



<Lone Coder>

<Effort Tracking System> Software requirement specification

Version <1.0>

Submitted in Partial Fulfillment for the Award of Degree of Bachelor of Technology in Information
Technology from Rajasthan Technical University, Kota

MENTOR:

Mr. Manoj Raman
(Dept. of Information Technology)

SUBMITTED BY:

Apeksha Pitaliya
(21ESKIT019)

COORDINATOR:

Dr. Priyanka Yadav
(Dept. of Information Technology)

DEPARTMENT OF INFORMATION TECHNOLOGY

SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN
Ramnagar (Jagatpura), Jaipur – 302017

SESSION 2024-25

Effort Tracking System	Version: 1.0
Software Requirement Specification	Date: 15/10/2024

Revision History

Version	Date	Author	Description
1.0	01-Sep-2024	Initial Developer	Initial project setup and documentation created.
1.0	10-Oct-2024	Backend Team	Added employee and attendance modules.
1.0	11-Nov-2024	Frontend Developer	Integrated React frontend with Django backend.
1.0	15-Nov-2024	QA Team	Performed unit and integration testing.
1.0	20-Dec-2024	DevOps Engineer	Deployed system and fixed minor deployment bugs.
1.0	01-Jan-2025	Project Manager	Finalized documentation and user manual.

Effort Tracking System	Version: 1.0
Software Requirement Specification	Date: 15/10/2024

Table of Contents

1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms and Abbreviations
- 1.4 References
- 1.5 Technologies to be used
- 1.6 Overview

2. Literature survey

- 2.1 Review of Related Work
- 2.2 Knowledge gaps
- 2.3 Comparative Analysis
- 2.4 Summary

3. Specific Requirements

- 3.1 Functional Requirements
- 3.2 Non- Functional Requirements
- 3.3 Hardware Requirements
- 3.4 Software Requirements
- 3.5 Agile Methodology
- 3.6 Business Process Model
- 3.7 Supplementary Requirements

4. System Architecture

- 4.1 Client-Server Architecture
- 4.2 Communications Interfaces

5. Design and Implementation

- 5.1 Product feature
- 5.2 Data Flow diagram
- 5.3 E-R Diagram
- 5.4 Class Diagram
- 5.5 Use-Case Model Survey
- 5.6 Behaviors Diagrams
- 5.7 Database Diagram
- 5.8 Assumptions and Dependencies

6. Supporting Information

7. Conclusion & Future scope

8. Concerns / Queries / Doubts if any:

Effort Tracking System	Version: 1.0
Software Requirement Specification	Date: 15/10/2024

1. Introduction

The **Django-React HR System** is a full-stack web application designed to assist HR departments in managing employees and tracking attendance efficiently. Built using Django for the backend and React for the frontend, the system offers a secure login for HR personnel, features to add or update employee details, and manage daily attendance records. With a clean user interface and robust API communication, the system simplifies HR workflows and enhances data handling. It is ideal for organizations looking to automate their HR tasks and improve workforce management.

1.1 Purpose

The purpose of the Django-React Tracking System is to streamline and digitize HR operations such as employee management and attendance tracking. Built with Django for the backend and React for the frontend, the system allows HR personnel to securely log in, manage employee records, mark attendance, and view insights through a user-friendly dashboard. It aims to reduce manual work, enhance data accuracy, and improve overall efficiency in HR processes.

1.2 Scope

The scope of the **Django-React Tracking System** includes the streamlined management of employee records and attendance tracking within an organization. It allows HR professionals to securely log in, add, view, edit, and delete employee information, as well as mark and monitor attendance. The system is scalable for small to medium-sized organizations and can be extended to include features like payroll integration, leave management, and reporting. It supports web-based access, ensuring ease of use and availability from any location with internet access. The system aims to minimize manual HR tasks and enhance overall operational efficiency.

1.3 Definitions, Acronyms and Abbreviations

- **SRS:** Software Requirements Specification
- **API:** Application Programming Interface
- **UI:** User Interface
- **DBMS:** Database Management System

Effort Tracking system	Version: 1.0
Software Requirements Specification	Date: 15/10/2024

1.4 References

1. Django and React official documentation were referenced for backend and frontend development practices respectively.
2. Tutorials and community forums like Stack Overflow and GitHub were consulted for implementation guidance and troubleshooting.

1.5 Technologies to be used

- **Frontend:** React
- **Backend:** Django
- **Database:** SQLite

1.6 Overview

The project contains a complete **Effort Tracking System** built using **Django** for the backend and **React** for the frontend. This system is designed to help HR personnel efficiently manage employee information and attendance records. It includes key features like user authentication (for HR roles), employee creation and editing, attendance tracking, and a dashboard overview. The backend uses Django REST Framework to expose APIs, while the frontend consumes these APIs to provide a responsive and user-friendly interface. This project demonstrates full-stack development and integrates technologies such as Axios, Zustand, Chart.js, and JWT for secure and smooth operations.

2. Literature survey

Many companies still manage tracking tasks manually, which causes delays and errors. Modern web apps use frameworks like Django (for backend) and React (for frontend) for better performance.

Django helps in handling user data, security, and APIs easily.

React makes the UI fast and user-friendly for HR activities.

This project combines both to build an efficient and easy-to-use Effort tracking System.

2.1 Review of Related Work

This section reviews existing effort tracking systems to understand the landscape:

2.1.1 Overview of Existing Systems:

Many existing tracking systems are complex and require training to use.

Some are paid software with limited customization options.

Basic tools like Excel or Google Sheets are still used in small companies.

Effort Tracking system	Version: 1.0
Software Requirements Specification	Date: 15/10/2024

Older systems often lack features like real-time attendance or online user management. These limitations highlight the need for a modern, user-friendly Tracking system like this one

2.1.2 Key Functionalities:

The Django-React tracking System offers secure login for HR users with role-based access. It allows easy management of employee profiles, including personal and job-related details. HR can record, update, and view employee attendance for specific dates. A dashboard provides quick stats on employees and attendance. The system uses Django for backend APIs and React for a smooth, responsive frontend interface.

