

A
Project Report
On
Online Expense Tracker

By
Pratik Kadam (19)
Abhijeet Singh (48)

Under the Guidance of
Prof. Sushma Sumant

For
Mini Project
In partial fulfillment of
MASTER OF COMPUTER APPLICATION
Semester II
UNIVERSITY OF MUMBAI



NCRD's Sterling Institute of Management Studies

Nerul, NaviMumbai

2023-2024



NCRD's Sterling Institute of Management Studies
Nerul, NaviMumbai

Certificate of Approval

This is to certify that the Summer project titled **Online Expense Tracker** successfully completed by **Pratik Kadam & Abhijeet Singh** for Semester-II (Academic year 2023-24) in partial fulfillment of **Masters of Computer Application, University of Mumbai, Mumbai** through the NCRD's Sterling Institute of Management Studies Nerul, Navi Mumbai, carried out by him/her under our guidance and supervision.

Date: / /20

Internal Guide

Prof. Sushma Sumant

HOD

Dr. Pragati Goel

Examiner

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: / /20

Pratik Kadam

Abhijeet Singh

ACKNOWLEDGMENT

It gives us immense pleasure in presenting this summer report for the project *Online Expense Tracker*. I profoundly thank our **Director Dr. Murlidhar Dhanawade**, for giving us support throughout the course and thus made us capable of being worthy of recognition and extended every facility to us for making and completing this project smoothly.

I would like to express my sincere thanks to **Dr. Pragati Goel, Professor & HOD (MCA)** for his constant encouragement, which made this project a success.

I owe deep gratitude to **Prof. Sushma Sumant** my project guide, for rendering his/her valuable guidance with a touch of inspiration and motivation. He/She has guided me quite a lot in negotiating through the hurdles by giving plenty of early ideas and which resulted in the present fine work.

I would like to thank all the faculty members & staff of NCRD's Sterling Institute of Management Studies, Nerul, Navi Mumbai, for providing us sufficient information which helped me to complete my project successfully. Their guidance has always inculcated confidence in me. And last but not the least, I wish to thank all my friends and well-wishers who are directly or indirectly linked with the success of my project.

FY MCA A
Pratik Kadam (19)
Abhijeet Singh (48)

TABLE OF CONTENTS

Sr. No	Topic	Page No.
	Abstract	7
1.	Introduction	
	1.1 Introduction	8
	1.1.1 Problem definition	8
	1.1.2 Objectives of Project	8
	1.1.3 Scope of Project	8
	1.2 Technical Details	9
	1.2.1.Overview of the Front End	9
	1.2.2.Overview of the Back End	10
2.	System Study and Planning	
	2.1 System Study	11
	2.1.1 Existing System	11
	2.1.2 Disadvantages of Existing system	11
	2.1.3 Proposed System	12
	2.2 System Planning and Schedule	12
	2.2.1 S/W development Model	13
	2.2.2 GANTT Chart	14
3.	System Design	
	3.1 Software Requirement Specification(SRS)	15
	3.1.1 Introduction of SRS	15
	3.1.2 Technology Requirements	16
	3.1.2.1 Hardware to be used	16
	3.1.2.2 Software/tools to be used	17
	3.2 Detailed life Cycle of the Project	19
	3.2.1 Modules	19
	3.2.2 Object Oriented Analysis & Design Diagrams	20
	3.2.2.1 Use Case Diagram	20
	3.2.2.2 Activity Diagram	20
	3.2.2.3 Class Diagram	21

	3.2.2.4 Sequence Diagram	21
	3.2.2.5 Flowchart/DFD/ER diagram	22
	3.2.3 Database	23
	3.2.3.1 Database Table	24
	3.2.4 I/O Screen Layout	25
4.	Coding	30
5.	Testing	33
	5.1 Methodologies used for testing	33
	5.2 Types of Testing	34
6.	Conclusion	36
7.	Limitations	37
8.	Future Enchantements	38
9.	References	39

ABSTRACT

The online expense tracker system is a web-based application designed to help users manage their personal finances by tracking their expenses and income. This system provides a comprehensive and user-friendly platform where users can input, categorize, and monitor their financial transactions in real time. The primary objective of this project is to enable users to have a clear understanding of their spending patterns and financial status, thereby facilitating better financial planning and management.

The system leverages modern web technologies and follows best practices in software development to provide a reliable, scalable, and secure solution. By using this online expense tracker, users can gain better control over their financial health, identify areas where they can save money, and make informed financial decisions. This project aims to enhance the financial literacy and discipline of its users, ultimately leading to improved financial well-being.

Effective financial management is a cornerstone of personal and economic well-being. By simplifying the process of tracking and managing expenses and income, the online expense tracker system empowers users to take control of their finances. This system not only aids in day-to-day financial activities but also fosters long-term financial planning and savings. Ultimately, the online expense tracker system aims to enhance financial literacy, promote responsible spending, and help users achieve their financial goals.

1. INTRODUCTION

1.1 Introduction

1.1.1 Problem Definition

The “online expense tracker system” is a web-based application designed to help users manage their personal finances by tracking their expenses and income. This system provides a comprehensive and user-friendly platform where users can input, categorize, and monitor their financial transactions in real time. The primary objective of this project is to enable users to have a clear understanding of their spending patterns and financial status, thereby facilitating better financial planning and management.

1.1.2 Objectives of Project

The online expense tracker system aims to provide users with a comprehensive and user-friendly platform to manage their finances effectively. The primary objective is to enhance financial awareness by enabling users to understand their spending habits and income sources through detailed insights and visualizations. Simplifying budgeting and expense management is another key goal, allowing users to set, track, and adjust budgets for various categories and receive alerts to avoid overspending.

1.1.3 Scope of Project

Anybody who is an Account holder in this bank can become a member of Bank Account Management System. He has to fill a form with his personal details and Account Number. Bank is the place where customers feel the sense of safety for their property. In the bank, customers deposit and withdraw their money. Transaction of money also is a part where customer takes shelter of the bank. Now to keep the belief and trust of customers, there is the positive need for management of the bank, which can handle all this with comfort and ease. Smooth and efficient management affects the satisfaction of the customers and staff members, indirectly. And of course, it encourages management committee in taking some needed decision for future enhancement of the bank.

1.2 Technical Details

1.2.1. Overview of the Front End

- Console Interface:
 - The frontend provides a simple console-based interface for user interaction.
 - Users can choose options such as creating accounts and exiting the system.
- Features:
 - Create Account:
 - Users can create a new User account by providing Email Address, Email Password.
 - Exit:
 - Users can exit the system.
- User Input Handling:
 - Utilizes the Scanner class for accepting user input.
 - Ensures proper validation of input data to maintain data integrity.

