

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Screen 5](#)

[Screen 6](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any edge or corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services or other external services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Implement Settings](#)

[Task 4: Implement Notification](#)

[Task 5: Implement Widget](#)

[Task 6: Implement Ad Service](#)

[Task 7: Implement Location Service](#)

[Task 8: Final task](#)

GitHub Username: ashj (<https://github.com/ashj/Capstone-Project>)

Daily Ten-Minutes Tasks

Description

Create and manage short task (maximum 10 minutes) to be accomplished one by one each day. For those who have lots to do but almost no time to do anything else.

Intended User

Users who have several tasks (domestic tasks, for example) to be done, but that almost never end up doing, usually because of lack of free time.

The user registers several relatively short tasks (maximum 10 minutes to complete). The application then reports a task to be performed daily.

In this way, users end up doing a lot of short tasks, instead doing nothing.

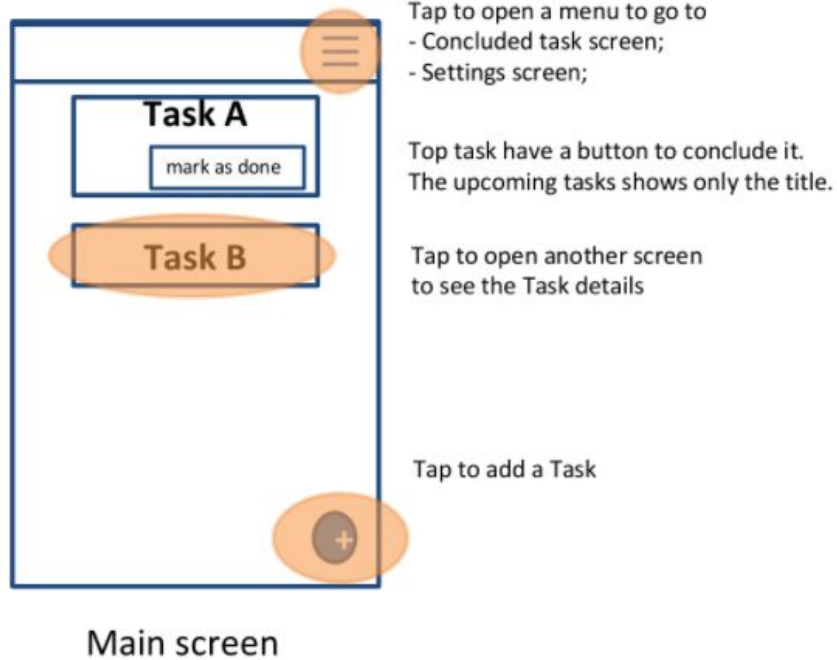
Features

- Tasks sorted by priority;
- History of tasks done;
- Daily notification about the task to be done.

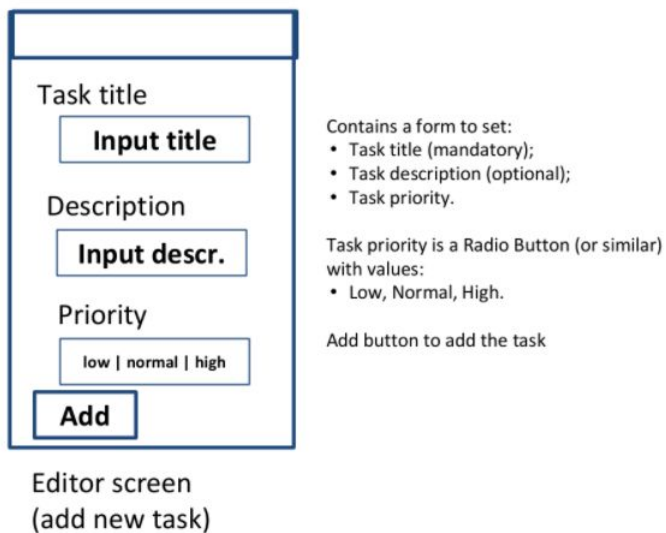
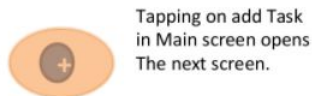
User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



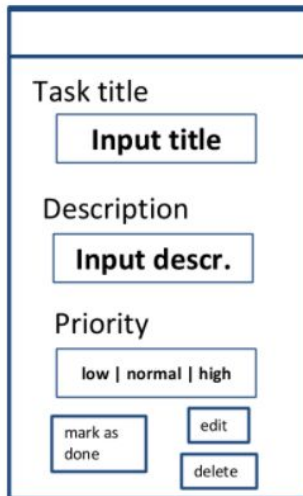
Screen 2



Screen 3

Task B

Tap to open another screen to see the Task details



The diagram shows a mobile app screen for task details. It has a white background with a blue border. At the top, there's a header bar. Below it, the text 'Task title' is followed by an input field containing 'Input title'. Then 'Description' is followed by an input field containing 'Input descr.'. Below that, 'Priority' is followed by a button labeled 'low | normal | high'. At the bottom, there are three buttons: 'mark as done', 'edit', and 'delete'.