[A] Limitations in Current Systems:

- **Limited Customization:** Many systems lack flexibility to adapt to specific organizational workflows or employee structures.
- **Poor User Interface:** Some older systems have outdated, non-intuitive interfaces, making navigation difficult.
- **Lack of Real-Time Updates:** Attendance and employee data updates are not always reflected instantly.
- **Restricted Access Control:** Most systems don't support detailed role-based access for different HR levels.
- **Integration Issues:** They often don't integrate well with modern tools like payroll systems, communication apps, or analytics dashboards.

[B] Relevance to the Effort Tracker System:

- **Efficient Time Management:** It allows employees to log hours against specific tasks, helping track productivity in real time.
- **Enhanced Transparency:** Managers can monitor progress, ensuring accountability and fair distribution of work.
- **Data-Driven Decisions:** Insights from logged efforts help in performance evaluation and resource planning.
- **Improved Task Allocation:** HR and team leads can assign tasks based on availability and effort history.
- **Support for Remote Work:** Especially useful in hybrid/remote setups to ensure consistent workflow tracking.

2.2 Knowledge gaps

Here are some **Knowledge Gaps** identified in the context of existing employee and effort tracking systems:

- **Lack of Real-Time Analytics:** Many systems don't provide immediate insights into how time is spent across tasks or projects.

Effort Tracking system	Version: 1.0
Software Requirements Specification	Date: 15/10/2024

- **Limited Integration:** Existing tools often fail to integrate seamlessly with HR, payroll, or project management systems.
- **User Experience Issues:** Interfaces may be complex or unintuitive, especially for non-technical users.
- **Customization Constraints:** Many systems offer fixed features, making it hard to adapt to unique organizational needs.
- **Inadequate Tracking of Non-Task Activities:** Informal efforts like mentoring, meetings, or research often go untracked.

2.3 Comparative Analysis

A comparison of your Effort Tracker System with existing tools:

Feature	Existing Systems	Effort Tracker System
Employee Management	Basic profiles with limited data	Detailed profiles with full name, position, status, education, gender, image, etc.
Attendance Tracking	Manual or biometric-based, limited tracking	Digital tracking with status options (Present, Absent, Leave) and time-stamped entries
User Access Control	Limited role-based access	Role-based access control; only HR can manage employees and attendance
Frontend Interface	Outdated or less interactive	Modern, responsive UI using React
Technology Stack	Legacy or monolithic systems	Django REST Framework (backend) + React (frontend)
Customization & Scaling	Rigid structure, hard to extend	Modular design, easy to extend (e.g., adding task logging, reports)

Effort Tracking system	Version: 1.0
Software Requirements Specification	Date: 15/10/2024

2.4 Summary

The project is a full-stack **Effort Tracking System** designed using **Django (backend)** and **React (frontend)**. It allows **HR users** to manage employees and attendance effectively. The system includes secure **user authentication**, employee CRUD operations, attendance tracking, and dashboard analytics. The backend is powered by Django REST Framework and uses a relational **database** to store data. The frontend provides a responsive interface built with React and popular libraries. This project serves as a practical solution for digitizing and streamlining HR operations, making it scalable, maintainable, and user-friendly.

3. Specific Requirements

3.1 Functional Requirement:

1. Authentication and Authorization:

- Only registered HR users can log in.
- JWT tokens are used for secure authentication.
- Unauthenticated users cannot access protected routes or APIs.

2. User Management:

- HR can create user accounts for employees (username, email, password).
- Users are stored in a separate User model (accounts app).
- Users are linked with employee profiles via foreign keys.

3. Employee Management:

- Add new employees along with their full details:
 - First Name, Last Name
 - Gender
 - Phone Number
 - Position
 - Education
 - Employment Type (Full-Time, Part-Time, Intern, Contract)
 - Status (Active/NotActive)
- Upload and display profile pictures (ImageField support).
- Edit employee details using Update APIs.
- View employee list in tabular format.

4. Attendance Management:

- HR can record attendance for each employee:
 - Select employee
 - Choose a date

Effort Tracking system	Version: 1.0
Software Requirements Specification	Date: 15/10/2024

- Mark status: Present, Absent, or Leave
- View existing attendance records via API or frontend list.
- Each attendance entry is linked to an employee.

5. Dashboard:

- A summary dashboard displays:
 - Total number of employees
 - Total attendance entries
- Easy to access for quick HR decision-making.

6. REST API Endpoints:

- APIs follow RESTful principles using Django REST Framework:
 - /api/v1/employee/ - GET/POST (List & Create)
 - /api/v1/employee/<id>/ - GET/PATCH (Details & Update)
 - /api/v1/attendance/ - GET/POST (Manage Attendance)
 - /api/v1/dashboard/ - GET (Summary data)
- CORS support is configured for React integration.

7. Frontend (React) Features:

- Uses React Router for page navigation.
- Axios handles all HTTP requests.
- React Hook Form manages form inputs efficiently.
- SweetAlert2 provides user-friendly alerts and confirmations.

8. File & Media Handling:

- Employee images are uploaded to a media folder and served with proper URLs.
- Default images are used if no picture is provided.

9. Security:

- All API routes are secured using authentication.
- Only HR users can access views and make changes.
- Passwords are hashed and never stored in plain text.

10. Validation and Error Handling:

- Validations exist on both frontend and backend.
- Errors like duplicate emails, missing fields, or unauthorized actions return helpful messages.

Effort Tracking system	Version: 1.0
Software Requirements Specification	Date: 15/10/2024

3.2 Non Functional Requirements

1. Performance:

- The system responds quickly to user interactions and API calls.
- Optimized queries ensure efficient data retrieval for large employee records.

2.Security:

- Authentication uses JWT tokens to protect endpoints.
- Passwords are encrypted and user access is restricted based on roles.
- Sensitive data is not exposed in APIs.

3. Scalability:

- The system is built to scale with more users and employee records over time.
- Modular Django and React architecture supports easy extension.