1.2.2. Back end Overview:

- Java with JDBC:
 - Backend logic is implemented in Java, using JDBC for database connectivity.
 - JDBC is employed to establish a connection to the MySQL database, execute SQL queries, and manage transactions.
- Database Connection:
 - The backend connects to a MySQL database using a JDBC URL, username, and password.
 - Checks if the accounts table exists; if not, creates it.
- Exception Handling:

- Incorporates exception handling to deal with SQL-related errors.
- Security Measures:
 - Assumes secure database credentials handling for real-world applications.
 - In a production environment, password encryption and other security measures would be implemented.

1.2.3. MySQL Workbench Connectivity:

- Database Structure:
 - Assumes a simple database structure with a single table named accounts.
 - The accounts table contains columns for account number, account holder name, and balance.
- SQL Script:
 - Provided an SQL script to create the necessary accounts table.

2. SYSTEM STUDY & PLANNING

2.1 System Study

2.1.1 Existing System

Before the development of the online expense tracker system, traditional methods and standalone software applications were commonly used for personal financial management. These methods often involved manual entry of transactions into spreadsheets or desktop-based software like Microsoft Excel or Quicken. Users typically relied on paper receipts and bank statements to reconcile their expenses and income, which was a time-consuming and error-prone process.

2.1.2 Disadvantages of Existing System

The existing systems for personal financial management, such as manual entry into spreadsheets or desktop software like Microsoft Excel, suffer from several disadvantages. Firstly, these methods are labor-intensive and prone to errors, relying heavily on manual data entry from receipts and bank statements. This manual process is not only time-consuming but also increases the likelihood of inaccuracies in financial records. Additionally, traditional systems lack real-time updates and accessibility across different devices, limiting users' ability to manage their finances conveniently on the go. Security concerns also arise as sensitive financial data stored locally on personal computers or physical documents may be vulnerable to loss, theft, or unauthorized access. Moreover, these systems often lack advanced features such as automated categorization of expenses, budget tracking, and dynamic reporting, which are crucial for effective financial planning and decision-making. Overall, the limitations of the existing systems underscore the need for a more modern and integrated solution like the online expense tracker system, which aims to overcome these drawbacks and provide users with a secure, efficient, and user-friendly platform for managing their finance.

2.1.3 Proposed System

The proposed online expense tracker system aims to revolutionize personal financial management by offering a modern, integrated, and user-friendly platform. Unlike traditional methods, which rely on manual entry and standalone software, the proposed system leverages web-based technologies to provide real-time updates, accessibility from any device with internet connectivity, and robust security measures.

Key features of the proposed system include:

1. **User Authentication and Security:** Secure user registration, login, and password recovery processes to protect sensitive financial information.
2. **Transaction Management:** Streamlined entry, categorization, and management of income and expenses with options for automated categorization and tagging.
3. **Budgeting Tools:** Ability for users to set budgets for various expense categories, track expenditures against these budgets, and receive alerts for overspending.
4. **Interactive Dashboard:** A dynamic dashboard displaying summarized financial data through charts, graphs, and key metrics for quick insights into financial health.
5. **Reporting and Analytics:** Comprehensive reporting capabilities with options to generate detailed reports based on various criteria (e.g., date range, categories) and export them in formats like PDF or CSV for further analysis.
6. **Accessibility and Multi-Device Support:** Accessible via web browsers on desktops, tablets, and smartphones, ensuring users can manage their finances anytime and anywhere.
7. **Data Backup and Security:** Regular automated backups of user data to prevent loss, coupled with encryption and secure transmission protocols (SSL/TLS) to safeguard data integrity.
8. **Educational Resources:** Integration of financial literacy resources such as tips, articles, and tutorials to empower users with knowledge and skills for better financial management.

By implementing these features, the proposed online expense tracker system aims to streamline financial management processes, enhance user convenience and security, promote financial transparency and accountability, and ultimately empower users to achieve their financial goals effectively. The system's modern approach and comprehensive functionalities aim to address the limitations of existing methods and provide a robust solution tailored to the needs of today's digitally-connected user.

2.2 System Planning & Schedule

The system planning and schedule for the online expense tracker project involve a structured approach to ensure the timely and successful development, testing, and deployment of the application. This phase encompasses several key activities and milestones:

1. **Requirements Gathering:** Detailed gathering of functional and non-functional requirements from stakeholders and potential users regarding financial management needs, security requirements, and user interface preferences.
2. **Feasibility Study:** Evaluation of the technical feasibility, economic viability, and operational feasibility of the project. This includes assessing the compatibility of chosen technologies (Java, Spring Boot, Hibernate), database options (MySQL or PostgreSQL), and estimating costs and resources required.
3. **Project Plan Development:** Creation of a comprehensive project plan outlining:

- **Project Scope:** Defined functionalities and features, including user authentication, transaction management, budgeting tools, reporting, and data security measures.
 - **Milestone Definition:** Identification of key milestones such as completion of backend development, frontend implementation, testing phases, and deployment.
 - **Resource Allocation:** Allocation of human resources, including developers, testers, and project managers, as well as hardware and software resources needed for development and testing.
 - **Timeline:** Development of a detailed timeline with specific deadlines for each phase of the project, ensuring adherence to schedules and timely completion.
4. **System Architecture Design:** Designing the system architecture based on the MVC (Model-View-Controller) pattern, ensuring scalability, maintainability, and integration of chosen technologies (Spring Boot for backend, Thymeleaf or Angular/React for frontend).
 5. **Security Planning:** Planning and implementation of robust security measures including:
 - **Authentication and Authorization:** Secure user registration, login, and role-based access control using Spring Security.
 - **Data Encryption:** Encryption of sensitive data such as passwords and financial transactions using strong encryption algorithms.
 - **Secure Communication:** Implementation of SSL/TLS protocols to ensure secure communication between clients and servers.
 6. **Development Environment Setup:** Configuration of development tools and environments including:
 - **IDE:** Selection of IntelliJ IDEA or Eclipse for Java development.
 - **Version Control:** Implementation of Git for version control, with repositories hosted on GitHub or GitLab.
 - **Build Automation:** Use of Maven or Gradle for build automation and dependency management.
 7. **Testing Strategy:** Development of a comprehensive testing strategy including:
 - **Unit Testing:** Using JUnit for testing individual components and classes.
 - **Integration Testing:** Testing interactions between different modules and components.
 - **End-to-End Testing:** Conducting automated browser testing using tools like Selenium to ensure the application functions correctly from the user's perspective.
 8. **Deployment Plan:** Preparation of a deployment plan for deploying the application on a cloud platform such as AWS or Heroku, including:
 - **Containerization:** Dockerizing the application for consistency in deployment across different environments.
 - **Continuous Integration/Continuous Deployment (CI/CD):** Setting up Jenkins or GitHub Actions for automated testing and deployment processes to streamline deployment and ensure software quality.
 9. **Documentation and Training:** Creation of user manuals, technical documentation, and training materials for end-users and support staff to facilitate smooth adoption and operation of the system.
-

2.2.1 Gantt Chart



3. SYSTEM DESIGN

3.1 Software Requirement Specification(SRS)

1. Introduction

1.1 Purpose

The purpose of this Software Requirement Specification (SRS) document is to outline the functional and non-functional requirements for the development of the online expense tracker system. This document serves as a guide for the development team to ensure the system meets the needs of its users for managing their personal finances efficiently.

