
[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

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

[Describe any 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.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

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

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: [ShivamArora](#)

Note Maker

Description

With Note Maker, you can create notes containing text, organize them by tags and share notes to your friends and family via Social Networking Services.

You can use Note Maker to quickly capture what's on your mind and get a reminder for it later at the right time.

Intended User

The app is intended for any kind of user.

Features

- Create Notes containing text
- Organize notes by tags
- Share notes with friends and family
- Get a reminder for your notes.

User Interface Mocks

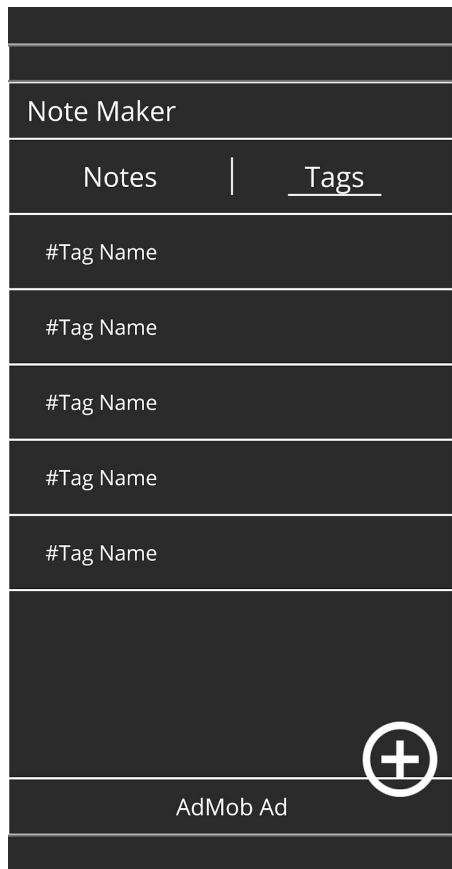
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



The launch screen for the app where you can access all your notes and tags. Click the + button to create a new note.

Screen 2



The second tab portrays list of tags. Clicking on a tag displays all the notes carrying the selected tag.

Screen 3



On selecting a tag you can view all the notes carrying the selected tag.

Screen 4

The mockup shows a mobile app screen for adding a new note. It features a dark theme with white text and icons. At the top, there's a header bar with a back arrow, the text 'Add new note', an alarm clock icon, and a checkmark. Below this is a large text input field for the title. Underneath the title field are three tag input fields, each containing the text 'Tag 1'. The main body of the screen is a large text area for the note content, with a placeholder text 'Write your note here...'. At the bottom, there's a dark bar with the text 'AdMob Ad'.

← Add new note ⌚ ✓

Title

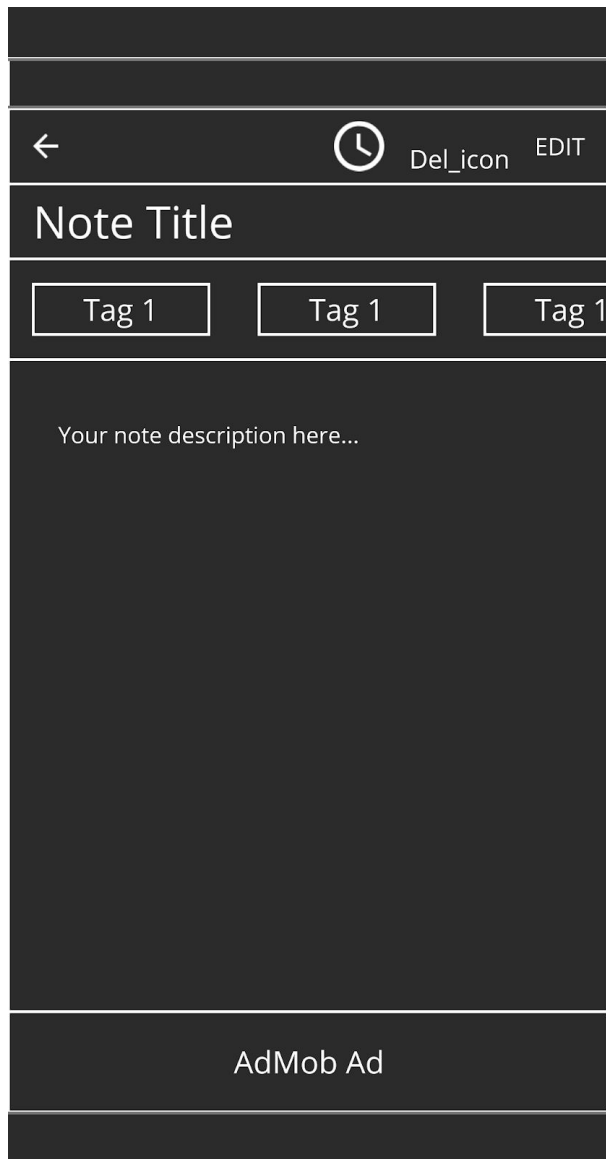
Tag 1 Tag 1 Tag 1

Write your note here...

