a policy and clicks “Submit Claim”.
4. Fills claim details and uploads documents.
5. The system stores the claim and shows a confirmation message.

- **Postconditions**: Claim is stored in the database with status “Pending”.

7. Assumptions and Constraints

Assumptions

- Users have access to a stable internet connection.
- Third-party APIs (payment/compliance) are available and functional.
- Users are expected to provide genuine information.

Constraints

- The application must follow local insurance regulations.
- User data must be securely stored.
- The system should be scalable.
- Claims must be submitted only against active policies.

8. Non-Functional Requirements

Category	Requirement
Security	All user data should be encrypted.
Performance	System should respond to user actions within 2 seconds.
Scalability	Applications should support scaling with increasing users.
Availability	System should be available 99.9% of the time.
Maintainability	Code must be well structured for easy updates.
Compliance	System must follow GDPR and IRDA (India) regulations for data protection.

9. Functional Work-Flow

10. Workflow

User Flow

1. Start - starts the process
2. User Registration - system take data from user for registration
3. Registered - user data is registered in database
4. View Policy Option - user see different insurance policies to choose from
5. Select Policy - user picks there desired policy
6. View Policy Options - user gets detailed options for there selected policy.
7. Check Claim Status - users can track the status of their insurance claim.
8. End - The process ends here.

System Flow

1. Call Auth API - System verifies user credentials.
2. User Database - it stores user registration and login data.
3. Call Payment Gateway - starts the payment process for the selected policy.
4. Payment Successful?
    - Yes → Proceed to submit a claim and update the Policy DB.
    - No → Stay on the payment page or retry.
5. Submit Claim - User files an insurance claim.
6. Check Claim Status - System checks status from the Claim DB.
7. Policy DB - update and store policy data after payment.

Admin Flow

1. View User - Admin can view registered users and their information.
2. Manage Policies - Admin can add, edit, or delete insurance policies.
3. Generate Reports - System generates analytical reports for review.
4. Manage Database - Admin manages Policy DB and Claim DB.