Mobile Application Development

Project 2025-2026

In this project, you should develop an Android Application that enables users to manage their daily tasks (i.e., tasks created at the beginning of a day that should be completed in the following hours). In particular, the Android Application should enable users to perform the following operations:

(A) [1.0]: **Create** a new Task, by providing the appropriate data through a user-friendly interface. The data provided should be stored in a **Relational Database**, using **Room** (*see comment no. 1* below).

(B) [0.5]: The user should be capable of specifying the **default** duration and difficulty of new tasks in a separate "page" using Shared Preferences.

(C) [0.5]: **Periodically** (e.g., every 1 hour) **check** the existing tasks and **update their status** (if being necessary) taking into account a few parameters such as their «start-date» and «duration», as described below (*see comment no. 2*). Tasks from previous days should be automatically **removed**.

(D) [1.0]: **View** ALL Tasks that are **NOT** already COMPLETED with the appropriate order, so that the «urgent» tasks (e.g., tasks «expired») appear on the top of their screen. For user convenience the App should initially provide an overview of existing non-completed tasks (using a Recycler View) and accordingly (on click) more information about the selected one (*see comment no. 3* below).

(E) [1.0]: In the second case of question "D" (i.e., in the screen presenting all the available data for each task), the user should be capable of changing their **status** to COMPETED, if being necessary, by pressing a button. Also, in case the location is not empty, the App should enable users (by pressing another button) to see the location provided on **Goolge Maps**.

(F) [0.5]: The application should enable users to **export tasks** not completed yet (including all the data recorded for each one of them) in a **TEXT** (or ideally HTML) file stored in a shared folder (e.g., Downloads directory), so that the user can accordingly open it using another Android Application.

(G) [0.5]: The App should enable Other Android Applications to update the content of the datasource, through the implementation of a **Content Provider**. For testing purposes, you can use the **Content Resolver** for performing the CRUD operations in the same App (on request).

Deliverable: A ZIP or RAR file with the following data:

(a) **Source Code [5 points],** including app manifest xml file, java classes created and res files created/updated, organized in subfolders (e.g., one folder for each one of these three categories)

(b) **Report and Video [1 point]** that demonstrates the functionality provided by the application developed. At the beginning of the document (report) you should provide your Full Name and your HUA-DIT code. Also you should clearly state the functionalities covered by your application (i.e., A, B, C, D, E, F, G). Accordingly, for each one of them, you should provide a short description, example(s) of usage and relevant screenshots. The video should be in accordance with your report and presents the functionality provided. You can either upload the video (inside the ZIP/RAR file or provide a link to this file, at the beginning of your report. Functionality not covered by the video will be ignored.

If something is missing (e.g., video) you won't be graded ! ! !

Comments

Comment #1: Data Storage and App UI

For each Task, the user should provide the following parameters (apart from UID, date and status).

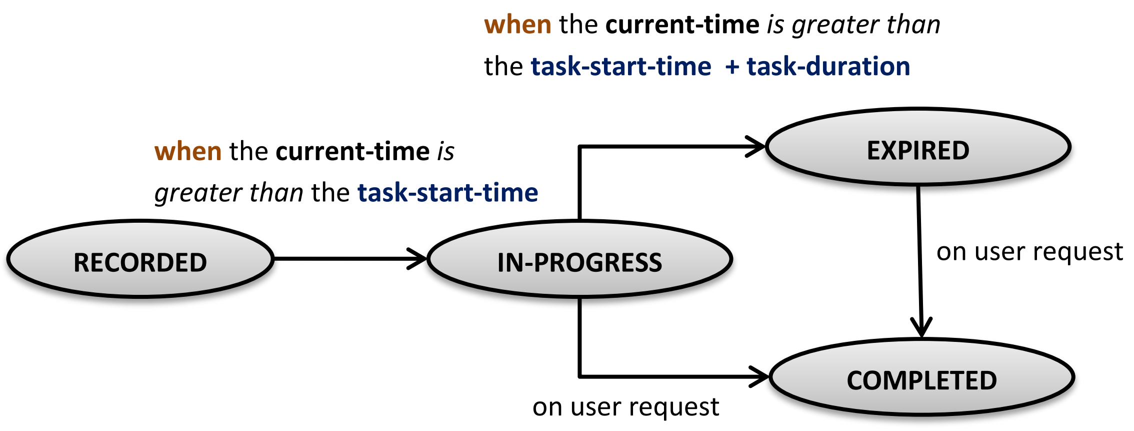
- Unique ID aka UID (an integer, automatically generated)
- Short Name: A String that does not exceed 20 characters.
- Brief Description: A String that does not exceed 150 characters.
- Difficulty: An integer in range [0,10]. 0: very easy task, 10: very difficult task
- Task Date: The date that the user created a task
- Start Time: The time that the user should start doing this task (hours & minutes using a Picker)
- Duration: The number of hours required for completing this task
- Status: a term from a pre-defined list of terms (see below – comment no. 2)
- Location: The address of particular interest for this task – could be empty

Regarding the data storage, the tasks and the possible statuses should be stored in two separate tables that are linked through a *foreign key*. The App should also check that the data provided by the user are valid (e.g., ensures that there is a short name and the duration provided is a positive integer). When creating a new Task, the status should be "recorded" and the date should be the current date.

Comment #2: Task State Transition Diagram

In this project, a Task can be in one of the following four states: recorded, in-progress, expired, completed. The status of a new task should be "recorded". Provided that the current time is greater than the start time of a task, its status should *automatically* change to "in-progress". Accordingly, the status can change to "expired" on condition that the current time is greater than the one required for its completion. The status of a task can change to "completed" *on user's request*.

The above are depicted in the following state transition diagram:



when the current-time *is greater than*
the task-start-time + task-duration
when the current-time *is*
greater than the task-start-time
RECORDED
IN-PROGRESS
EXPIRED
on user request
COMPLETED
on user request

Comment #3: Task Overview and Task Details

The application should initially present a few parameters about each Task (e.g., ID, Short Name and Status) using **RecyclerView** and accordingly when the users selects the desirable one (e.g., by clicking on it) the application should present all the available data recorded about the particular task. In the second case, the application can provide additional elements (e.g, buttons) so that the user can further process this task (e.g., change it status to **completed**) or explore the data provided (e.g., visualization of the location in **GoogleMap**, if not empty) as already described in question "E".