End to End Project for Business Analysis – Insurance Domain

Sagar Sahu

PROJECT SCOPE DOCUMENT

Project Title:

Insurance Application Development

Project Objective:

The objective of this project is to build a customer centric insurance application where users can explore various insurance products such as auto, home, health, life, and property insurance. The application will allow customers to get quotes, compare insurance products, and purchase policies online.

Project Justification:

The insurance market is evolving with increased customer demand for digital solutions that offer convenience and transparency. With the rise of mobile and web technologies, insurance companies must adapt by providing platforms where users can access information, receive quotes, and complete transactions with ease. This application will enhance customer satisfaction, improve market reach, and reduce overhead associated with in-person policy sales.

Project Stakeholders:

- **Primary Stakeholders:**
    - Project Sponsor: Insurance Company CEO
    - Product Manager
    - Marketing Team
    - Sales Team
    - Customer Support Team
    - Business Analysts
    - IT Development Team
    - QA Team

- **Secondary Stakeholders:**
    - Regulatory Agencies
    - Insurance Agents and Brokers
    - End-users (Customers)
    - Partner Banks for insurance tie-ups

AS-IS State:

Currently, the insurance company primarily relies on in-person or agent-based sales, which limits customer flexibility. Online engagement is minimal, and customers have to contact agents for quotes and to purchase policies. The company's digital presence is limited to informational web pages with no interactive features.

TO-BE State:

The future state involves an interactive insurance application where customers can:

- View details about insurance products (auto, home, health, life, property).
- Get instant insurance quotes by filling out basic forms.
- Compare different insurance policies.
- Purchase insurance policies through an online payment gateway.

- Manage their purchased policies within their accounts (e.g., view policy documents, renew, and claim status).
- The system will include a customer support chatbot and other assistance features.

Project In Scope Use Case:

1. **View Insurance Products:** Customers can explore different products such as auto, home, health, life, and property insurance.
2. **Get Real-time Quotes:** Customers can get instant quotes by entering personal information and insurance preferences.
3. **Compare Insurance Products:** The system will allow users to compare different policies side by side.
4. **Purchase Insurance Policy:** Customers can select a policy and complete the purchase through an integrated payment gateway.
5. **Manage Policies:** Customers can renew, view, and update their insurance policies.
6. **Claims Filing and Tracking:** Users can file claims and track their progress through the application.

Project Out of Scope Use Case:

1. **Agent and Broker Management:** The application will not include functionalities for managing agents and brokers.
2. **Customizable Policy Building:** The system will only offer predefined packages; fully customizable policies are out of scope.
3. **Third-Party Insurance Add-ons:** Integration with external services like roadside assistance or health add-ons is excluded.
4. **Advanced Analytics:** Detailed analytics and reporting on user behaviour will be part of a separate project phase.

Other In-Scope Deliverables:

- **User Interface:** A responsive web and mobile interface for seamless customer navigation.

- **Quote Generation Module:** A real-time insurance quote generator.
- **Policy Purchase Workflow:** A secure workflow for purchasing insurance with an integrated payment gateway.
- **Customer Dashboard:** A section where users can view purchased policies, premium dues, and claim statuses.
- **Customer Support Chatbot:** Automated chatbot support for basic queries.
- **Admin Dashboard:** Admin panel for internal users to track policies and manage customer requests.

Other Out-of-Scope Deliverables:

- **Backend CRM Development:** Integration of the insurance application with the company's existing CRM will be handled separately.
- **Advanced Analytics Dashboard:** Any advanced analytics beyond customer engagement reports is out of scope.

- **External API Integration:** Integrations with third-party analytics or marketing platforms will be excluded from this phase.

Project Risks:

1. **Regulatory Compliance:** The project may face delays if insurance regulations change or if compliance is not met.
2. **Data Security:** Risks related to customer data breaches or payment gateway vulnerabilities.
3. **Third-Party Dependencies:** Integration issues with external systems (e.g., payment gateways).
4. **Customer Adoption:** Potential risk of low customer adoption due to technical challenges or insufficient user experience.
5. **Scope Creep:** If additional features are requested mid-project, it may lead to delays and cost overruns.

Project Assumptions:

1. The insurance company will provide all necessary product information and regulations.

2. Customers have access to the internet and are familiar with using digital applications.
3. The payment gateway will be integrated without issues from third-party vendors.
4. The IT team has the technical resources and skills to build and maintain the application.

Project Constraints:

1. The project must be completed within the set timeline (6 months).
2. The application must comply with local insurance regulations.
3. The scope must adhere to the in-scope features outlined above.

Project Issues:

1. **Lack of Customer Engagement Data:** The company lacks existing data on digital user behaviour, which may hinder design decisions.
2. **Team Resource Availability:** Limited IT resources may delay development.