1.2 Scope

The online expense tracker system is a web-based application designed to help users track their income and expenses, set budgets, and generate financial reports. The system will provide real-time updates and be accessible from various devices including desktops, tablets, and smartphones. Key functionalities include user authentication, transaction management, budgeting tools, and data visualization.

1.3 Definitions, Acronyms, and Abbreviations

- **MVC**: Model-View-Controller
- **REST**: Representational State Transfer
- **JSON**: JavaScript Object Notation
- **SSL/TLS**: Secure Sockets Layer / Transport Layer Security
- **CI/CD**: Continuous Integration / Continuous Deployment
- **IDE**: Integrated Development Environment

1.4 References

- IEEE Standard for Software Requirements Specifications (IEEE Std 830-1998)

1.5 Overview

This SRS document is organized into the following sections:

- Introduction
- Overall Description
- Specific Requirements

2. Overall Description

2.1 Product Perspective

The online expense tracker system is an independent web-based application that interacts with external services for features like email notifications and backup storage. The system will follow the MVC architecture, with Spring Boot as the backend framework, Hibernate for ORM, and Thymeleaf or Angular/React for the frontend.

2.2 Product Functions

- **User Authentication:** Registration, login, password recovery, and secure access control.
- **Transaction Management:** Adding, editing, deleting, and categorizing income and expense transactions.
- **Budgeting Tools:** Setting and tracking budgets for various categories.
- **Reporting and Analytics:** Generating financial reports, visualizations, and exporting data.
- **Data Backup:** Regular backups and restoration of user data.

2.3 User Characteristics

The primary users of the system are individuals looking to manage their personal finances. They are expected to have basic computer skills and access to internet-enabled devices.

2.4 Constraints

- The system must be accessible on various devices and screen sizes.
- Data security and user privacy must be ensured.
- The system should handle concurrent users efficiently.

2.5 Assumptions and Dependencies

- Users have reliable internet access.
- External services (e.g., email, cloud storage) are available and operational.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Authentication

- **FR1.1:** The system shall allow users to register with a unique username and password.
- **FR1.2:** The system shall provide a login mechanism for registered users.
- **FR1.3:** The system shall allow users to recover their password via email.
- **FR1.4:** The system shall use secure authentication protocols to protect user credentials.

3.1.2 Transaction Management

- **FR2.1:** The system shall allow users to add new income and expense transactions.
- **FR2.2:** The system shall allow users to edit and delete existing transactions.
- **FR2.3:** The system shall categorize transactions based on user-defined categories.
- **FR2.4:** The system shall provide a search and filter functionality for transactions.

3.1.3 Budgeting Tools

- **FR3.1:** The system shall allow users to set monthly and yearly budgets for different categories.
- **FR3.2:** The system shall track actual spending against set budgets.
- **FR3.3:** The system shall notify users when they are close to exceeding their budget limits.

3.1.4 Reporting and Analytics

- **FR4.1:** The system shall provide a dashboard with visual representations (charts, graphs) of income, expenses, and budgets.
- **FR4.2:** The system shall generate detailed financial reports based on user-selected criteria.
- **FR4.3:** The system shall allow users to export reports in PDF and CSV formats.

3.1.5 Data Backup and Restoration

- **FR5.1:** The system shall perform regular automated backups of user data.
- **FR5.2:** The system shall allow users to manually initiate data backups.
- **FR5.3:** The system shall provide data restoration functionality.

3.2 Non-Functional Requirements

3.2.1 Performance

- **NFR1.1:** The system shall support at least 1000 concurrent users without performance degradation.
- **NFR1.2:** The system shall respond to user actions within 2 seconds on average.

3.2.2 Security

- **NFR2.1:** The system shall use SSL/TLS for secure communication.
- **NFR2.2:** The system shall encrypt sensitive data such as passwords and financial information.

3.2.3 Usability

- **NFR3.1:** The system shall have an intuitive and user-friendly interface.
- **NFR3.2:** The system shall be accessible on various devices, including desktops, tablets, and smartphones.

3.2.4 Reliability

- **NFR4.1:** The system shall have an uptime of 99.9%.
- **NFR4.2:** The system shall recover from failures without data loss.

3.2.5 Maintainability

- **NFR5.1:** The system shall be developed using modular components to facilitate maintenance and updates.
- **NFR5.2:** The system documentation shall be comprehensive and up-to-date.

3.2 Detailed life Cycle of the Project

The development of the online expense tracker system follows a structured life cycle model, encompassing several phases to ensure thorough planning, development, testing, and deployment. Below is a detailed description of each phase in the project life cycle:

1. Initiation Phase

- **Project Proposal and Approval:** Define the project objectives, scope, and feasibility. Present the project proposal to stakeholders for approval.
- **Requirement Gathering:** Conduct interviews, surveys, and meetings with potential users to gather detailed requirements. Document both functional and non-functional requirements.

2. Planning Phase

- **Project Plan Development:** Create a comprehensive project plan detailing the project schedule, milestones, resource allocation, and risk management strategies.
- **System Design Planning:** Outline the system architecture based on the MVC model. Define the technologies and frameworks to be used (e.g., Spring Boot, Hibernate, Thymeleaf/Angular/React).
- **Security Planning:** Plan for robust security measures, including user authentication, data encryption, and secure communication protocols.

3. System Design Phase

- **Architecture Design:** Develop the overall system architecture, including the backend, frontend, database, and integration points.
- **Detailed Design:** Create detailed design documents for each system component, specifying the interactions between the Model, View, and Controller layers. Design database schema and API endpoints.
- **UI/UX Design:** Design wireframes and prototypes for the user interface, ensuring a user-friendly and intuitive experience.

