



**INSTITUTE FOR ADVANCED COMPUTING AND
SOFTWARE DEVELOPMENT, AKURDI, PUNE**

“SpendWise”

PG-DAC Feb 2025

Submitted By:

Group No: 42

Roll No.	Name of Student
252215	Rohit Thakare
252073	Pratik More

Mr. Harshal Waghchaure

Project Guide

Mr. Prashant Deshpande

Centre Coordinator

ABSTRACT

SpendWise is a comprehensive Expense Management System designed to streamline expense tracking for both individuals and organizations. It replaces traditional paper-based recording methods with a modern, digital platform that enables quick, accurate, and efficient management of income and expenses. The system offers an intuitive interface for adding, categorizing, and analyzing financial transactions, helping users gain real-time insights into their spending patterns and improve financial decision-making. The purpose of SpendWise is to define the business, functional, and non-functional requirements for an integrated expense tracking platform. This document outlines the system's scope, objectives, and constraints to ensure a unified understanding for stakeholders. It aims to provide a secure, scalable, and user-friendly solution for personal and organizational financial management.

ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my heartfelt thanks to our esteemed guide, **Mr. Harshal Waghchaure** for providing me with the right guidance and advice at the crucial juncture and showing me the right way. I sincerely thank our respected Centre Co-Ordinator, **Mr. Prashant Deshpande**, for allowing us to use the available facilities. I would also like to thank the other faculty members at this occasion. Last but not least, I would like to thank my friends and family for the support and encouragement they have given me during our work.

Rohit Thakare (250241220204)

Pratik More (250241220148)

Table of Contents

Sr.No	Description	Page No.
1	Introduction	1
2	SRS	13
3	Diagrams	18
3.1	ER Diagram	18
3.2	Use Case Diagram	19
3.3	Data Flow Diagram	20
3.4	Activity Diagram	21
3.5	Class Diagram	23
3.6	Sequence Diagram	24
4	Database Design	25
5	Snapshots	28
6	Conclusion	33
7	Future Scope	34
8	References	35

1. INTRODUCTION

SpendWise is a comprehensive expense management system designed to revolutionize how individuals and businesses track and analyze their finances. This web-based platform provides an all-in-one solution for recording transactions, monitoring spending patterns, and generating insightful financial reports. Built with a modern **React.js frontend** and **Spring Boot backend**, SpendWise delivers a seamless user experience with real-time data synchronization and responsive design. The system incorporates robust **Spring Security** with JWT authentication to ensure complete protection of sensitive financial data.

Unlike basic expense trackers, SpendWise offers advanced features like automated transaction categorization, customizable budget alerts, and multi-user access for organizational accounts. The platform supports various user roles including individual budgeters, financial managers, and system administrators, each with tailored functionalities. With its intuitive dashboard and powerful analytics, SpendWise transforms complex financial data into actionable insights, helping users make smarter spending decisions and achieve their financial goals. The system's architecture prioritizes security, scalability, and user-friendliness, making it an ideal solution for both personal finance management and business accounting needs.

Admin Role

The Admin is the highest authority in the Expense Management System. Admins are typically platform owners or system operators who oversee all modules, users, and configurations.

❖ Responsibilities:

• Login & Authentication

- Secure admin access with multi-level checks

• User Authentication

- Verify and approve user and business accounts
- Monitor user activity and ensure compliance

• User Categorization

- Assign roles (individual user, organization)
- Manage permissions and access levels

• System Management

- Monitor overall system activity
- Manage accounts and ensure platform security
- Enforce access control policies via Spring Security
 - Manage website content like banners, announcements, and promotions.
- **Security Management**
 - Monitor security logs and apply access control policies via Spring Security
 - Investigate and respond to unauthorized access attempts or policy violations

❖ Access Level:

- Full access to all system modules and sensitive data
 - Authority to override or delete user-generated content
 - Can access and update global configurations and financial data
- **Message Acknowledgment**
 - Send notifications after registration, transactions, and major actions
- ❖ **Access Level:**
 - Full access to all modules and user data
 - Authority to override or delete user-generated content
 - Can configure platform-wide settings and security policies

User Role

Users are individuals who manage their personal expenses. They interact with modules related to budgeting, expense tracking, and group sharing.

- ❖ **Responsibilities:**
 - Register/login and manage personal account
 - Recover password via security questions and email
 - Update profile details and security settings
 - Add, edit, or delete expense entries
 - Categorize expenses (e.g., food, travel, groceries)
 - Generate mini statements by date range or summary
 - Set budgets and analyze spending patterns
 - Create groups for shared expenses (e.g., trips, households)
 - Split expenses among group members (equally, by percentage, or custom)
 - Upload and scan receipts to extract expense details
- ❖ **Access Level:**
 - Access to personal dashboard, expense records, and budget tools
 - Cannot manage other users or system-wide settings

Small Business Role

Small Businesses are organizational users who manage both income and expenses. They utilize advanced features for departmental tracking and financial reporting.

❖ Responsibilities:

- Register/login and manage business account
- Recover password via security questions and email
- Update business profile and security settings
- Track and manage income and expenses
- Organize expenses by departments or sub-heads
- Generate mini statements for various timeframes
- Categorize expenses for detailed reporting
- Verify expenses submitted by employees or contractors
- Analyze over-expenditure against set limits
- Process reimbursements for approved expenses
- Generate detailed financial reports
- Set and monitor departmental budgets

❖ Access Level:

- Access to business dashboard, financial tools, and reporting features
- Cannot manage other businesses or platform-wide configurations

Role-Based Access Control (RBAC)

The Expense Management System uses Spring Security to enforce RBAC, ensuring users only access features permitted by their role.

❖ Key Access Rules:

- **Admin:** Full access to all modules, users, and system configurations
- **User:** Can manage personal expenses and groups, but not other users
- **Small Business:** Can manage organizational finances, but not platform settings or other businesses

❖ Security Considerations:

- **Authentication:** Secure login using Spring Security and session tokens
- **Authorization:** Route and action-level access restrictions
- **Session Management:** Timed session expiry, secure cookies, and logout mechanisms
- **Data Protection:** No access to others' sensitive or financial data unless permitted

1.1 Purpose:

The purpose of the **SpendWise** system is to provide a comprehensive and user-centric platform for managing personal and organizational finances. It is designed to streamline operations such as expense tracking, budget planning, financial reporting, and collaborative expense sharing. By integrating **role-based access control (RBAC)** and modern web technologies like **React.js**, **Spring Boot**, and **JWT-based Spring Security**, SpendWise ensures secure, scalable, and efficient financial management.

