|  |
| --- |
| User Management System |
| * **Parth Kansara** * **(M) +91 8849291401** * **(E) parthkansara11@gmail.com** |

## **High Level System Design**

* **User Management Service (UMS)**: Responsible for managing users, their roles, permissions, and profile details. This service should offer CRUD operations for user management.
* **Authentication Service**: Ensures that users can log in securely using a username/password (e.g., OAuth, JWT). This should be capable of integrating with enterprise Single Sign-On (SSO) systems.
* **Role & Permission Management**: Responsible for assigning roles to users, and for managing the permissions tied to those roles.
* **Audit Service**: Tracks system events (logins, role changes, password changes) and stores them for security compliance.
* **Notification Service**: Handles notifications to users, such as password change notifications, account creation, etc.
* **Database**: Relational database (e.g., PostgreSQL, MySQL) for storing users, roles, permissions, and audit logs.
* **Deployment:** Deployment will be done via GIT actions. AWS services ECS will be used to deploy the code in the containerized manner.

## **Core Feature and API Design**

**Key Features of User Management System:**

* **User Creation and Update**: Allows for the addition, modification, and removal of users.
* **Role Assignment and Permission Management**: Allows admins to assign roles to users and define what actions are allowed based on those roles.
* **Authentication**: Secure login and token generation (JWT/OAuth2).
* **Password Management**: Secure password creation, hashing, and management.
* **Audit Logging**: Tracks important actions for auditing (e.g., login attempts, password changes).
* **Permission Checking**: Enforces access control by checking user roles and permissions before granting access to resources.