# Expense Management System – Salesforce Project Documentation

## 1. Project Overview

The Expense Management System is a beginner-friendly Salesforce project designed to simulate a real-world organizational expense workflow. The project combines both administrative (declarative) and developer (programmatic) features of Salesforce to model how companies manage employee expenses through validation, automation, and approval mechanisms.

The system allows employees to create expense records, submit them for approval, and enables managers to approve or reject requests. The project focuses on implementing realistic business rules and automation logic commonly used in enterprise Salesforce environments.

---

## 2. Objectives of the Project

• Understand Salesforce data modeling using custom objects and fields
• Implement data validation and integrity rules
• Design automation using Record-Triggered Flows
• Configure Approval Processes
• Enforce business logic using Apex Triggers
• Gain practical exposure to profiles, roles, permissions, and security model

---

## 3. Key Features Implemented

• Custom Expense Object • Validation Rules for data integrity • Record-Triggered Flow Automation • Approval Process Workflow • Automated Approved Date Handling
• Email Notification for Approvals • Apex Trigger for Edit Restrictions

---

## 4. Concepts Covered

### 4.1 Salesforce Admin Concepts

• Custom Object Creation (Expense__c) • Custom Fields (Amount, Category, Expense Date, Status, Approved Date) • Page Layout Design • Validation Rules • Profiles & Permissions • Role Hierarchy • User Creation & Manager Mapping • Tab Visibility & Object Access • Email Templates

### 4.2 Salesforce Developer Concepts

• Record-Triggered Flows • Approval Process Configuration • Entry Criteria Logic • Field Update Actions • Apex Trigger Development • Trigger Context Variables (Trigger.new, Trigger.oldMap) • Business Logic Enforcement • Conflict Resolution Between Automation Layers

---

## 5. Business Relevance / Use Cases

Expense management workflows are widely used in organizations to:

• Control employee spending • Ensure financial transparency • Implement approval hierarchies • Maintain audit trails • Prevent unauthorized modifications • Automate reimbursement processes

This type of system is commonly implemented in corporate ERP and financial management applications.

---

## 6. Development Approach

### 6.1 Data Model Design

A custom object (Expense__c) was created to represent expense records. Fields were added to capture relevant expense information.

Key Fields: • Expense Name
• Amount
• Category
• Expense Date
• Status
• Approved Date

---

### 6.2 Admin (Declarative) Implementation

#### Object & Fields

• Created Expense custom object • Added business-related fields

#### Validation Rules

Implemented rules such as: • Category cannot be empty • Expense Date cannot be in the future • Approved Date cannot be manually edited • Status controlled by approval process

#### Security Configuration

• Created Employee Profile • Created Manager Profile • Configured Object Permissions • Enabled Tab Visibility • Created Role Hierarchy (Manager → Employee) • Created Users • Assigned Manager Field

#### Approval Process

• Entry Criteria: Status = Submitted • Approver: Manager of Record Submitter • Final Approval Actions:

- Status → Approved
- Approved Date → Today • Final Rejection Actions:
- Status → Rejected

#### Email Notifications

• Created Approval Email Template • Embedded Approval URLs

---

### 6.3 Developer (Programmatic & Automation) Implementation

#### Record-Triggered Flows

Flow 1 – Auto Submit Expense • Trigger: Record Created • Action: Status → Submitted

Flow 2 – Set Approved Date • Trigger: Record Updated • Condition: Status Changed → Approved • Action: Approved Date → Today

#### Apex Trigger

Implemented logic to prevent editing approved expenses while allowing system updates:

• Trigger Type: Before Update • Logic: Compare old vs new values • Block modification of key business fields after approval

---

## 7. Challenges Faced & Resolutions

### Challenge 1 – Record Read-Only Errors

Issue: Attempting updates in after triggers Resolution: Converted logic to before triggers

### Challenge 2 – Automation Conflicts

Issue: Flow + Approval Process both updating Status Resolution: Clearly separated responsibilities

### Challenge 3 – Validation Rule Conflicts

Issue: Rules blocking system updates Resolution: Added PRIORVALUE logic

### Challenge 4 – Approval Field Updates Not Working

Issue: Missing Final Approval Actions Resolution: Added explicit Field Updates

### Challenge 5 – Email Delivery Issues

Issue: Emails routed to spam / links restricted Resolution: Verified deliverability & used text templates  after successful delivery changed to custon HTML template.

---

## 8. Key Learnings

• Difference between Approval Engine & Field Updates • Importance of separating automation responsibilities • Understanding Trigger Execution Context • Designing conflict-free automation architecture • Salesforce Security Model (Profiles vs Roles) • Practical debugging of real-world Salesforce issues

---

## 9. Final System Behavior

• Employee creates expense → Auto Submitted • Expense enters approval workflow • Manager approves/rejects • Status updated accordingly • Approved Date populated automatically • Approved records protected from edits

## 10. Conclusion

This project provided a comprehensive introduction to Salesforce’s declarative and programmatic capabilities. By combining Flows, Validation Rules, Approval Processes, and Apex Triggers, the system successfully models a realistic enterprise expense workflow.

The project strengthened practical understanding of Salesforce automation design, business logic enforcement, and platform behavior.