The platform empowers users—from individuals to small businesses—to gain actionable insights into their spending habits, set financial goals, and make informed decisions. With features like automated categorization, multi-user grouping, and real-time analytics, SpendWise bridges the gap between raw financial data and strategic financial planning.

1.2 Scope:

The scope of the **SpendWise** project includes the complete design, development, deployment, and maintenance of a responsive web-based expense management system. The platform supports role-specific functionalities for **Users**, **Small Businesses**, and **Admins**:

- **Users** can record and categorize expenses, set budgets, analyze spending patterns, and collaborate in shared expense groups.
- **Small Businesses** can manage both income and expenses, organize departmental budgets, verify employee submissions, and generate detailed financial reports.
- **Admins** oversee system-wide operations including user authentication, role assignment, security enforcement, content management, and platform configuration.

SpendWise is built to support secure multi-role access, real-time data synchronization, and scalable architecture, making it suitable for both personal finance and business accounting needs.

system incorporates subscription-based access, secure login mechanisms, and plans for future integration with payment gateways and advanced search features like location-based filters.

1.3 Objectives:

The objectives of the **SpendWise** platform define its functional and strategic goals, ensuring the system effectively serves its diverse user base while maintaining security, scalability, and performance.

- 2 **Efficient Expense Management** – To enable users and businesses to record, categorize, and manage financial transactions with detailed metadata such as type, date, amount, and description.
- 3 **Streamlined Financial Tracking** – To provide individuals and organizations with intuitive interfaces for tracking expenses, generating statements, and analyzing spending behavior.
- 4 **Enhanced Role-Based Experience** – To deliver customized dashboards and functionalities tailored to each role—User, Small Business, and Admin.

- 5 **Robust Security and Access Control** – To ensure secure authentication, authorization, and data protection using Spring Security and JWT-based RBAC.
- 6 **Subscription-Based Monetization** – To implement feature-based subscription plans that support scalable monetization and controlled access to premium tools.
- 7 **Scalable and Maintainable Architecture** – To utilize a modular technology stack (React.js, Spring Boot) that supports future enhancements, mobile app integration, and third-party API connectivity.
 - **Comprehensive Admin Oversight** - To provide the Admin with full control over user roles, plans, property listings, and system settings.
 - **Enhanced User Engagement** -To deliver an intuitive and responsive UI/UX experience across all devices, improving user satisfaction.
 - **Transparent Communication and Trust Building** -To facilitate transparent communication between parties, improving trust and reducing delays in deals.
 - **Preparation for Future Enhancements** -To lay the groundwork for upcoming features such as real-time chat, document upload, and AI-based property recommendations.

7.2 Functionalities Provided by SpendWise

1. User Management

- **User Registration and Login** Users (individuals and businesses) can create accounts, securely log in, and manage their profiles. Authentication and authorization are handled using Spring Security with session management and encrypted credentials.
- **Role-Based Access Control (RBAC)** The system defines specific roles: Admin, User, and Small Business. Each role has predefined access privileges:
 - Users can manage personal expenses and groups.
 - Small Businesses can manage income, expenses, and departmental budgets.
 - Admins oversee all data, users, and system configurations.

- **Profile Management** Users can update contact details, preferences, and passwords. Features include password reset, profile image upload, and role-based dashboard customization.

2. Expense and Budget Management

- **Expense Entry and Modification** Users and businesses can add, edit, or delete expense records with details like category, amount, date, and description.
- **Multi-Categorization and Grouping** Expenses can be categorized (e.g., food, travel, utilities) and grouped by departments or shared user groups.
- **Mini Statement Generation** Users can generate summaries of transactions based on custom date ranges or monthly/annual views.
- **Budget Setting and Analysis** Users can define budgets and compare actual spending against targets to identify patterns and control overspending.
- **Receipt Scanning** Upload scanned receipts and extract expense details using OCR-based tools.

4. Admin Panel Features

- **System Monitoring** Admins can view user activity, approve business accounts, manage complaints, and enforce platform policies.
- **Report Generation** Auto-generated reports include:
 - Total expenses per user
 - Budget usage
 - Business activity
 - Platform engagement metrics
- **User Role and Plan Management** Admins can assign roles, block/unblock users, and update subscription validity.

5. Customer Engagement and Support

- **Search and Filter Options** Users can filter expenses by category, date, amount, and group.
- **Feedback & Review System (Future Scope)** Users may rate or review shared expense groups or business features for transparency.
- **Group Communication Tools (Optional Enhancement)** Secure messaging or comment threads within shared groups for collaborative expense tracking.

6. Reporting and Analytics

- **User Analytics** Insights into user behavior such as frequent categories, spending trends, and group activity.
- **Expense Analytics** Metrics like most common expense types, highest spending periods, and budget adherence.
- **Financial Analytics** Subscription revenue, plan performance, and monthly earnings for admin oversight.

7. Security and Compliance

- **Data Protection and Encryption** All sensitive data including credentials and financial records are encrypted and securely stored.
- **Audit Logs** Tracks changes to expenses, login events, subscription activity, and admin actions.
- **Compliance** Designed to comply with legal and data protection standards, including GDPR principles.

8. User Experience Enhancement

- **Responsive Design** Built with responsive layouts using React.js, ensuring compatibility across mobile, tablet, and desktop devices.
- **Saved Reports and Alerts (Future Enhancement)** Users can save report configurations and receive alerts for budget breaches or new group activity.

9. Integration and Extensibility

- **RESTful APIs** Built-in APIs support integration with mobile apps, external accounting tools, or analytics platforms.
- **Modular Architecture** Developed using Spring Boot and MVC architecture, allowing seamless scaling and future feature integration.

2. SOFTWARE REQUIREMENT SPECIFICATION

The functional requirements for the **SpendWise** platform define the essential features and capabilities that the system must deliver to meet the expectations of its users, which include individuals, small businesses, and administrators. These requirements serve as a blueprint for the development lifecycle, ensuring that the platform fulfills its core mission of streamlined financial management, secure role-based access, and scalable operations.

User Management SpendWise shall allow new users to register by providing necessary details such as name, email, password, contact number, and role (User, Small Business, or Admin). Authentication will be enforced using Spring Security, validating credentials during login and maintaining secure session management. The system will implement Role-Based Access Control (RBAC) to differentiate permissions based on user roles: Admins will have full access to manage users, subscriptions, and monitor system activity; Users will manage personal expenses and groups; Small Businesses will handle organizational finances and reporting. All users will be able to view and update their profile information, including contact details, password, and subscription status.

