Expense Claim & Approval System

Documentation

1. Introduction

Employees in the company submit reimbursement claims (travel, food, equipment, etc.), but the current process is handled via email and Excel leading to problems like lost requests, duplicate claims, no approval tracking, no audit trail, poor reporting visibility. This tracker system aims at solving these issues in

2. System requirements

2.1. Functional Requirements

- **Roll-based login and employee creation –** Users can login with their company email to perform a roll-based access to the system. This is achieved using the JWT tokens and cookies.
- **Claims Submission and Update –** Employees can update their claims autosaved in the form of drafts until submission.
- **Multi-level Claims Review and Approval –** Employee requests will be handled through layers like Employee to Manager to Finance, to prevent discrepancies and faulty reimbursements.
- **Maintenance of records and logs –** This will improve transparency and faults by promoting better auditing and information access to all employees with a centralized claims management system.
- **CSV Downloadable –** All the application users can download their history in a CSV file format.

2.2. Non-Functional Requirements

- **Security –** Ensures multi-level authentication and authorization through proper setup and login.
- **Performance –** The system handles concurrent users efficiently
- **Usability –** A user-friendly interface and intuitive interface.
- **Scalability –** The system is scalable and made keeping in mind to transition into the micro-services architecture for reduced cloud costs.

3. System Architecture & Design

3.1. UI Design

- **Login Page --**
- **Employee Dashboard --**
- **Manager Dashboard --**
- **Admin Dashboard --**
- **Financial Dashboard --**