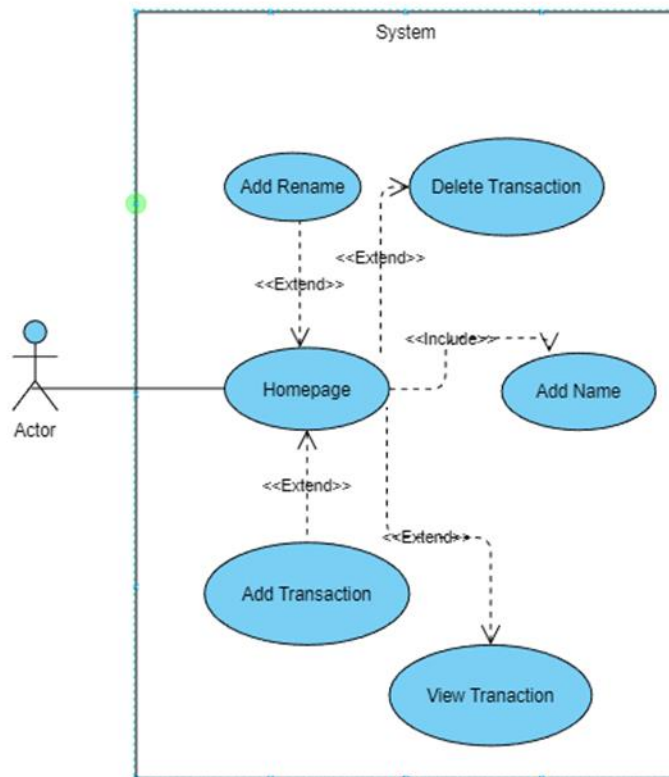
4. Development Phase

- **Setup Development Environment:** Configure development tools and environments, including IDEs (IntelliJ IDEA/Eclipse), version control (Git), and build automation tools (Maven/Gradle).
- **Backend Development:** Implement the backend logic using Spring Boot. Develop RESTful APIs for handling user requests and interactions with the database.
- **Frontend Development:** Build the user interface using Thymeleaf for server-side rendering or Angular/React for client-side rendering. Ensure responsive design for various devices.
- **Database Integration:** Implement database models and configure Hibernate for ORM. Develop CRUD operations for managing transactions and user data.
- **Security Implementation:** Integrate Spring Security for authentication and authorization. Implement data encryption and secure communication protocols (SSL/TLS).

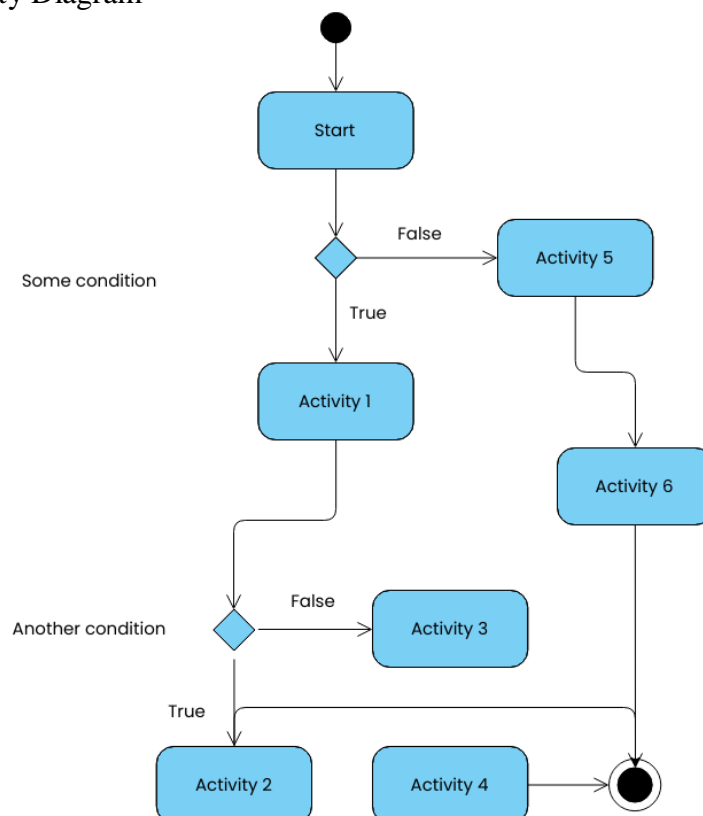
5. Testing Phase

- **Unit Testing:** Write and execute unit tests for individual components using JUnit.
 - **Integration Testing:** Test interactions between system components and modules to ensure they work together seamlessly.
-

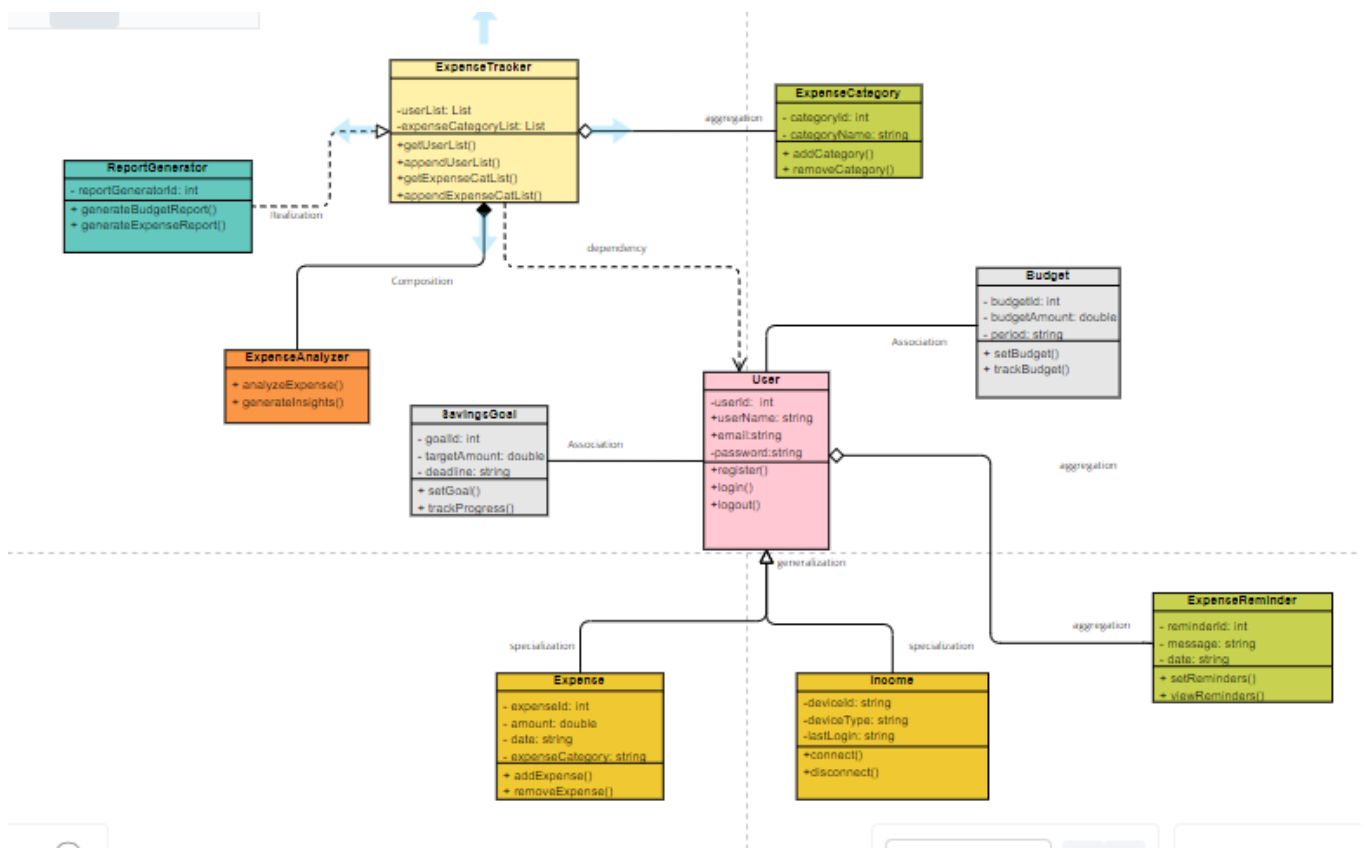
3.2.1 Object Oriented Analysis & Design Diagrams