4. Maintainability:

- Simple and intuitive React-based frontend for HR users.
- Validation feedback improves user practices in both backend and frontend.
- Clearly separated concerns: models, views, serializers, and components.
- Easy to update or enhance specific features without affecting others.
- interaction. Code is well-structured and follows best
- User-friendly forms and navigation.

5. Reliability:

- Handles exceptions gracefully and provides meaningful error messages.
- Uses database transactions to maintain data consistency.

6. Portability:

- Can be deployed on any platform supporting Django, PostgreSQL or SQLite, and Node.js/Vite for frontend.
- Backend and frontend are decoupled, allowing independent deployment.

Effort Tracking system	Version: 1.0
Software Requirements Specification	Date: 15/10/2024

3.3 Hardware Requirements:

1. Minimum Requirements:

- **Processor:** Dual Core 2.0 GHz
- **RAM:** 4 GB
- **Hard Disk:** 20 GB free space
- **Display:** 1024×768 resolution
- **Internet Connection:** Required for server communication and API integration

2. Recommended Requirements:

- **Processor:** Quad Core 2.5 GHz or higher
- **RAM:** 8 GB or more
- **Hard Disk:** 50 GB SSD for faster performance
- **Display:** 1920×1080 (Full HD)
- **Internet Connection:** High-speed broadband for smooth frontend/backend communication

3.4 Software Requirements:

1. Operating System:

- Windows 10/11, macOS, or any Linux distribution
- **Recommended:** Ubuntu 20.04+ or Windows 10+

2.Backend Environment:

- **Python** (version 3.8 or above)
- **Django** (version 4.x) – Web framework for backend
- **Django REST Framework** – For building RESTful APIs

3. Frontend Environment:

- **Node.js** (version 16 or above)
- **Vite** – Build tool for React frontend
- **React.js** – Frontend JavaScript library
- **NPM/Yarn** – Package managers to install dependencies

4. Database:

- **SQLite** (for development)
- **PostgreSQL/MySQL** (for production)

Effort Tracking system	Version: 1.0
Software Requirements Specification	Date: 15/10/2024

- **Version Control:**
 - **Git** – To manage source code versioning
- **IDE/Text Editor:**
 - **VS Code / PyCharm / Sublime Text** – For development
- **Browser:**
 - **Google Chrome / Mozilla Firefox** (latest version)

3.5 Agile Methodology

[1] Agile Principle:

- **Customer Satisfaction:**
Deliver valuable software early and continuously to satisfy customers.
- **Welcome Changing Requirements:**
Even late in development, changes are welcomed for the customer's competitive advantage.
- **Frequent Delivery:**
Deliver working software frequently—from a few weeks to a couple of months.
- **Collaboration:**
Developers and business people must work together daily.
- **Motivated Individuals:**
Build projects around motivated team members and give them the environment and trust they need.

[2] Agile Framework Overview:

The **Agile Framework** is a set of practices and tools that help teams work together efficiently to deliver high-quality software in small, manageable increments. It promotes continuous improvement, flexibility, and collaboration.

(a) Key Components of Agile Framework:

- 1. Scrum** – A popular Agile method with fixed-length sprints (usually 2–4 weeks), roles like Scrum Master, Product Owner, and daily stand-ups.
- 2. Kanban** – Visual workflow management system focusing on continuous delivery without fixed sprints.
- 3. Extreme Programming (XP)** – Emphasizes technical practices like test-driven development (TDD), pair programming, and frequent releases.
- 4. Lean** – Focuses on eliminating waste and maximizing value to the customer.

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

5. Crystal – Prioritizes team communication and project size/type adaptability.

(b) Common Agile Practices:

- Sprints/Iterations
- Daily Stand-ups
- Sprint Planning & Retrospective
- Backlog Grooming
- Continuous Integration & Delivery

3.6 Business Process Model

The **Business Process Model** outlines the key activities and flow of information in the tracking system, helping visualize how different components interact. Here's a simplified view based on the uploaded Django-React HR System:

[1] Key Business Processes:

1. HR Login & Authentication:

- HR logs in using secure credentials.
- System verifies access rights and redirects to dashboard.

2. Dashboard View:

- HR can see total employees and attendance records.
- Overview for quick decision-making.

3. Employee Management:

- Add new employee (personal + job details).
- Edit or update employee info.
- Upload profile image and maintain status (active/inactive).

4. Attendance Management:

- Select employee and date.
- Mark status (Present, Absent, Leave).
- View attendance history.

5. User & Employee Creation:

- HR creates login credentials for a new employee.
- Simultaneously adds employee details in the system.

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

3.7 Supplementary Requirements

Supplementary requirements are non-functional aspects that support the system's functionality, ensuring it runs efficiently, securely, and remains user-friendly. These requirements improve the overall quality and user experience of the application.

[1] Key Supplementary Requirements:

1. Performance Requirements:

- The system must respond to any user action within 2 seconds.
- It should handle multiple HR users simultaneously without delay.

2. Security Requirements:

- Only authenticated HR users can access the system.
- Passwords must be securely stored and encrypted.
- Sensitive data like employee details must be protected from unauthorized access.

3. Usability Requirements:

- The UI must be intuitive and easy to navigate for non-technical users.
- Form validations should provide clear error messages and guide the user.

4. Portability Requirements:

- The system should work on all major browsers (Chrome, Firefox, Edge).
- Should be accessible on both desktop and tablet devices.

5. Maintainability Requirements:

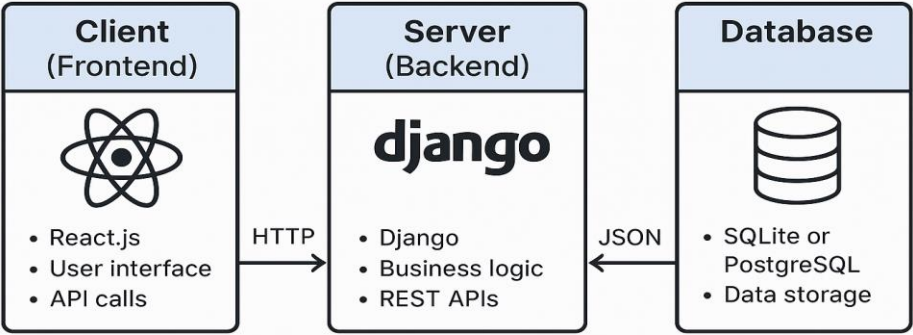
- The system should be modular to allow easy updates or feature additions.
- Code must follow clean and commented practices for future debugging.

