

Specification

Domain of Application

The domain of the application is an Android blogging social media application where users can submit written posts, follow other users, like others' post, and receive recommended posts from a suggested feed. The application will also be able to support multiple concurrent users.

Useful Terminology

- User: A user is a real-life person who has an account in the blogging application.
- Admin: A user with extra permissions to perform administrative tasks related to users and posts.
- Post: A post is an article that a user writes and adds to their profile.
- Profile: A profile consists of a user's biography and a collection of all posts written by the user.
- Suggested feed: A personalized page which collects all posts authored by followed users ordered by recency.

Application Functionality

Here is a more detailed overview of what a user can do with this application.

Users can sign up for new accounts. Upon signing up for a new account or logging in to an existing account, users can perform:

- Create operations like
 - writing posts which can be added to their profile
 - writing a biography about themselves
- Read operations like
 - querying user profiles based on usernames
 - querying posts based on keywords to read posts aligning with their interest
 - viewing posts in their suggested feeds
- Update operations like
 - updating their account biography
 - updating their following list by following or unfollowing other users
 - liking posts written by other users
- Delete operations like
 - deleting their own posts
 - deleting their own account

After finishing with their uses of the application, users can then log out.

Admins share the same functionality as users, but admins may additionally perform:

- Update operations like
 - promoting non-admin users to become admins
 - banning non-admin accounts
- Delete operations like
 - deleting posts written by non-admin users
 - deleting non-admin accounts

User Interface (UI)

- When the application is opened by a user, the user will be brought to a login and sign up page where they choose to either login or sign up.
- Once logged in or signed up, there will be 3 different pages that can be navigated to using a horizontal navigation bar at the bottom of the user's screen.
 - The first button leads users to their suggested feed page.
 - The second button leads users to their profile where they can manage their posts and perform admin commands if the user is an admin.
 - The last button leads users to a search page where they can search for user profiles or posts.

Complexity

To gauge the application's complexity, we explain what the architecture could look like for phase 1.

Entities

- *Account*: An account is created by a user. Accounts have getters and setters for data such as accounts followed, followers, a user biography, and a list of posts created by the user.
- *Admin*: An admin extends all functionalities of an account. In addition, an admin also stores an admin log to track commands that can only be made by admins.
- *Post*: A post is an article that a user can add to his or her profile. Posts have getters and setters for data such as post title, post content, date of publication, and the number of likes on the post.

Use Cases

- *AccountManager*: The AccountManager contains the implementation for the application logic for commands that can be made by regular accounts. The AccountManager stores a collection of all regular accounts.
- *AdminManager*: The AdminManager contains the implementation for the application logic for commands that can be made by admin accounts. The AdminManager stores a collection of all admin accounts.

Controllers

- We keep all controllers from phase 0 which are responsible for handling requests related to logging in, signing up, banning users, unbanning users, and deleting accounts.
- Additional controllers we believe can be implemented in phase 1 include:
 - *PostController*: This controller is responsible for handling user input for create, read, update, and delete (CRUD) operations related to posts (e.g. liking a post).
 - *AccountController*: This controller is responsible for handling user input for CRUD operations related to a user's account (e.g. updating account biography).
 - *AdminController*: This controller is responsible for handling user input for CRUD operations related to admin commands (e.g. banning a non-admin user).
 - *SearchController*: This controller is responsible for handling user input when users search for user profiles based on usernames or for posts based on keywords.
 - *NavigationController*: This controller is responsible for reading user input for navigating between the 3 different pages in the UI.