Information Architecture Document

Project Overview

This document defines the information architecture for a social networking platform that integrates professional user profiles, social interactions (likes, comments, follows), and settings management.

Technology Stack

Flask (Python), SQLAlchemy ORM, Keycloak (for Authentication), Relational Database (e.g., PostgreSQL)

Logical Data Model

- User: Authenticates via Keycloak (keycloak_id used as external identifier).
- Post: Created by users with optional images.
- Comment: Users can comment on posts.
- Like / Reaction: Engagement with posts.
- Follow: One user follows another.
- ProfessionalDetails: One-to-one with User.
- Chat: Direct messaging.
- EmailNotificationSettings: Preferences for notifications.
- ProfileVisibilitySettings: Controls visibility of profile data.

Entity Relationships

- One-to-Many: User -> Post, Comment, Like, Reaction, Follow, Chat
- One-to-One: User -> ProfessionalDetails
- Many-to-Many: User <-> User (via Follow model)

Authentication & Identification

External Identity Provider: Keycloak

keycloak_id used as external unique identifier and referenced in multiple settings tables.

Settings and Configurations

- EmailNotificationSettings: Boolean values tied to setting IDs.
- ProfileVisibilitySettings: Uses category and setting_id to manage profile field visibility.

Information Architecture Document

Normalization & Constraints

Schema is in 3rd Normal Form (3NF).

- Unique constraints prevent duplicate follows and likes.

- Check constraint in Chat prevents users from messaging themselves.

Scalability Considerations

- Indexed fields for performance.

- CASCADE deletions to maintain integrity.

- Optimized for high-volume social interactions.

Future-Proofing

- Easy extension to group features, reactions on comments, media uploads via CDN.