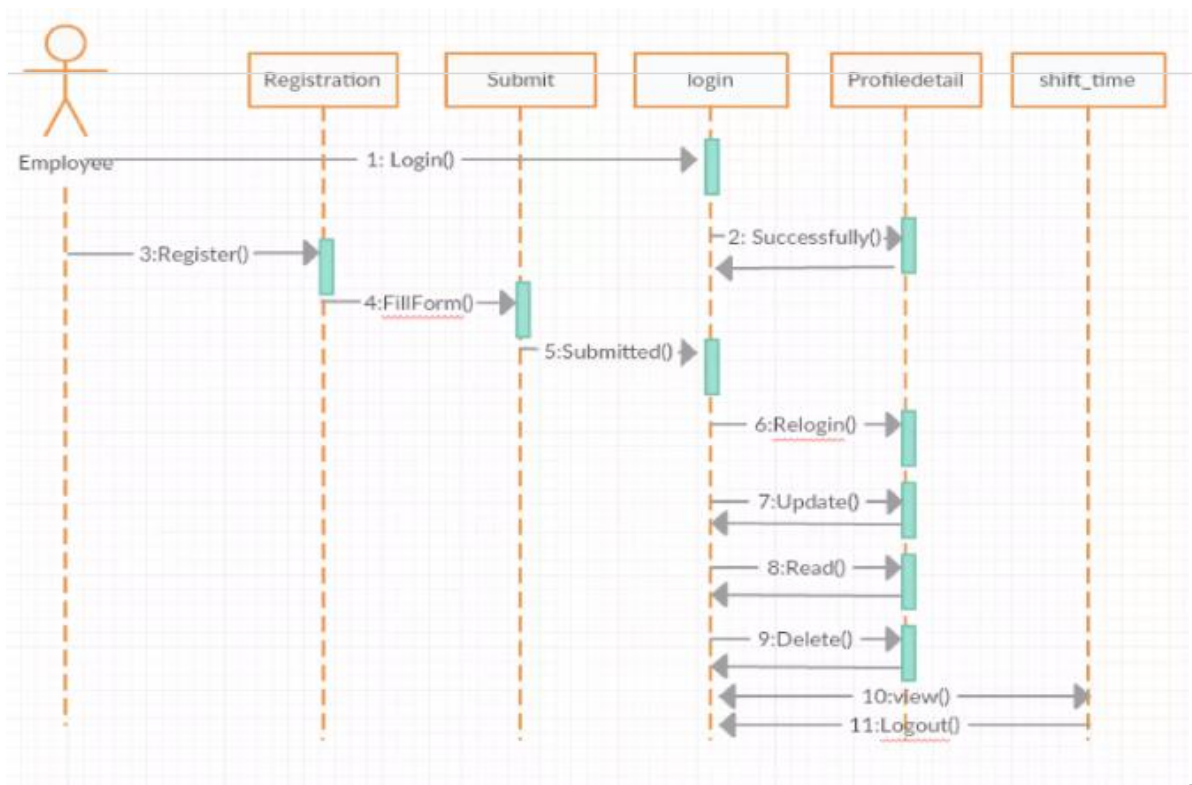
3.2.1.1 Activity Diagram



3.2.1.2 Class Diagram

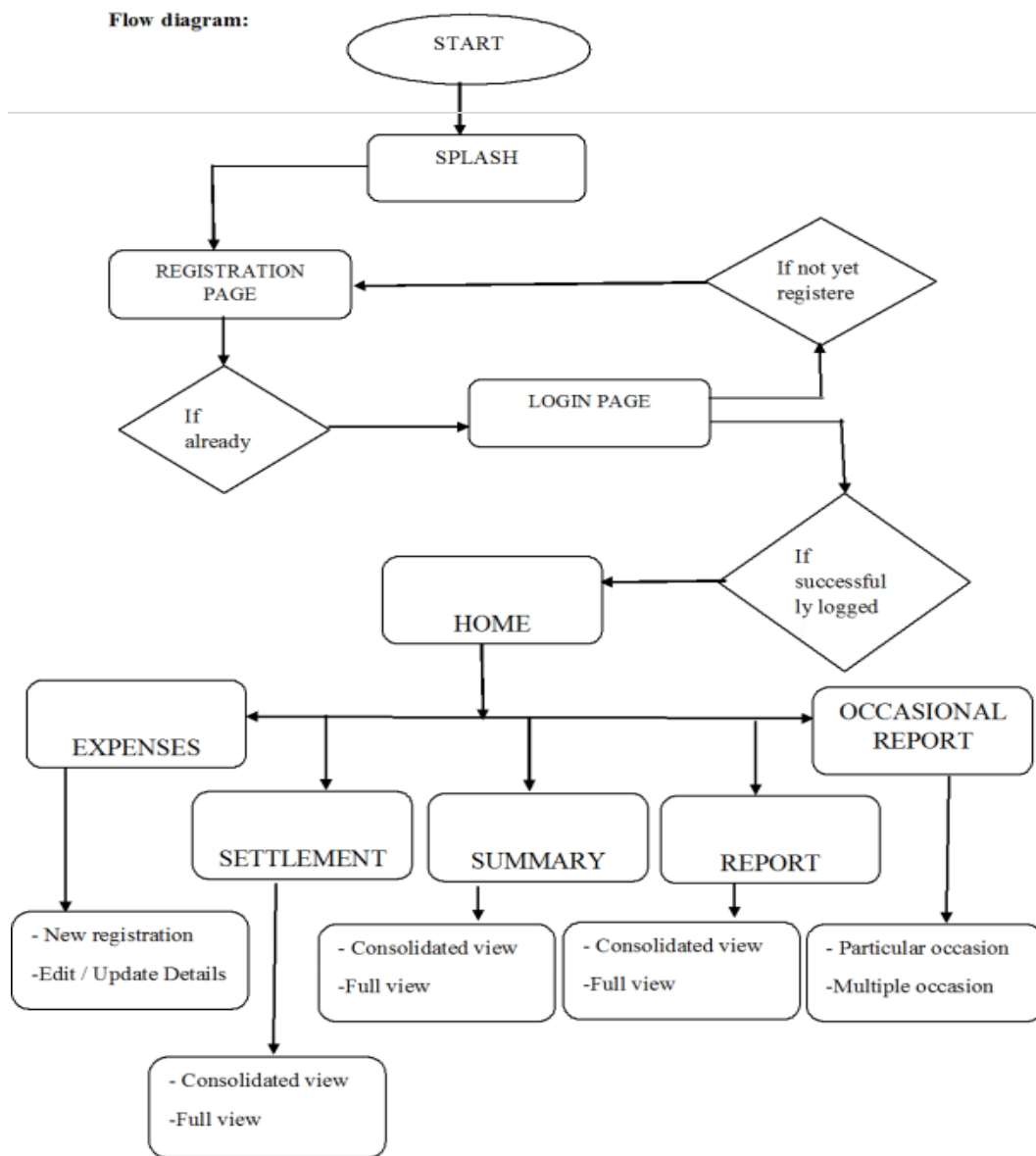


3.2.1.3 Sequence Diagram



3.2.1.4 Flowchart diagram

1. Flowchart:



3.2.2 Database

Creating a database for a Online Expense Tracker involves designing tables to store information such as User details, Expense information, Expense history.

Below is a simplified example of a relational database schema for a basic Online Expense Tracker:

1. User Table:

- CustomerID (Primary Key)
- FirstName
- LastName
- DateOfBirth
- Address
- ContactNumber
- Email

2. Expense Table:

- User Id(Primary Key)
- Date
- Price
- Time
- Description

3. Hibernate Sequence Table:

- Hibernate Sequence Id

3.2.2.1 Database Table

User Details

Result Grid		Filter Rows:		Edit:			
	id	email	Name	number	pass		
▶	1	pk@gmail.com	pratik	08732327455	asdfa		
	2	abhi123@gmail.com	abhijit	87324737832	abhi123		
	4	raj123@gmail.com	raj	08976542398	dxg		
	5	raj123@gmail.com	raj	08976542398	dxg		
	7	pk12345@gmail.com	Pratik Kadam	983538265282	pk@123		
	8	pratik123@gmail.com	Pratik Kadam	8975623354	pk123		
	9	pankaj12@gamil.com	pankaj	743747843874	747		
•	NULL	NULL	NULL	NULL	NULL		