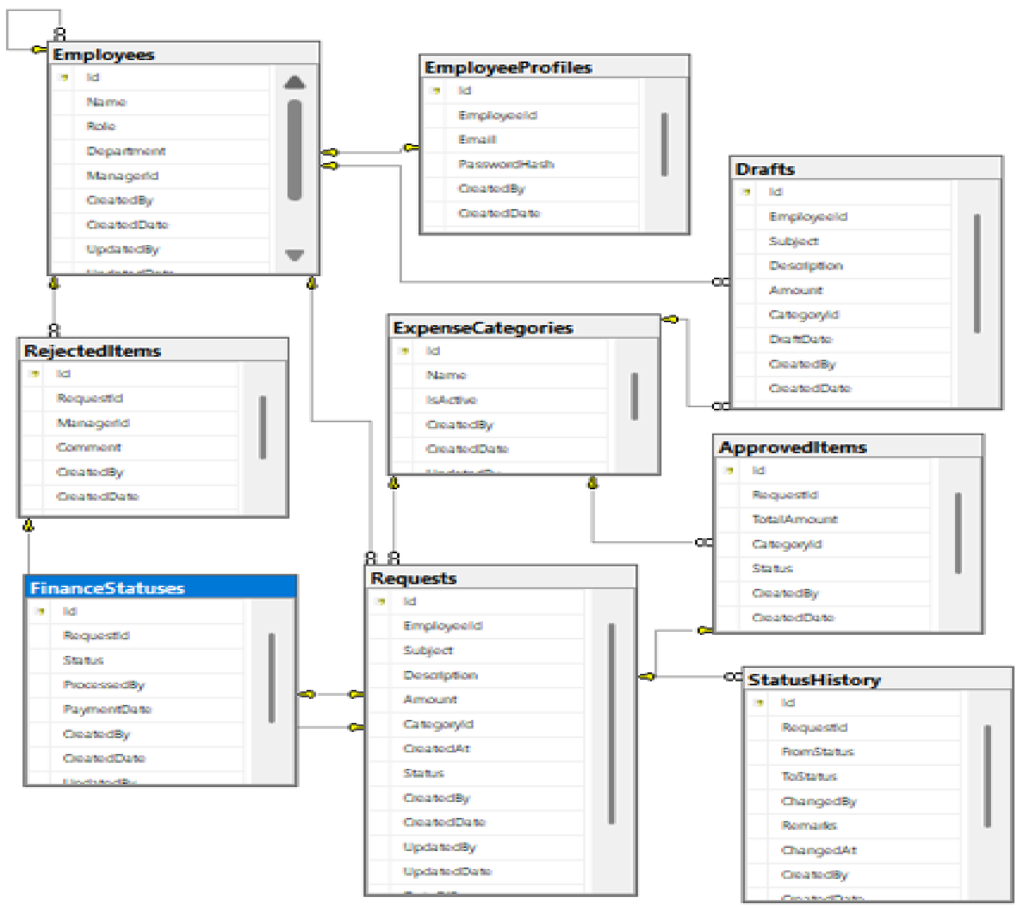
3.2. Database Design

- **Employees Table --**

3.3 API Design

- **Authentication API --**

- **Roll-based APIs**
  - **Manager Dashboard API --**
  - **Admin Dashboard API --**
  - **Financial Dashboard API --**
  - **Employee Dashboard --**
  - **Employees Management API –**



4. Code

This section explains the implementation of the **Expense Claim & Approval Management System**, including backend development, frontend implementation, security mechanisms, and system functionality.

The system is developed using a **full-stack architecture** to ensure modularity, maintainability, and scalability.

Technology Stack

Component	**Technology**
Frontend	React.js
Backend	ASP.NET Core Web API
Database	SQL Server
ORM	Entity Framework Core
Authentication	JWT (JSON Web Token)
UI Library	Material UI

The system follows modern development practices such as **layered architecture, API-based communication, and secure authentication mechanisms**.

4.1 Backend Implementation

The backend of the system is developed using **ASP.NET Core Web API**. It acts as the central processing unit of the system, handling business logic, database operations, authentication, and authorization.

The backend communicates with the frontend through **RESTful APIs**.

The backend performs the following major tasks:

- User authentication and authorization
- Expense claim processing
- Multi-level approval workflow
- Database communication
- Security and validation
- Audit logging

Each request from the frontend is processed through APIs, validated, and then stored or retrieved from the database.

4.1.1 Authentication Implementation

Authentication is implemented using **JSON Web Tokens (JWT)**.

When a user logs in:

1. The user provides their email and password.
2. The backend validates the credentials against the database.
3. If the credentials are correct, a **JWT token** is generated.
4. The token is returned to the client.
5. The client includes the token in the header for all future API requests.

Example request header:

```
Authorization: Bearer <JWT_TOKEN>
```

This ensures that only authenticated users can access protected endpoints.

4.1.2 Role-Based Authorization

The system implements **role-based access control (RBAC)** to restrict actions based on user roles.

Different roles have different privileges within the system.

Employee Permissions

Employees can:

- Create expense claims
- Save claims as drafts
- Edit draft claims
- Submit claims for approval
- View claim history
- Track claim status
- Download claim reports

Manager Permissions

Managers can:

- View claims submitted by their team members
- Approve claims
- Reject claims with comments
- Track approval history

Managers act as the **first level of approval** in the workflow.

Finance Permissions

Finance users can:

- Review manager-approved claims
- Verify expense details
- Approve reimbursements
- Reject suspicious claims
- Generate financial reports

Finance acts as the **final approval authority**.

Admin Permissions

Administrators have full system control and can:

- Create employee accounts
- Assign roles
- Manage employees
- Monitor system activity
- Access audit logs

4.2 API Implementation

The backend exposes several **REST APIs** that allow the frontend to interact with the system.

These APIs handle authentication, claim management, approval workflows, and employee management.

Authentication API

Handles login and token generation.

Functions include:

- User login
- Token generation
- Session management

Employee Claim APIs

These APIs allow employees to manage their expense claims.

Key operations include:

- Create new claim
- Update draft claim
- Submit claim
- View claim history
- Delete draft claims

These APIs ensure that employees can manage their claims efficiently.

Manager APIs

Manager APIs enable managers to process claims submitted by their team members.