Project Dependencies:

1. Availability of third-party payment gateways.
2. Timely delivery of policy data from internal systems.
3. Regulatory approvals for digital insurance sales.
4. Availability of the IT team for integration with existing systems.

Glossary:

1. **Quote:** A calculated estimate of the insurance premium based on user-provided information.
2. **Premium:** The amount customers pay for their insurance policy.
3. **Policy:** A contract between the insurer and the insured detailing the terms of coverage.
4. **Claim:** A request by the insured for compensation due to a covered loss.
5. **Customer Dashboard:** The user interface where customers can manage their policies.

GAP ANALYSIS

Current State:

1. **Limited Online Presence:**
    a. The insurance company has a basic website that only provides static information about products. Customers must contact an agent to get quotes or purchase insurance.
    b. No functionality for comparing insurance products or managing purchased policies online.
    c. Customers must visit physical offices or interact with agents for policy renewal, claims, and other services.
2. **Manual Quote and Purchase Process:**
    a. Customers must call or visit an office to get quotes or purchase insurance policies.
    b. Quotes are generated manually by agents based on customer inputs, leading to potential delays.
3. **No Online Claims Processing:**
    a. Claims must be filed in person or over the phone with no online tracking or management system available for customers.
4. **Lack of Customer Self-Service:**

a. Customers cannot manage their policies (renewal, updating contact information, etc.) independently.

b. No online account creation or dashboard for viewing insurance policies and claims.

Future State:

1. **Full Digital Insurance Application:**

   a. A fully functional insurance application that allows customers to view, compare, and purchase insurance products online.

   b. Self-service features for policy management and claim filing.

2. **Automated Quote Generation:**

   a. Real-time insurance quote generation based on customer-provided details.

   b. Customers can instantly compare different products and select the most suitable policy.

3. **Online Claims Processing and Management:**

   a. Customers can file and track claims directly through the application.

   b. Policyholders can view claim statuses and receive updates without needing to contact support.

4. **Customer Self-Service Portal:**

   a. A customer account management system where users can view, renew, and modify their policies.

   b. Online access to policy documents, premium payment schedules, and claims.

Gap Identified:

1. **Limited Online Functionality:**

   a. Current system does not allow online quoting, product comparison, or policy purchasing.

2. **Manual Processes:**

   a. Most transactions (quotes, purchases, claims) are done manually, leading to inefficiencies and delays.

3. **Lack of Self-Service Capabilities:**

   a. Customers cannot independently manage policies or track claims.

4. **No Real-Time Data Processing:**

   a. There is no infrastructure for real-time quote generation and processing of policy information.

Steps to Cover the Gap:

1. **Implement Digital Platform for Insurance Services:**
   a. Develop a web and mobile application that allows customers to browse insurance products, get quotes, compare policies, and make purchases online.
   b. Ensure the user interface is intuitive and accessible across devices.
2. **Automate Quote Generation and Policy Comparison:**
   a. Build a real-time quote generation engine that uses customer data to generate quotes instantly.
   b. Incorporate comparison features that allow customers to evaluate different insurance policies side by side.
3. **Enable Online Claims Management:**
   a. Develop functionality for customers to file, manage, and track claims online.

   b. Set up a customer dashboard where they can view claim history and receive updates.

4. **Establish a Self-Service Customer Portal:**
   a. Build a secure login-based customer account management system where users can view, renew, and manage their policies.
   b. Allow policyholders to download policy documents, check premium due dates, and update their personal information.
5. **Enhance Data Processing Capabilities:**
   a. Integrate the application with the company's backend systems for real-time policy and customer data processing.
   b. Implement secure payment gateways for online policy purchases and renewals.
6. **Improve Customer Support Features:**
   a. Add chatbot support for answering basic queries and providing assistance during the insurance purchase process.
   b. Allow customers to directly interact with support teams through the app for more complex issues.

ROOT CAUSE ANALYSIS

Problem:

Customers are reporting inaccurate or delayed insurance quotes when trying to get a quote through the application. The quotes provided either do not match their input criteria or take too long to generate, resulting in a poor user experience.

Root Cause Analysis using the 5 Whys Technique

1. **Why are the insurance quotes inaccurate or delayed?**
   - The quote generation system is not consistently pulling the correct data from the underlying data sources, or the system experiences significant lag in processing requests.
2. **Why is the system not pulling the correct data or experiencing lag?**
   - The data integration between the frontend user interface and the backend quote generation engine

is faulty, and the communication between them is slow or inconsistent.