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

4. System Architecture

4.1. Client-Server Architecture:

- **Client (Frontend):**
 - **Technology:** React.js (JavaScript framework)
 - **Role:**
 - Acts as the **user interface** layer.
 - Sends HTTP requests (using axios) to the backend APIs.
 - Renders data and provides user interactions (forms, dashboards, attendance input).
 - Handles routing using react-router-dom.
- **Server (Backend):**
 - **Technology:** Django (Python framework) with Django REST Framework
 - **Role:**
 - Acts as the **API provider** and logic processor.
 - Handles business logic, authentication, authorization (for HR roles).
 - Communicates with the database (SQLite or PostgreSQL) for CRUD operations on employees, users, and attendance.
 - Uses REST APIs to send data back to the frontend.
- **Database:**
 - **Technology:** SQLite (default, can be switched to PostgreSQL)
 - **Role:**
 - Stores persistent data (employee records, user credentials, attendance logs).



Django-React HR System

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

4.2. Communication Interfaces

1. Frontend to Backend (Client to Server)

- **Technology Used:** HTTP/HTTPS via RESTful APIs.
- **Format:** JSON.
- **Tools:** Axios library in React handles API requests.
- **Examples:**
 - Login request: POST /api/token/
 - Fetch employees: GET /employee/
 - Submit attendance: POST /attendance/

2. Backend to Database

- **Technology Used:** Django ORM.
- **Format:** SQL (abstracted by ORM).
- **Database Operations:** CRUD operations for users, employees, attendance.

3. Authentication Flow

- **Type:** Token-based (JWT).
- **Purpose:** Secure API access.
- **Process:** Token generated at login, stored in browser cookies or local storage, sent with every API call.

4. Cross-Origin Communication

- **CORS Handling:** Configured in Django settings using django-cors-headers to allow React frontend to interact with the backend on a different port or domain.

5. Overall Description

[5.1] Product Feature

1. User Authentication

- Secure login system using JWT.
- Only HR users can access the dashboard and other admin functionalities.

2. Employee Management

- Add, edit, update, and view employee details.
- Upload employee images and manage employee roles like full-time/part-time.

3. Attendance Management

- HR can record attendance for employees.
- View attendance logs by date and employee.

4. Dashboard Insights

- Dashboard displays key statistics like total employees and total attendance entries.

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

5. Role-Based Access

- Access restricted to authenticated HR users only, ensuring data privacy and security.

6. Interactive UI

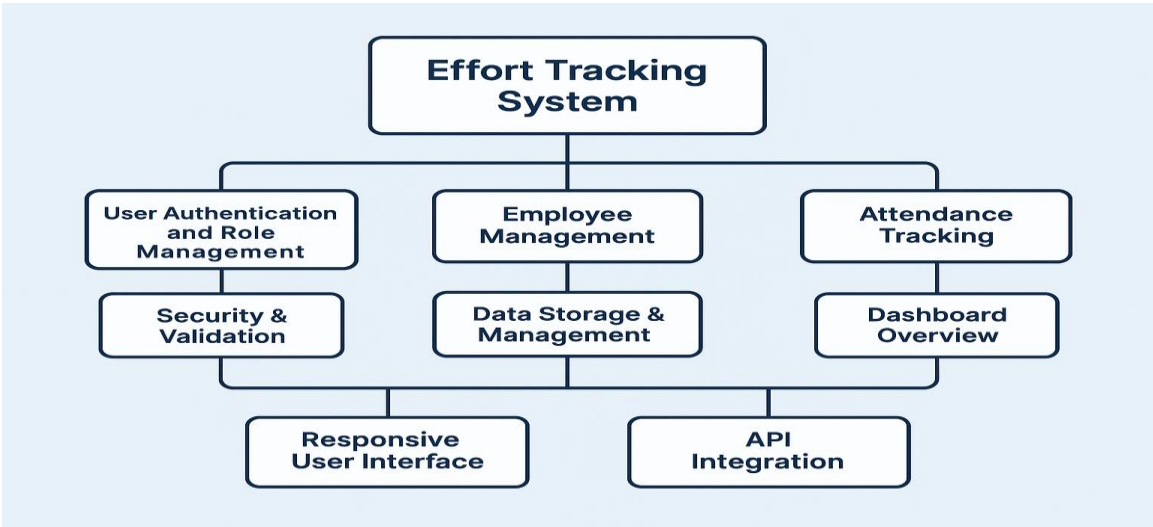
- Clean and user-friendly React frontend.
- Uses SweetAlert for confirmations and alerts, and charts for data visualization.

7. API-Driven Communication

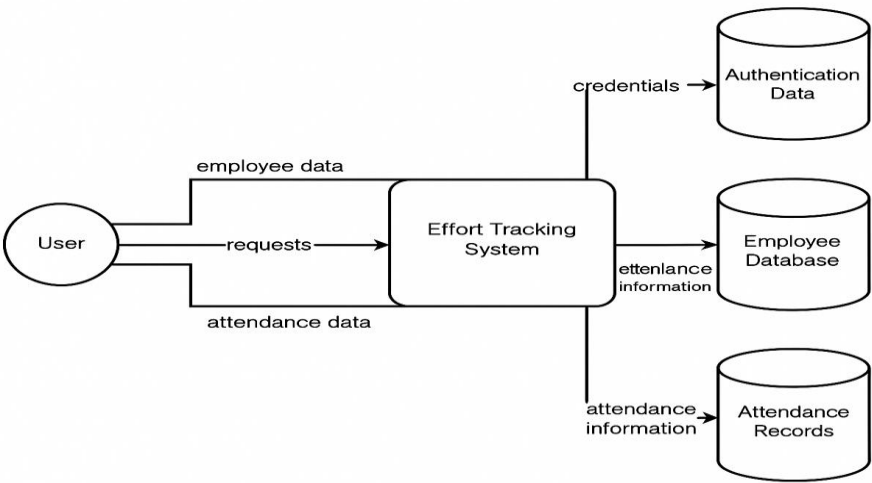
- Frontend and backend communicate through RESTful APIs for scalability and flexibility.

