Darshan University

A Project Report on

"Expense Management System"

Under the subject

Software Engineering (2301CS405)

B. Tech, Semester – IV

Computer Science & Engineering Department

Submitted By

Student Name: Dhol Namra Enrollment No.: 23010101407

Academic Year

(2024-2025)

Internal Guide	Dean-DIET
Prof. R. B. Gondaliya	Dr. Gopi Sanghani
Darshan University	Darshan University

Computer Science & Engineering Department
Darshan University

DECLARATION

We hereby declare that the SRS, submitted along with the **Software Engineering (2301CS405)** for entitled **"Expense Management System"** submitted in partial fulfilment for the Semester-5 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Department to Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of R. B. Gondaliya and that no part of any of report has been directly copied from any students' reports, without providing due reference.

Dhol Namra

Student's Signature

Date: ___________

Computer Science & Engineering
Department
Darshan University

CERTIFICATE

This is to certify that the SRS on **"Expense Management System" has** been satisfactorily prepared by **Dhol Namra (23010101407)** under my guidance in the fulfillment of the course **Software Engineering (2301CS405)** work during the academic year 2024-2025.

Internal Guide	Dean-DIET
Prof. R. B. Gondaliya	Dr. Gopi Sanghani
Darshan University	Darshan University

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to my project guide Prof. R. B. Gondaliya and all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co-ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

Dhol Namra

ABSTRACT

The Expense Management System is a comprehensive tool designed to automate the management of employee expenses across various roles, including Admin, Manager, Chartered Accountant (CA), and Employee. The system allows employees to submit expense reports with receipts, while managers and CAs review, audit, and ensure compliance with financial regulations. Admins manage user roles and configure system-wide settings such as approval workflows and expense categories. Key features include budget monitoring, expense tracking, real-time notifications, and the ability to generate detailed financial reports. The system improves organizational efficiency by streamlining approval processes, reducing manual errors, and ensuring expenses align with company policies and tax regulations. Main purpose of this system is to reduce human efforts as much as possible.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Product perspective

The Expense Management System is designed as an independent application that integrates seamlessly with existing financial systems and workflows. It provides a centralized platform for managing, tracking, and reporting employee expenses, offering scalability and customization based on organizational needs.

1.2 Product features

1.2.1 There are three different users who will be using this product

- Admin – Responsible for managing user roles, system configurations, and overseeing the entire system.
- Manager – Reviews and approves or rejects employee expense submissions.
- Chartered Accountant (CA) – Manages financial records, audits expense reports, and ensures compliance with tax regulations.
- Employee – Submits expenses for approval and tracks reimbursement status. The features that are required for the Librarian are:

1.2.2 The features that are required for the Admin are:

- **User Management** – Create, edit, and delete user accounts and assign roles (e.g., Manager, Employee, CA).
- **System Configuration** – Customize system settings, such as approval workflows, expense categories, and reimbursement policies.
- **Budget Management** – Set and monitor budgets for departments or individual employees.
- **Access Control** – Manage permissions for different users and ensure data security.
- **Audit Trail** – Track all actions within the system for transparency and compliance.
- **Reporting** – Generate detailed reports on expenses, budgets, and system usage.

1.2.3 The features that are required for the Manager are:

- **Expense Approval/Review** – Review, approve, or reject expense reports submitted by employees.
- **Expense Tracking** – Monitor the status of expenses within their team or department.
- **Budget Monitoring** – Ensure that expenses stay within allocated budgets and provide feedback if limits are exceeded.
- **Reporting** – Generate reports on team or department expenses for analysis and compliance.
- **Commenting/Feedback** – Provide feedback or request additional details on submitted expense reports before approval.

1.2.4 The features that are required for the CA are:

- **Financial Auditing** – Review and audit expense reports for accuracy and compliance with financial regulations.
- **Tax Compliance** – Ensure that all expenses meet tax regulations and generate reports for tax filing purposes.
- **Expense Verification** – Verify the legitimacy of expenses and ensure they align with the company's financial policies.
- **Reporting** – Generate detailed financial reports, including tax deductions, expense trends, and compliance reports.

1.2.5 The features that are required for the Employee are:

- **Expense Submission** – Submit expense reports with receipts and required documentation for approval.
- **Expense Tracking** – Track the status of submitted expenses (e.g., pending, approved, or rejected).
- **Expense History** – View the history of all submitted expenses, including past approvals and reimbursements.
- **Receipt Upload** – Attach receipts and other proof of expenses for validation.
- **Notifications** – Receive alerts for approvals, rejections, or requests for additional information on submitted expenses.

1.3 Functional Requirement

1.3.1 Admin

- Add User: Add accounts for Manager, CA and Employees
- Update User: Update accounts for Manager, CA and Employees
- Delete User: Delete accounts for Manager, CA and Employees View Expense Report: View detailed reports and analytics on expense
- System Settings:
  - Configure system-wide settings such as expense categories, limits, and approval workflow.
  - Set up and manage company policies related to expenses
- Dashboard and Analytics:
  - Access a comprehensive dashboard with visual analytics on expenses, approvals, and budget usage.
  - Customize dashboard views and reports based on specific criteria (e.g., department, time period).
- Budget Management: o Set and manage budgets for different departments and projects.
  - Monitor budget utilization and receive alerts when limits are approached or exceeded.
- Expense Policy Management:
  - Define and update company expense policies.
  - Ensure policies are communicated and enforced across the system.
- Notifications and Alert:

  o Configure and receive notifications and alerts for various events (e.g., large expense submissions, budget overruns, policy violations).
- Role-Based Access Control:
  - Define and manage access control policies to ensure users only access data and functionality relevant to their roles.
  - Audit and monitor user activities to ensure compliance with access control policies.
- Customizable Approval Workflows:
  - Create and customize approval workflow for different types of expenses and departments.
  - Allow multi-level approvals and define specific approvers for each level.
- Data Export:

  o Export expense data in various forms (e.g., CSV, Excel, PDF) for further analysis and reporting.

1.3.2 Manager

- Expense Submission:
  - Submit expenses for approval to the CA.

- Expense Review:
  - Review and approve or reject expenses submitted by the Employees under their supervision.
  - Provide feedback or request additional information on submitted expenses.
- Reporting:
  - Generate and view reports on expenses for their team.
  - Monitor expenses trends and identify any anomalies or areas of concern.
- Team Expense Overview:
  - View a consolidated overview of all expenses submitted by team members.
  - Filter and sort expenses by various criteria (e.g., data, type, employee).
- Expense prediction:
  - prediction future expenses based on historical data and planned activities.
  - Adjust team budgets and expense limits accordingly.
- Expense Compare:
  - Compare current expenses with past periods to identify spending patterns and areas for improvement.
  - Generate comparison reports to visualize expense trends.
- Delegation of Approval:
  - Delegate approval authority to other team members during absences of high workload periods.
  - Track and manage delegated approvals to maintain accountability.
- Expense Policy Execution:
  - Ensure team members adhere to company expense policies by implementing automatic checks and validations during expense submission.
  - Provide feedback and guidance to team members on policy compliance.
- Training and Support:
  - Provide training materials and support resources for team members on how to use the expense management system.

1.3.3 Charter Accountant

- Financial Reporting:
  - Generate and view detailed financial reports on expenses.
  - Prepare financial statements and ensure accurate accounting of expenses.
- Audit:
  - Conduct audits of expenses to ensure accuracy and compliance.
  - Identify and report any discrepancies or fraudulent activities.
- Tax Calculation:

  - Automatic Tax Calculation: The system should automatically calculate applicable taxes for each expense based on predefined tax rules.
  - Manual Tax Adjustments: Allow the CA to manually adjust tax calculations when necessary Tax Reporting
  - Generate Tax Reports: Generate detailed tax reports for internal review and external compliance. Reports should include a summary of taxes paid categorized by expense type, date, and department.
- Tax Documentation:
  - Maintain Tax Records: Store digital copies of all receipts and tax-related documents and ensure that all necessary documentation is available for tax audits and review.
  - Audit Trail: Keep a detailed audit trail of all tax-related transactions, adjustments, and approvals and allow auditors to trace and verify tax calculations and compliance.
- Expense Verification:
  - Verify the accuracy and legitimacy of submitted expenses.
  - Cross-check expenses against receipts and other supporting documentation.
- Financial Analysis:
  - Perform in-depth financial analysis on expense data in identifying trends, inefficiencies, and opportunities for cost savings.
  - Provide insights and recommendations to the Admin and Management for improving expense management.
- Expense Categorization:
  - Categorize expenses according to accounting standards and company policies.
  - Ensure consistency in categorization for accurate financial reporting and analysis.
- Tax Compliance:
  - Ensure Compliance with Tax Laws: Ensure that all expenses comply with relevant local, state, and federal tax laws. Automatically update tax rules and rates in the system to reflect changes in tax regulations.

