**Admin Workflow Documentation**

* **Parrot Analyzer**

**Date: 27-02-2025**

**Table of Contents**

1. Introduction & Overview
2. Purpose of This Documentation
3. Role-Based Architecture  
    3.1 Super Admin  
    3.2 Management Personnel  
    3.3 Group Admin
4. Key Functional Modules  
    4.1 Live Tracking & Attendance  
    4.2 Leave Management  
    4.3 Expense Management  
    4.4 Task & Schedule Management  
    4.5 Reports & Analytics  
    4.6 System Administration
5. High-Level Data Flow  
    5.1 Front-End (Mobile App)  
    5.2 Backend (Node.js + Express)  
    5.3 Database (PostgreSQL)  
    5.4 Communication
6. Why Role-Based Workflows?
7. Role-Specific Text-Based Workflows in Parrot Analyzer  
    7.1 Super Admin Workflow  
     7.1.1 Overview  
     7.1.2 Workflow Steps  
      7.1.2.1 Authentication & Login  
      7.1.2.2 Company Management  
      7.1.2.3 User Management  
      7.1.2.4 Global Settings & Security  
      7.1.2.5 Reporting & Audit  
    7.2 Management Personnel Workflow  
     7.2.1 Overview  
     7.2.2 Workflow Steps  
      7.2.2.1 Authentication & Login  
      7.2.2.2 Leave Policy Configuration  
      7.2.2.3 Aggregated Analytics & Reporting  
      7.2.2.4 Escalation Handling  
      7.2.2.5 Group Admin Management  
    7.3 Group Admin Workflow  
     7.3.1 Overview  
     7.3.2 Workflow Steps  
      7.3.2.1 Authentication & Login  
      7.3.2.2 Employee Management  
      7.3.2.3 Leave Request Processing  
      7.3.2.4 Expense & Task Management  
      7.3.2.5 Attendance & Live Tracking  
      7.3.2.6 Reporting & Analytics  
      7.3.2.7 Personal Dashboard & Profile Management
8. Workflow Diagram
   1. All Admin Workflow
   2. Super Admin Workflow
   3. Management Personal Workflow
   4. Group Admin Workflow
9. Database Schema

**1. Introduction & Overview**

Parrot Analyzer is a comprehensive mobile application designed to streamline various aspects of workforce management. The application focuses on:

* **Real-time employee tracking**
* **Attendance logging**
* **Leave and expense management**
* **Live chatbot support**

It leverages:

* **React Native with Expo** for the frontend,
* **Node.js + Express** for the backend, and
* **PostgreSQL** for data storage.

Real-time functionalities are powered by **Socket.io**, ensuring live updates and efficient communication across the system.

**2. Purpose of This Documentation**

This document aims to:

* Provide a clear understanding of the administrative roles within Parrot Analyzer.
* Detail the workflows and responsibilities for **Super Admin**, **Management Personnel**, and **Group Admin**.
* Serve as a comprehensive reference for developers, stakeholders, and end-users regarding system architecture and operational workflows.

**3. Role-Based Architecture**

Parrot Analyzer uses a role-based access control (RBAC) system, ensuring that users only have access to functionalities pertinent to their role.

**3.1 Super Admin**

* Oversees system-wide settings
* Manages company information
* Sets user limits and maintains global security

**3.2 Management Personnel**

* Configures leave policies
* Reviews escalated requests
* Accesses high-level analytics and reporting

**3.3 Group Admin**

* Manages employees within specific groups
* Handles daily approvals for leave and expenses
* Oversees task assignments and attendance

**4. Key Functional Modules**

**4.1 Live Tracking & Attendance**

* **GPS-based Tracking:** Captures real-time employee locations
* **Geofencing:** Filters indoor movements to accurately compute travel distance and time
* **Shift Logging:** Automates attendance by recording start/end times and location data

**4.2 Leave Management**

* **Policy Configuration:** Management Personnel define entitlements and rules
* **Leave Requests:** Processed by Group Admins (approval, rejection, escalation)
* **Leave Insights:** Employees can view policies, balances, and request statuses

**4.3 Expense Management**

* **Submissions:** Employees submit travel details, supporting documents, and expense breakdowns
* **Approval Workflow:** Group Admins review or reject expense submissions
* **Financial Summaries:** Maintains records for audit and reimbursement purposes

