Mobile Application Development 2023-24

# A teamwork management application

## Lab3 – Manage tasks

## Learning objectives

- Designing early UX layers
- Using JepackCompose to build interactive screens
- Modelling the task entity and its related information
- Modelling the state of a task
- Creating 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 app will support the following features:*

- *Managing tasks of a team*
    - *Ability to create, edit, and delegate tasks to team members*
    - *Ability to define recurring tasks*
    - *Task categorization and tagging for easy identification*
    - *Real-time task status tracking (e.g., pending, in-progress, on-hold, completed, overdue)*
    - *Filter and sort options for easy navigation*
- *Task Details:*
    - *Detailed view of a selected task*
    - *Task title, description, assigned team member(s), and due date*
    - *Task state and history*
    - *Comments section for team communication*
    - *Attachments and links to related documents*"

The **task** is the heart of the application: the work of a group is managed by *dividing* the work into tasks and *assigning* them to the actual members. Visualising a task and a task list is therefore very important for the effectiveness of the app.
A task has associated **information**, a **history of actions** performed on it (status changes/member who performed the action/members to whom it was assigned), and **comments**. It is essential to model the visualisation of these data and, of course, to be able to edit it properly.

Members of a group can view the **task list**, **sort** it by creation/expiry date, **filter** it by assigned member, status, tag/category, and of course **add new** tasks.

Finally, each member can view the page with their *assigned tasks* to better organise their work.

Remember to pay attention to usability and effectiveness in displaying the most important information for a potential user of the application.

# Steps

1. To accept the assignment, use GitHub Classroom as you did in previous labs: https://classroom.github.com/a/BiM75ODi
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 GroupTasksScreen component.
   Start creating the **ShowTaskDetails** pane component
    a. A non-exhaustive list of task details includes: title, description, tag, category, assigned team member(s), due date, and state
    b. Allow for the inclusion of *comments*, facilitating users to access all comments linked to a task
    c. Status and assigned member changes over time (task history) are very important. Show the *history*, identifying who made modifications and when.
4. Add the **NewTask** pane component

    a. This component shows up upon user initiation to *create* a new task
    b. It should be possible to provide all the *information* displayed in the previous component
    c. Ensure that the inserted data withstands *configuration ch*anges
    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 User Information and EditProfile components, craft an **EditTask** component for modifying information pertaining to an existing task
    a. Make sure 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
    b. Allow the *status* to be changed and the *member*(s) to whom a task is *assigned* to be modified (or do it in ShowTaskDetails component, based on your group choice)
6. Create the **TaskList** pane component to display the list of tasks created by group members
    a. Enable sorting and filtering functionalities
    b. Task list may contain a *large* number of items and not all of them could be visible in the component's viewport. Implement a suitable approach, as proposed in Android/Compose, for efficiently displaying a collection of items. Consider adding *animation* to give users visual feedback of actions performed on the task list
    c. Integrate a Floating Action Button (FAB) for streamlined creation of new tasks
7. Create **PersonalTaskList** pane component. This component is visually similar to the general task list but shows only the tasks of a specific user.

# Submission rules

- The work must be submitted by May, 13 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”.