8. Responsive Design

- Built with modern React libraries to ensure smooth performance on different devices.



[5.2] Data Flow diagram

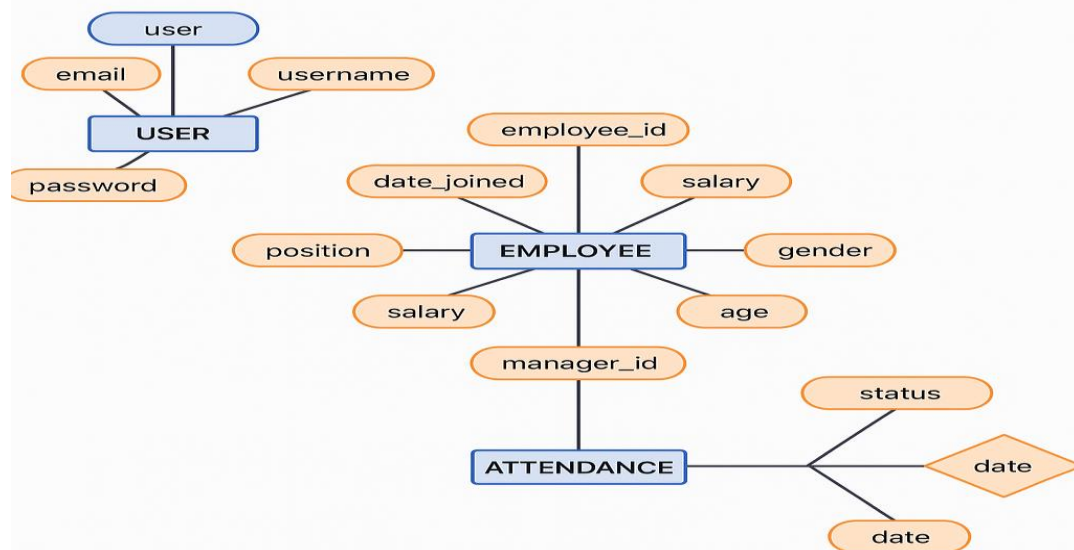


Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

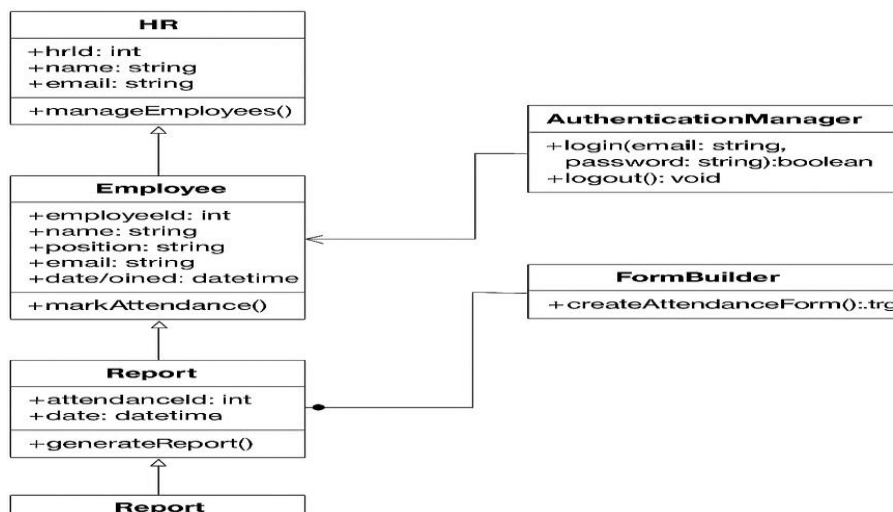
[5.3] E-R Diagram

Entities:

- **User:** Stores login credentials and personal info for HR.
- **Employee:** Contains employee profile linked to a user.
- **Attendance:** Linked to an employee and stores presence info.