**4.4 Task & Schedule Management**

* **Task Creation:** Group Admins create and assign tasks
* **Status Tracking:** Monitors progress with progress bars and deadlines
* **Scheduling:** Helps employees plan their work calendars

**4.5 Reports & Analytics**

* **Aggregated Reports:** Summarizes data on attendance, leave usage, expenses, and performance
* **Visual Analytics:** Uses charts and graphs to provide insights
* **Export Options:** Allows exporting reports in PDF/Excel formats for compliance and review

**4.6 System Administration**

* **Company Management:** Super Admins add or manage companies and set user limits
* **User Management:** Involves user creation, approval, and role assignment
* **Security & Audit Logs:** Tracks changes to maintain system integrity

**5. High-Level Data Flow**

**5.1 Front-End (Mobile App)**

* Built using **React Native (Expo)** and styled with **NativeWind**
* Connects to the backend via RESTful APIs and **Socket.io**
* Custom dashboards for Employees, Group Admins, and Management Personnel

**5.2 Backend (Node.js + Express)**

* Processes requests from the mobile app
* Implements authentication, validation, and business logic
* Uses middleware to enforce role-based access
* Supports real-time data transfers with **Socket.io**

**5.3 Database (PostgreSQL)**

* Stores user data, company information, leave requests, expense records, task assignments, and schedules
* Maintains referential integrity using foreign keys to link employees, groups, and companies

**5.4 Communication**

* **API Endpoints:** Manage CRUD operations and complex queries (see API-DOCUMENTATION.md)
* **Real-time Communication:** Uses **Socket.io** for location updates, notifications, and chat functionalities

**6. Why Role-Based Workflows?**

Implementing role-based workflows provides:

* **Security:** Limits access to only the data and features needed by each role
* **Clarity:** Clearly defines each role’s responsibilities, reducing confusion
* **Scalability:** Facilitates the addition of new roles (e.g., Group Admins or Management Personnel) without disrupting existing workflows
* **Maintainability:** Allows developers to update role-specific features independently, ensuring efficient system evolution

**7. Role-Specific Text-Based Workflows in Parrot Analyzer**

This section details the end-to-end workflows for each administrative role within Parrot Analyzer. Each workflow outlines the steps that a user in that role takes from interacting with the front-end interface, through API calls and back-end processing, to the updating of the database. The workflows ensure that each role’s responsibilities are handled securely and efficiently.

**7.1 Super Admin Workflow**

**7.1.1 Overview**

Super Admins have system-wide responsibilities. They manage company records, oversee all user accounts, configure global settings, and generate high-level reports. Their workflow involves creating and updating companies, managing user roles, and enforcing system configurations.

**7.1.2 Workflow Steps**

**7.1.2.1 Authentication & Login**

* **Front-End:** Super Admin logs in via the login interface, entering credentials.
* **Back-End:** /auth/login endpoint verifies credentials and returns a JWT token.
* **Database:** The users table is queried to authenticate the Super Admin.

**7.1.2.2 Company Management**

* **Action:** Access the Company Management dashboard.
* **Front-End:** Super Admin views a list of companies, with options to add, update, or revoke access.
* **Back-End:** Calls endpoints such as GET /api/companies to fetch company data and POST /api/companies/create to add a new company.
* **Database:** Inserts/updates occur in the companies table.

**7.1.2.3 User Management**

* **Action:** Manage users across companies.
* **Front-End:** Super Admin navigates to the user management panel to view, approve, or toggle user statuses.
* **Back-End:** Endpoints like GET /api/super-admin/users and PATCH /api/super-admin/users/:id/toggle-status are called.
* **Database:** The users table is updated accordingly.

**7.1.2.4 Global Settings & Security**

* **Action:** Update system settings (e.g., geofencing parameters, shift timings).
* **Front-End:** A settings panel allows Super Admin to adjust configurations.
* **Back-End:** Relevant endpoints (e.g., /api/admin/settings) process these updates.
* **Database:** Global configuration tables and audit\_logs are updated to reflect changes.