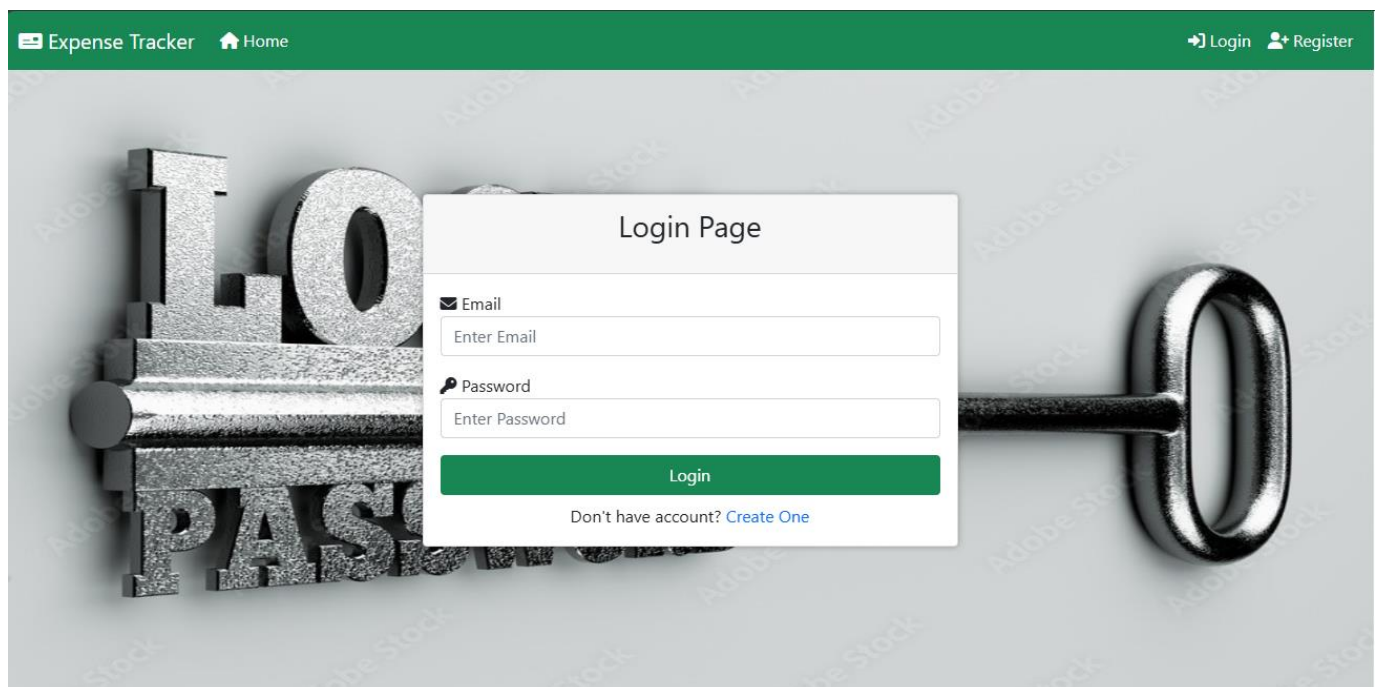
Expense Details

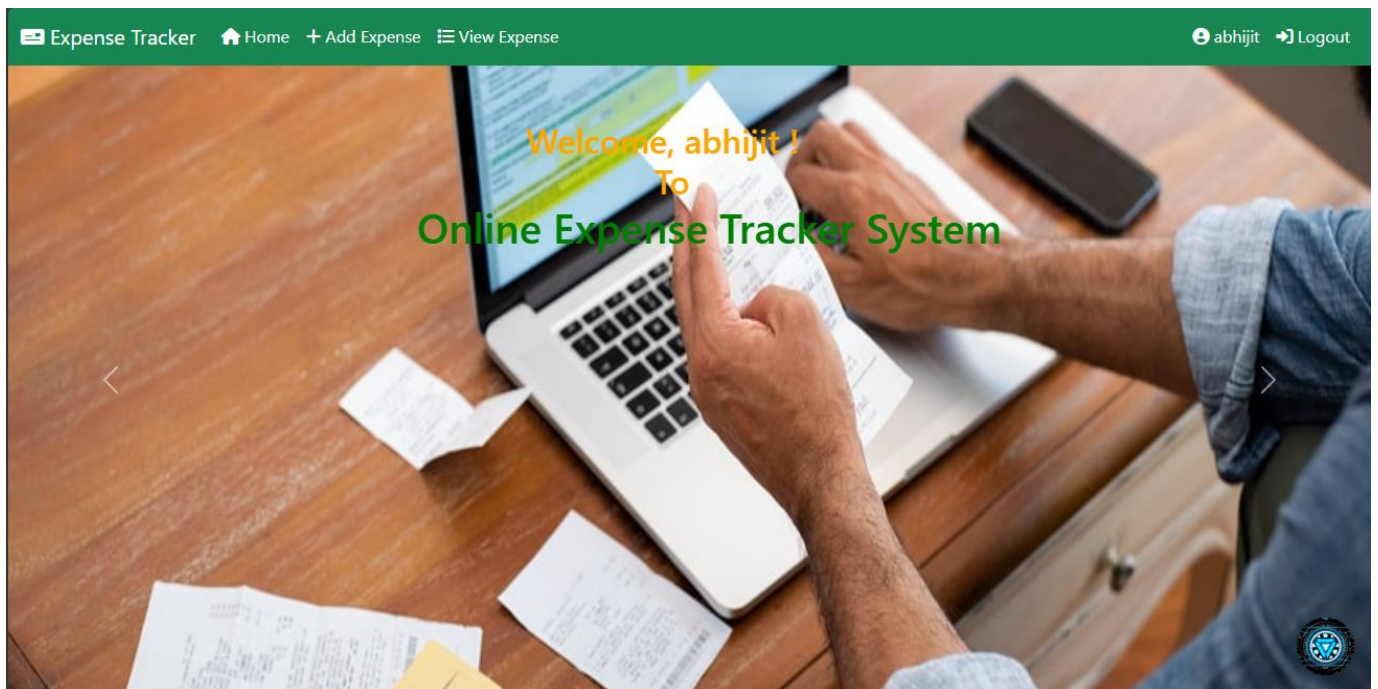
Result Grid		Filter Rows:		Edit:			Export/Import:	
	id	date	description	price	time	title	user_id	totalprice
▶	1	2024-05-23	burger	35	15:45	Food	1	NULL
	2	2024-05-01	yamaha v3 blue	1,65,000	03:10	bike	1	NULL
	3	2024-05-06	biscuit	10	23:49	food	1	NULL
	6	2024-05-02	best for body	342	11:08	protein	2	NULL
	8	2024-05-09	try	435	23:28	try	1	NULL
	10	2024-04-05	yamaha v3 blue	200980	09:40	Bike	2	NULL
	12	2024-05-07	Anniversary	1	00:01	Tritorc	7	NULL
	13	2024-05-11	best bike	12050	09:40	bike	1	NULL
	22	2024-05-16	burger	342	21:15	Food	2	NULL
	23	2024-05-17	nasta	20	17:25	biscuit	8	NULL

Hibernate Sequence

Result Grid	
	next_val
▶	13

3.2.3 I/O Screen Layout





The screenshot displays the 'Add Expense' form overlay on top of a background image showing a person holding a clipboard with a bar chart. The form is white with a green header bar that says 'Add Expense'. It contains the following fields and controls:

- Title:** A text input field with the placeholder 'Enter Title'.
- Date:** A date picker field with the placeholder 'dd-mm-yyyy' and a calendar icon.
- Time:** A time picker field with the placeholder '--:-- --' and a clock icon.
- Description:** A text input field with the placeholder 'Enter Description'.
- Price:** A text input field with the placeholder 'Enter Price' and a rupee symbol (₹) to its left.
- Add:** A green button at the bottom of the form.

Expense Tracker

Home

+ Add Expense

View Expense

abhi123

Logout

All Expenses

Title	Description	Date	Time	Price(Rs)	Action
protein	best for body	2024-05-02	11:08	342	<div>EditDelete</div>
Bike	yamaha v3 blue	2024-04-05	09:40	200980	<div>EditDelete</div>
Food	burger	2024-05-16	21:15	342	<div>EditDelete</div>

Total Amount: 201664.0

Start Date:

dd-mm-yyyy

End Date:

dd-mm-yyyy

Filter

Expense Tracker

Home

abhi123

Logout

User Information

Name

