

Low Level Design Document

For

Clone Social Media App (Backend)

Version 1.0

This is a backend part of social media kind of application with some functionality.

1. Introduction

This is a clone of a social media app where user authentication, adding posts, and discussions among users on the posts are facilitated, with interaction modes including likes and comments.

2. There are 3 components totally:

a. User

1. In this component, we have taken care of all the operations related to users. As per our current requirement.
2. The functioning is divided into 2 blocks further: Auth, User
3. In the Auth, we are having the login and signup functionality
4. Where as in the User, we have add following functionality
 - Edit or Update User -> updating user details
 - Get particular user details
 - Delete user. Which I have made inactive in the db for future reference
 - List users -> All user list
 - Search users
 - Follow users
 - Unfollow users

b. Post

1. In this component, we have handled the operations related to post and discussion in the post.
2. Since the post can be a text or text with image. Those things are managed from this component
3. The following function are taken care from this components

- Create post -> For adding a new post
- Increment post view -> Every time some view a post this route will be hit
- Get post by tags
- Search post by tags
- Get particular post
- Update post
- Delete post

c. Interaction

1. Since the user is adding a post. It will be leading to further discussion among the other available user.
2. The interaction can be done in likes or comments and replies of comments.
3. The following functionalities are there currently
 - Like and unlike a post
 - Like and unlike a comment
 - Add comment
 - Edit comment
 - Delete comment
 - Add a reply to comment

3. Models Details:

a. User Component

- Model: "users"
- Field:
 - name: String, required
 - mobile: String, required, unique,
 - email: String, required, unique
 - password: String, required
 - followers: ObjectId from User collection
 - following: ObjectId from User collection
 - status: Enum [ACTIVE, INACTIVE]

b. Post Component

- Model: "posts"
- Fields:
 - text: String, required
 - image: String
 - hashtags: [String]
 - likes: ObjectId from Like Collection
 - comments: ObjectId from Like Collection
 - createdBy: ObjectId from User Collection, required

- status: Enum [ACTIVE, INACTIVE]
- viewCount: Number

c. Interaction Components

1. Comment Component:

- Model: "comments"
- Fields:
 - userId: ObjectId from User Collection, required
 - postId: ObjectId from Post Collection, required
 - text: String, required
 - replies: Array of ObjectId from Reply Collection
 - likes: Array of ObjectId from User Collection

2. Likes Component

- Model: "likes"
- Fields:
 - userId: ObjectId from User Collection, required
 - postId: ObjectId from Post Collection, required

3. Reply Component

- Model: "replies"
- Fields:
 - userId: ObjectId from User Collection, required
 - commentid: ObjectId from Post Collection, required
 - text: String, texts

Diagram Illustrating the System Architecture