**7.1.2.5 Reporting & Audit**

* **Action:** Generate reports on overall system performance.
* **Front-End:** Super Admin selects filters (date range, company, etc.) to generate reports.
* **Back-End:** Calls endpoints such as POST /api/reports/generate.
* **Database:** Data is aggregated from various tables (e.g., attendance\_records, leave\_requests, expenses).

**7.2 Management Personnel Workflow**

**7.2.1 Overview**

Management Personnel are responsible for strategic oversight, including configuring leave policies, handling escalated requests, and monitoring aggregated analytics. They bridge operational tasks with high-level analytics.

**7.2.2 Workflow Steps**

**7.2.2.1 Authentication & Login**

* **Front-End:** Management Personnel log in via the same authentication interface.
* **Back-End:** Uses /auth/login to verify and issue JWT tokens.
* **Database:** The users table confirms the role and credentials.

**7.2.2.2 Leave Policy Configuration**

* **Action:** Access the Leave Policy Configuration section.
* **Front-End:** A dedicated interface allows the management team to create or update leave types (EL, SL, ML, CL, etc.) with specific rules.
* **Back-End:** Calls endpoints such as POST /api/leave-policies (or equivalent) to configure policies.
* **Database:** Updates occur in the leave\_policies and leave\_types tables.

**7.2.2.3 Aggregated Analytics & Reporting**

* **Action:** Generate high-level reports and review performance metrics.
* **Front-End:** The dashboard displays aggregated data using charts and summary cards.
* **Back-End:** Endpoints like POST /api/reports/generate are used to fetch report data.
* **Database:** Data is aggregated from multiple tables, such as leave, requests, attendance, records, and expenses.

**7.2.2.4 Escalation Handling**

* **Action:** Review and process escalated leave or expense requests forwarded by Group Admins.
* **Front-End:** A section of the dashboard lists escalated cases for review.
* **Back-End:** Specific endpoints (e.g., GET /api/management/escalated-requests) retrieve these requests.
* **Database:** The leave\_requests table is queried for records marked as escalated.

**7.2.2.5 Group Admin Management**

* **Action:** View and manage Group Admin accounts.
* **Front-End:** The management panel lists Group Admins with options to create new accounts (individually or via CSV import).
* **Back-End:** Calls endpoints such as /api/management/group-admins to fetch and manage data.
* **Database:** Relevant records in the users table are filtered by role.

**7.3 Group Admin Workflow**

**7.3.1 Overview**

Group Admins handle day-to-day operations within their specific group. They manage employees, process leave and expense requests, assign tasks, track attendance, and monitor live tracking data. Their workflow is focused on operational efficiency and real-time decision-making.

**7.3.2 Workflow Steps**

**7.3.2.1 Authentication & Login**

* **Front-End:** Group Admin logs in using the standard login interface.
* **Back-End:** Authentication is handled via /auth/login, issuing a JWT token.
* **Database:** The users table verifies that the user’s role is "group\_admin."

**7.3.2.2 Employee Management**

* **Action:** Add, view, search, and delete employees.
* **Front-End:** The Employee Management section allows for individual employee creation or bulk import using CSV.
* **Back-End:** Endpoints such as POST /api/group-admin/employees and GET /api/group-admin/employees are used.
* **Database:** Data is managed in the users and employee\_profiles tables.

**7.3.2.3 Leave Request Processing**

* **Action:** Review and process leave requests submitted by employees.
* **Front-End:** A dedicated dashboard displays a list of leave requests with filtering options (by employee name, leave type, date).
* **Back-End:** Group Admin calls GET /api/group-admin-leave/requests to retrieve requests and uses PATCH /api/group-admin-leave/requests/:id/status to update the request status.
* **Database:** The leave\_requests table is updated with approval/rejection decisions.

**7.3.2.4 Expense & Task Management**

* **Action:** Review expense submissions, approve/reject expenses, and assign tasks to employees.
* **Front-End:** Separate interfaces allow viewing expense summaries and creating tasks using forms (capturing task title, description, priority, and due date).
* **Back-End:** Endpoints such as GET /api/expenses/group and POST /api/tasks/create facilitate these operations.
* **Database:** Operations update the expenses and tasks tables.