abhi123

Email

abhi123@gmail.com

Password

abhi123

Mobile Number

87324737832

Back

'Warning, This account will be permanently deleted.'

Delete Account

4. CODING

```
{ User Page Code}

package com.entity;

import
javax.persistence.Column
;

import
javax.persistence.Entity
;

import
javax.persistence.Genera
tedValue;

import
javax.persistence.Genera
tionType;

import
javax.persistence.Id;

import
javax.persistence.Table;

@Entity

@Table(name =
"User_Details")

public class User

{

    @Id

    @GeneratedValue(strat
egy =
GenerationType.AUTO)

    private int id;

    @Column(name =
"Name")

    private String name;

    private String email;

    private String pass;

    private String
number;

    public User()
```

```

        {

            super();

        }

        public User(String
name, String email,
String pass, String
number)

        {

            this.name =
name;

            this.email =
email;

            this.pass =
pass;

            this.number =
number;

        }

        public int getId() {

            return id;

        }

        public void setId(int
id) {

            this.id = id;

        }

        public String
getName() {

            return name;

        }

        public void
setName(String name) {

            this.name =
name;

        }

```

```

        public String
getEmail() {

            return email;

        }

        public void
setEmail(String email) {

            this.email =
email;

        }

        public String
getPass() {

            return pass;

        }

        public void
setPass(String pass) {

            this.pass =
pass;

        }

        public String
getNumber() {

            return number;

        }

        public void
setNumber(String number)
{

            this.number =
number;

        }

        @Override

        public String
toString() {

            return "User
[id=" + id + ", name=" +
name + ", email=" +