Expense and Budget Management Users and Small Businesses shall be able to add new expense entries with details such as category, amount, date, and description. They will also have the ability to edit or delete existing records. The system will support browsing and filtering of expenses based on various criteria like category, date range, and group. Users can categorize expenses and generate mini statements for financial analysis. Wishlist-like functionality will allow users to mark specific transactions or categories for future review or budgeting.

Subscription and Plans Admins shall be able to create, update, and delete subscription plans that define access to premium features such as advanced analytics, group sharing, and receipt scanning. Users and Small Businesses must purchase a subscription plan to unlock these features. The system will track subscription status, including active, expired, or upgraded plans, and notify users accordingly.

Communication and Collaboration SpendWise will support internal messaging features to facilitate communication within shared expense groups or between business users and their teams. Users may also request meetings or collaborative sessions for financial planning or expense verification.

Reporting and Monitoring (Admin Only) Admins shall have access to detailed reports on user registrations, subscription usage, and platform activity. They will be able to view analytics related to total expenses, most active users, and plan performance. These insights will help monitor engagement and optimize system operations.

Security The platform shall enforce secure authentication using encrypted credentials and robust session management. Authorization will be strictly controlled through Spring Security's RBAC mechanisms, ensuring users only access permitted features. All sensitive data, including personal information and financial records, shall be encrypted to prevent unauthorized access and ensure compliance with data protection standards.

Non Functional Requirements

1. The non-functional requirements for the **SpendWise** platform define the quality attributes that ensure the system performs efficiently, remains reliable, and is easy to use and maintain. These requirements are critical to delivering a robust and scalable solution that meets user expectations and business continuity goals.
2. **Performance** SpendWise shall maintain optimal performance under normal operating conditions, with system response times not exceeding 2 seconds for user actions such as property search, listing updates, or subscription purchases. The platform shall be scalable to support growth in user base and data volume, accommodating at least 10,000 concurrent users without degradation in performance. During peak usage periods, the system shall sustain a throughput of at least 100 subscription transactions or property searches per second, ensuring smooth

and uninterrupted service.

3. **Reliability** The system shall maintain high availability, with an uptime of 99.9% over a 12-month period, guaranteeing consistent access for buyers, sellers, and traders. SpendWise shall be fault-tolerant, capable of continuing operations in the event of hardware or software failures with minimal disruption. Robust error handling mechanisms shall be implemented to gracefully manage system issues, providing users with clear and actionable feedback in cases such as failed payments or unavailable property data.
4. **Usability** SpendWise shall offer a user-friendly interface that is intuitive and easy to navigate. Key functionalities such as property search, listing creation, subscription management, and profile updates shall be clearly accessible, ensuring a seamless user experience across all roles.
5. **Maintainability**
6. The system shall be developed using a modular architecture, allowing individual components—such as property listings, subscription plans, or payment integration—to be updated or enhanced independently without impacting the overall system. Code quality shall adhere to industry best practices, with clean, well-documented, and maintainable code to facilitate future development. Comprehensive testing shall be conducted, including unit tests, integration tests, and user acceptance tests, to ensure system stability and reliability throughout its lifecycle.

7. Other Requirements

Hardware and Network Interfaces:

Back-end Server Configuration:

- Intel Pentium-IV Processor
- 8 GB RAM

Front-end Client Configuration:

- AMD Ryzen 5 Processor
- 128 MB SDRAM

- 10 GB Hard Disk Drive
- 104 Keys Keyboard
- PS2 Mouse with pad

Software Interfaces:

Software configuration for Back-end Services:

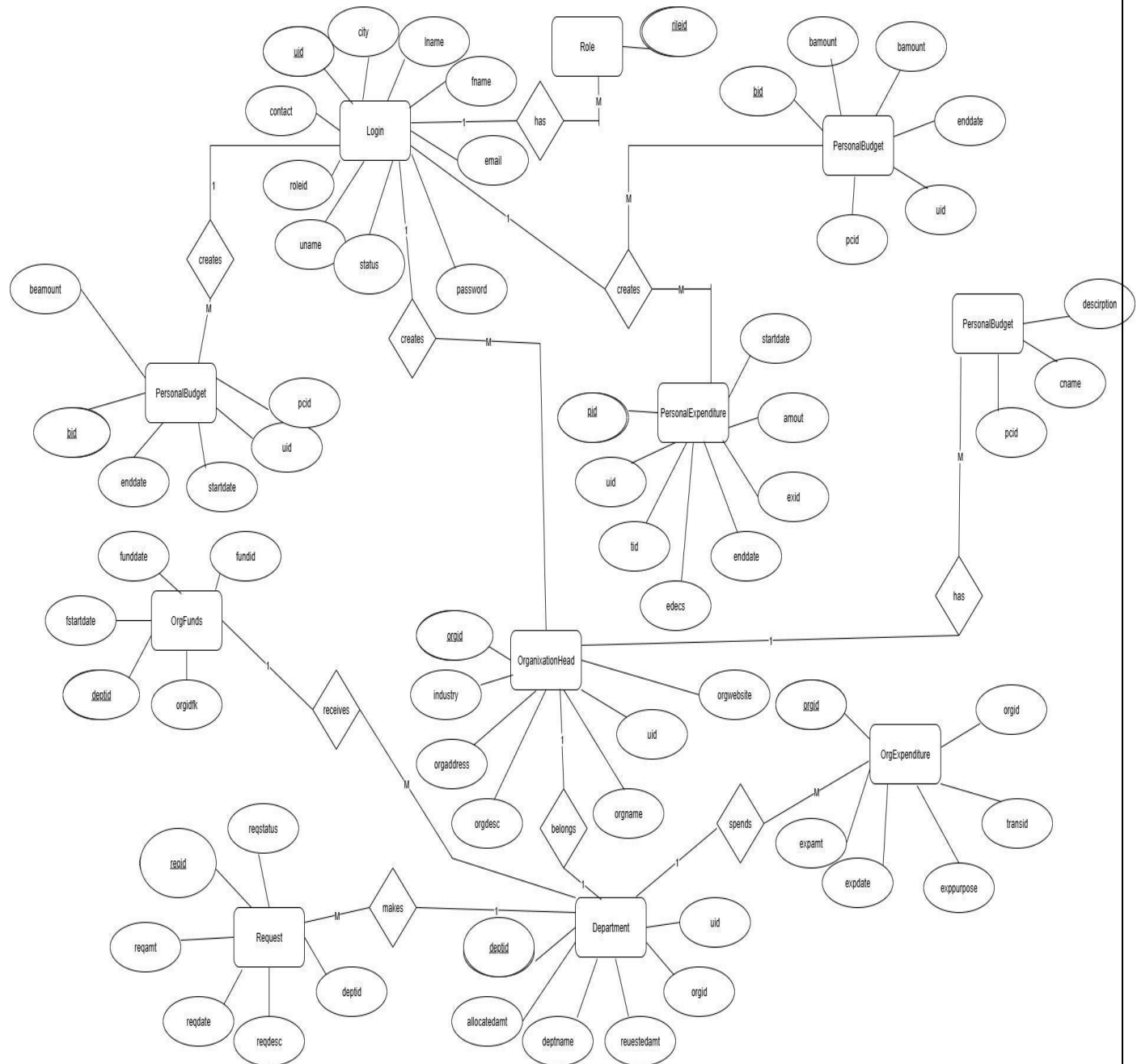
- Java 21
- .Net 9
- Spring Boot 3.4.8, Spring Security 6.4.2, JPA 3.4.8
- MySQL 8.0
- STS 4.3