1.3.4 Employee

- Expense Submission:
  - Submit expenses for approval to their manager.
  - Upload receipts and documentation for expenses.
- Expense Tracking:
  - Track the status of submitted expenses.
  - Receive notifications on expense approval or rejection.
- Expense Limits:
  - View personal expense limits and ensure submission are within the allowed limits. Request adjustments to limits if necessary.
- Expense History:
  - View a detailed history of all submitted expenses, including approval status and comments from reviewers.
  - Filter and sort expense history by various criteria (e.g., date, type, status).
- Expense Templates:
  - Use predefined templates for common types of expenses (e.g., travel, meals, office supplies) to streamline the submission process.

- o Customize templates for individual needs.

1.4 Address and resolve issues faced by team members in submitting or managing expenses. Non-Functional Requirement

1.4.1 Usability:

- The UI should be simple enough for everyone to understand and get the relevant information without any special training. Different languages can be provided based on the requirements.

1.4.2 Accuracy:

- The data stored about the books and the fines calculated should be correct, consistent, and reliable.

1.4.3 Availability:

- The System should be available for the duration when the library operates and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

1.4.4 Maintainability:

- The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible. In addition to this, the software must also be portable.

2 Design and Implementation Constraints

2.1 Use case diagram

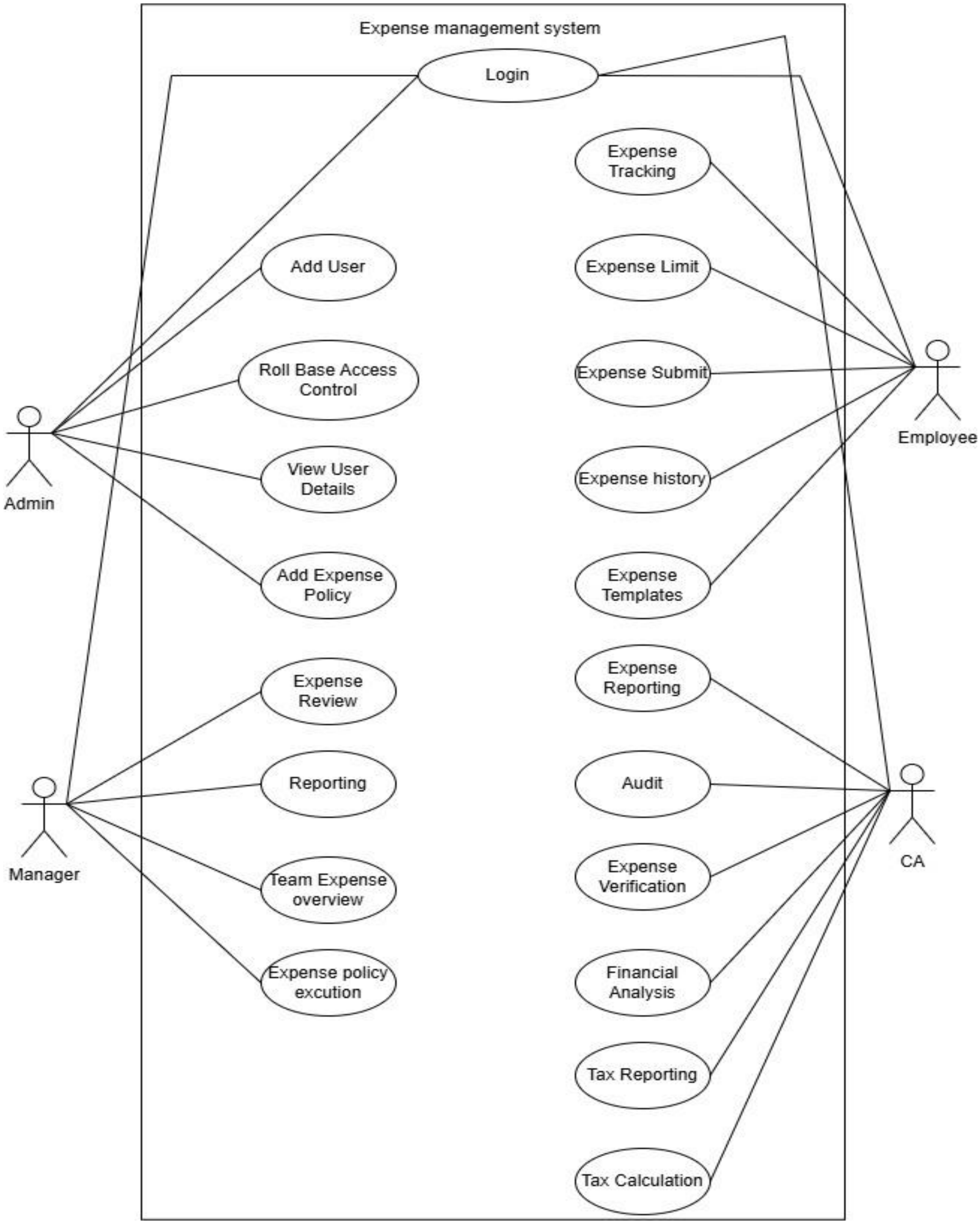

Figure 2.1-1 Use case diagram for Expense management system

2.2 Activity diagram and Swimlane diagram

Figure 2.2-1 Activity diagram for employee expense submission

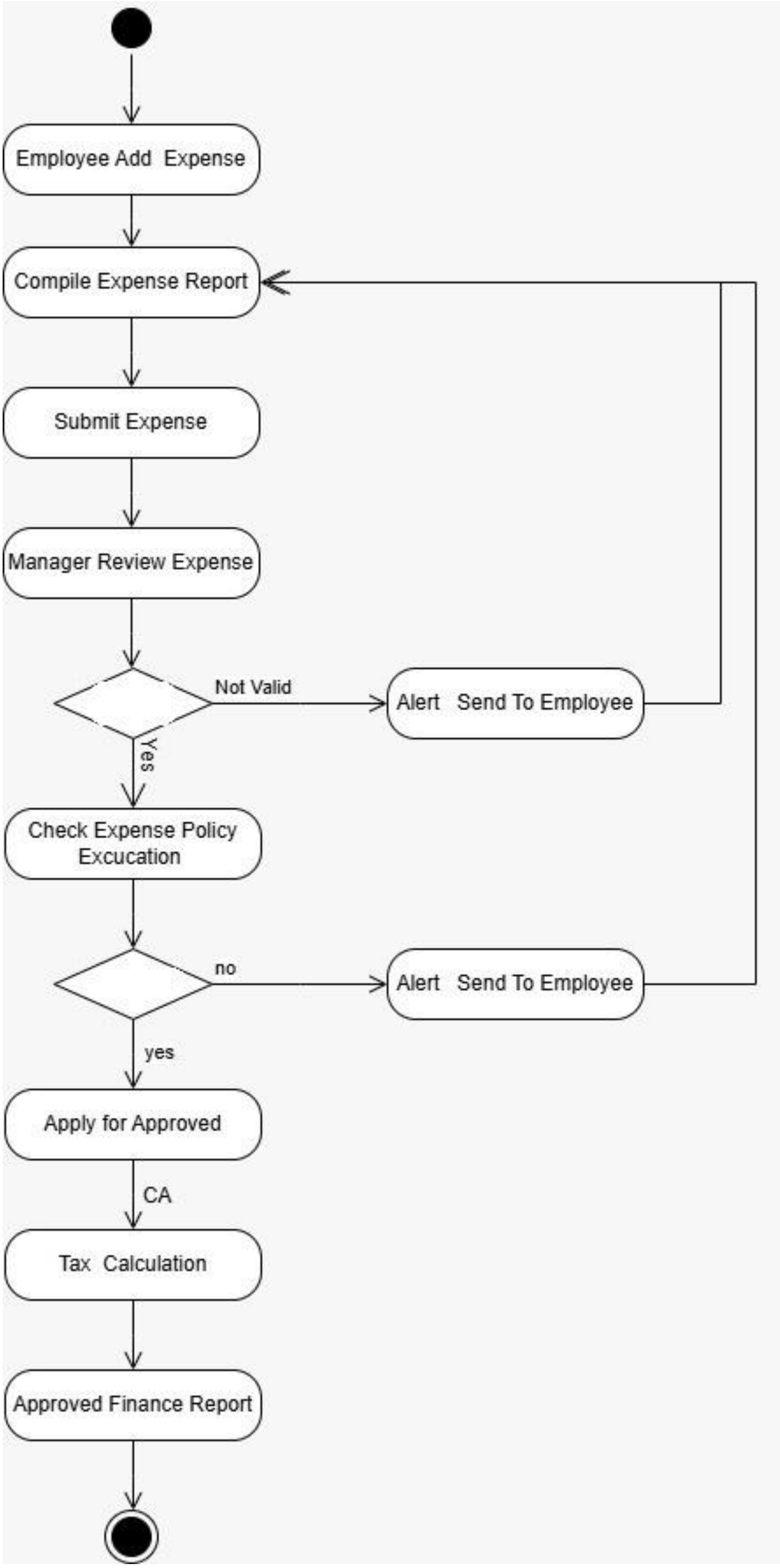

Figure 2.2-2 Activity diagram for approval CA

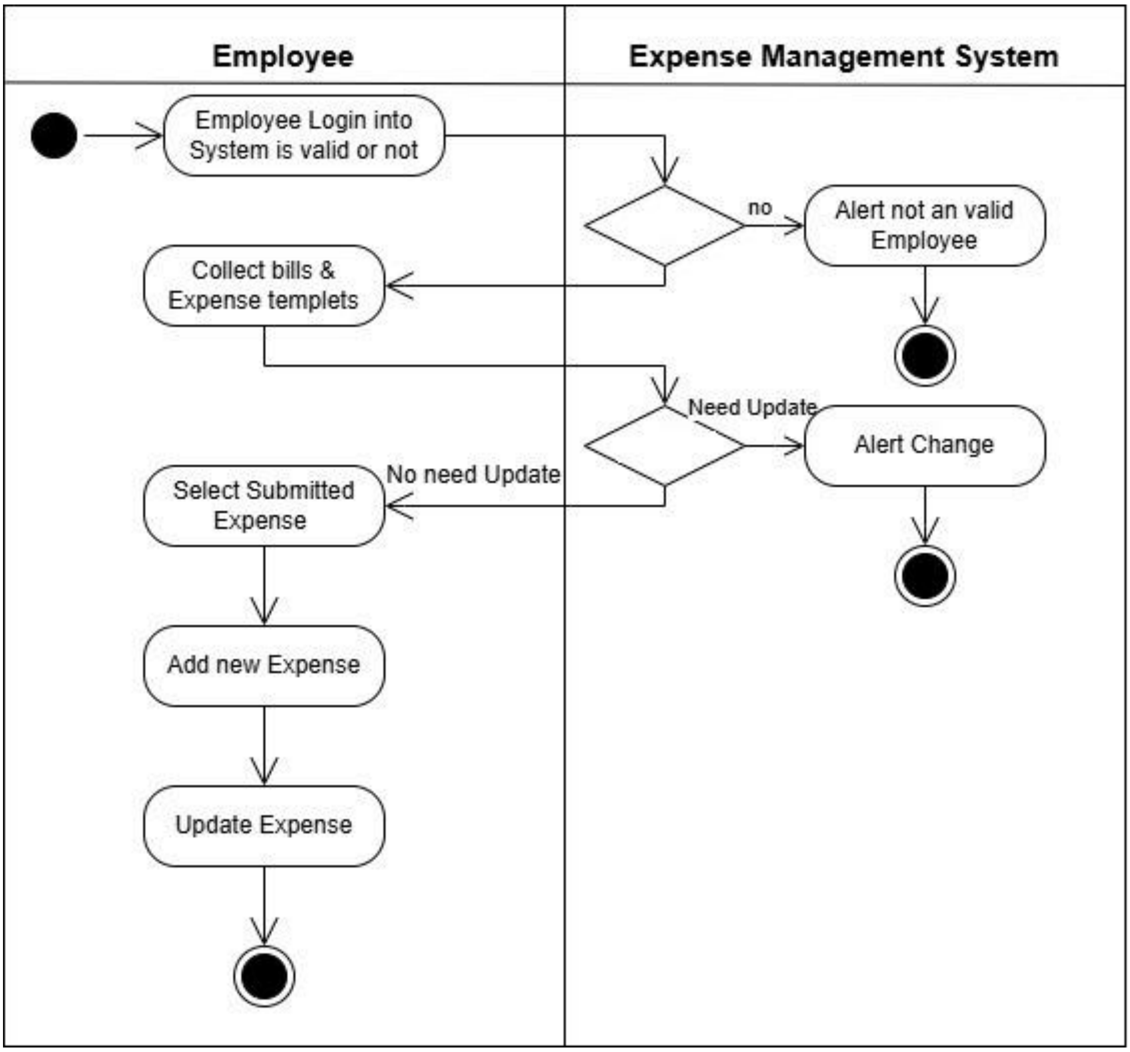

Figure 2.2-3 Swimlane diagram for Employee Add Expense

2.3 Sequence diagram

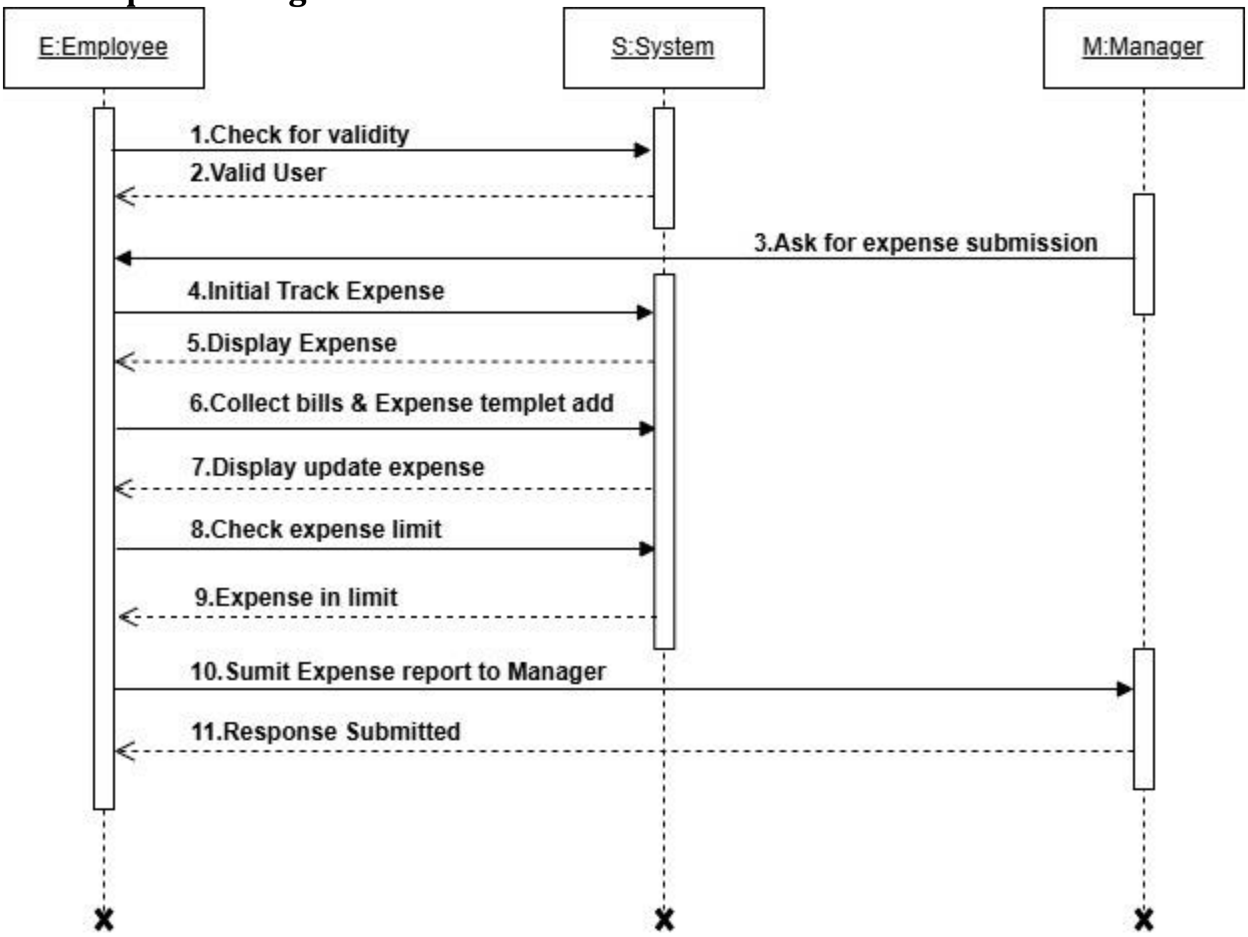

Figure 2.3-1 Sequence diagram for Expense Submit

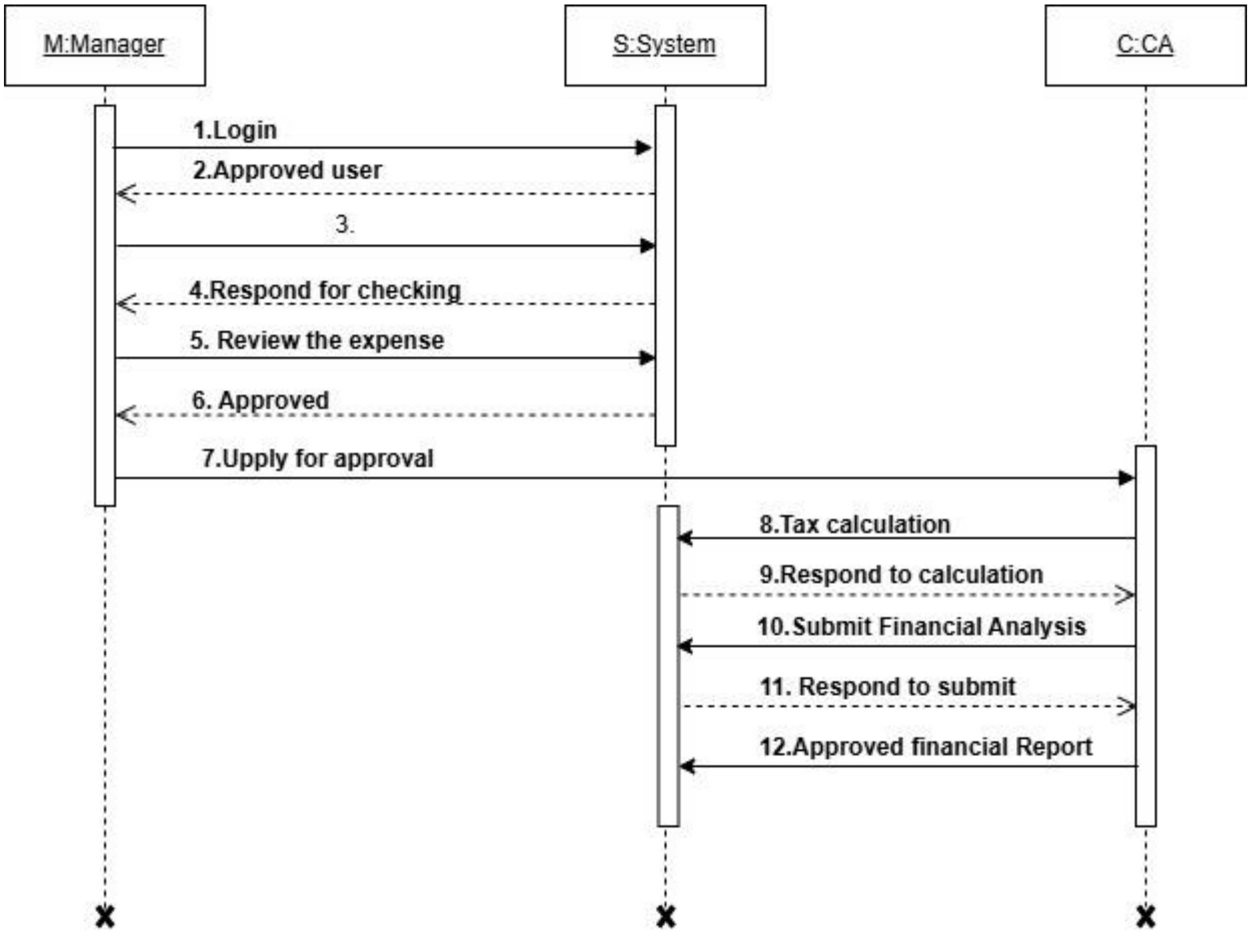

Figure 2.3-2 Sequence diagram for CA Aproval

2.4 State diagram

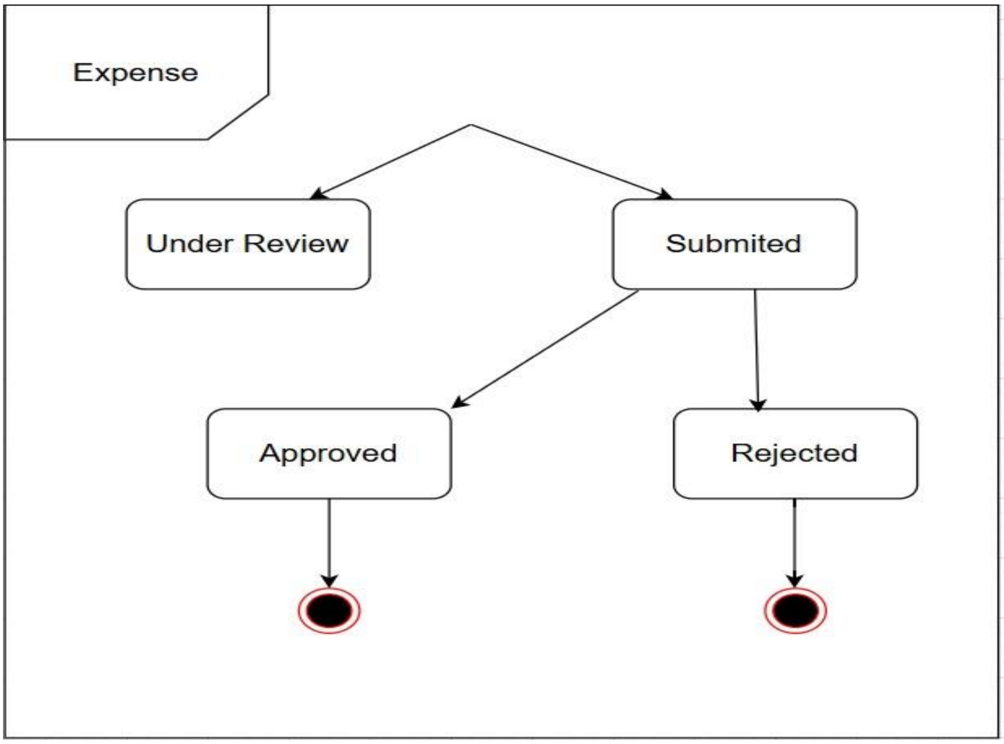

Figure 2.4-1 State diagram of Expense

2.5 Class diagram

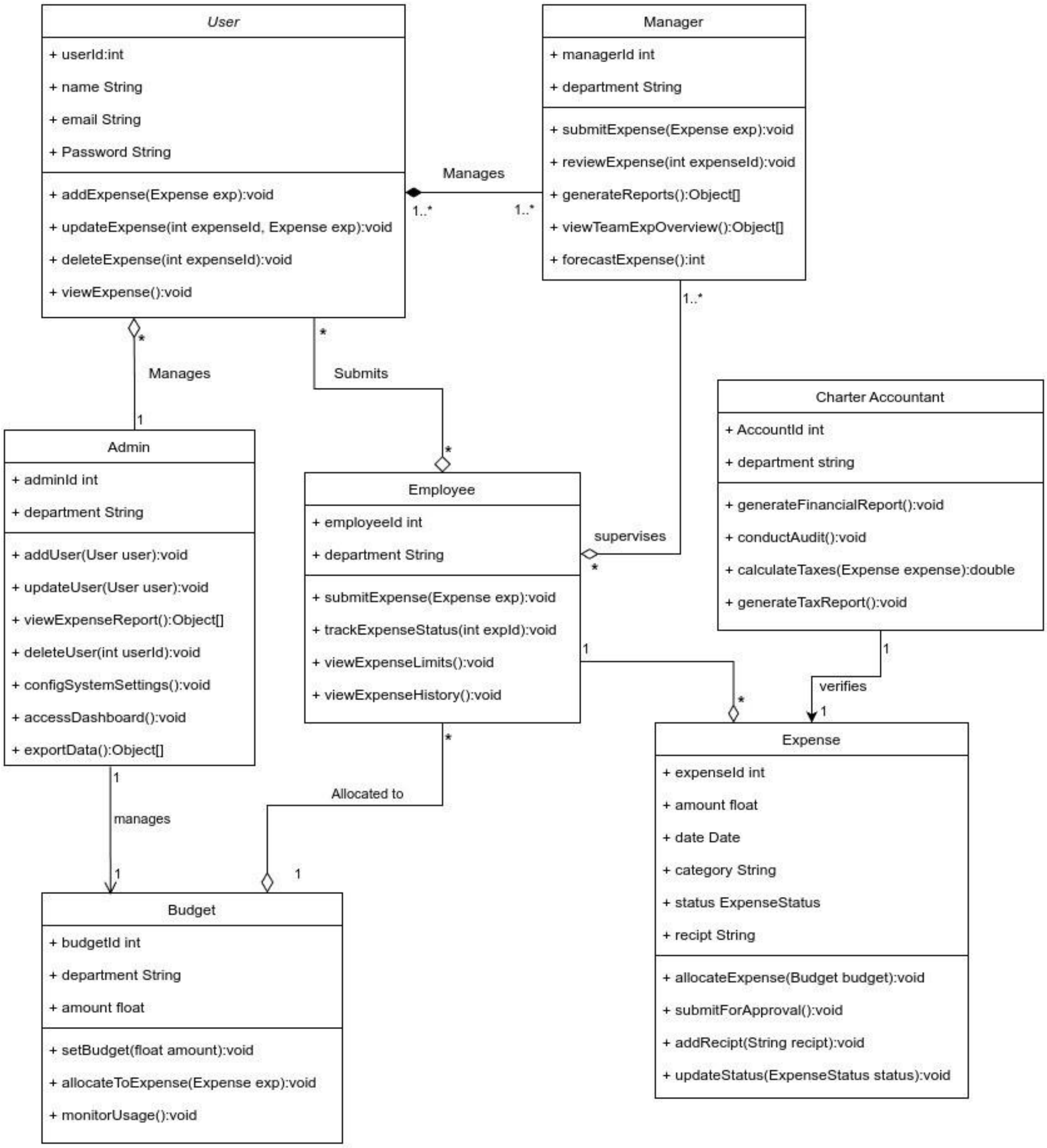

Figure 2.5-1 Class diagram for Expense management system

2.6 Data flow diagram

2.6.1 Context diagram (level-0)

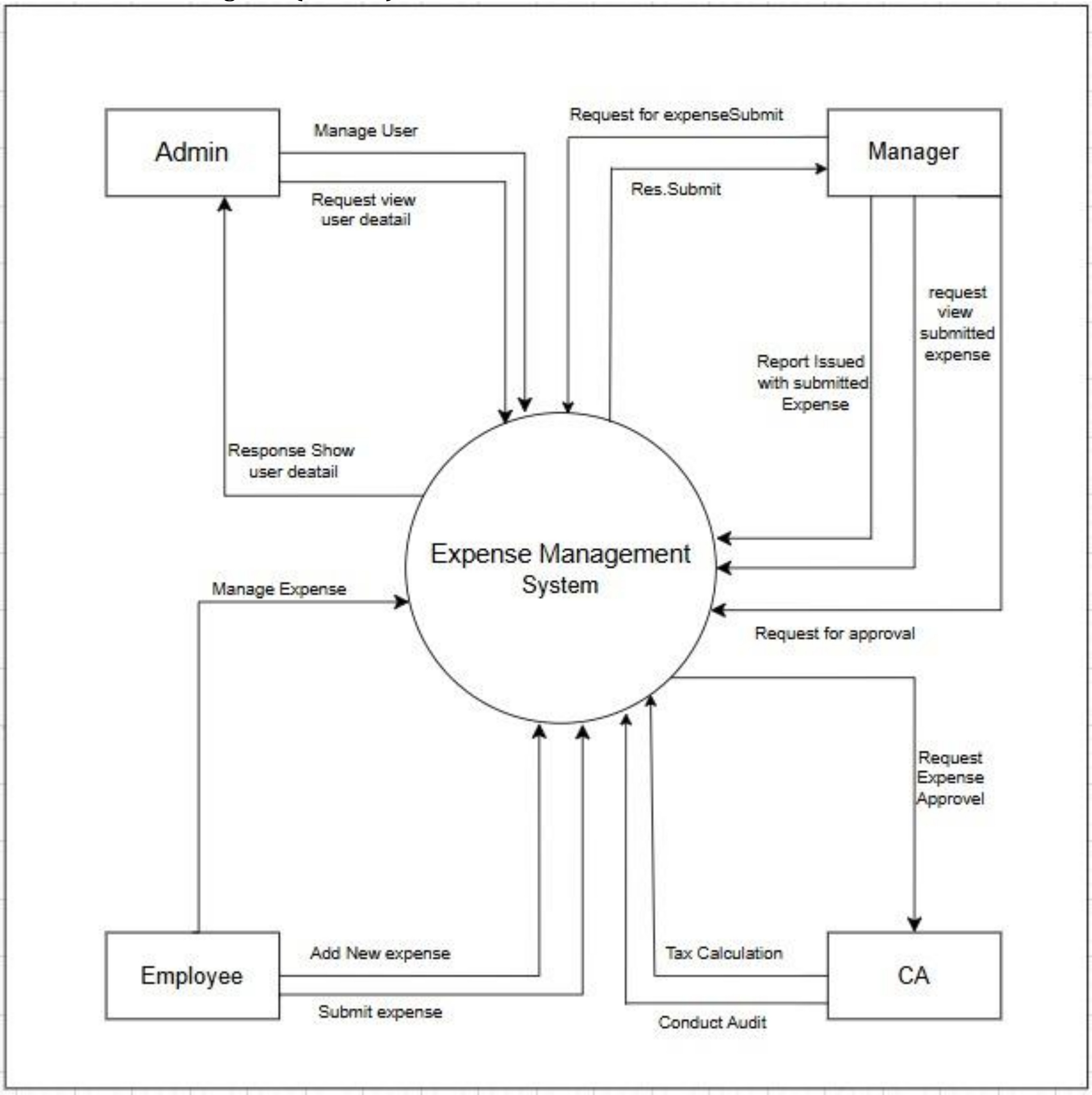

Figure 2.6-1 Context diagram for Expense management system

2.6.2 DFD Level-1

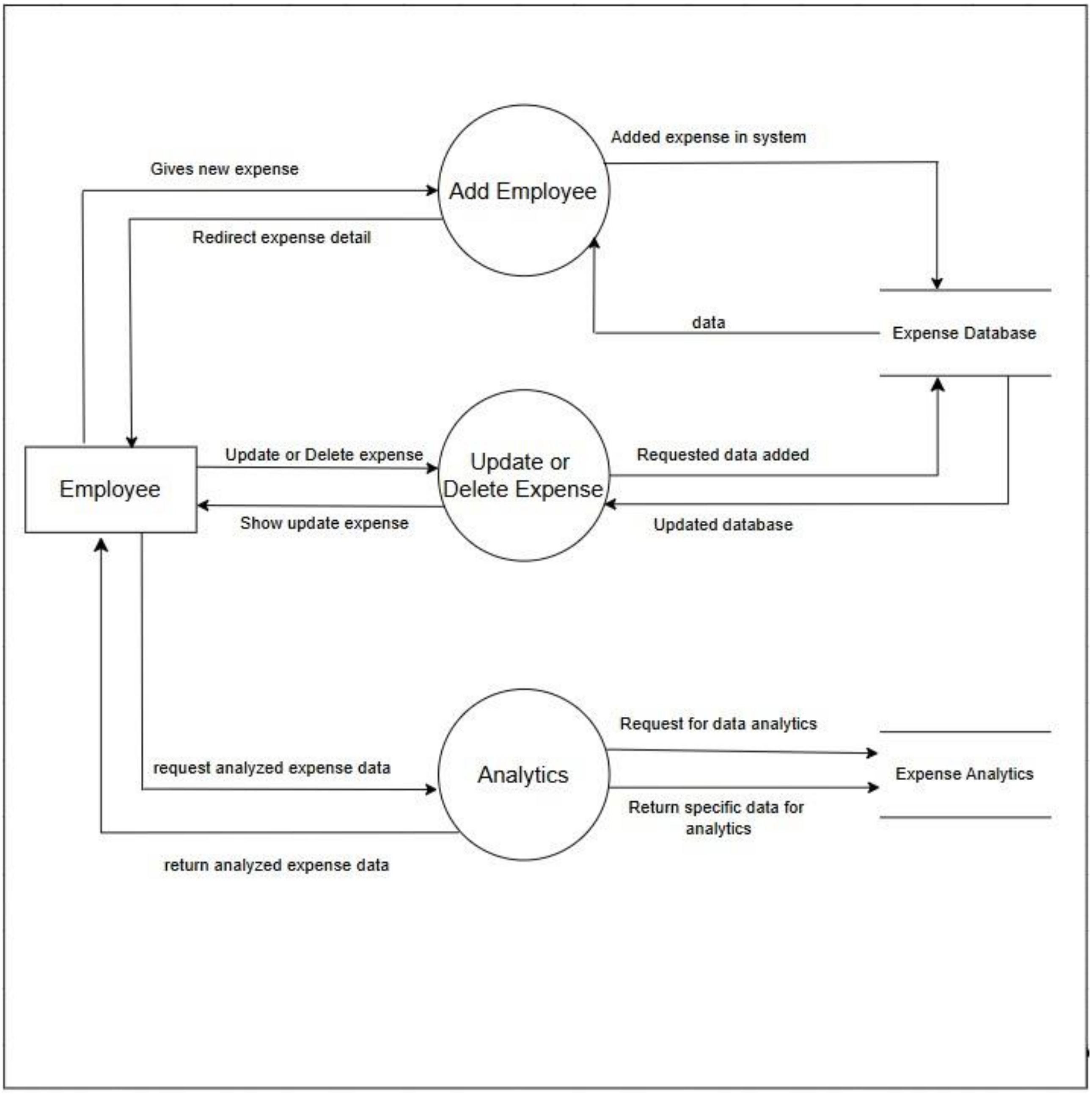

Figure 2.6-2 DFD level-1 for Expense management system

3 External interface requirement (Screens)

3.1 Screen-1: Add New User Form

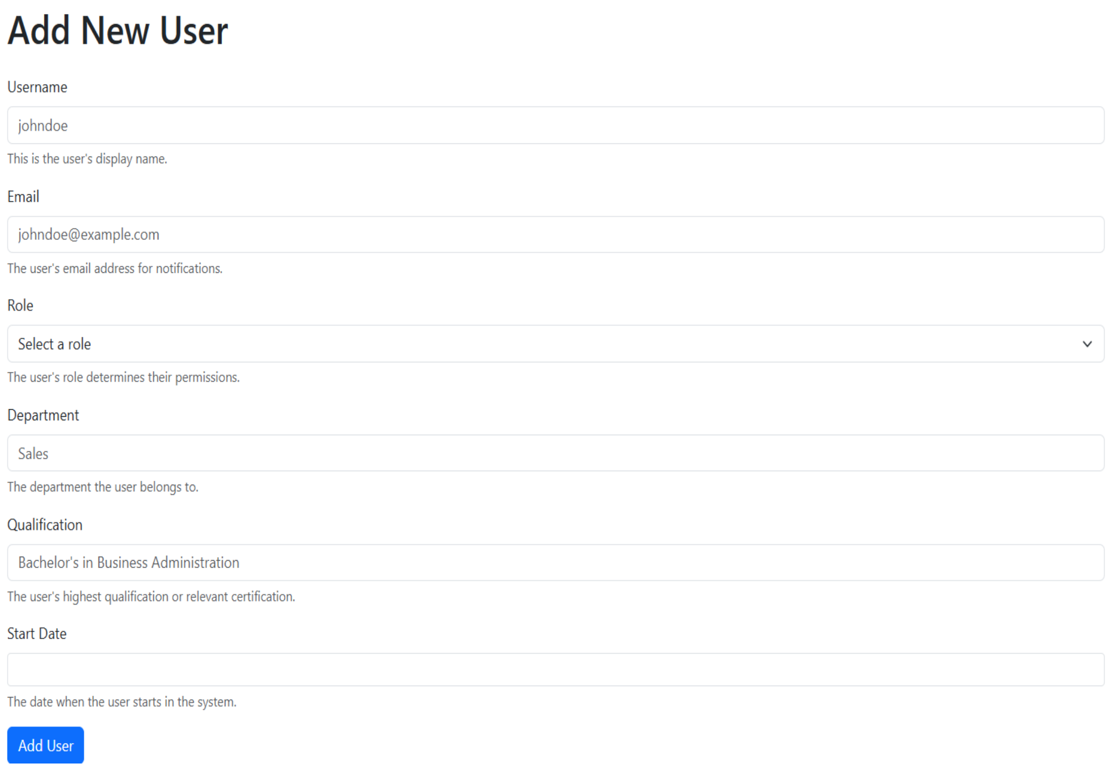

Figure 3.1-1 Screen-1: Add New User Form