3. **Why is the data integration faulty?**

   - The backend system relies on legacy infrastructure that is not optimized for real-time data processing. The API calls to fetch data from legacy databases are inefficient and slow, leading to delays and errors.

4. **Why is the backend infrastructure not optimized for real-time processing?**

   - The project did not allocate sufficient resources or time to upgrade the backend infrastructure during development, and legacy systems were repurposed instead of being replaced or optimized.

5. **Why was there insufficient resource allocation for infrastructure upgrades?**

   - During the project planning phase, there was a focus on frontend features and customer-facing functionality, with insufficient attention given to the importance of upgrading backend systems for seamless integration and real-time data processing.

Root Cause Identified:

The primary root cause is the **use of legacy backend systems** that are not optimized for real-time data processing. These systems are slow in retrieving and processing data, and the API integration with the frontend is inefficient, leading to inaccurate or delayed insurance quotes. Additionally, during the project planning phase, inadequate focus was placed on upgrading these backend systems, which has now led to performance issues.

Steps to Address the Root Cause:

1. **Backend System Upgrade:**
    a. Upgrade the backend infrastructure to modern systems capable of real-time data processing. Implement faster, more efficient databases and ensure the infrastructure can handle high volumes of requests without lag.

2. **Optimize API Integration:**
    a. Redesign the API integration between the frontend and backend systems for more efficient communication. Ensure that the APIs are optimized to handle real-time quote generation and data retrieval.
3. **Load Testing and Performance Tuning:**
    a. Perform load testing on the application to simulate high user traffic and stress-test the system. Use the insights gained to optimize system performance and reduce latency.
4. **Monitoring and Error Handling:**
    a. Implement real-time monitoring and error logging to identify performance bottlenecks and integration issues early. Ensure that there are mechanisms to handle errors gracefully without affecting user experience.
5. **Reprioritize Project Scope:**
    a. Revisit the project scope to allocate resources and time for addressing backend system performance. Ensure that both frontend functionality and backend performance are

prioritized equally in future development phases.

The root cause of the inaccurate and delayed insurance quotes lies in the reliance on legacy backend systems that were not upgraded or optimized during the project. By upgrading the backend, optimizing API integration, and conducting proper load testing, the performance of the insurance application can be improved, leading to accurate, real-time insurance quotes and a better user experience.

BUSINESS REQUIREMENT DOCUMENT

Business Problem:

Currently, the insurance company relies on a largely manual system for quoting and purchasing insurance, leading to inefficiencies, customer dissatisfaction, and reduced market competitiveness. Customers cannot easily compare products or manage their policies, resulting in high operational costs and lower sales conversion rates.

AS-IS State:

- The current system allows customers to view static information about insurance products, but they must contact an agent to get quotes and purchase policies.
- The process for generating quotes, policy purchase, and claims management is manual, slow, and prone to errors.

- No online self-service option exists for managing policies or filing claims.

To-Be State:

- The new system will provide customers with a web and mobile application to explore insurance products, receive real-time quotes, compare policies, purchase insurance online, and manage their policies.
- Customers will be able to file and track claims, renew policies, and access policy details through a self-service portal.
- The application will be integrated with secure payment gateways and backend systems for real-time data processing.

RACI Matrix

Task	Responsible (R)	Accountable (A)	Consulted (C)	Informed (I)
Requirement Gathering	Business Analyst	Product Manager	Marketing, Sales	Development Team
System Design	Development Team	IT Manager	Product Manager	QA Team
Backend Integration	Development Team	IT Manager	Business Analyst	Sales, Marketing
Testing and QA	QA Team	QA Manager	Business Analyst	Product Manager
Deployment	Development Team	IT Manager	Product Manager	Sales, Marketing
User Training	Customer Support	Training Manager	Business Analyst	End Users

Business Requirements:

1. The system must allow customers to view all insurance products (auto, home, health, life, property).
2. The system must generate real-time quotes based on customer input.
3. Customers should be able to compare insurance policies and select the most appropriate one.
4. The system must enable users to purchase insurance policies online securely.
5. The application must provide a user account portal where customers can manage their policies and file claims.
6. The system must provide notifications (via email/SMS) to customers regarding renewals, policy updates, and claims statuses.

Functional Requirements:

1. **User Interface:**
    a. The system must be accessible via web and mobile applications.
    b. It should provide an intuitive user interface for product exploration and policy management.
2. **Quote Generation:**
    a. The system must generate real-time quotes based on inputs like vehicle details, home value, health information, etc.
3. **Policy Comparison:**
    a. The application must allow customers to compare up to 3 policies simultaneously.
4. **Purchase Process:**
    a. The system must offer a secure, integrated payment gateway for completing policy purchases.
5. **Policy Management:**
    a. The system must allow users to view policy details, renew policies, and update personal information.
6. **Claims Management:**

a. The system must provide a way for customers to file claims and track the progress of claims in real time.