**7.3.2.5 Attendance & Live Tracking**

* **Action:** Monitor real-time employee locations and daily attendance records.
* **Front-End:** An interactive map shows live tracking data, and a calendar view displays attendance details.
* **Back-End:** Endpoints like GET /api/attendance/group and GET /api/live-tracking/group (if applicable) are used.
* **Database:** Data is stored and retrieved from the attendance\_records and live\_tracking\_data tables.

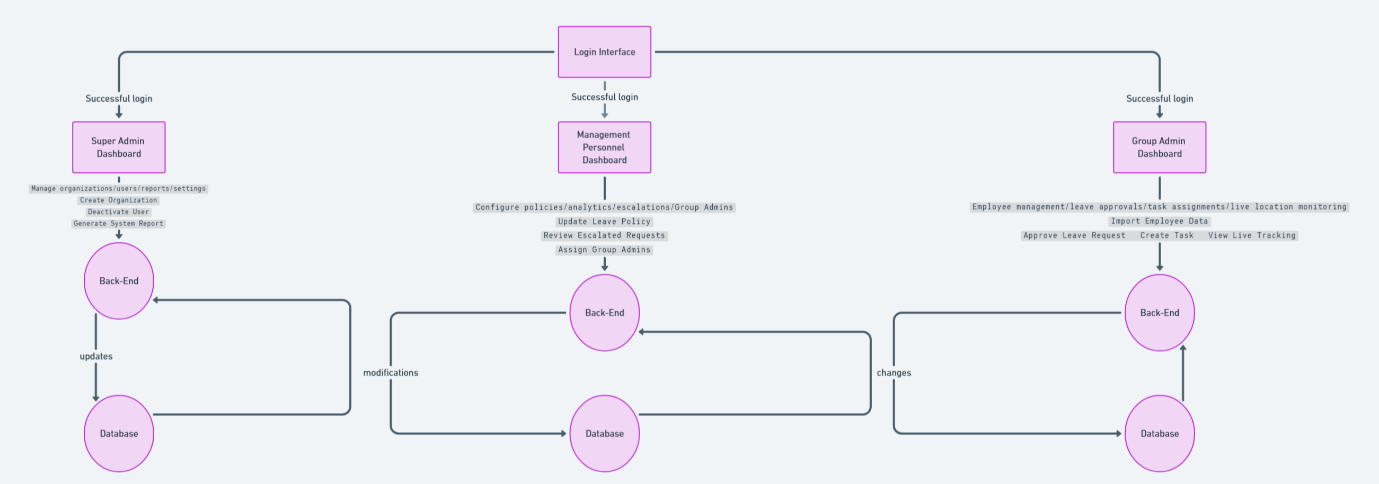
**7.3.2.6 Reporting & Analytics**

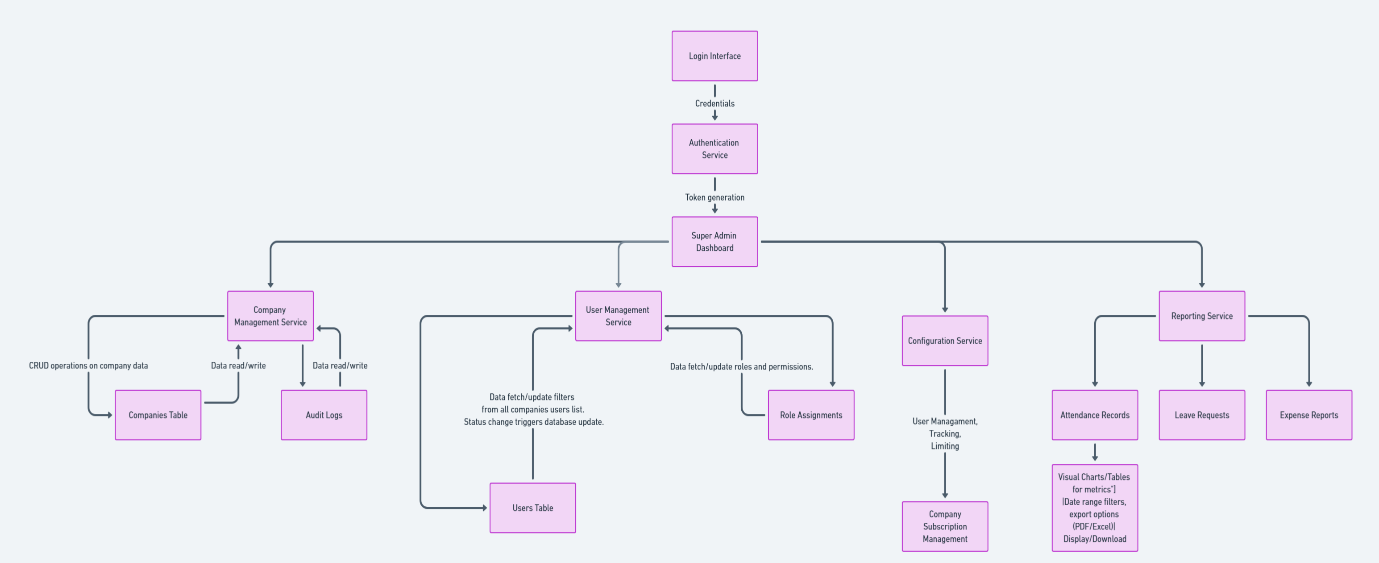
* **Action:** Generate group-specific reports on attendance, expenses, tasks, travel, and performance.
* **Front-End:** Reports are displayed as charts (line, bar, pie) with filtering and export options (PDF).
* **Back-End:** Calls endpoints such as POST /api/reports/generate with group-level filters.
* **Database:** Aggregated data from multiple tables (e.g., leave\_requests, attendance\_records, expenses) is used.