Purpose: This form will allow the target end-users to register in the system. To register , the following information will be encoded in the system.

3.1.1 Screen element of : Add New User Form

Sr.	Screen Element	Input Type	O/M	1/N	Description
1	Username	Textbox	M	1	Username field should be editable and accept the Username.
2	Email	Textbox	M	1	Email field should be editable and accept the email with proper format.
3	Role	Dropdown	M	1	Role field should be selected and accepted from dropdown.
4	Department	Dropdown	M	1	Department field should be selected and accepted from dropdown.
5	Qualification	Textbox	M	1	Qualification field should be editable and accept the Qualification.
6	Start date	Textbox	------	------	Start date field should be editable and accept the start date.
7	Add user	Button	------	------	Add user is a button for store the entered data into database.

3.2 Screen-2: Expense Review

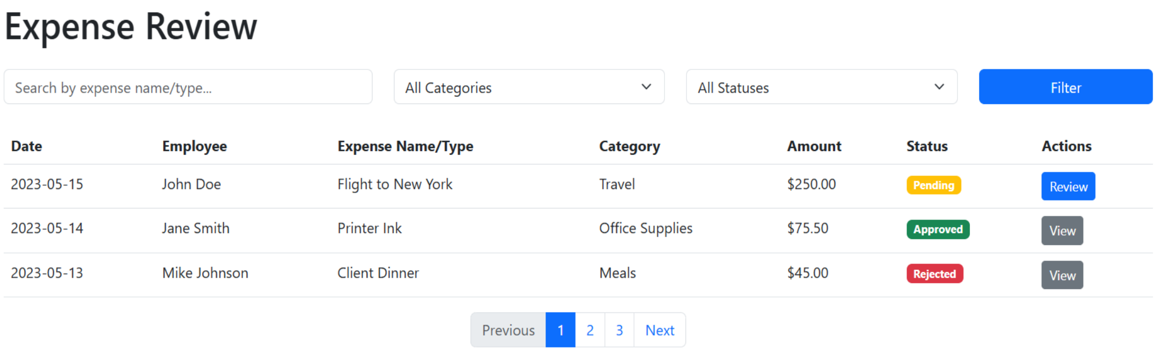

Figure 3.2-1 Screen-2: Expense Review

Purpose: This form will be used by the system's users to access records and features of the system. The users will input the correct combination of their username and password to be able to login to the system.

3.2.1 Table 3.2-1 Screen element of Expense Review

Sr.	Screen Element	Input Type	O/M	1/N	Description
1	Search expense	Textbox	M	1	Search expense field should be search expense by name or it's type
2	All Categories	Dropdown	M	1	All categories field should be select categories from drop down
