Task Manager Application Design Document

Stephan Peters

Colorado State University Global

24WD-CSC475-1: Software Engineering

Herbert Pensado

9 March, 2025

Task Manager Application Design Document

Finding good personal task management software can be difficult. There are many approaches, some take the form of a game (gamification) such as Habitica or Forest (Loayza, 2017), some are just a simple task list, such as Todoist or Google Tasks (Pot, 2022), but finding one that adequately handles subtasks and subtasks of subtasks for larger or long-term personal projects (Wren, 2023).

This design document provides a comprehensive overview of the To-Do application I am designing that includes tasks and sub-tasks, including its main components, task structure, priority system, and key features.

Overview and Main Components

1. Application Overview

The ToDo application is a task management system with a hierarchical structure, allowing users to create, manage, and track tasks and subtasks. The application features a priority-based ordering system and a user-friendly interface for easy task management.

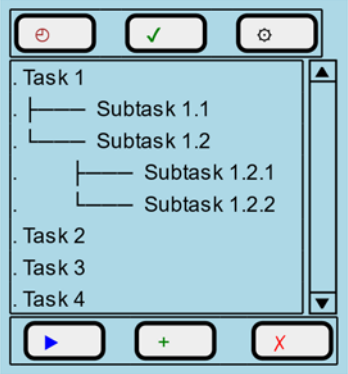
2. Main Components

2.1 User Interface

The application consists of three main screens, a ToDo Screen, a Done Screen, and a Settings Screen. The main screen will show main tasks that the user should complete, pressing on a task will reveal subtasks (if any). The Done screen will allow the user to view completed tasks. Completed subtasks of incomplete tasks will also show here under a greyed-out task, indicating the main task is not complete.

Figure 1

Main UI Structure for the ToDo and Done screens

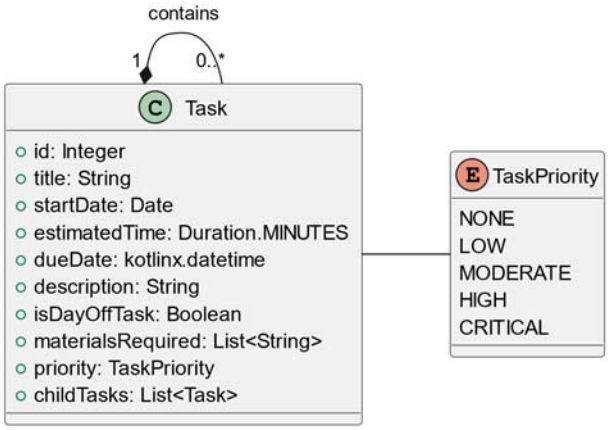


2.2 Task Structure

The application organizes tasks in a hierarchical structure, parent tasks, child tasks, and grandchild tasks.

Figure 2:

Task Hierarchy



2.3 Priority System

The priority of each task is determined by an internal formula based on the importance of the task, time since the last update, estimated completion time, and due date. For example, if the

user does not start a “Whenever” task by a configurable deadline, the app will assign it a higher priority.

Color visually represents task priority:

- Green: Low priority
- Yellow: Moderate priority
- Orange: High priority
- Red: Critical priority

3. Navigation

The top tab bar switches between the ToDo (◴), Done (✓), and Settings (⚙) screens. The bottom navigation bar has buttons for details (▶) which opens the task details dialog, add (+) which creates a new task or subtask, and exit (✗) which closes the application.

4. Task Details

Each task contains the following information, a start date which can be a specific date, “Now,” or “Whenever,” an estimated time to complete in hours and minutes, a due date which can be a specific date, “Now,” or “Whenever,” a description of the task, a day off task flag (Boolean), and materials required (List<String>).

5. Task Management, Creation and Editing

A list on the screen displays tasks, ordered by priority. Parent tasks expand to show child tasks. Child tasks inherit properties from parent tasks, but the user can modify them. They can also expand to show grandchild tasks. The user creates new tasks using the Add button. If the user has not selected a task, the app creates a new task. If the user selected a task, it creates a new subtask under the selected task. The Details dialog edits tasks.

6. Data Synchronization

The child tasks synchronize materials required to parent tasks upon creation or editing. Child tasks do not automatically re-sync with parent tasks after initial creation

7. Settings

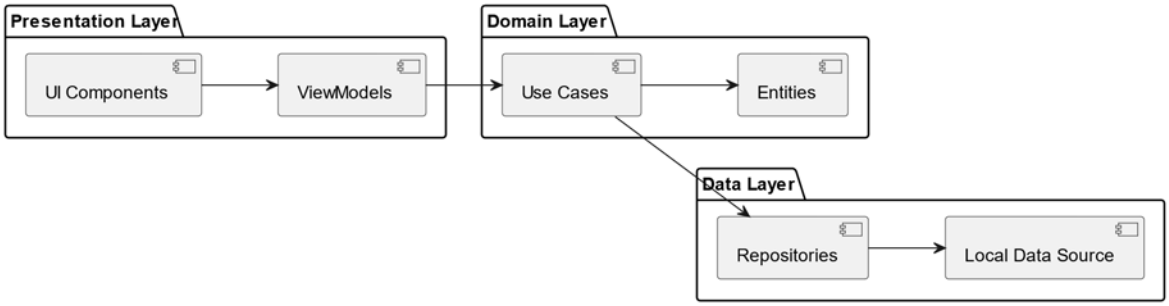
The Settings screen allows users to configure application preferences, such as default start and due dates, normal work hours, sleep hours, and days off. The user may also edit notification preferences.

9. Implementation Considerations

The application will use a local database (e.g., Room (Google, 2019)) for persistent storage of tasks, it will use Jetpack Compose for building the user interface (Tarasov, 2024) and implement a WorkManager for background task prioritization and notifications (Android Developers, 2025).

Figure 3:

High-Level Architecture



Conclusion

This application design will offer a task management application that allows for tasks, subtasks, and subtasks of subtasks. For example, if a user is designing a garden plot, a “finish garden” task is not enough. The user will need to complete subtasks such as digging weeds, laying out landscape fabric, erecting bean climbing towers, redirecting gutters to plot area, &c. &c. This application will solve this problem.

References

Android Developers. (2025a). *Define work requests*. Android Developers. https://developer.android.com/develop/background-work/background-tasks/persistent/getting-started/define-work

Google. (2019). *Save data in a local database using Room*. Android Developers. https://developer.android.com/training/data-storage/room

Horton, J. (2019). *Android programming with Kotlin for beginners*. Packt Publishing Ltd.

Loayza, J. (2017, February 24). *The 10 best productivity apps that use gamification*. Yu-Kai Chou: Gamification & Behavioral Design. https://yukaichou.com/lifestyle-gamification/the-top-ten-gamified-productivity-apps/

Pot, J. (2022, November 18). *9 best to do list apps*. Zapier.com. https://zapier.com/blog/best-todo-list-apps/

Tarasov, V. (2024, April 10). *Jetpack Compose: A new approach to Android UI development*. Medium. https://medium.com/@vptarasov/jetpack-compose-a-new-approach-to-android-ui-development-4ca03d7ee261

Wren, C. (2023, October 28). *Trying to find the perfect todo list app*. Medium. https://colinwren.medium.com/trying-to-find-the-perfect-todo-list-app-ccbe0ca46070