Non-Functional Requirements:

1. **Performance:**
    a. The system must handle up to 10,000 simultaneous users without performance degradation.
2. **Security:**
    a. The system must adhere to industry-standard data encryption protocols (e.g., SSL, TLS) to ensure secure transactions and data privacy.
3. **Usability:**
    a. The interface must be user-friendly and accessible to users with limited technical knowledge.
4. **Availability:**
    a. The system must be available 99.9% of the time, with maintenance windows planned during off-peak hours.

API Requirements:

1. **Real-time Quote API:**
   a. The application must integrate with third-party data providers (e.g., for vehicle or property data) to provide real-time quotes.
2. **Payment Gateway API:**
   a. The system must integrate with a secure payment gateway (e.g., Stripe, PayPal) for handling online transactions.
3. **Claims API:**
   a. The system must expose an API for internal systems to track and manage claims processing.

Integration Requirements:

1. **Backend System Integration:**
   a. The application must integrate with the existing policy management and CRM systems for real-time data processing.
2. **Payment Gateway Integration:**

   a. The application must integrate with third-party payment systems to process online policy purchases securely.

3. **Customer Support System:**
   a. The system must integrate with a customer support system (e.g., Zendesk) to manage inquiries and assist with claims.

Database Requirements:

1. The database must store customer details, policy data, and transaction history.
2. The database must be optimized for real-time read/write operations to support high-volume transactions.
3. The database must ensure data consistency and integrity during policy updates, purchases, and claims processing.

Transition Requirements:

1. The system must ensure seamless migration from the existing manual process to the new digital platform without disrupting current operations.

2. Data from the legacy system (customer records, policy information) must be migrated into the new system.
3. Customers must be informed and educated on how to use the new platform through proper onboarding and tutorials.

Data Dictionary:

FIELD NAME	DESCRIPTION	DATA TYPE	EXAMPLE
CUSTOMER_ID	Unique identifier for a customer	Integer	12345
POLICY_I D	Unique identifier for an insurance policy	Integer	67890
QUOTE_AMOUNT	Amount for the insurance quote	Decimal	1500.00
POLICY_START_DATE	Date the policy coverage begins	Date	2024-01-01
CLAIM_STATUS	Status of an insurance claim	String	Pending, Approved, Rejected

SOFTWARE REQUIREMENT SPECIFICATION(SRS)

Purpose:

The purpose of this Software Requirements Specification (SRS) is to define the detailed software requirements for the development of the Insurance Application, which will allow customers to explore, compare, and purchase insurance products such as auto, home, health, life, and property insurance. This document serves as a comprehensive guide for developers, testers, and stakeholders to ensure that the software meets the business and functional needs outlined in the Business Requirement Document (BRD).

Scope:

This SRS covers the functional, non-functional, API, integration, database, and system requirements necessary for the implementation of the Insurance Application. It includes features such as product exploration, quote generation, policy comparison, online purchase, policy management, claims processing, and integration with third-party payment gateways and internal insurance systems.

Definitions, Acronyms, and Abbreviations:

- **BRD:** Business Requirements Document
- **API:** Application Programming Interface
- **SLA:** Service Level Agreement
- **UI:** User Interface
- **UX:** User Experience
- **CRM:** Customer Relationship Management system
- **Quote:** Estimated insurance cost based on customer data

References:

- Business Requirement Document (BRD) for Insurance Application Development.
- Insurance Industry Regulatory Guidelines
- API Documentation for third-party Payment Gateway (e.g., Stripe/PayPal).

Overall Description

Product Perspective:

The Insurance Application will replace the current manual system for policy generation, purchasing, and claims processing. The new system will integrate with the company's backend CRM, policy management systems, and third-party payment gateways. The application will provide both a web-based interface and a mobile app for ease of use.

Product Features:

- **Insurance Product Exploration**: Customers can view details of various insurance products (auto, home, health, life, property).
- **Real-Time Quote Generation**: Customers can obtain instant quotes based on inputs like vehicle information, property details, or health data.
- **Policy Comparison**: Users can compare insurance products side by side.
- **Online Purchase**: Customers can buy insurance policies securely via a payment gateway.
- **Policy Management**: Customers can view, renew, and manage their policies.

- **Claims Filing and Tracking**: Customers can submit and track claims through the application.

User Classes and Characteristics:

- **Customers**: End users of the application who explore products, get quotes, purchase insurance, and file claims.
- **Admin**: Internal users responsible for customer support, managing requests, and handling claims.
- **Third-party Payment Gateway**: Integration of external payment systems for policy transactions.