Software configuration for Front-end Services:

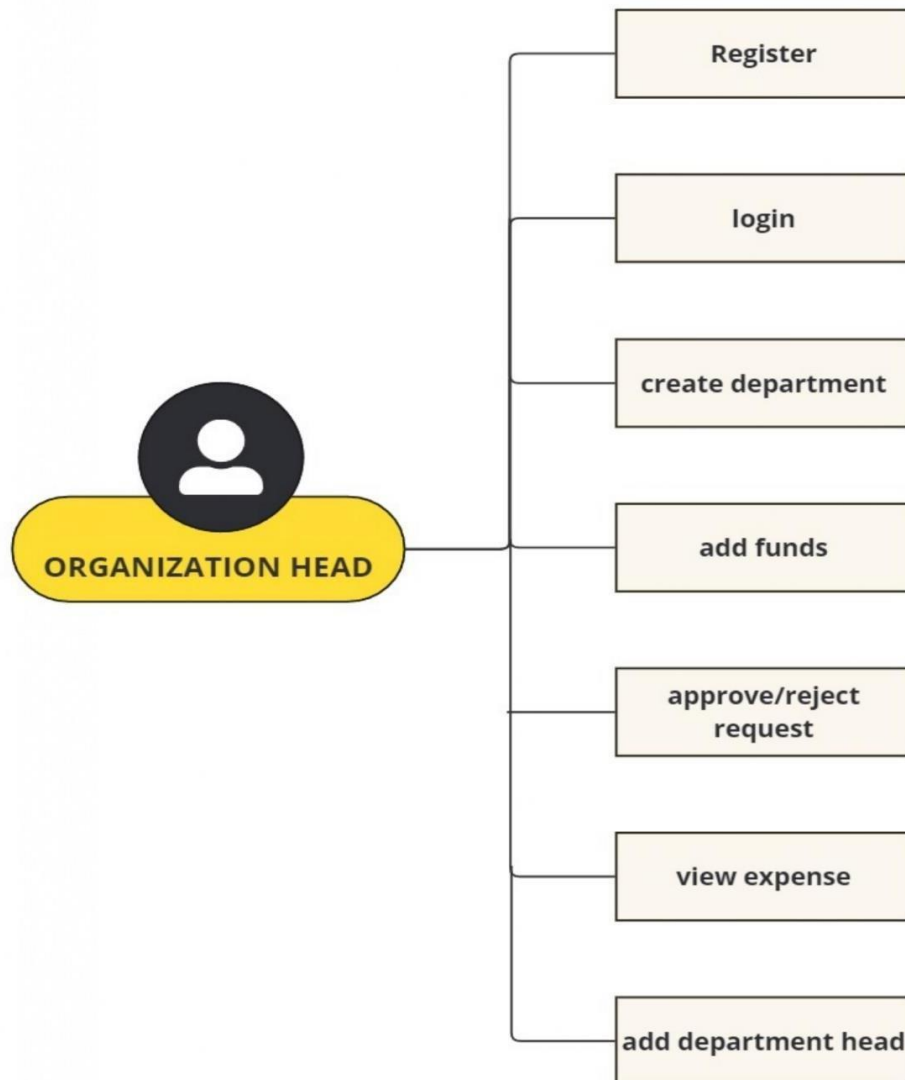
- ReactJS 19.1.1
- HTML5, CSS3, JavaScript ES16
- VS Code 1.103.0