3	All Status	Dropdown	M	1	All Status field should be select Status from drop down
4	filter	button	------	------	Filter is a button for apply filter on expense from Search expense ,All categories, All Status.
5	Review	button	------	------	Review is a button for review expense and set the status.
6	--	--	------	------	---

3.3 Screen-3: Add tax calculation

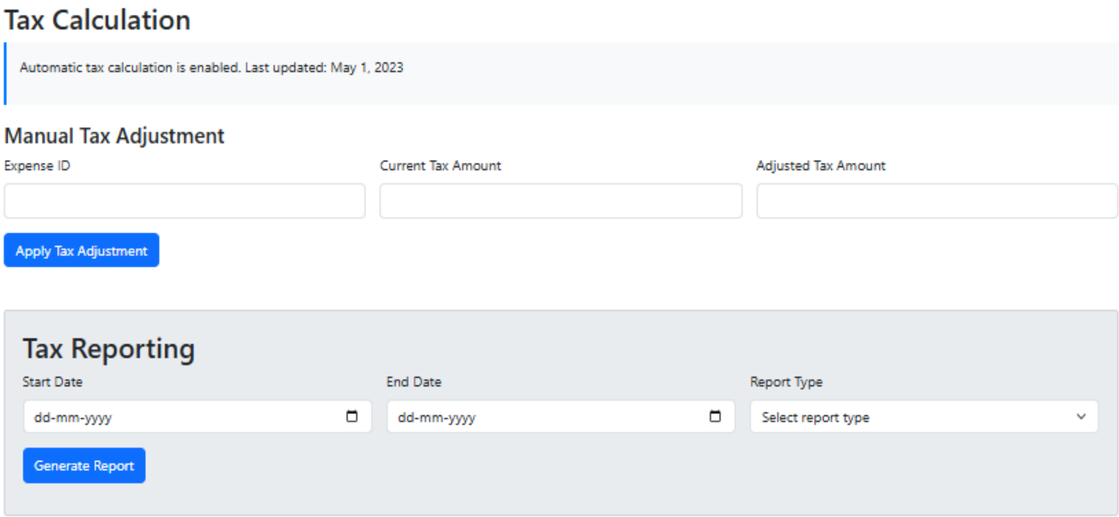

Figure 3.3-1 Screen-3: Add tax calculation

Purpose: This module will allow the system administrator to add, edit, update or delete borrowers of book. The admin can add borrower information and manage it.

3.3.1 Table 3.3-1 Screen element of Add tax calculation

Sr.	Screen Element	Input Type	O/M	1/N	Description
1	Expense ID	Textbox	M	1	Expense ID field should be editable and accept alphanumeric values.
2	CurrentTax Amount	Textbox	M	1	Current Tax Amount field should be editable and accept only numeric values.
3	AdjustedTax Amount	Textbox	M	1	Adjusted Tax Amount field should be editable and accept only numeric values.
4	ApplyTax Adjustment	Button	—	—	Button to apply the manual tax adjustment.
5	Start Date	Datetime	M	1	Start Date field should be selectable using a date picker.
6	End Date	Datetime	M	1	End Date field should be selectable using a date picker.
7	Report Type	Dropdown	M	1	Dropdown to select the type of tax report to generate.
8	Generate Report	Button	—	—	Button to generate the tax report based on selected filters.

