

Final Assignment

Android Programming Class

Topic: Forum-like Application

Features to implement:

- **User Authentication:**
 - Users can create an account and log in to the application.
 - Users can reset their password if they forget it.
 - Users can sign in using Google.
 - Authentication is only necessary when user posts. Reading posts does not require Authentication.
- **Forum Posting:**
 - Users can create new posts and reply to existing posts.
 - Posts can contain text, images, and links.
 - Posts should belong to topics. (When a user signs in, he sees Topics, then sub-topics etc.
- **Notifications ([Bonus +3](#)):**
 - Users can receive notifications when someone replies to their post or mentions them in a post.
- **User Profiles:**
 - Users can create and edit their profile, including their username, display name, and profile picture.
 - Users can view other users' profiles.
- **Search:**
 - Users can search for posts, users, and categories.
- **Moderation:**
 - Moderators can delete posts and comments that violate the forum rules.
- **Real-time Communication:**
 - Forum Topics are updated in real time.
 - ([Bonus + 3](#)) Chat messages can be sent to individuals or groups.
- **Photo Attachment (Only for teams with 3 members. 2 member teams can skip these requirement):**
 - Users can attach photos to their posts and comments.
 - Photos can be uploaded from the user's device..

Team Members:

- *2-3 Persons*

Implementation Details:

- **Database/Backend:** Firebase FireStore, (Real Time Database)

- **Authentication:** Firebase Authentication
- **Front-end:** Android Jetpack Compose, Material You (M3)
- **Architecture:** Unidirectional Data Flow, Repository Pattern, ViewModels
- **Photo Attachment:** Use Firebase Storage to upload photos.

Evaluation:

The final assignment will be evaluated based on the following criteria:

- **Code Quality:** The code should be well-written, organized, and easy to read.
- **Architecture:** The application should have a clean and modular architecture.
- **Design (UI/UX):** The application should have a user-friendly and intuitive UI/UX.
- **Feature Completeness:** The application should implement all of the required features.
- **Stability & Performance:** The application should be stable and performant.

Submission:

The repository should include a README file that describes the application and its features.

Deadline:

See on Github Classroom.