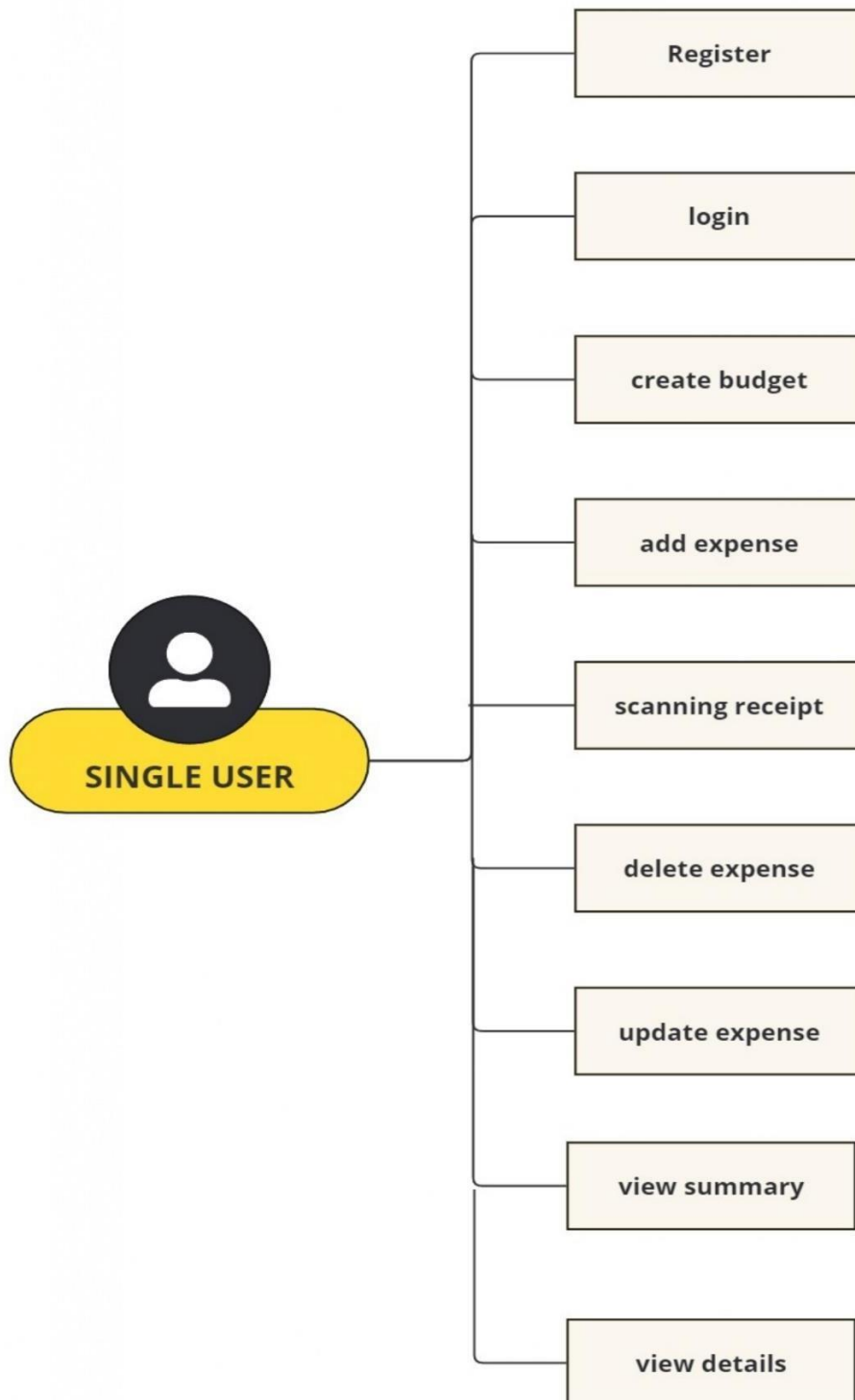
3. DIAGRAMS

3.1 Entity Relationship Diagram:



3.2 Use Case Diagram:



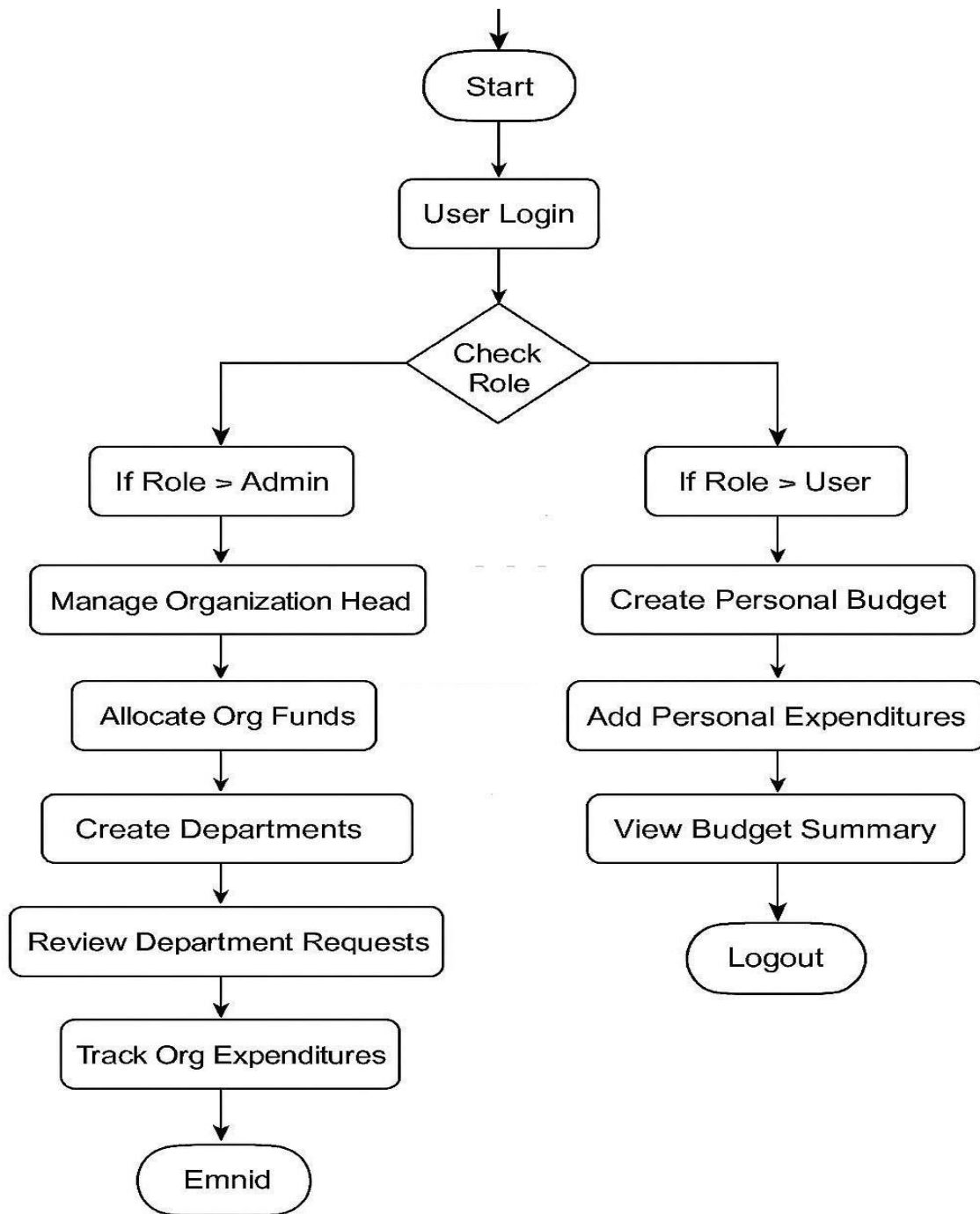