**7.3.2.7 Personal Dashboard & Profile Management**

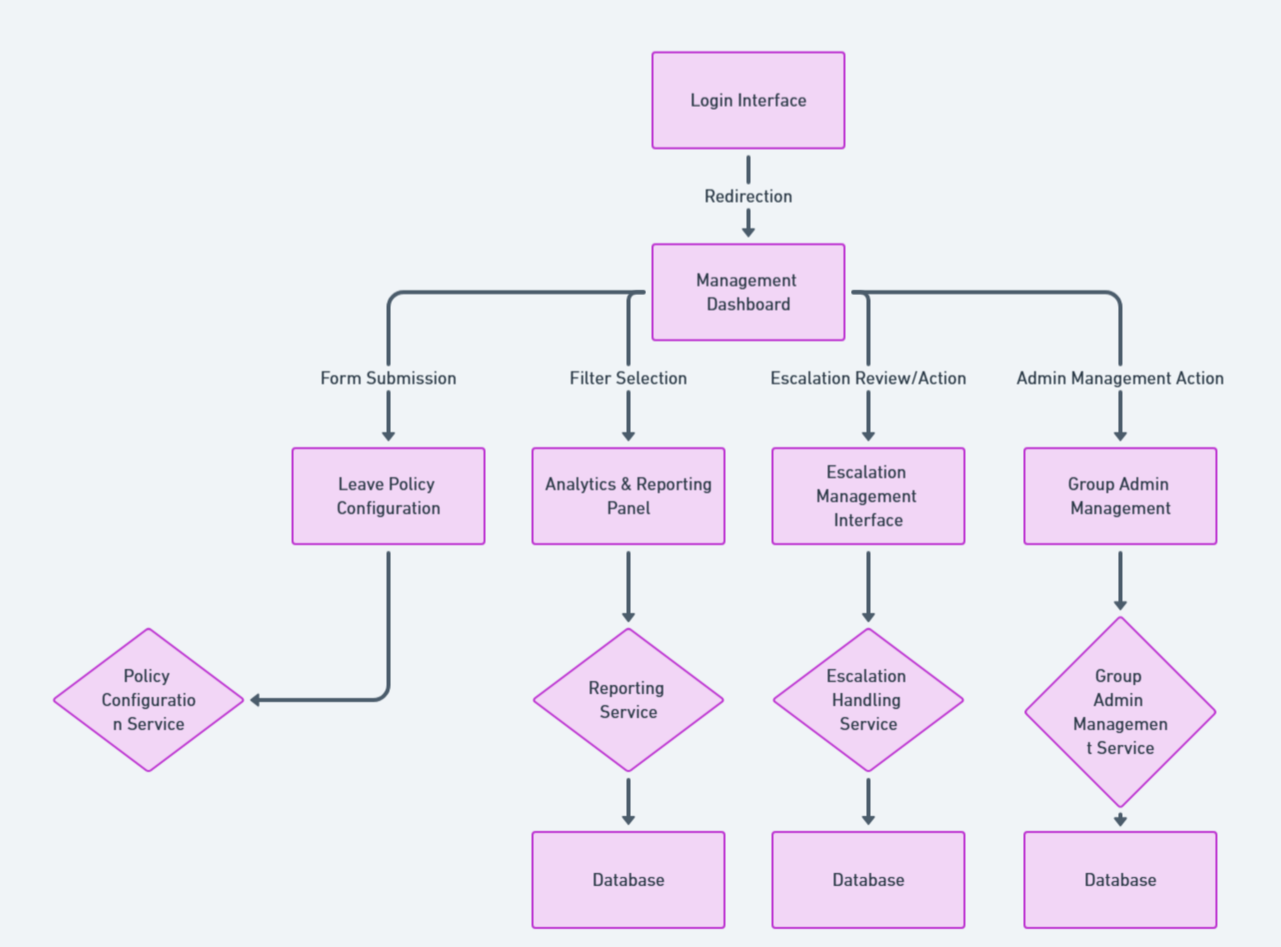
* **Action:** Access personal profile details, recent activities, and customization options (e.g., theme switch).
* **Front-End:** A profile section shows personal details, performance summaries (total hours worked, expense claims, task progress), and navigation to support.
* **Back-End:** Endpoints such as GET /api/users/profile and PUT /api/users/profile update personal information.
* **Database:** The users table is used for storing profile data.