3.4 Screen-4: Submit Expense

Figure 3.4-1 Screen-4: Submit Expense

3.4.1 Table 3.4-1 Screen element of Submit Expense

Sr.	Screen Element	Input Type	O/M	1/N	Description
1	Expense Date	Datetime	M	1	Select the date when the expense occurred.
2	Expense Category	Dropdown	M	1	Select an expense category from a predefined list.
3	Amount	Textbox	M	1	Enter the expense amount in numeric format.
4	Description	Textarea	O	1	Provide a brief description of the expense.
5	Receipt	File Upload	O	1	Upload a photo or scan of the receipt if available.
6	Payment Method	Dropdown	M	1	Select the payment method used for the expense.
7	Billable to Client	Checkbox	O	1	Mark if the expense is billable to a client.
8	Submit Expense	Button	—	—	Button to submit the expense details.

3.5 Screen-5: Financial Report Generator

Figure 3.5-1 Screen-5: Financial Report Generator

3.5.1 Table 3.5-1 Screen element of Financial Report Generator

Sr.	Screen Element	Input Type	O/M	1/N	Description
1	Fund Amount	Textbox	M	1	Enter the total fund amount in numeric format.
2	Equity Amount	Textbox	M	1	Enter the equity amount in numeric format.
3	Report Date	Datepicker	M	1	Select the report generation date using a date picker.
4	Current Ratio	Textbox	O	1	Enter the current ratio in numeric format.
5	Sales Asset Ratio	Textbox	O	1	Enter the sales asset ratio in numeric format.
6	Generate Report	Button	—	—	Button to generate the financial report.