Task details screen

Shows the task details:

- Task title;
- Task description;
- Task priority.

Mark as done concludes the task.

Edit button allows to edit the task.

Delete button deletes the task from the database.

Screen 4



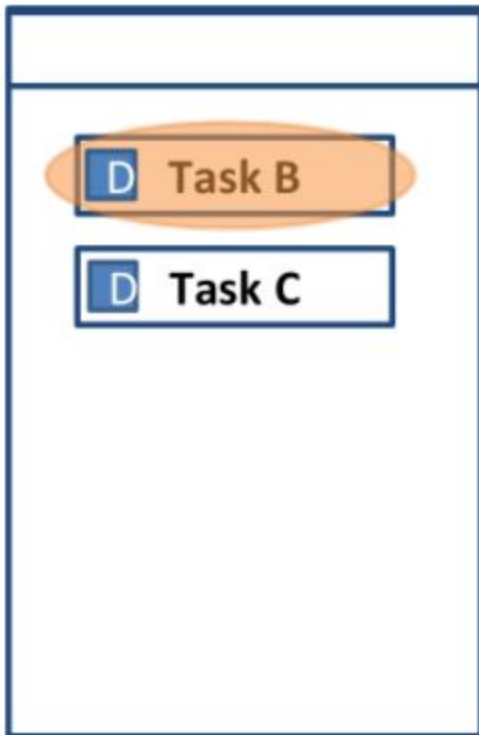
The diagram shows a mobile app settings screen. It has a white background with a blue border. At the top, there's a header bar. Below it, there's a checkbox labeled 'Show daily notification at' followed by a time input field showing '01:22'. Below that, there's a button labeled 'Location option'.

Settings screen

Settings screen contains a checkbox
To enable notification at a specified time.

It will also have an option
to set a location via Play
Services, so the
notification will only be
shown in the set area.

Screen 5



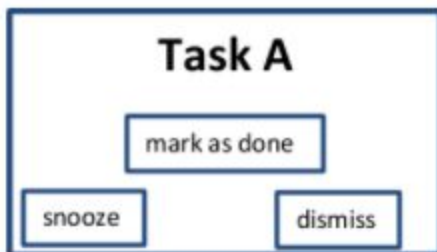
This screen shows the concluded tasks.

Tap to open another screen
to see the Task details

Delete button deletes
a concluded task.

Concluded tasks screen

Screen 6



Notification



widget

Key Considerations

How will your app handle data persistence?

Data persistence: local database for tasks; shared preferences for user settings.

Describe any edge or corner cases in the UX.

Back button does not save a Task when creation / editing; Only the submit button performs this action.

Describe any libraries you'll be using and share your reasoning for including them.

- Timber - logging tool.
- Schematic - to implements Content Providers easily.

Describe how you will implement Google Play Services or other external services.

- Ad service: FREE (with ads) and PAID (no ads) versions.
- Location: The possibility of indicating a location where notifications can be displayed (the user's residence, for example).

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

Target: to have the basics of running Android project: a blank screen.

- Creation of the GIT repository;
- Project creation in Android Studio;
- Basic configuration of gradle;
- Main Activity (empty).

Task 2: Implement UI for Each Activity and Fragment

Implementation of screens with basic functionalities.

- UI for MainActivity;
- UI for the Tasks editor;
- Database / Content Provider for data persistence;
 - [Implement with Schematic lib.](#)
- UI to display Tasks details;
- UI for completed task history.

[Some basic input validation shall be done before inserting them into the database.](#)

[Database operations shall be done in a background thread \(via AsyncTask or Loader or a Service \(to be defined\)\).](#)

Task 3: Implement Settings

- Settings screen implementation

Task 4: Implement Notification

- Daily notification implementations (without Location service).

Task 5: Implement Widget

- Widget implementation

Task 6: Implement Ad Service

- Ad service implementation;
- Separate between FREE and PAID versions.

Task 7: Implement Location Service

- Location Service implementation: notify only when the user is in the determined location area.

Task 8: Final task

- Signed APK;
- [Implement installRelease task in Gradle](#)

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
 - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"