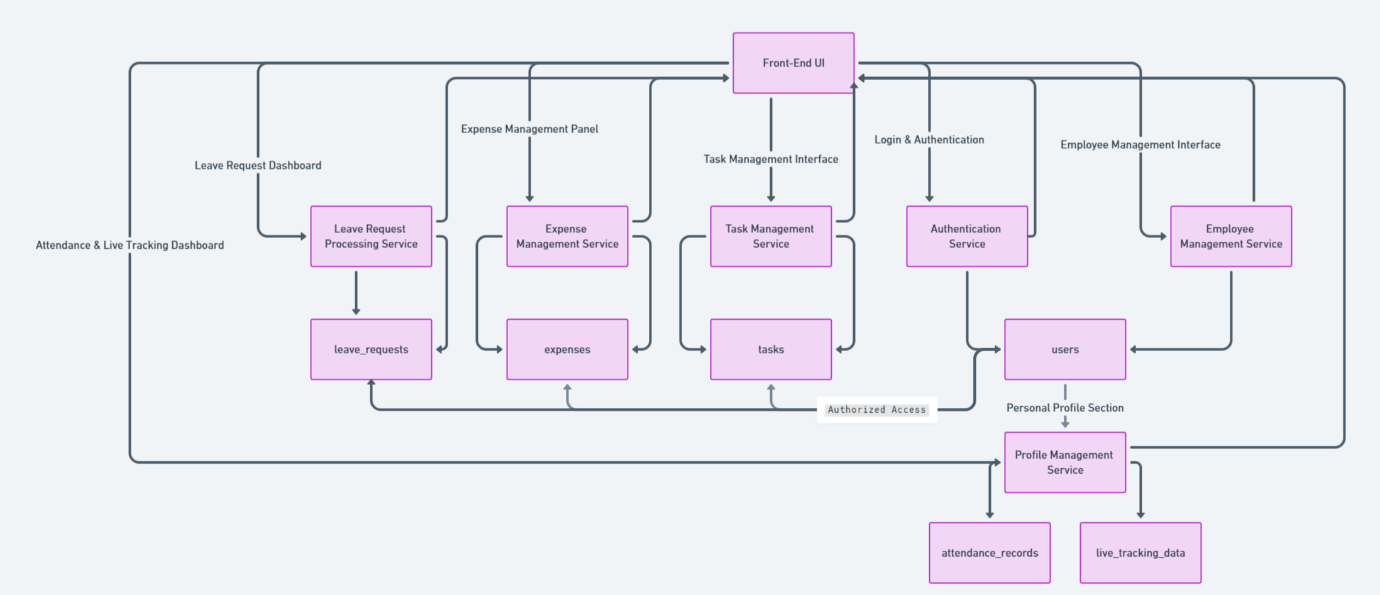
**8. Workflow Diagram**

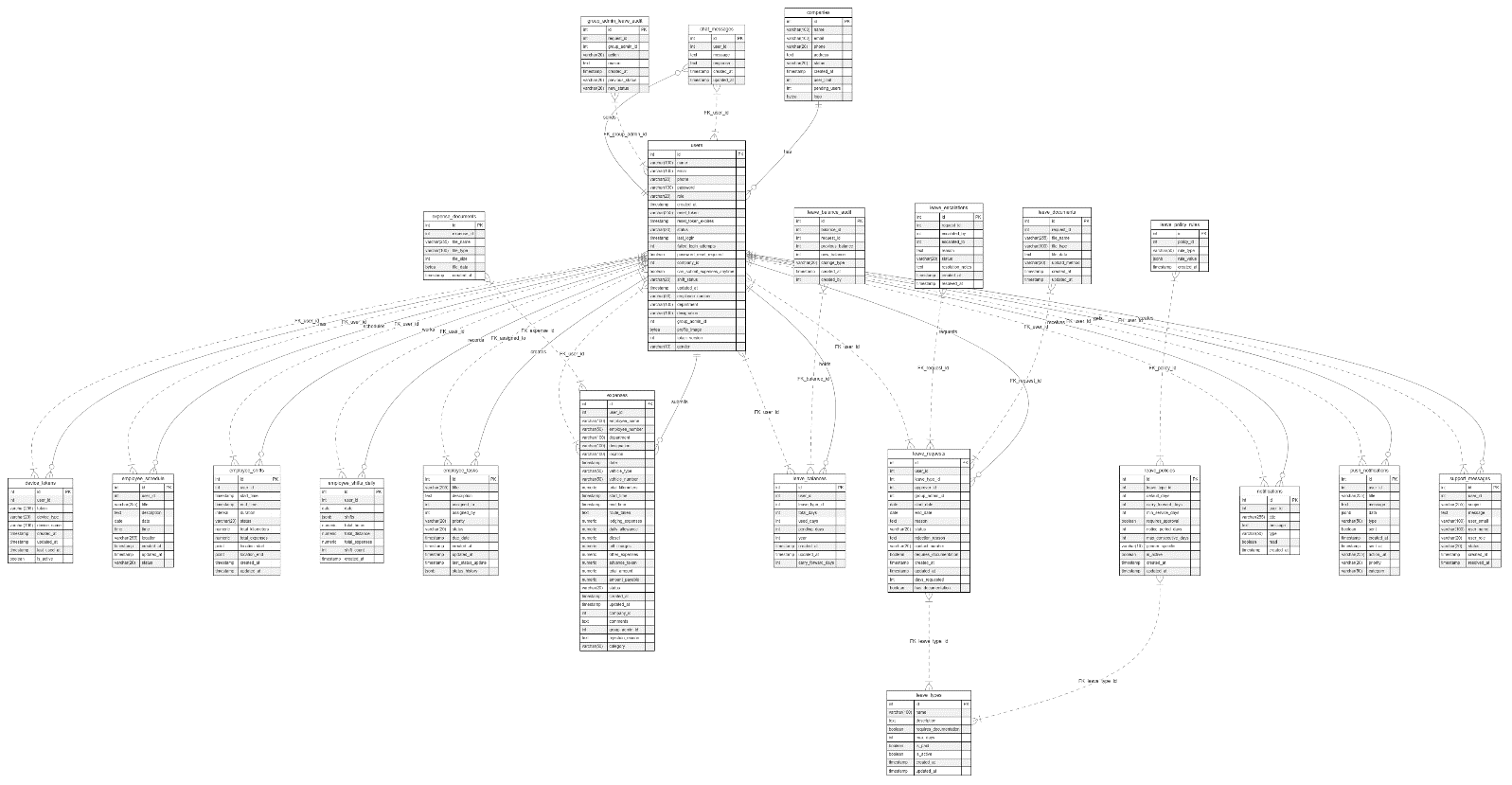
**8.1. All Admin Workflow**

**8.2. Super Admin Workflow**

**8.3. Management Personal Workflow**



**8.4. Group Admin Workflow**

**9.Database Schema**