3.3 Data Flow Diagram:

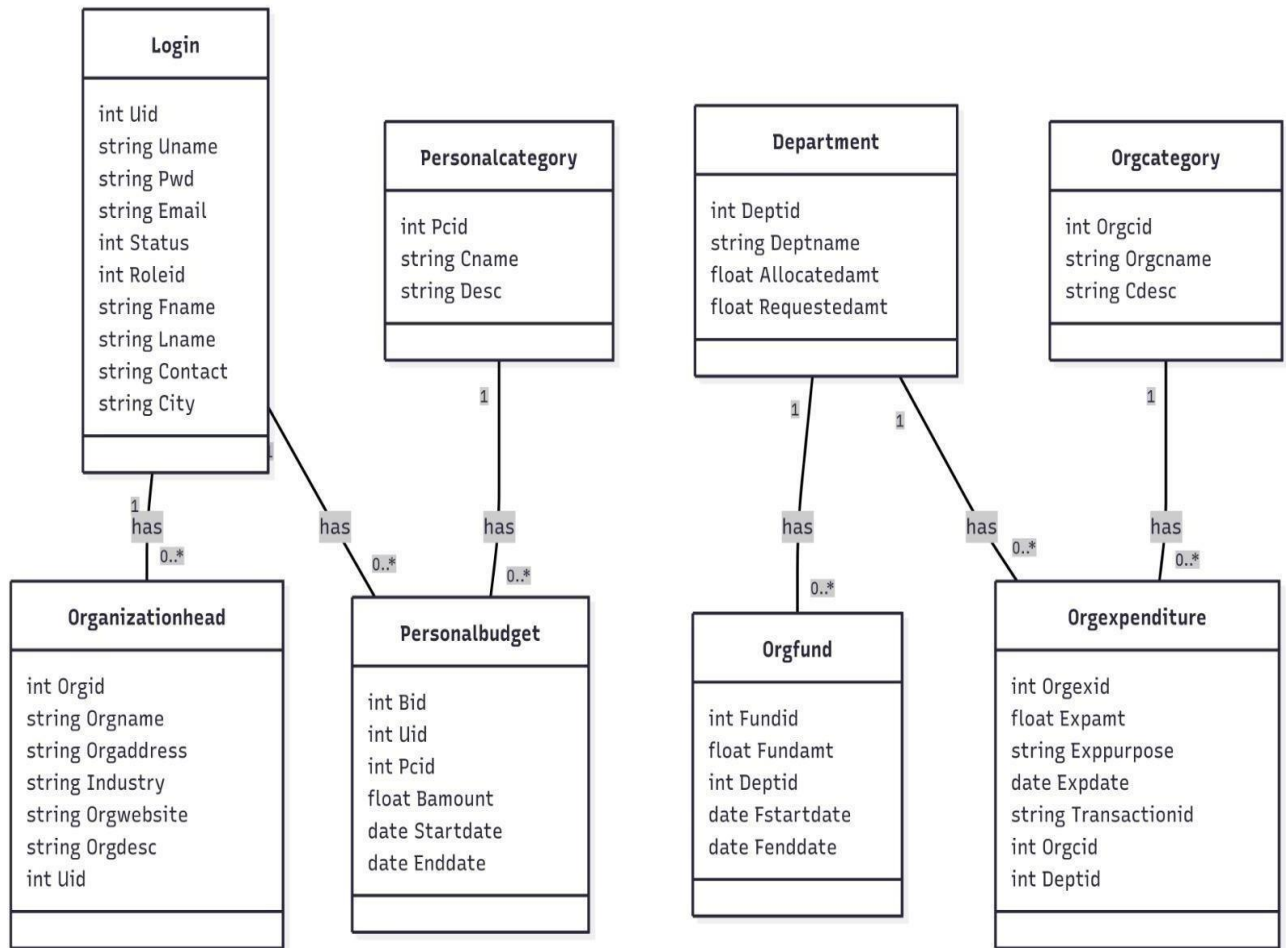
DFD Level 0:



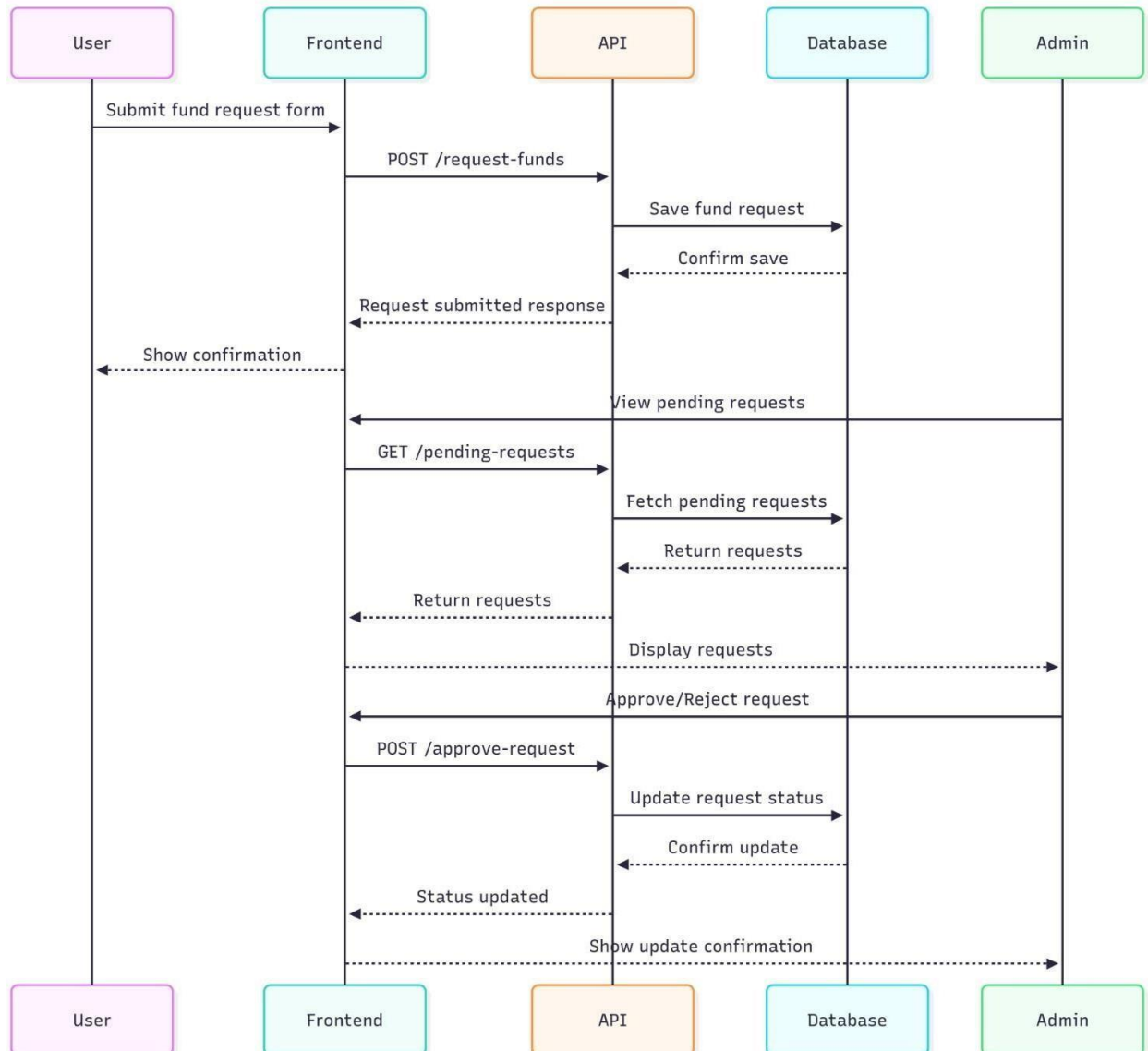
3.4 Activity Diagram :