4 Database design

4.1 List of Tables

- User
- Employee
- Expense
- Budget
- Manager

4.1.1 Table 4.1-1 Table: User

Column	Data Type	Null	Constraint	Description
userId	int	Not Null	Primary Key	Auto Increment
name	Varchar (50)	Not Null	-	Name of user
email	Varchar (100)	Allow Null	-	Email of user
Password	Varchar (256)	Not Null	-	Password of account
Column	**Data Type**	**Null**	**Constraint**	**Description**

4.1.2 Table 4.1-2 Table: Employee

Column	Data Type	Null	Constraint	Description
employeeId	int	Not Null	Primary Key	Auto Increment
departmentId	int	Not Null	Foreign Key	Reference to department table
employeeName	Varchar (100)	Not Null	-	Name of employee
employeeEmail	Varchar (100)	Not Null	-	Email of employee
contactNo	Number(10,0)	Allow Null	-	Contact no of employee
address	Varchar (256)	Allow Null	-	Address of employee

4.1.3 Table 4.1-3 Table: Expense

Column	Data Type	Null	Constraint	Description
expenseId	int	Not Null	Primary Key	Auto Increment
amount	int	Not Null	-	Amount of expense
date	Date	Not Null	-	Date of issue
category	Varchar (50)	Not Null	-	Category of expense
status	Varchar (50)	Not Null	-	Status of expense
receipt	Varchar (100)	Allow Null	-	Receipt of expense

4.1.4 Table 4.1-4 Table: Manager

Column	**Data Type**	**Null**	**Constraint**	**Description**
managerId	int	Not Null	Primary Key	Auto Increment
departmentId	int	Not Null	Foreign Key	Reference to Department Table
managerName	Varchar (100)	Not Null	-	Name of manager
email	Varchar (100)	Not Null	-	Email of manager
Address	Varchar (256)	Allow Null	-	Address of manager

4.1.5 Table 4.1-5 Table: Budget

Column	**Data Type**	**Null**	**Constraint**	**Description**
budgetId	int	Not Null	Primary Key	Auto Increment
departmentId	int	Not Null	Foreign Key	Reference to Department Table
amount	int	Not Null	-	Budget amount allocated
dateStart	Date	Not Null	-	-
dateEnd	Date	Not Null	-	-

5 Stories and Scenario

5.1 Story-1: Add New User in Employee details

*Story # **S1***	:	**As an** Admin, **I want to** add a new user in employee details **So that** everyone can divided by their role
Priority	:	High
Estimate	:	XL
Reason	:	The addition of a new user to the employee catalogue is crucial for ensuring that the employees do their work as per role.

5.1.1 Scenario# S1.1

*Scenario# **S1.1***	:	Adding a New User with Valid Information
Prerequisite	:	Admin is logged in to the Expense management system.

Acceptance Criteria	:	**Given:** The Admin is navigated to the Employee details management page. Valid user information , including role, qualification and other relevant details is added. **When:** The Admin selects the "Add New User" option And The Admin enters valid User details The librarian clicks the "Save" button to add the user to the employee details. **Then t**he system successfully adds the User to the employee details and the admin receives a confirmation message with the User's identification number.

5.1.2 Scenario# S1.2

*Scenario# **S1.2***	:	Adding a New User with Invalid Information.
Prerequisite	:	The Admin is logged into the Expense management system.
Acceptance Criteria	:	**Given:** The Admin is on the employee details page **When:** The Admin selects the "Add New User" option and the Admin enters an incomplete or incorrect User details and Admin clicks the "Save" button to add the User to the Employee Details. **Then t**he system displays error messages for the incorrect or missing information and the book is not added to the employee details.

5.2 Story-2: Manage receipt for add expense

*Story # **S2***	:	**As** Employee, **I want** to manage Receipt for add expense **So that** I can ensure that my expenses are properly documented without issues.
Priority	:	High
Estimate	:	X
Reason	:	Proper due date management is crucial for maintaining the library's collection and ensuring that books are available for all members.

5.2.1 Scenario# S2.1

Scenario # S2.1	:	Adding a valid Receipt to an Expense
Prerequisite	:	The employee is logged into the expense management system.
Acceptance Criteria	:	**Given**: The employee is adding a new expense, and the receipt is valid (clear, readable, and correct). **When**: The employee uploads the receipt using the "Add Receipt" button in the expense report form. **Then**:

		• The system successfully attaches the receipt to the expense, confirming its validity, and the employee receives a notification that the receipt has been added.

5.2.2 Scenario# S2.2

Scenario # S2.2	:	Adding an Invalid Receipt to an Expense
Prerequisite	:	The Admin is logged into the Expense management system.
Acceptance Criteria	:	**Given**: The employee is adding a new expense, but the receipt is invalid (e.g., blurry, incomplete, or not matching the expense). **When**: The employee uploads the invalid receipt and clicks "Submit." **Then**: • The system rejects the receipt, displays an error message (e.g., "Invalid receipt: blurry image, or receipt does not match expense"), and the employee cannot proceed without fixing the issue.

5.3 Story-3: Approve financial report

*Story # **S3***	:	**As CA**, **I want** to Approve financial report **so that I can ensure all expenses comply with financial policies before finalizing them.**
Priority	:	High
Estimate	:	XL
Reason	:	Approving financial reports is crucial for maintaining accurate records, ensuring compliance with regulations, and preparing for audits.

5.3.1 Scenario# S3.1

Scenario # S3.1	:	Approving a Financial Report with Valid Data
Prerequisite	:	The CA is logged into the Expense Management System and has access to pending financial reports.
Acceptance Criteria	:	**When**: • The CA clicks on a financial report. • The CA verifies the transactions and ensures they align with company policies. • The CA clicks the "Approve" button. **Then**: • The system updates the report status to "Approved." • A confirmation message is sent to relevant stakeholders. • The report becomes available for record-keeping and audit purposes.

5.3.2 Scenario# S3.2

Scenario # S3.2	:	Rejecting a Financial Report Due to Errors

Prerequisite	**:**	The CA is logged into the Expense Management System and has access to pending financial reports.
Acceptance Criteria	**:**	Acceptance Criteria: Given: The CA is on the "Pending Financial Reports" page and identifies inconsistencies or errors in a report (e.g., missing documentation, incorrect amounts, policy violations). When: • The CA clicks on the financial report. • The CA identifies errors and clicks the "Reject" button. • The CA enters a reason for rejection. Then: • The system updates the report status to "Rejected." • A notification is sent to the finance team or the responsible employee to make corrections. • The report cannot proceed for final approval until it is corrected and resubmitted.

6 Test cases

Project Name:	**EMI Calculator**	**Test Designed by:**	**P. U. Jadeja**
Module Name:	**Add User**	**Test Designed date:**	
Release Version:	**1.0**	**Test Executed by:**	
		Test Execution date:	

Pre-condition: Web application should be accessible

Test Case ID	**Test Title**	**Test Type**	**Description**	**Test Case ID**
TC_001	Verify that the form is submitted successfully with all valid inputs	Functional	Fill out all fields correctly and click 'Add User'	TC_001
TC_002	Verify that all fields are properly aligned	GUI	Open the form and check field alignment	TC_002

Test Case Title	Add user with valid credential
Test Type	Functional
Test Priority	High
Pre-condition	Web application should be accessible

Test Step	Test Case Description	Expected Result	Actual Result	Status	Comment	Data	BUG ID
1	Enter valid Username in username field	Username field should be editable and accept the Username	Username input accepted	Pass			
2	Verify that the email field accepts valid email addresses	Enter a valid email (e.g., namradhol @gmail.com)	The email should be accepted	Pass			
3	Verify that the department field accepts valid input	Enter a valid department name and submit	The department should be accepted	pass			
4	Verify that the role field accepts valid input	Select valid role form dropdoen	Selected role accepted	Pass			
5	Click on adduser button	User should added into system.	User added successfully .	pass			

Test Case Title	Add user page elements
Test Type	GUI
Test Priority	Medium
Pre-condition	Web application should be accessible

Test Step	Test Case Description	Expected Result	Actual Result	Status	Comment	Data	Bug ID
1	Verify that all fields are properly aligned	Open the form and check field alignment	Fields should be properly aligned	Pass			
2	Verify that placeholder text is present in all input fields	Check each input field for placeholder text	Each field should have appropriate placeholder text	Pass			
3	Verify that role dropdown all list selectable	All display list are selectable	Display an error Username not found	Pass			
4	Verify that the 'Add User' button is enabled only after all required fields are filled	Leave a required field empty and check button state	The button should remain disabled until all fields are filled	pass			

Project Name:	**EMI Calculator**	**Test Designed by:**	**P. U. Jadeja**
Module Name:	**Submit Expense**	**Test Designed date:**	
Release Version:	**1.0**	**Test Executed by:**	
		Test Execution date:	

Pre-condition: Web application should be accessible				
Test Case ID	**Test Title**	**Test Type**	**Description**	**Test Case ID**
TC_001	Verify that the form is submitted successfully with all valid inputs	Functional	Fill out all fields correctly and click 'Submit Expense'	TC_001
TC_002	Verify that all fields are properly aligned	GUI	Open the expense submission form.	TC_002

Test Case Title	Submit Expense with valid credential
Test Type	Functional
Test Priority	High
Pre-condition	Web application should be accessible

