Mobile Application Development 2023-24

A teamwork management application

Lab4 – Manage teams

Learning objectives

- Appling MVVM architecture
- Implementing navigation
- Designing UX layers
- Using JepackCompose to build interactive screens
- Modelling the team entity and its related information
- Modelling the team member entity
- Displaying a dynamic list of objects

Description

From the Lab1 description:
"*A teamwork management app is designed to facilitate collaboration, communication, and coordination among **team members** to enhance users' overall productivity and efficiency.*
*By using this app, a user may manage her/his participation in one or more **teams**, get and set information about tasks to be performed by **team members**, document her/his own progress and achievements, report **contributed efforts**, and gather **feedback** from other team members as well as analytic information derived from collected data.*

*The goal is to streamline **communication**, enhance collaboration, and improve overall team performance by providing an easy to use app that keeps track of all the details of the team activity.*

The app will support the following features:

- *Managing teams*
    - *Ability to create, edit, and delete teams*
    - *Sending invitations to new members via deep links and/or QR-Codes*
    - *Accessing the list of team members, showing profile pictures and roles*
    - *Option to send direct messages to other team members or to the team as a whole*

    - *Ability to set the degree of one's participation in a team or to withdraw from it*
- *Feedback and Performance:*
    - *Performance metrics and analytics for individual team members*
    - *Visual representations of team achievements*”

Users are inclined to use the app as they seek to oversee their team activities efficiently. It's important that users can easily navigate their team memberships, access teammates' profiles, engage in chats with them, and adjust their participation within groups, other than exploring and managing team tasks.

Furthermore, the app should allow users to organize their teams smoothly, enabling them to create, customize, invite other members, and leave teams effortlessly.

Moreover, the integration of an achievements section for each team adds considerable value to the app. This feature assists in tracking team contributions and goal achievements, thereby enhancing collaboration and productivity. Thoughtful design of this section is essential to ensure it serves its purpose effectively.

Steps

1. To accept the assignment, use GitHub Classroom as you did in previous labs: https://classroom.github.com/a/I9qcx08w

2. Create a new project on Android Studio

    a. *Phone and Tablet → Empty Activity*

    b. Commit project. Push it on the remote repository

3. Create and customize the TeamsScreen component.
   Start creating the **ShowTeamDetails** pane component

    a. A non-exhaustive list of team details includes: team name, group image, description, category, team member(s), creation date

    b. Manage the list of team members: users can invite new members and can eject existing ones. Users can also view a **member’s profile** in detail

    c. Give the possibility to a user to change his/her time participation to a group

4. Add the **NewTeam** pane component

a. This component shows up upon user initiation to *create* a new team

a. It should be possible to provide all the *information* displayed in the previous component

b. Ensure that the inserted data withstands *configuration* changes

a. Perform *validation* on saving. Display useful information to the user if an error occurs during validation

5. As you did in the previous lab with the ShowTaskDetails and EditTask components, craft an **EditTeam** component for modifying information about an existing team

   a. Ensure that the components are filled with the *previous* values and that the validation is performed on saving.
   Display useful information to the user if an error occurs during validation

6. Create the **TeamList** pane component to display the list of teams of which a user is a member

   a. Enable filtering functionalities

   b. Team list may contain a *large* number of items and not all of them could be visible in the component's viewport. As you did in the previous lab with the tasks list, implement a suitable approach, as proposed in Android/Compose, for efficiently displaying a collection of items. Consider adding *animation* to give users visual feedback on actions performed on the task list

   c. Integrate a Floating Action Button (FAB) for streamlined creation of a new team

7. Develop a **Chat** component enabling team members to communicate with each other

   a. The team chat displays messages including the *date* and *time* they were sent, as well as the *sender's* identity

   b. Users can send *messages* to the entire group or direct them to specific individuals by tagging them

   c. Incorporate a visual *notification* system (such as a badge on the team within the team list or a new section in the app) to help users easily identify new messages in the team chat

8. Create an **Achievement** component allowing team members to monitor team contributions and accomplishments

    a. Select and display relevant *Key Performance Indicators* (KPIs) and information (e.g., the number of completed tasks or the member who completed the most tasks in the previous month…)

9. Enable **navigation** across the various components you've developed: users can move from the list of teams to accessing a specific team, viewing its details, checking the task list, accessing the chat, viewing member profiles, and reviewing achievements

Submission rules

- The work must be submitted by June, 3 23:59
- The design and the code of the user interface will be evaluated.
- The last commit before the deadline will be evaluated. Alternatively, create a release and label it “completed”.