3.5 Class Diagram:

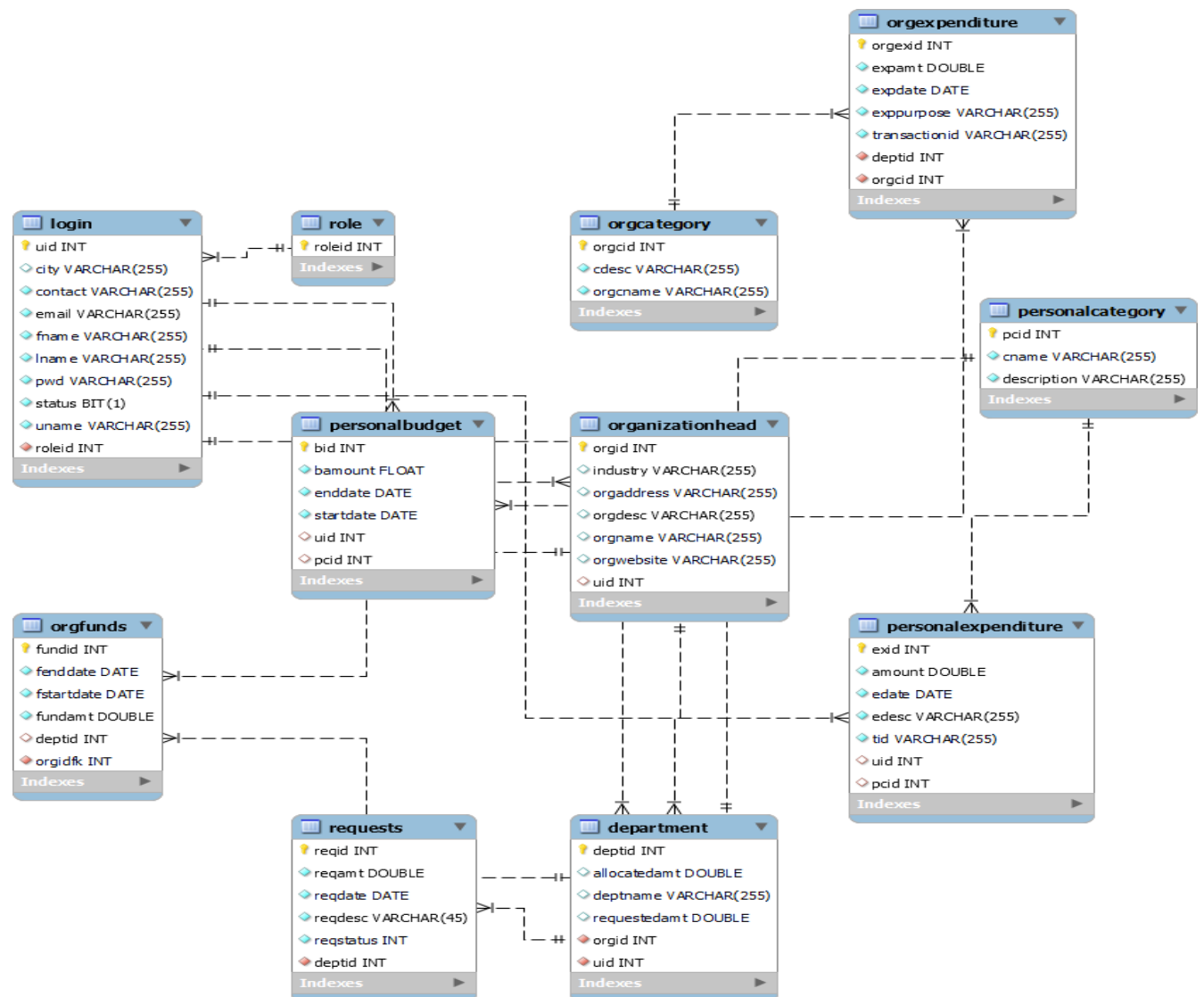


3.6 Sequence Diagram:



4. DATABASE DESIGN

4.1 Design:



4.2 Tables:

The following table structures depict the database design.

```
mysql> use projectdb;
Database changed
mysql> show tables;
+-----+
| Tables_in_projectdb |
+-----+
| department           |
| login                |
| organizationhead      |
| orgcategory           |
| orgexpenditure        |
| orgfunds              |
| personalbudget        |
| personalcategory      |
| personalexpenditure    |
| requests              |
| role                  |
+-----+
11 rows in set (0.00 sec)

mysql> desc department;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| deptid     | int           | NO   | PRI | NULL    | auto_increment |
| allocatedamt | double        | YES  |     | NULL    |                 |
| deptname    | varchar(255)  | YES  |     | NULL    |                 |
| requestedamt | double        | YES  |     | NULL    |                 |
| orgid      | int           | NO   | MUL | NULL    |                 |
| uid        | int           | NO   | MUL | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

```
mysql> desc login;
```

Field	Type	Null	Key	Default	Extra
uid	int	NO	PRI	NULL	auto_increment
city	varchar(255)	YES		NULL	
contact	varchar(255)	NO	UNI	NULL	
email	varchar(255)	NO	UNI	NULL	
fname	varchar(255)	NO		NULL	
lname	varchar(255)	NO		NULL	
pwd	varchar(255)	NO	UNI	NULL	
status	bit(1)	NO		NULL	
uname	varchar(255)	NO	UNI	NULL	
roleid	int	NO	MUL	NULL	

10 rows in set (0.00 sec)

```
mysql> desc organizationhead;
```

Field	Type	Null	Key	Default	Extra
orgid	int	NO	PRI	NULL	auto_increment
industry	varchar(255)	YES		NULL	
orgaddress	varchar(255)	YES		NULL	
orgdesc	varchar(255)	YES		NULL	
orgname	varchar(255)	YES		NULL	
orgwebsite	varchar(255)	YES		NULL	
uid	int	YES	UNI	NULL	

7 rows in set (0.00 sec)

```
mysql> desc orgcategory;
```

Field	Type	Null	Key	Default	Extra
orgcid	int	NO	PRI	NULL	auto_increment
cdesc	varchar(255)	NO		NULL	
orgcname	varchar(255)	NO	UNI	NULL	

3 rows in set (0.00 sec)

```
mysql> desc personalbudget;
```

Field	Type	Null	Key	Default	Extra
bid	int	NO	PRI	NULL	auto_increment
bamount	float	NO		NULL	
enddate	date	NO		NULL	
startdate	date	NO		NULL	
uid	int	YES	MUL	NULL	
pcid	int	YES	MUL	NULL	

6 rows in set (0.00 sec)