Test Step	Test Case Description	Expected Result	Actual Result	Status	Comment	Data	BUG ID
1	Verify that the expense date field accepts valid input	The date should be accepted	Date input accepted	Pass			
2	Verify that the expense category dropdown functions properly	The user should be able to select a category	Selected expense category accepted	Pass			
3	Verify that the amount field only accepts numeric values	Only numbers should be accepted	Number field accepts numeric value	pass			
4	Verify that the description field accepts text input	Description should be entered successfully	Description input accepted	Pass			
5	Verify that the 'Submit Expense' button is	The button should remain disabled	Expense summited successfully .	pass			

	enabled only after all required fields are filled	until required fields are filled					

Test Case Title	Submit expense page elements
Test Type	GUI
Test Priority	Medium
Pre-condition	Web application should be accessible

Test Step	Test Case Description	Expected Result	Actual Result	Status	Comment	Data	Bug ID
1	Verify form layout	Check alignment and positioning of form fields	Properly aligned and readable	Pass			
2	Verify placeholder text	Check if all fields have proper placeholder text	Placeholder text should guide users	Pass			
3	Verify that dropdown lists contain selectable options	All displayed options should be selectable	All displayed options should be selectable	Pass			
4	Verify that the 'Submit Expense' button is enabled only after all required fields are filled	The button should remain disabled until all fields are filled	Button should be enabled when valid, disabled when invalid	pass			

7 References

- http://www.w3schools.com/html/html_intro.asp
- https://www.w3schools.com/php/default.asp
- https://www.javatpoint.com/uml

WetherApiDemo

GET get

https://api.openweathermap.org/data/3.0/onecall?lat=33.44&lon=-94.04&exclude=hourly,daily&appid={99f21
70137c629cbf4e69c9eab09e9ca }

StartFragment

Product concept

Get essential weather data, short-term and long-term forecasts and aggregated weather data is easy with our OpenWeather **One Call API 3.0**. This product designed to ensure easy migration from the Dark Sky API.

One Call API 3.0 contains 4 endpoints and provides access to various data:

- and government weather alerts
  - minute forecast for 1 hour
  - hourly forecast for 48 hours
  - daily forecast for 8 days
- **Weather data for any timestamp** for 46+ years historical archive and 4 days ahead forecast
- **Daily aggregation** of weather data for 46+ years archive and 1.5 years ahead forecast
- **Weather overview** with a human-readable weather summary for today and tomorrow's forecast, utilizing OpenWeather AI technologies

One Call API 3.0 is based on the proprietary OpenWeather Model and is updated every 10 minutes. Thus, in order to receive the most accurate and up-to-date weather data, we recommend you request One Call API 3.0 every 10 minutes.

EndFragment

PARAMS

lat	33.44 Latitude, decimal (-90; 90). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
lon	-94.04 Longitude, decimal (-180; 180). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API

exclude	hourly,daily By using this parameter you can exclude some parts of the weather data from the API response. It should be a comma-delimited list (without spaces). Available values: current minutely hourly daily alerts
appid	{99f2170137c629cbf4e69c9eab09e9ca } our unique API key (you can always find it on your account page under the "API key" tab)

POST Post

https://api.openweathermap.org/data/3.0/onecall?lat=33.44&lon=-94.04&exclude=hourly,daily&appid={99f2170137c629cbf4e69c9eab09e9ca }

StartFragment

Product concept

Get essential weather data, short-term and long-term forecasts and aggregated weather data is easy with our OpenWeather **One Call API 3.0**. This product designed to ensure easy migration from the Dark Sky API.

One Call API 3.0 contains 4 endpoints and provides access to various data:

- and government weather alerts
  - minute forecast for 1 hour
  - hourly forecast for 48 hours
  - daily forecast for 8 days
- **Weather data for any timestamp** for 46+ years historical archive and 4 days ahead forecast
- **Daily aggregation** of weather data for 46+ years archive and 1.5 years ahead forecast
- **Weather overview** with a human-readable weather summary for today and tomorrow's forecast, utilizing OpenWeather AI technologies

One Call API 3.0 is based on the proprietary OpenWeather Model and is updated every 10 minutes. Thus, in order to receive the most accurate and up-to-date weather data, we recommend you request One Call API 3.0 every 10 minutes.

EndFragment

PARAMS

lat	33.44 Latitude, decimal (-90; 90). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API

lon	-94.04 Longitude, decimal (-180; 180). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
exclude	hourly,daily By using this parameter you can exclude some parts of the weather data from the API response. It should be a comma-delimited list (without spaces). Available values: current minutely hourly daily alerts
appid	{99f2170137c629cbf4e69c9eab09e9ca } our unique API key (you can always find it on your account page under the "API key" tab)

PATCH patch

https://api.openweathermap.org/data/3.0/onecall?lat=33.44&lon=-94.04&exclude=hourly,daily&appid={99f2170137c629cbf4e69c9eab09e9ca }

StartFragment

Product concept

Get essential weather data, short-term and long-term forecasts and aggregated weather data is easy with our OpenWeather **One Call API 3.0**. This product designed to ensure easy migration from the Dark Sky API.

One Call API 3.0 contains 4 endpoints and provides access to various data:

- and government weather alerts
  - minute forecast for 1 hour
  - hourly forecast for 48 hours
  - daily forecast for 8 days
- **Weather data for any timestamp** for 46+ years historical archive and 4 days ahead forecast
- **Daily aggregation** of weather data for 46+ years archive and 1.5 years ahead forecast
- **Weather overview** with a human-readable weather summary for today and tomorrow's forecast, utilizing OpenWeather AI technologies

One Call API 3.0 is based on the proprietary OpenWeather Model and is updated every 10 minutes. Thus, in order to receive the most accurate and up-to-date weather data, we recommend you request One Call API 3.0 every 10 minutes.

EndFragment

PARAMS

lat	33.44
	Latitude, decimal (-90; 90). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
lon	-94.04
	Longitude, decimal (-180; 180). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
exclude	hourly,daily
	By using this parameter you can exclude some parts of the weather data from the API response. It should be a comma-delimited list (without spaces). Available values: current minutely hourly daily alerts
appid	{99f2170137c629cbf4e69c9eab09e9ca }
	our unique API key (you can always find it on your account page under the "API key" tab)

PUT put

https://api.openweathermap.org/data/3.0/onecall?lat=33.44&lon=-94.04&exclude=hourly,daily&appid={99f2170137c629cbf4e69c9eab09e9ca }

StartFragment

Product concept

Get essential weather data, short-term and long-term forecasts and aggregated weather data is easy with our OpenWeather **One Call API 3.0**. This product designed to ensure easy migration from the Dark Sky API.

One Call API 3.0 contains 4 endpoints and provides access to various data:

- and government weather alerts
  - minute forecast for 1 hour
  - hourly forecast for 48 hours
  - daily forecast for 8 days
- **Weather data for any timestamp** for 46+ years historical archive and 4 days ahead forecast
- **Daily aggregation** of weather data for 46+ years archive and 1.5 years ahead forecast
- **Weather overview** with a human-readable weather summary for today and tomorrow's forecast, utilizing OpenWeather AI technologies

One Call API 3.0 is based on the proprietary OpenWeather Model and is updated every 10 minutes. Thus, in order to receive the most accurate and up-to-date weather data, we recommend you request One Call API 3.0 every 10 minutes.

EndFragment

PARAMS

lat	33.44	Latitude, decimal (-90; 90). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
lon	-94.04	Longitude, decimal (-180; 180). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
exclude	hourly,daily	By using this parameter you can exclude some parts of the weather data from the API response. It should be a comma-delimited list (without spaces). Available values: current minutely hourly daily alerts
appid	{99f2170137c629cbf4e69c9eab09e9ca }	our unique API key (you can always find it on your account page under the "API key" tab)

DELETE delete

```
https://api.openweathermap.org/data/3.0/onecall?lat=33.44&lon=-94.04&exclude=hourly,daily&appid={99f21
70137c629cbf4e69c9eab09e9ca }
```

StartFragment

Product concept

Get essential weather data, short-term and long-term forecasts and aggregated weather data is easy with our OpenWeather **One Call API 3.0**. This product designed to ensure easy migration from the Dark Sky API.

One Call API 3.0 contains 4 endpoints and provides access to various data:

- and government weather alerts
  - minute forecast for 1 hour
  - hourly forecast for 48 hours
  - daily forecast for 8 days
- **Weather data for any timestamp** for 46+ years historical archive and 4 days ahead forecast
- **Daily aggregation** of weather data for 46+ years archive and 1.5 years ahead forecast
- **Weather overview** with a human-readable weather summary for today and tomorrow's forecast, utilizing OpenWeather AI technologies

One Call API 3.0 is based on the proprietary OpenWeather Model and is updated every 10 minutes. Thus, in order to receive the most accurate and up-to-date weather data, we recommend you request One Call API 3.0 every 10 minutes.

EndFragment

PARAMS

lat	33.44	Latitude, decimal (-90; 90). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
lon	-94.04	Longitude, decimal (-180; 180). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
exclude	hourly,daily	By using this parameter you can exclude some parts of the weather data from the API response. It should be a comma-delimited list (without spaces). Available values: current minutely hourly daily alerts
appid	{99f2170137c629cbf4e69c9eab09e9ca }	our unique API key (you can always find it on your account page under the "API key" tab)