Operating Environment:

- **Web Application**: Accessible on major web browsers (Chrome, Firefox, Safari, Edge).

  Mobile Application: Available on both Android (version 10 and above) and iOS (version 12 and above).
- **Backend System**: The application will be deployed on a cloud infrastructure with APIs integrated for real-time quote generation and payment processing.

Design and Implementation Constraints:

- Compliance with data privacy and security regulations (GDPR, HIPAA, PCI-DSS).
- Scalability to handle up to 10,000 concurrent users.
- Integration with existing legacy backend systems.

Functional Requirements

Insurance Product Exploration:

- **Description**: The system will display the details of all available insurance products (auto, home, health, life, property).

  Inputs: User selects the category of insurance they want to explore.
- **Outputs**: The system displays available products with features, coverage details, and pricing.
- **User Interaction**: Users can click on individual products for detailed information.

Real-Time Quote Generation:

- **Description**: Customers will input personal details and insurance preferences (e.g., car model, home location), and the system will generate an instant quote.
- **Inputs**: Customer details (e.g., age, property information, health details).
- **Outputs**: Insurance quote displayed with breakdown and comparison options.
- **Business Rules**: The system must calculate quotes using predefined algorithms based on customer input data.
- **Dependencies**: Integration with backend systems for real-time data retrieval.

Policy Comparison:

- **Description**: Customers will be able to select and compare up to 3 insurance products side by side.
- **Inputs**: Customer selects insurance products for comparison.
- **Outputs**: A comparison table showing features, coverage, and costs of the selected policies.

Online Purchase:

- **Description**: After choosing a policy, the customer will proceed with the purchase by making a payment through the integrated payment gateway.
- **Inputs**: Customer-selected policy and payment details (credit/debit card, bank transfer, etc.).
- **Outputs**: Confirmation of the policy purchase and receipt.
- **Business Rules**: Ensure compliance with secure payment protocols (PCI-DSS).
- **Dependencies**: Integration with third-party payment gateway for processing payments.

Policy Management:

- **Description**: Customers can manage their purchased policies, including renewal, view policy details, and update personal information.
- **Inputs**: Customer account details, policy number.

- **Outputs**: Policy details, premium due date, renewal options.
- **Business Rules**: Customers must be authenticated before accessing their policy data.

Claims Filing and Tracking:

- **Description**: The system allows customers to file insurance claims and track the status in real time.
- **Inputs**: Claim details (e.g., incident date, description, supporting documents).
- **Outputs**: Claim status updates (e.g., pending, approved, rejected).
- **Business Rules**: Claims must follow the company's internal validation process before approval.

Non-Functional Requirements

Performance Requirements

- The system should handle up to 10,000 concurrent users with less than 2 seconds of response time for key operations (quote generation, policy purchase).
- The system must be scalable to accommodate growing customer traffic.

Security Requirements

- Data encryption (SSL, TLS) must be implemented for all transactions.
- Multi-factor authentication (MFA) should be required for customer login and policy management.
- The system must comply with **GDPR**, **HIPAA**, and **PCI-DSS** to ensure user privacy and payment security.

Usability Requirements

- The user interface must be intuitive, allowing users to navigate the application with minimal technical knowledge.
- The mobile app must support swipe and touch features for ease of use.

Availability Requirements

- The system must have 99.9% uptime with maintenance windows scheduled during off-peak hours.
- Automatic failover should be in place to ensure system availability during hardware failures.

Reliability

- The system should ensure data integrity during transactions, especially when generating quotes, processing payments, and updating policy details.
- All transactions must be logged for auditing and troubleshooting purposes.

API Requirements

Real-Time Quote API

- The application must integrate with external databases (e.g., vehicle registry, property records) to retrieve real-time data for quote generation.
- The API must respond within 1 second to ensure a smooth user experience.

Payment Gateway API

- The system must integrate with third-party payment providers (e.g., Stripe, PayPal) to handle secure transactions.
- The API must support multiple payment methods and currencies.

Claims API

- The system must expose an internal API for claim status tracking, allowing customers to view realtime updates on their claims.
- The API must be capable of handling large data sets for image uploads and document submissions.

Integration Requirements

Backend System Integration

- The system must integrate with existing CRM and policy management systems to retrieve and update policy data in real time.
- The integration must support real-time synchronization of customer data, quotes, and policy statuses.

Payment Gateway Integration

- The system must integrate with third-party payment gateways (e.g., Stripe, PayPal) for secure transaction processing.
- The integration must support payment authentication, refunds, and receipts.

Customer Support Integration

- The system must integrate with the company's customer support platform (e.g., Zendesk) for tracking and resolving customer issues.

