Level 1 DFD

A Level 1 DFD decomposes the single process from the Level 0 diagram into multiple sub-processes. It reveals major functional areas within the system and shows how data moves between them, external entities, and data stores. This level provides more detail, offering insight into how the system handles different operations and the logic behind each process.

Level 2 DFD

The Level 2 DFD further breaks down individual processes from the Level 1 diagram into more specific sub-processes. It gives a detailed look into each functional component, showing precise data inputs, processing steps, and outputs. This level is often used by developers and system analysts to understand intricate workflows and design efficient system components.

Figure 2.4: DFD Level 2

2.1.3.2 Entity-Relationship Diagram (ERD)

The Entity-Relationship Diagram (ERD) illustrates the structure of the database by defining the entities, their attributes, and the relationships between them. In this system, the key entities are Users, Employees, and Leave Requests.

Each user has a role (either HR or Employee), and is linked to an employee record. Employees can submit multiple leave requests, which are tracked and managed by HR. The ERD helps ensure that the database is well-organized, supports role-based access, and maintains data integrity.

By mapping out these relationships, the ERD provides a clear foundation for designing efficient queries and maintaining consistency across the system.

System Requirements Specification (SRS)

The System Requirements Specification (SRS) lists the hardware and software needed for the system to work.

Design

Planning the system’s structure and user interface is essential to create a clear and organized solution. This involves deciding how components connect and how users will interact with the system to meet all requirements effectively.

System Architecture

The Employee Analytics and Churn Prediction System is structured using a three-tier architecture, which separates the system into three logical layers: the Presentation Tier, Application Tier, and Data Tier. This design enhances modularity, scalability, and maintainability.

1. Presentation Tier (Frontend): The presentation layer is the user interface of the system, accessible via a web browser. It is developed using HTML, Tailwind CSS, JavaScript, and Chart.js. It allows HR and Employee users to interact with the system, including logging in, managing records, requesting leave, and viewing analytics dashboards. This tier handles all user input and displays visual feedback from backend responses.

2. Application Tier (Backend/Logic Layer): The application logic is handled by the

Flask framework. It processes user requests, performs authentication, manages sessions, handles CRUD operations for employee and leave data, and serves as the communication bridge between the frontend and backend.

Database Design

A database schema is the structural design that shows how data is organized in a database, including tables, fields, relationships, and rules. It works as a blueprint that defines how data is stored, linked, and accessed inside the system.

In the Employee Analytics and Churn Prediction System, the database schema includes four main tables: employees, users, leave requests, and applicants. The employees table stores detailed information about each staff member, such as department, salary, experience, and other fields used in analytics and churn prediction. The users table manages system login and access, storing credentials and assigning roles (HR or employee), and is linked to each employee through a foreign key. The leave requests table records leave applications submitted by employees, including type, duration, and approval status. The applicants table saves the job application details submitted by users applying for positions, and once an applicant is hired, their data is moved to the employees table and a user account is created.
These tables work together to ensure smooth system operation and data consistency, supporting key features like authentication, leave handling, analytics, and the hiring process.

User Interface Design

A user interface (UI) diagram shows how the screens or pages are arranged in a system and how users move between them. It helps visualize the navigation flow and where key elements like forms, buttons, and dashboards are placed.
In the Employee Analytics and Churn Prediction System, the UI diagram includes three main roles, HR, Employee, and Applicant. For HR, the interface has analytics dashboards, employee management pages (add, edit, view, delete), leave approval sections, churn prediction results, and views to manage job applications. Employees have access to a personal profile page, a leave request form, leave history, and feedback options. Applicants can fill out and submit job application forms through a dedicated view.
This layout helps keep the system simple and user-friendly, making it easier for all users to access features relevant to their role.

Algorithm Design

Algorithm design is the process of creating a clear and logical set of steps to solve problems or complete tasks in a system. It helps define how the system should process inputs to reach the correct output.

2.3.1.3 Machine Learning

The prediction model was built with scikit-learn, a Python tool for machine learning. The Random Forest Classifier model was used to predict which employees might leave the company.

2.3.1.4 Database

MySQL was used to store important data like employee details, user accounts, and leave requests. This keeps all information safe and easy to access.

2.3.1.5 Development Tools

The development of the project utilized several essential tools to streamline the workflow. Visual Studio Code (VS Code) served as the primary integrated development environment (IDE) for writing code, managing project files, and debugging, with helpful extensions for Python and Flask. Postman was used to test API endpoints, ensuring proper communication between the frontend and backend components. Git functioned as the version control tool, allowing for efficient tracking of changes and collaboration during development. Additionally, the Command Line Interface (CLI) was used frequently to run Flask servers, manage virtual environments, and execute Python scripts.

Module Description

Modules in code are separate components that group related functions, routes, or logic into organized sections. They help in breaking down a large application into manageable parts, making the code easier to understand, test, and maintain. Using modules improves reusability and allows different team members to work on separate features without conflicts. This structure enhances scalability and supports a clean development workflow.

Churn Prediction Module

The Churn Prediction Module is designed to help HR identify employees who are likely to leave the company, allowing for timely interventions. This is achieved using machine learning models trained on the IBM HR Analytics dataset. The five

different classification algorithms that were evaluated are: Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes, and Random

Forest.

The models were assessed using four primary performance metrics:

1. Accuracy: The percentage of total predictions that were correct.
2. Precision: The proportion of predicted positives that were actually positive.
3. Recall: The proportion of actual positives that were correctly identified.
4. F1-Score: The harmonic mean of precision and recall; useful when classes are imbalanced.

Note: Class 0 indicates no attrition (employee stays), while class 1 indicates attrition (employee likely to leave).