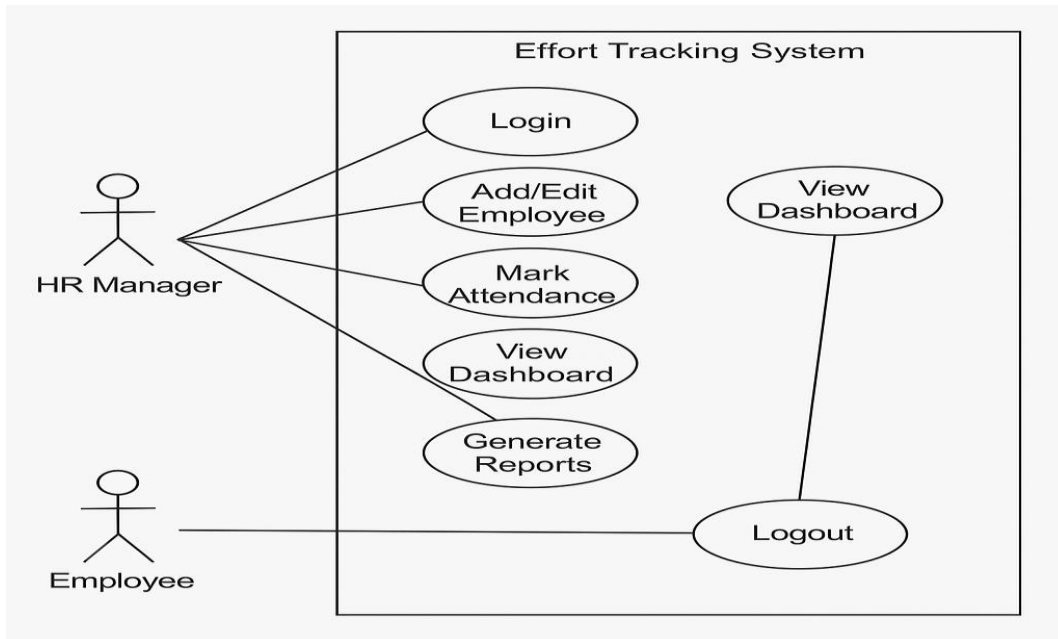


[5.4] Class Diagram



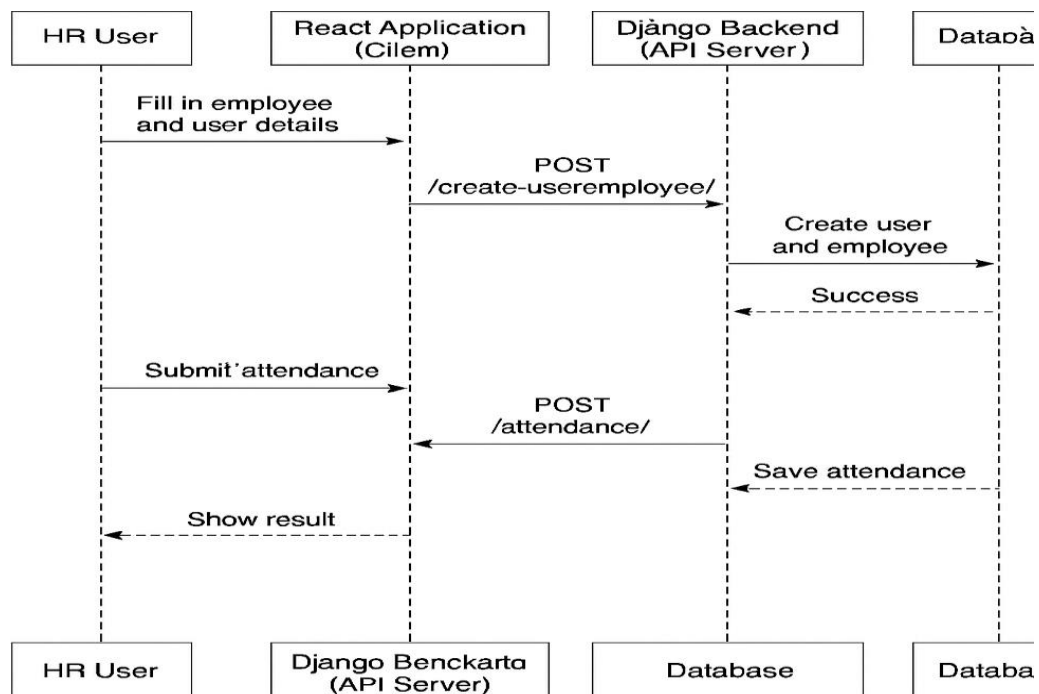
Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