Database Requirements

Customer Database

- Store customer details, including personal information, policy data, and transaction history.
- Ensure real-time updates and retrieval of customer data with encryption.

Policy Database

- Store policy details, premium amounts, and renewal schedules.
- Ensure data integrity during policy updates and claims processing.

Claims Database

- Store claim details, supporting documents, and status updates.

- The system must allow for large file uploads and efficient data retrieval.

Transition Requirements

- Migrate data from the legacy system into the new platform without disrupting operations.
- Conduct training for internal teams (admin, customer support) on how to use the new system.
- Implement customer onboarding tutorials to help users transition to the new platform.

USER STORY AND ACCEPTANCE CRITERIA

User Story 1: View Insurance Products

As a **customer**, I want to view the details of different insurance products (auto, home, health, life, property), so that I can explore my options and choose a suitable policy.

Acceptance Criteria:

1. The system must display a list of available insurance products.
2. Each product should show a summary (name, description, coverage, starting premium).
3. The customer must be able to click on a product to view more details (policy terms, coverage limits, exclusions).
4. The UI should be responsive on both web and mobile devices.
5. The customer must be able to navigate back to the product list after viewing details.

User Story 2: Get Real-Time Quote

As a **customer**, I want to receive an instant, real-time quote for my insurance policy based on my input, so that I can make an informed decision about purchasing a policy.

Acceptance Criteria:

1. The customer must be able to enter personal details (e.g., car model, home address, age, etc.).
2. The system must calculate a real-time quote based on the inputs.
3. The quote should be displayed within 2 seconds after form submission.
4. The system should show a breakdown of the premium, including any discounts or add-ons.
5. If the system cannot calculate a quote, a message should be displayed with further instructions or alternative actions.
6. The customer must have the option to save the quote for future reference.

User Story 3: Compare Insurance Products

As a **customer**, I want to compare multiple insurance products, so that I can select the best one based on coverage and price.

Acceptance Criteria:

1. The customer must be able to select up to 3 insurance products for comparison.
2. The system must display a side-by-side comparison of the selected products.
3. Comparison criteria should include coverage amount, premium, exclusions, and additional features.
4. The customer must be able to remove or add products to the comparison list before proceeding.
5. The comparison screen should be mobile responsive and allow for easy navigation.

User Story 4: Purchase Insurance Policy

As a **customer**, I want to securely purchase an insurance policy online after selecting it, so that I can complete my transaction without needing to visit an agent.

Acceptance Criteria:

1. The customer must be able to select a policy and proceed to payment.
2. The system must provide payment options (credit card, debit card, bank transfer, etc.).
3. The payment gateway must be integrated and adhere to PCI-DSS standards for secure transactions.
4. After successful payment, the system must send an email confirmation with the policy details.
5. The system should display a success message and allow the customer to download the policy document.
6. If the payment fails, an error message must be displayed with possible actions (e.g., retry, contact support).

User Story 5: Manage Purchased Policies

As a **customer**, I want to view, renew, and manage my purchased policies in one place, so that I can stay informed and maintain my insurance coverage.

Acceptance Criteria:

1. The system must allow customers to view a list of their active and expired policies.
2. Each policy must display key details such as premium due date, renewal options, and policy coverage.
3. The customer must be able to renew a policy by initiating a payment process through the application.
4. The system should allow the customer to update personal details (e.g., contact information) linked to the policy.
5. Notifications should be sent to the customer regarding upcoming renewals.
6. The customer must be able to download policy documents.

User Story 6: File and Track Claims

As a **customer**, I want to file a claim and track its status, so that I can easily manage my claims process without needing to contact an agent.

Acceptance Criteria:

1. The customer must be able to select a policy and file a new claim by filling out a form with incident details.
2. The system must allow customers to upload supporting documents (e.g., photos, receipts) with the claim.
3. Once submitted, the customer should receive a confirmation email with claim details.
4. The system must allow customers to view the status of the claim (e.g., pending, approved, rejected) on a dashboard.
5. The system must display status updates for each claim (e.g., under review, additional documentation required).
6. If the claim is approved, the system must display the payout information.

User Story 7: Customer Registration and Authentication

As a **customer**, I want to register for an account and securely log in, so that I can manage my policies and access my quotes and claims.

Acceptance Criteria:

1. The customer must be able to register by providing personal information (name, email, phone number).
2. The system must validate the customer’s email via a confirmation link sent during registration.
3. The customer must be able to log in using email and password.
4. Multi-factor authentication (MFA) must be enabled for additional security.
5. The customer should be able to reset their password via a "Forgot Password" option.
6. The customer must be logged out after 15 minutes of inactivity.