```

```

        email + ", pass=" + pass
        + ", number=" + number +
        "];

    }

}

{ Database Page Code}

package com.db;

import java.util.Properties;

import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.cfg.Environment;
import org.hibernate.service.ServiceRegistry;

import com.entity.Expense;
import com.entity.User;

public class HibernateUtil
{
    private static SessionFactory sessionFactory;

    public static SessionFactory getSessionFactory()
    {
        if(sessionFactory == null)
        {
            Configuration configuration = new Configuration();

            Properties properties = new Properties();

            properties.put(Environment.DRIVER,
"com.mysql.cj.jdbc.Driver");
            properties.put(Environment.URL,
"jdbc:mysql://localhost:3306/online_expense_tracker");
            // properties.put(Environment.URL,
"jdbc:mysql://localhost:3306/testing");

            properties.put(Environment.USER, "root");
            properties.put(Environment.PASS, "12345");
            properties.put(Environment.DIALECT,
"org.hibernate.dialect.MySQL8Dialect");
            properties.put(Environment.HBM2DDL_AUTO, "update");
            properties.put(Environment.SHOW_SQL, true);

            configuration.setProperties(properties);
            configuration.addAnnotatedClass(User.class);
            configuration.addAnnotatedClass(Expense.class);

            ServiceRegistry serviceRegistry = new
StandardServiceRegistryBuilder().applySettings(configuration.getProperties()).b
uild();

            sessionFactory =
configuration.buildSessionFactory(serviceRegistry);
        }
        return sessionFactory;
    }
}

```

5. Testing

5.1 Methodologies used for testing

In a Online Expense Tracker project, various testing methodologies are essential to ensure the system's functionality, security, and reliability. Here's a brief summary of the testing methodologies commonly used:

1. Unit Testing:

- Purpose: Verify individual units or components.
- Scope: Functions, methods, or procedures.

2. Integration Testing:

- Purpose: Verify interactions between integrated components.
- Scope: Combined components.

3. System Testing:

- Purpose: Validate the entire system's compliance.
- Scope: System as a whole.

4. Acceptance Testing:

- Purpose: Ensure the system meets business requirements.
- Scope: Entire system.

5. Regression Testing:

- Purpose: Ensure new changes don't impact existing functionalities.
- Scope: Previously executed test cases.

6. Performance Testing:

- Purpose: Assess responsiveness and scalability.
- Types: Load testing, stress testing.

7. Security Testing:

- Purpose: Identify vulnerabilities.
- Types: Penetration testing, vulnerability scanning.

8. UI Testing:

- Purpose: Validate usability and interface.
- Scope: Graphical User Interface.

9. Database Testing:

- Purpose: Verify database operations.
- Scope: Data retrieval, storage, and manipulation.

10. Usability Testing:

- Purpose: Assess user-friendliness and experience.
- Scope: Overall user interface.

5.2 Types of Testing

In a online Expense Tracker project, various types of testing are crucial to ensure the system's quality and reliability. Here's a brief overview of the types of testing commonly conducted:

1. Unit Testing:

- Purpose: Verify individual units or components.
- Scope: Functions, methods, or procedures.

2. Integration Testing:

- Purpose: Verify interactions between integrated components.
- Scope: Combined components.

3. System Testing:

- Purpose: Validate the entire system's compliance.
- Scope: System as a whole.

4. Acceptance Testing:

- Purpose: Ensure the system meets business requirements.
- Scope: Entire system.

5. Regression Testing:

- Purpose: Ensure new changes don't impact existing functionalities.
- Scope: Previously executed test cases.

6. Performance Testing:

- Purpose: Assess responsiveness and scalability.
- Types: Load testing, stress testing.

7. Security Testing:

- Purpose: Identify vulnerabilities.
- Types: Penetration testing, vulnerability scanning.

8. User Interface (UI) Testing:

- Purpose: Validate usability and interface.
- Scope: Graphical User Interface.

9. Database Testing:

- Purpose: Verify database operations.
- Scope: Data retrieval, storage, and manipulation.

10. Usability Testing:

- Purpose: Assess user-friendliness and experience.
- Scope: Overall user interface.

6. CONCLUSION

In conclusion, an expense tracker serves as an essential tool for managing personal finances effectively. By providing a clear and comprehensive overview of income, expenses, and spending patterns, it empowers individuals to make informed financial decisions, set realistic budgets, and achieve their financial goals. The systematic recording and categorization of expenditures not only help in identifying unnecessary spending but also in planning for future financial needs. Overall, an expense tracker promotes financial discipline, enhances savings, and contributes to a more secure and organized financial future.

7. LIMITATIONS

Despite its numerous benefits, an expense tracker has certain limitations that users should be aware of. Firstly, the accuracy of the tracker heavily depends on the user's consistency and diligence in recording every transaction, which can be time-consuming and prone to human error.

Additionally, categorizing expenses can sometimes be subjective, leading to potential misclassification that affects the accuracy of financial insights. Moreover, many expense trackers require manual data entry, which may not capture all expenses, especially those made in cash.

Privacy concerns also arise, as storing financial data digitally can expose users to risks of data breaches and unauthorized access. Finally, while an expense tracker provides valuable insights, it does not automatically lead to better financial habits; users must actively use the information provided to make sound financial decisions.

8. FUTURE ENCHANMENTS

Future enhancements for expense trackers can significantly improve their functionality and user experience.

Integration with banking and financial institutions can automate the data entry process by syncing transactions directly from users' accounts, thereby reducing manual input and minimizing errors.

Advanced artificial intelligence and machine learning algorithms can offer personalized financial advice, predict future spending patterns, and provide actionable insights based on users' financial behaviors.

Enhanced data security measures, such as end-to-end encryption and multi-factor authentication, can address privacy concerns and protect sensitive financial information. Additionally, incorporating features like receipt scanning and optical character recognition (OCR) can streamline the recording of cash transactions.

Multi-platform accessibility, including seamless synchronization across devices, ensures that users can track their expenses anytime and anywhere. Finally, incorporating gamification elements, such as rewards for meeting budgeting goals, can make financial management more engaging and motivate users to maintain better financial habits.

9. REFERENCES

1. Unique Developer YouTube Channel (<https://www.youtube.com/watch?v=zdWfyBXO2iU>)
2. Online Expense Tracker System
Website: https://ijirt.org/master/publishedpaper/IJIRT150860_PAPER.pdf
3. Learning MYSQL, JavaScript, jQuery,, HTML, CSS3,
Website: <http://www.w3schools.com>
4. MySQL video tutorials
Website: <http://www.freehinditutorial.com>, <http://www.youtube.com>
5. JavaScript validation for empty input field
Website: <http://stackoverflow.com/questions/3937513/javascript-validation-for-empty-input-field> ,