AdMob Ad

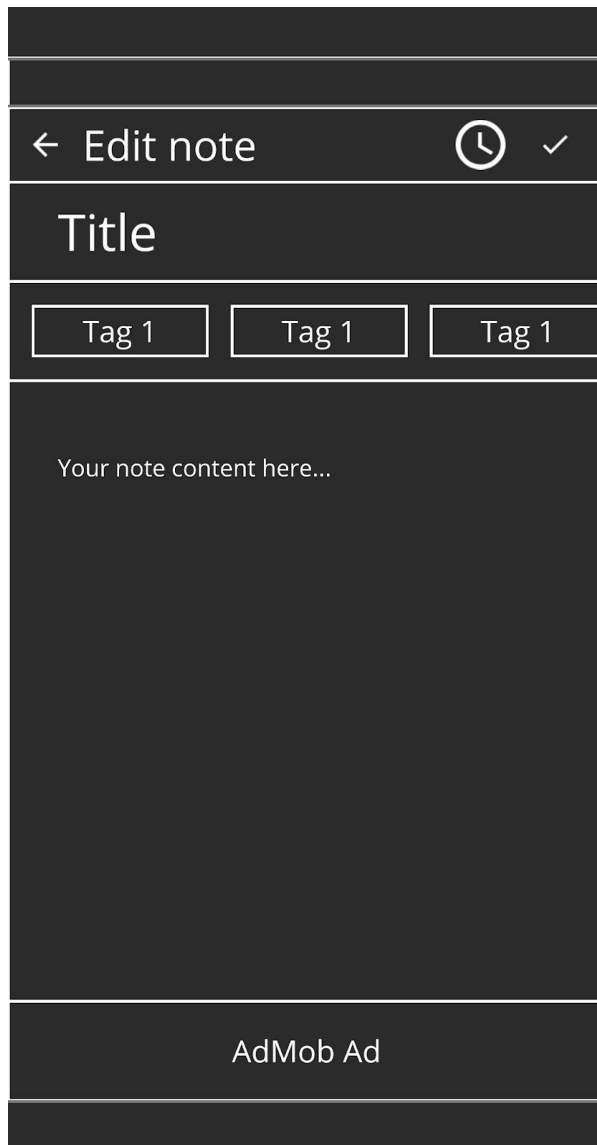
The add new note screen lets you add a title, tags and the note content. You can also set a reminder for the note by pressing the alarm button in the appbar.

Screen 5



This is how a note looks when it is viewed.
You can delete or edit a note via the appbar actions.

Screen 6



The mockup shows a mobile application screen for editing a note. At the top, there is a dark header bar with a back arrow, the text 'Edit note', a clock icon, and a checkmark icon. Below the header is a text input field labeled 'Title'. Underneath the title field are three rectangular buttons, each labeled 'Tag 1'. The main body of the screen is a large text area with the placeholder text 'Your note content here...'. At the bottom of the screen is a dark footer bar with the text 'AdMob Ad'.

This is how you can edit the note.

You can save the note by the tick mark action in the appbar.

Widget

Note content

Note content

Note content

Note content

Key Considerations

How will your app handle data persistence?

The app will handle data persistence by using the Room library in addition with LiveData and ViewModel.

Describe any edge or corner cases in the UX.

There will be no network usage.

On pressing back button without saving the note from the edit note screen, the user will get a confirmation dialog asking if you want to save the note or discard the changes.

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

I'll be using the Nachos library v1.1.1 for displaying Chips for the tags.
And the Firebase Job Dispatcher library v0.8.5 for handling reminders.

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

I'll be using AdMob, Firebase Analytics and Firebase Crashlytics for crash reporting services.
I'll be using v17.0.0 for all these libraries.

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

- Add the required libraries to gradle.
- Add firebase to the project.
- Will be using Gradle 4.6 with Gradle Android Plugin v3.2.1 and Android Studio 3.2.1
- Setup JDK before Android Studio as the project will be using Java

Task 2: Implement UI for Each Activity and Fragment

- Build UI for NotesAndTagsActivity
- Build UI for NotesListFragment
- Build UI for TagsListFragment
- Build UI for NotesByTagActivity
- Build UI for AddNewNoteActivity
- Build UI for View/EditNoteActivity

Task 3: Implement Room, LiveData and ViewModel

- Implement Room for Handling Database Operations
- Implement LiveData for handling data changes
- Implement ViewModel for handling orientation changes

Task 4: Implement Widget

- Create the widget layout
- Implement the WidgetProvider

Task 5: Implement AdMob

- Create an AdUnit on AdMob
- Add an AdView with the respective AppId and AdUnit to the activities

Task 6: Implement Analytics

- Enable Analytics in Firebase
- Add singleton instance for analytics in the project and use it to collect user analytics data.

Task 7: Implement Crashlytics

- Enable Crashlytics on Firebase
- Add and implement Crashlytics in the project

Task 8: Provide Localisation and Accessibility support

- The app will be providing Localisation support by using strings.xml file for all the strings in the project.
- The app will also provide support for accessibility by using content description attributes and support for D-pad navigation.

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**"