User Story 8: Integration with Payment Gateway

As a **system**, I need to securely process customer payments using a third-party payment gateway, so that customers can complete their policy purchase transactions.

Acceptance Criteria:

1. The system must integrate with a third-party payment gateway (e.g., Stripe, PayPal) to handle secure transactions.
2. Payment transactions must be encrypted using SSL/TLS.
3. The system must return payment success or failure statuses from the gateway and display appropriate messages to the customer.
4. In case of failed payments, the system must allow the customer to retry or choose a different payment method.
5. A payment receipt should be automatically generated and emailed to the customer upon success.

User Story 9: System Notifications

As a **customer**, I want to receive notifications about important updates like policy renewals and claim status changes, so that I can stay informed and take necessary actions.

Acceptance Criteria:

1. The system must send notifications for upcoming policy renewals, policy purchases, claim submissions, and claim status updates.
2. Notifications must be sent via email and SMS (if provided by the customer).
3. The notification content must be personalized with the customer's name, policy details, and clear action items (e.g., payment due, claim approved).
4. The system must allow customers to configure their notification preferences (email, SMS, or both).
5. Notifications should be triggered automatically based on system events (e.g., payment completion, claim approval).

User Story 10: Admin Dashboard for Claims Management

As an **admin**, I want to access a dashboard where I can view and manage customer claims, so that I can ensure timely and accurate processing of claims.

Acceptance Criteria:

1. The system must provide an admin dashboard that lists all customer claims with their status (pending, under review, approved, rejected).
2. The admin must be able to filter claims based on status, policy type, or customer.
3. The admin should be able to approve, reject, or request additional information for a claim.
4. The system must log all actions taken by the admin for auditing purposes.
5. The admin should receive notifications when new claims are submitted or require attention.
6. The system must allow the admin to export claim data for reporting purposes.

USE CASE

Use Case 1: View Insurance Products

- **Use Case ID:** UC001
- **Use Case Name:** View Insurance Products
- **Use Case Description:** The customer views available insurance products (auto, home, health, life, and property) to explore and choose a suitable option.
- **Primary Actor:** Customer
- **Supporting Actor:** None
- **Pre-Condition:** Customer has accessed the application or website.
- **Post Condition:** Customer is presented with the details of various insurance products.
- **Main Flow:**
  - **Actor Step:** Customer navigates too the insurance products section.
  - **System Response:** The system displays a list of available insurance products.

  - **Actor Step:** Customer clicks on a product to view more details.
  - **System Response:** The system presents detailed information about the selected insurance product (coverage, exclusions, premium, etc.).
  - **Actor Step:** Customer can choose to return to the product list or proceed with other actions like getting a quote.
  - **System Response:** The system navigates back to the product list or takes the customer to the relevant next page.
- **Alternate Flow:** If the customer is not logged in, they can still view the product list but will be prompted to log in when trying to proceed with personalized actions (e.g., getting a quote).
- **Exception Flow:** If there is a technical error fetching product data, the system displays a message: "Unable to retrieve product information at this time. Please try again later."

- **Additional Requirements:**
    - **Functional:** The system must allow customers to view insurance products even without logging in.
    - **Non-Functional:** The product information must load within 2 seconds.
    - **Database:** The system should retrieve product data from the insurance product database.
    - **Technical:** The system must be optimized to handle multiple requests for product data concurrently.

Use Case 2: Get Real-Time Quote

- **Use Case ID:** UC002
- **Use Case Name:** Get Real-Time Quote
- **Use Case Description:** The customer requests a real-time quote for their desired insurance product by entering personal and policy-related details.
- **Primary Actor:** Customer
- **Supporting Actor:** External Data Source (e.g., vehicle or property database)
- **Pre-Condition:** Customer is logged in and has selected an insurance product.
- **Post Condition:** The customer receives a real-time quote based on their input.
- **Main Flow:**
    - **Actor Step:** Customer selects "Get a Quote" for an insurance product.
    - **System Response:** The system presents a form requesting personal details (e.g., vehicle make/model, home address, age).
    - **Actor Step:** Customer fills out the form and submits it.
    - **System Response:** The system sends the

    data to external sources and retrieves relevant information (e.g., car value, location risk).
  - **System Response:** The system calculates the quote and displays it to the customer, including the premium breakdown.
  - **Actor Step:** Customer can save the quote or proceed to purchase the policy.

- **Alternate Flow:**
  - If the customer cancels the process, the system discards the entered data and returns to the product page.
- **Exception Flow:**
  - If external data sources are unavailable, the system displays a message: "Unable to retrieve data for quote generation. Please try again later."
  - If invalid inputs are provided (e.g., incomplete data), the system prompts the customer to complete the missing fields.

- **Additional Requirements:**
    - **Functional:** The system must calculate quotes in real-time based on customer inputs.
    - **Non-Functional:** The quote must be generated within 2 seconds after data submission.
    - **Database:** The system must store quote request data for auditing and future reference.
    - **Technical:** API calls to external data sources must be optimized for performance and reliability.

Use Case 3: Compare Insurance Products

- **Use Case ID:** UC003
- **Use Case Name:** Compare Insurance Products
- **Use Case Description:** The customer compares multiple insurance products to make an informed choice.
- **Primary Actor:** Customer
- **Supporting Actor:** None
- **Pre-Condition:** The customer has selected multiple insurance products to compare.
- **Post Condition:** The customer views a detailed comparison of the selected products.
- **Main Flow:**
  - **Actor Step:** Customer selects multiple insurance products and clicks "Compare."
  - **System Response:** The system retrieves product details and presents a side-by-side comparison (coverage, premium, exclusions, etc.).
  - **Actor Step:** Customer reviews the comparison.

  - **System Response:** The system allows the customer to make a selection for further actions (e.g., get a quote or purchase a policy).
- **Alternate Flow:**
  - If the customer chooses fewer than two products, the system displays a message prompting them to select at least two products for comparison.
- **Exception Flow:**
  - If product details cannot be retrieved, the system shows an error: "Unable to compare products at this time. Please try again later."
- **Additional Requirements:**
  - **Functional:** The system must allow customers to compare at least 3 insurance products.
  - **Non-Functional:** The comparison page must load within 3 seconds.
  - **Database:** The system retrieves product data from the product database and caches it for performance improvement.

- **Technical:** The comparison feature must be optimized for quick retrieval and display of multiple products.

Use Case 4: Purchase Insurance Policy

- **Use Case ID:** UC004
- **Use Case Name:** Purchase Insurance Policy
- **Use Case Description:** The customer purchases an insurance policy after selecting and reviewing the product.
- **Primary Actor:** Customer
- **Supporting Actor:** Payment Gateway
- **Pre-Condition:** The customer has received a quote or selected a policy.
- **Post Condition:** The policy is purchased, and the customer receives confirmation.
- **Main Flow:**
  - **Actor Step:** Customer clicks "Purchase Policy" after selecting a product or receiving a quote.
  - **System Response:** The system displays the payment page with available payment options.

  - **Actor Step:** Customer selects a payment method and enters payment details.
  - **System Response:** The system submits the payment to the payment gateway for processing.
  - **System Response:** Upon successful payment, the system confirms the purchase, sends an email with policy details, and displays a success message to the customer.
- **Alternate Flow:**
  - If the customer cancels the payment, the system redirects back to the product or quote page without finalizing the transaction.
- **Exception Flow:**

  If the payment fails, the system displays a message: "Payment failed. Please try again or use a different method."
- **Additional Requirements:**
  - **Functional:** The system must integrate with a third-party payment gateway for secure transactions.

    - **Non-Functional:** The payment process must complete within 5 seconds after payment submission.
    - **Database:** The system stores transaction details and updates the policy database upon successful purchase.
    - **Technical:** The system must comply with PCIDSS standards for payment security.

Use Case 5: File and Track Claims

- **Use Case ID:** UC005
- **Use Case Name:** File and Track Claims
- **Use Case Description:** The customer files an insurance claim and tracks its status through the system.
- **Primary Actor:** Customer
- **Supporting Actor:** Admin (Claims Handler)
- **Pre-Condition:** The customer has an active policy and is logged in.
- **Post Condition:** The claim is submitted, and the customer can track its status.

- **Main Flow:**
    - **Actor Step:** Customer navigates to the "File a Claim" section and selects a policy.
    - **System Response:** The system displays a form to collect claim details (incident date, description, supporting documents).
    - **Actor Step:** Customer fills out the form and submits the claim.
    - **System Response:** The system validates the claim and uploads any supporting documents.
    - **System Response:** The system submits the claim for processing and displays a confirmation message to the customer.
    - **Actor Step:** Customer can track the claim status via the dashboard.

- **Alternate Flow:**
    - If the customer cancels the process, the system discards the claim data and returns to the dashboard.

- **Exception Flow:**
    - If there is a validation error (e.g., missing required fields), the system prompts the customer to complete the form before submission.
- **Additional Requirements:**
    - **Functional:** The system must allow customers to upload documents as part of their claim.
    - **Non-Functional:** The claim submission must complete within 3 seconds, and document uploads must be optimized.
    - **Database:** The system stores claim data in the claims database and updates it as the claim progresses.
    - **Technical:** The system must provide secure document upload functionality and track the status of each claim.