[5.5] Use-Case Model survey



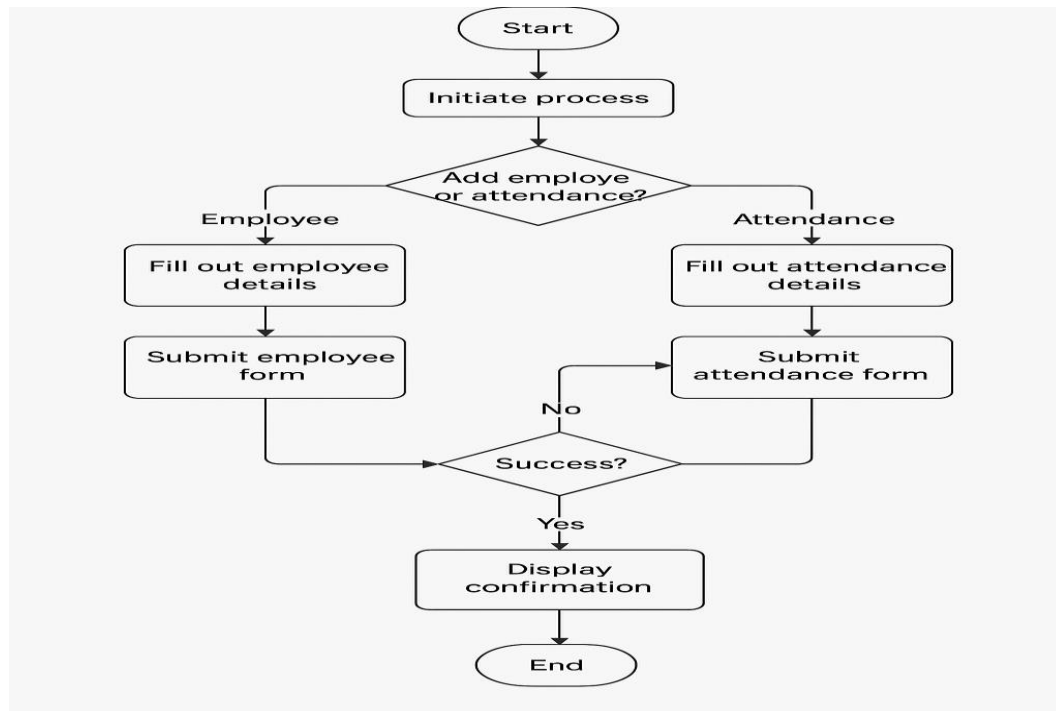
[5.6] Behaviors Diagram

1. Sequence Diagram

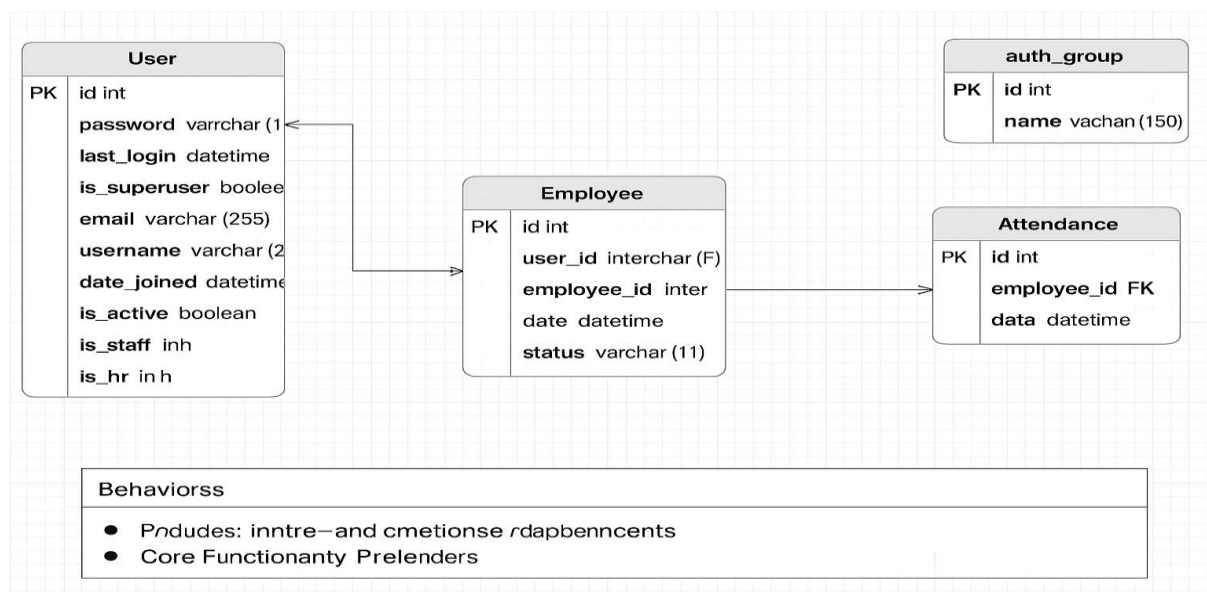


Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

2. Activity Diagram



[5.7] Database Diagram



Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

5.8 Assumptions and Dependencies

[1] Assumptions:

- HR users have valid credentials to log in and manage employee and attendance data.
- The server is configured correctly with Django, and static/media files are properly served.
- React frontend is deployed and correctly connected to the Django backend via API endpoints.
- Employees will not access the HR dashboard – access is restricted to HR personnel.
- The environment (development or production) supports image uploads and secure API communication.

[2] Dependencies:

Component	Dependency	Purpose
Backend	Django REST Framework	API creation & serialization
	Django	Core backend logic
	Pillow	Image processing for profile uploads
Frontend	React.js	User interface (HR dashboard)
	Axios	API requests to Django
	React Router DOM	Page navigation
	Zustand	State management
Database	SQLite (default) or PostgreSQL	Data storage
Hosting	Netlify (Frontend), Django Server (Backend)	Deployment platforms

6. Supporting Information

[A] Project Overview:

The Django-React HR System is a full-stack web application that helps HR personnel manage employee records and attendance efficiently. The backend is built with Django REST Framework, while the frontend uses React.js.

[1] User Roles:

- **HR:** Can log in, add/edit employees, and track attendance.
- **Employees:** Currently not designed to access the system directly.
- **Sample Credentials for Testing:**
- **Email:** testthr@gmail.com
- **Password:** Test12345
- **Development & Deployment Info:**

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

- Frontend is deployed using **Netlify**.
- Backend can be run locally or deployed via cloud hosting (e.g., Heroku, Render).
- Media/image uploads require correct configuration of `MEDIA_ROOT` and `MEDIA_URL` in Django settings.

[2] Useful Tools & Libraries:

- **Backend:** Django, DRF, Pillow
- **Frontend:** React, Axios, Zustand, Chart.js, SweetAlert2
- **Database:** SQLite (dev) / PostgreSQL (prod recommended)

[3] API Communication:

RESTful APIs enable smooth data exchange between frontend and backend. All secure operations require user authentication.

[4] Testing & Validation:

The Django test cases ensure the API endpoints work correctly for employee creation, retrieval, updating, and attendance tracking.

[B] Significance of an Effort Tracking System

An **Effort Tracking System** plays a crucial role in modern project and workforce management. Here's why it's important:

[1] Improves Productivity:

It helps track how much time employees spend on various tasks, enabling better time management and productivity analysis.

[2] Accurate Resource Allocation:

Managers can identify underutilized or overburdened resources and reassign work accordingly for efficiency.

[3] Performance Evaluation:

Individual and team performances can be measured based on the efforts logged, aiding in fair appraisals and rewards.

[4] Transparency & Accountability:

Employees become more accountable for their work when time and effort are regularly monitored and recorded.

[5] Informed Decision-Making:

With real-time effort data, project managers can make data-driven decisions regarding deadlines, budgets, and staffing.

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

2. Design Considerations

[1] User Roles & Permissions

- Define clear roles such as HR, employees, and admins.
- Ensure access control for sensitive data (e.g., only HR can modify attendance).

[2] Simple & Intuitive UI

- The interface should be user-friendly for non-technical users.
- Forms for logging hours or attendance should be quick and easy to fill.

[3] Modular Architecture

- Separate components for authentication, employee management, and attendance tracking.
- Makes the system easier to maintain and scale.

[4] Secure Authentication

- Implement token-based authentication using JWT.
- Passwords must be encrypted and securely stored.

[5] Real-Time Data Handling

- Use APIs that support real-time updates for attendance and effort logging.
- Minimize lag between actions and reflected data.

[6] Database Design

- Ensure normalized schema to avoid data redundancy.
- Use relationships (foreign keys) wisely for linking users, employees, and attendance records.

[7] Mobile Responsiveness

- The frontend should work seamlessly on desktops, tablets, and mobile devices.
- Consider using responsive design frameworks.

[8] Scalability & Extensibility

- The system should be able to handle an increasing number of users and data.
- Should support future modules like payroll, performance analysis, etc.

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

[9] Error Handling & Feedback

- Display meaningful error messages.
- Provide confirmation prompts and success feedback to users after actions.