Key operations include:

- View team claims
- Approve claims
- Reject claims
- Add approval comments

Managers act as the **first validation layer** in the approval process.

Finance APIs

Finance APIs allow finance staff to perform financial verification and reimbursement approvals.

Key operations include:

- View manager-approved claims
- Verify claims

- Approve reimbursements
- Reject invalid claims
- Generate expense reports

Admin APIs

Admin APIs allow administrators to manage employees and system configurations.

Key operations include:

- Create employees
- Update employee roles
- Activate or deactivate accounts
- View employee list

4.3 Frontend Implementation

The frontend of the system is developed using **React.js**, which provides a responsive and dynamic user interface.

The interface is designed using **Material UI components** to provide a modern and user-friendly experience.

The frontend communicates with the backend through **HTTP API requests** using libraries such as **Axios or Fetch**.

The frontend is responsible for:

- Rendering the user interface
- Managing user sessions
- Handling form submissions
- Displaying claim data
- Providing role-based dashboards

Login Page

The login page allows users to authenticate using their email and password.

The page performs the following actions:

1. Accepts user credentials.
2. Sends a login request to the backend API.
3. Stores the JWT token returned by the server.
4. Redirects the user to their respective dashboard.

Employee Dashboard

The employee dashboard allows employees to manage their expense claims.

Features include:

- Submit new claims
- Edit draft claims
- View claim history
- Track claim approval status
- Download claim data in CSV format

The dashboard provides employees with a clear overview of their reimbursement activities.

Manager Dashboard

The manager dashboard provides tools for reviewing and approving employee claims.

Managers can:

- View claims submitted by their team
- Filter claims by status
- Approve or reject claims
- Provide comments on rejected claims

Finance Dashboard

The finance dashboard focuses on financial verification and reimbursement management.

Finance users can:

- Review manager-approved claims
- Validate claim information
- Approve reimbursements
- Generate financial reports

Admin Dashboard

The admin dashboard provides administrative control over the system.

Administrators can:

- Manage employee accounts
- Assign roles
- Monitor system activity
- Access logs and reports

4.4 Security Implementation

Security is an essential aspect of the system.

The application implements several security mechanisms.

Authentication Security

- JWT token-based authentication
- Secure token validation for every request

Password Security

User passwords are stored securely using **hashing algorithms** to prevent exposure of sensitive data.

Passwords are never stored in plain text.

API Protection

All protected endpoints require **valid JWT tokens**.

Unauthorized users cannot access restricted APIs.

Role-Based Access Control

The system enforces role permissions to ensure that users can only perform actions allowed by their roles.

For example:

- Employees cannot approve claims.
- Managers cannot manage employee accounts.
- Finance users cannot create employees.

4.5 Error Handling

The system includes structured error handling to manage unexpected situations.

Common errors handled include:

- Invalid login credentials
- Unauthorized access
- Missing or incorrect data
- Server failures

The system returns standard **HTTP status codes**.

Status Code	Meaning
200	Successful request
400	Bad request
401	Unauthorized
404	Resource not found
500	Internal server error

4.6 Logging & Monitoring

Logging is implemented to monitor system activities and track errors.

The system records important events such as:

- Login attempts
- Claim submissions
- Approval actions
- System errors

Logs help administrators perform audits and troubleshoot issues.

5. Future Enhancements

The system can be further improved with additional features such as:

- Email notifications for claim approvals and rejections
- Receipt image upload functionality
- AI-based fraud detection
- Mobile application support
- Integration with payroll systems
- Advanced analytics dashboards

6. Conclusion

The **Expense Claim & Approval Management System** provides a secure, scalable, and efficient solution for managing employee reimbursement requests.By replacing manual processes with a centralized digital system, the application improves transparency, reduces errors, and ensures smooth claim processing.The system demonstrates modern web development practices and provides a strong foundation for future enhancements and enterprise deployment.