## **Functional Requirements Document (FRD)**

**Project Name:** E-Insurance Application  
**Date:** April 16, 2025  
**Prepared By:** [Kuber Seth](mailto:kuber0804.be21@chitkara.edu.in)

| **1. Introduction** |
| --- |

**Purpose:** This document outlines the functional, non-functional, and business requirements for a web-based E-Insurance Application. The purpose of this system is to streamline project planning, task assignment, and collaboration among team members.

**Scope:**

The application allows users to manage multiple projects with functionalities such as task tracking, team communication, file uploads, and dashboard reporting. It targets small to mid-sized teams (up to 100 users per project).

### **2. System Overview**

The E-Insurance Application will offer:

* A centralized dashboard for real-time task and project status.
* Role-based access for users including Project Managers, Team Members, and Viewers.
* Notifications and reminders for due tasks.
* A responsive interface compatible with mobile devices.

### **3. Business Requirements**

| **BR ID** | **Business Requirement** |
| --- | --- |
| BR-01 | Provide real-time visibility of task statuses to users. |
| BR-02 | Enable centralized communication and file sharing. |
| BR-03 | Automate notifications for deadlines and overdue tasks. |
| BR-04 | Offer analytics through dashboards and visual reports. |
| BR-05 | Support effective task delegation with role-based access. |

### **4. Functional Requirements**

| **FR ID** | **Functional Requirement** |
| --- | --- |
| FR-01 | Users shall be able to create, update, and delete tasks. |
| FR-02 | System shall send email notifications on task updates or status changes. |
| FR-03 | Dashboard shall visualize progress via Gantt charts. |
| FR-04 | System shall allow file uploads up to 10 MB per task. |
| FR-05 | Users shall be able to assign tasks to other team members. |
| FR-06 | System shall generate summary reports for each project. |

### **5. User Roles and Permissions**

| **Role** | **Permissions** |
| --- | --- |
| Project Manager | Full access to create/edit/delete tasks, manage users, and view reports. |
| Team Member | Can view and update assigned tasks, upload files, and add comments. |
| Viewer | Read-only access to all project information. |

### **6. Use Case Example**

**Use Case:** Assign a Task  
 **Actors:** Project Manager, System

**Steps:**

1. Project Manager logs into the system.
2. Selects a project from the dashboard.
3. Clicks “Assign Task” button.
4. Fills task details (title, description, due date).
5. Assigns the task to a team member.
6. System sends notification to the assigned member.
7. Task appears in the member’s task list.

### **7. Assumptions and Constraints**

**Assumptions:**

* Users have reliable internet access.
* Users are familiar with basic project management practices.
* System will handle teams of up to 100 users per project.

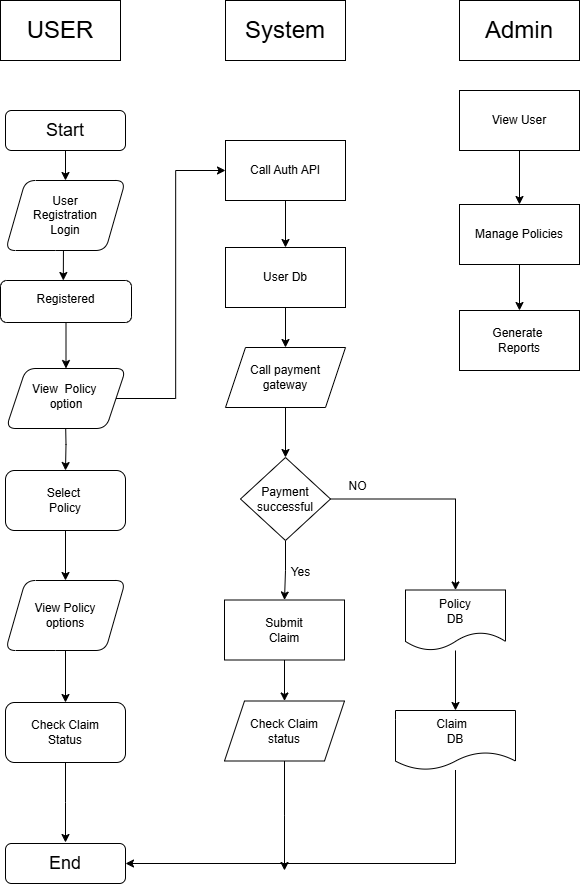
**Constraints:**

* File upload limit is 10 MB per task.
* Application supports only English (initial release).
* Integration with calendars and storage is limited to predefined APIs (e.g., Google Calendar, Dropbox).

### **8. Non-Functional Requirements**

|  |  |
| --- | --- |
| NFR-01 | Dashboards shall load within 3 seconds. |
| NFR-02 | Application shall be accessible on mobile browsers. |
| NFR-03 | System shall maintain 99.9% uptime annually. |
| NFR-04 | Data storage and backup must be secure and encrypted. |
| NFR-05 | Application shall support scalability for up to 100 concurrent users. |

### **9. Functional Workflow / Process Flow:**



### **User Side:** Start the process.

* Registers or logs into the system.
* Views available policy options.
* Selects a desired policy.
* Views policy details again for confirmation.
* Check the claim status.
* Ends the session.

### **System Side:**

* Authenticates users via **Auth API**.
* Stores/fetches user data from the **User DB**.
* Call the **payment gateway** when a policy is selected.
* Checks if **payment is successful**:  
  + **If No**:  
    - Policy info stored in **Policy DB**, but no claim submission.
  + **If Yes**:  
    - Allows users to **submit claims**.
    - Stores claim data in **Claim DB**.

### **Admin Side:**

* Views registered users.
* Manages policy details (add/update/delete).
* Generates reports for analysis and tracking.