[10] Backup & Data Recovery

- Regular data backups should be scheduled.
- Plan for recovery in case of accidental deletion or data loss

3. Real-World Applicability

The **Effort Tracking System** developed using Django (backend) and React (frontend) is highly applicable in various real-world scenarios where monitoring and optimizing workforce productivity is critical. Here's how it fits into real environments:

[1] Corporate HR Departments

- Helps HR teams track employee work hours, attendance, and productivity in real time.
- Streamlines employee management, reducing paperwork and manual entry.

[2] IT Companies & Project Teams

- Tracks how much effort each team member puts into assigned tasks or modules.
- Enhances project planning and billing accuracy, especially in client-based services.

[3] Startups & SMEs

- Small businesses can use it to monitor attendance and manage workforce even with limited resources.
- Affordable and scalable solution for growing teams.

[4] Remote Work & Freelancing Platforms

- Allows logging of remote employees' effort, attendance, and output digitally.
- Supports transparency and trust in distributed teams.

[5] Educational Institutions (Admin Staff)

- Used by educational administration to track working hours and leaves of faculty and staff.
- Ensures operational efficiency in managing non-teaching employees.

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

4. Example Scenarios

[1] Tech Company Project Team

Scenario:

A software development company needs to monitor the time developers spend on various client projects.

Use:

Each developer logs hours against specific tasks. The HR team uses reports for payroll and to assess performance.

[2] Remote Marketing Agency

Scenario:

A remote agency has freelancers and part-time marketers working across different time zones.

Use:

Employees mark attendance and log their daily work, helping managers track productivity and meet deadlines.

[3] Startup with Growing Staff

Scenario:

A startup adds new employees rapidly and wants to keep track of each employee's working hours and active status.

Use:

The HR can create users and employees, monitor attendance, and generate data for performance reviews.

[4] Educational Institution Admin Team

Scenario:

An institution needs to track non-teaching staff attendance for budgeting and work allocation.

Use:

The effort tracker helps the admin know who is present, log leaves, and plan shifts accordingly.

[5] Consulting Firm Handling Multiple Clients

Scenario:

A firm assigns consultants to multiple client sites and tracks hours worked at each site.

Use:

Each consultant enters hours per client, and the firm uses this data for client billing and reporting.

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

6. Technologies Supporting the System

[1] Backend – Django (Python)

- **Purpose:** Handles all business logic, data processing, and API endpoints.
- **Framework:** Django REST Framework (DRF) for building RESTful APIs.
- **Authentication:** Token-based or session-based for secure HR login.

[2] Frontend – ReactJS

- **Purpose:** Provides a user-friendly interface for HR to manage employees and attendance.
- **Features:** Dynamic UI, routing with react-router-dom, and component-based architecture.

[3] Database – SQLite (default) / PostgreSQL

- **Purpose:** Stores all persistent data like user info, employee records, attendance logs, etc.
- **Can be upgraded to PostgreSQL** for production environments.

[4] Axios

- **Purpose:** Handles HTTP requests between the React frontend and Django backend.
- **Usage:** CRUD operations like creating employees, fetching attendance, etc.

[5] State Management – Zustand

- **Purpose:** Manages global state in React efficiently without the overhead of Redux.

[6] UI Libraries & Tools

- **React Icons, SweetAlert2:** Enhance visuals and provide better user feedback.
- **Chart.js & react-chartjs-2:** For visualizing attendance or employee stats.
- **CKEditor:** (if enabled) for rich text editing in any content fields.

7. Conclusion and future scope

The Django-React Tracking System effectively combines a robust backend with an intuitive frontend to streamline HR tasks such as employee management and attendance tracking. With role-based access for HR users, it ensures data security and operational control. The system is responsive, scalable for small to medium organizations, and simplifies HR workflows. It demonstrates a practical implementation of modern web technologies for real-world HR challenges.

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

[1] Future scope:

- **Leave & Payroll Management:** Extend the system to handle leave requests, salary calculations, and monthly pay slips.
- **Multi-role Access:** Introduce access for Managers, Employees, and Admins with customized dashboards.
- **Performance Tracking:** Integrate employee performance metrics and feedback systems.
- **Reporting & Analytics:** Add graphical dashboards for attendance trends, hiring rates, etc.
- **Mobile App Integration:** Develop a mobile version using React Native for on-the-go HR operations.
- **Cloud Deployment & CI/CD:** Improve scalability with Docker, AWS/GCP deployment, and CI/CD pipelines.
- **Biometric or QR Attendance:** Integrate with biometric or QR systems for smart attendance marking.
- **Notification System:** Add email/SMS notifications for events like approvals, updates, or birthdays.

8. Concerns/Queries/Doubts if any

[1] Authentication and Security:

- Is user password securely hashed?
- Are API endpoints protected properly against unauthorized access?

[2] Role Management:

- Can the system easily extend to more roles beyond HR (e.g., Manager, Admin)?
- What happens if a non-HR user tries to access employee routes?

[3] Attendance Handling:

- Can multiple attendance entries be made for the same employee on the same date?
- Is there a way to mark half-days or work-from-home?

[4] Error Handling:

- Are API and form errors handled and shown properly to the user?
- Is there logging for backend exceptions?

[5] Scalability:

- Can the system handle 1000+ employees?
- What database would be recommended for production use?

Effort Tracking System	Version: 1.0
Software Requirements Specification	Date: 15/10/24

[6] Deployment:

- Is the backend deployed with proper CORS and HTTPS configuration?
- Are environment variables and secrets properly managed?

[7] UI/UX:

- Is the frontend fully responsive on mobile?
- Are users notified clearly after successful or failed actions?

[8] Data Validation:

- Are phone numbers, email addresses, and dates properly validated?
- What happens if invalid data is submitted?

[9] Performance:

- Are there performance optimizations for list views (pagination, filtering)?
- Does the dashboard aggregate data efficiently?

[10] Extensibility:

- How easy is it to add new modules like leave management, payroll, etc.?