```
mysql> desc personalcategory;
```

Field	Type	Null	Key	Default	Extra
pcid	int	NO	PRI	NULL	auto_increment
cname	varchar(255)	NO	UNI	NULL	
description	varchar(255)	NO		NULL	

3 rows in set (0.00 sec)

```
mysql> desc personalexpenditure;
```

Field	Type	Null	Key	Default	Extra
exid	int	NO	PRI	NULL	auto_increment
amount	double	NO		NULL	
edate	date	NO		NULL	
edesc	varchar(255)	NO		NULL	
tid	varchar(255)	NO	UNI	NULL	
uid	int	YES	MUL	NULL	
pcid	int	YES	MUL	NULL	

7 rows in set (0.00 sec)

```
mysql> desc orgfunds;
```

Field	Type	Null	Key	Default	Extra
fundid	int	NO	PRI	NULL	auto_increment
fenddate	date	NO		NULL	
fstartdate	date	NO		NULL	
fundamt	double	NO		NULL	
deptid	int	YES	MUL	NULL	
orgidfk	int	NO	MUL	NULL	

6 rows in set (0.00 sec)

```
mysql> desc requests;
```

Field	Type	Null	Key	Default	Extra
reqid	int	NO	PRI	NULL	auto_increment
reqamt	double	NO		NULL	
reqdate	date	NO		NULL	
reqdesc	varchar(45)	NO		NULL	
reqstatus	int	NO		NULL	
deptid	int	NO	MUL	NULL	

```
6 rows in set (0.00 sec)
```

```
mysql> desc role;
```

Field	Type	Null	Key	Default	Extra
roleid	int	NO	PRI	NULL	auto_increment

```
1 row in set (0.00 sec)
```

5. SNAPSHOTS

Home Page:

Welcome to SpendWise

Your comprehensive solution for tracking, managing, and analyzing expenses effortlessly. SpendWise empowers both individuals and organizations to streamline their expense management process, ensuring accuracy, efficiency, and valuable insights.

Monitor Your Expenses

Keep track of all your expenses in one place. Our intuitive dashboard provides a clear overview of your spending, helping you stay on top of your financial goals.

Manage Your Budget

Create and manage budgets with ease. Set spending limits, track progress, and make adjustments to ensure you stay within your budget.

Analyze Your Spending

Gain valuable insights into your spending patterns with our advanced analytics tools. Generate reports, visualize data, and make informed financial decisions.

Streamline Expense Reporting

Submit and approve expense reports seamlessly. Our system simplifies the process of reporting and approving expenses, reducing administrative overhead and improving efficiency.

[Home](#) [Features](#) [About us](#) [Contact](#)[Login](#)

Register As

Single User

Register as an individual user.

Organization

Register as an organization.

[Home](#) [Features](#) [About us](#) [Contact](#) [Login](#)

Register As

Single User

Register as an individual user.

Organization

Register as an organization.

[Home](#) [Features](#) [About us](#) [Contact](#) [Login](#)

Single User Registration

First Name:

Last Name:

City:

Contact:

Email:

Username:

Password:

Confirm Password:

☐ Show Password

Registration Form

Username:	Password:	First Name:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Username is required	Password is required	
Last Name:	Organization Name:	Organization Description:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Company Mail:	Contact:	Industry:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Organization Website:	City:	Organization Address:
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Register"/>		

[Home](#) [Features](#) [About us](#) [Contact](#)[Login](#)

Login Page

Enter UID:

Enter Password:

[new user?](#)

Add Department

User Name :

Password :

Email ID :

First Name :

Last Name :

Contact :

City :

Department Name :

Allocated Amt :

[Add Expenses](#)[Set Budget](#)[Expense History](#)[Update Profile](#)[Logout](#)

Welcome, pratik More!

No Budget available.

[Home](#) [Add Expense](#) [Expense History](#) [Update Profile](#) [Logout](#)

Budget Creating

Budget Amount:

Personal Category

Start Date:

End Date:

[Set Budget](#)

[Home](#) [Add Expenses](#) [Set Budget](#) [Update Profile](#) [Logout](#)

Date	Description	Amount	Transaction Id	Category	Update
------	-------------	--------	----------------	----------	--------

[Home](#) [Set Budget](#) [Expense History](#) [Update Profile](#)[Logout](#)

Input Form

Description

Amount

Date

dd-mm-yyyy



Personal Category

Select a category

Transaction ID

Optional

[Submit](#)[Add Expenses](#) [Set Budget](#) [Expense History](#) [Update Profile](#)[Logout](#)

Username

Helopnada

Password

Email

rohith7@gmail.com

First Name

pratik

Last Name

More

Contact

9607012400

City

Pune

[Update](#)

6. CONCLUSION

SpendWise successfully addresses the limitations of traditional expense tracking by providing a secure, efficient, and user-friendly digital platform for individuals and organizations. With its modular architecture, real-time insights, and customization features, the system streamlines financial management, enabling users to make informed spending decisions. By integrating React, .NET, and Java, SpendWise ensures robust performance, security, and scalability for long-term usability.

7. Future Scope

- Future scope includes payment integration.
 - Enable direct communication between buyers, sellers, and traders.
 - Allow users to bookmark properties for easy access later.
 - Implement a secure and efficient image upload functionality in the backend to allow users to store and manage images.
 - Implement email-based OTP verification to enhance account security and ensure valid user registrations.
 - Implement an AI-powered recommendation engine to suggest relevant content, property, or services to users.
-
- **Mobile Application** – Develop native Android and iOS applications for enhanced portability and offline access.
 - **AI-Powered Insights** – Integrate machine learning algorithms to predict spending trends and suggest saving strategies.
 - **Multi-Currency Support** – Allow expense tracking in different currencies for global users.
 - **Bank API Integration** – Automate expense tracking by importing transactions directly from bank accounts.
 - **Advanced Reporting** – Include customizable dashboards with export options (PDF, Excel).
 - **Voice Command Integration** – Enable expense entry and analysis through voice assistants.

8. REFERENCES

1. Spring Security Documentation. Retrieved from:
2. <https://docs.spring.io/spring-security/reference/>
3. Spring Boot Documentation. Retrieved from:
<https://docs.spring.io/spring-boot/index.html>
4. Martin, R. C. (2002). *Agile Software Development: Principles, Patterns, and Practices*. Prentice Hall.
5. Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). *Database Systems: The Complete Book*. Prentice Hall.
6. Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13(6), 377–387.
7. Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
8. Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley.