How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: apatil88

Make A Note

Description

Simple note taking app that allows you to take notes, store and retrieve them in an organized manner.
Key Features:

- Store notes on Google Drive or Dropbox
- Using your phone's camera, click and store photos with your note.

- Set reminders on your note so that you can get your tasks done at the right time.
- Compatible with phones and tablets.

Intended User

Make A Note is an Android application intended for anyone who likes to take notes in order to perform their tasks.

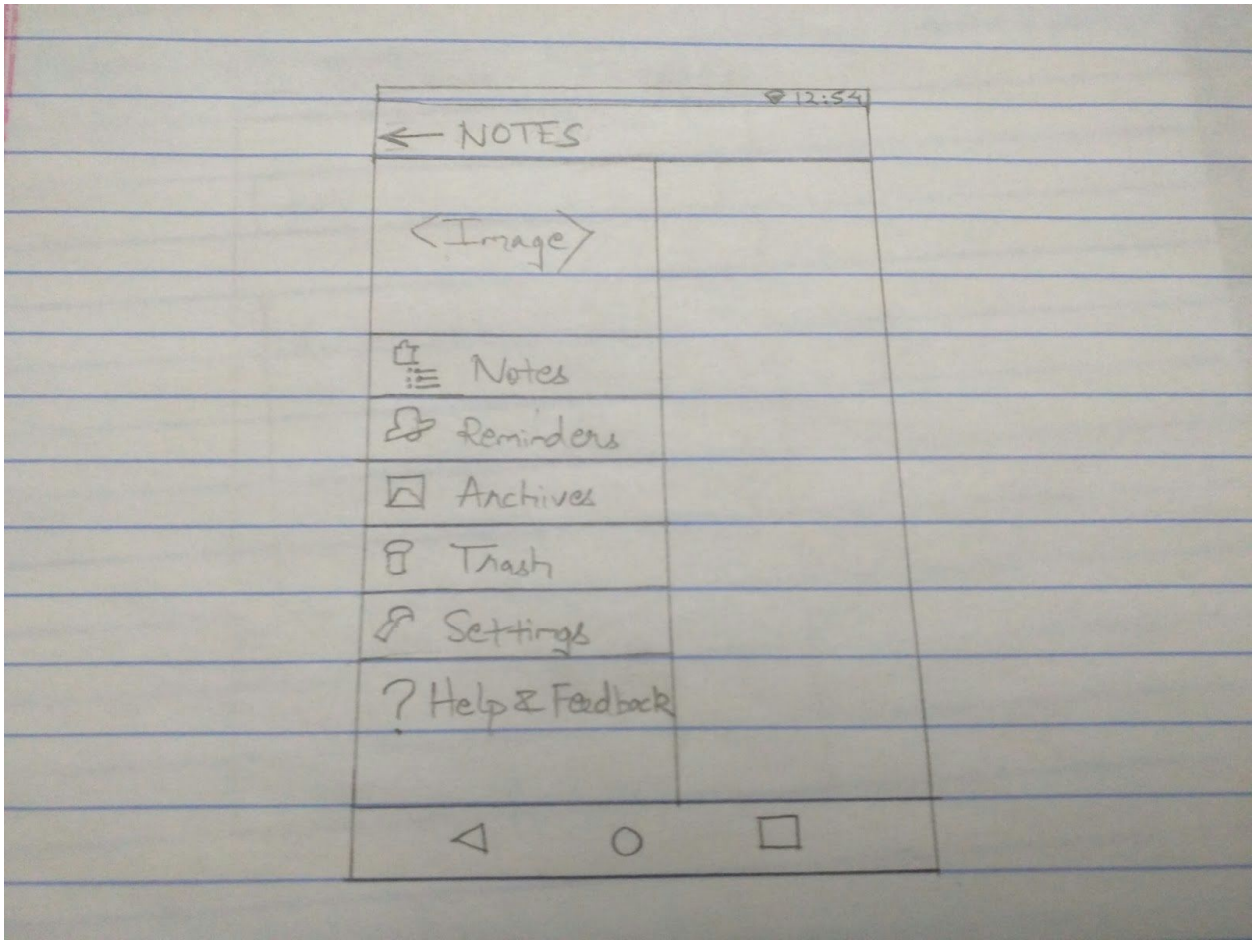
Features

The main features of Make A Note are as follows:

- This app allows you to save notes on the local device and Cloud platform services like Google Drive and Dropbox.
- This app allows you to take pictures and save it along with your notes.
- This app also allows you to set reminder notifications along with your notes.

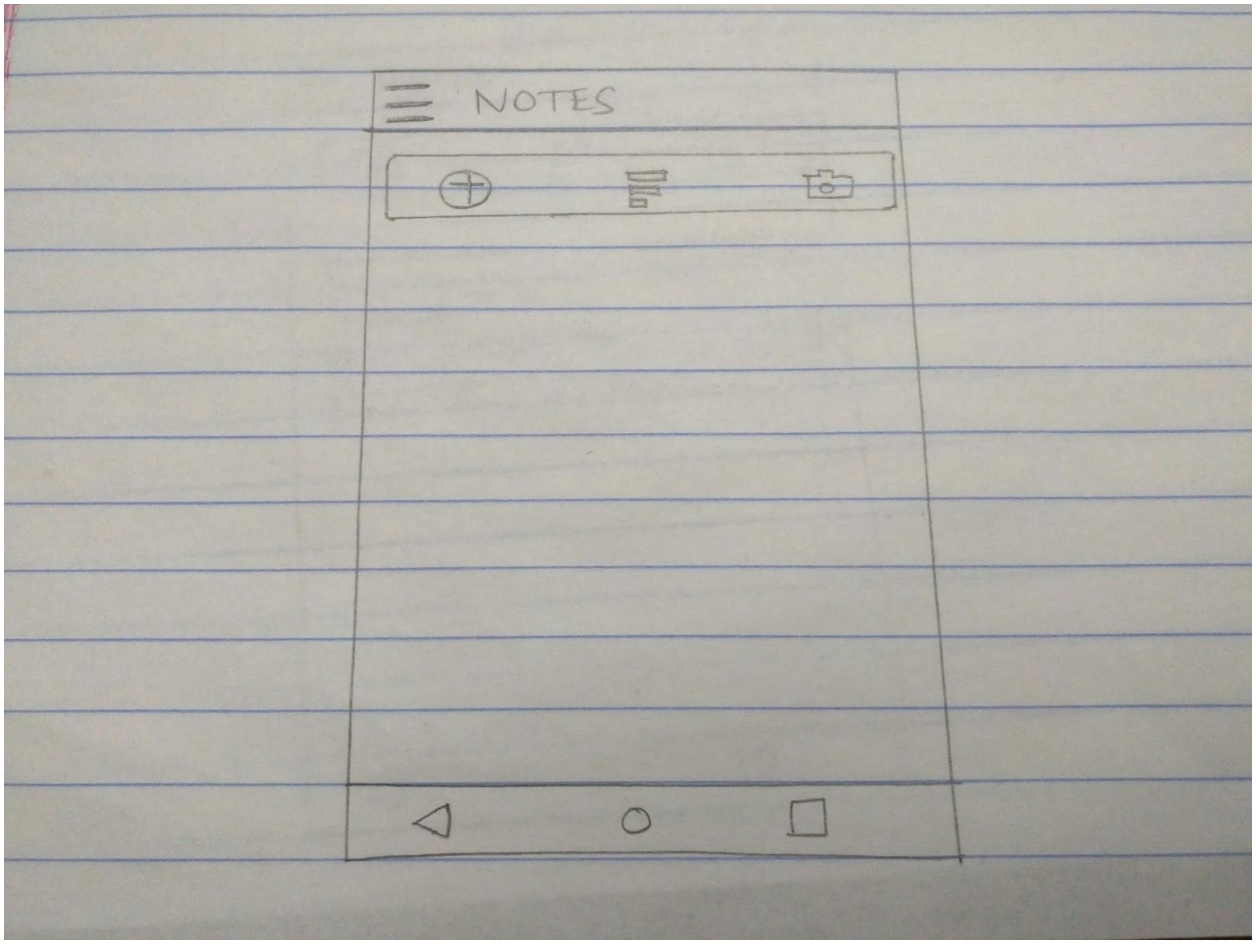
User Interface Mocks

Screen 1



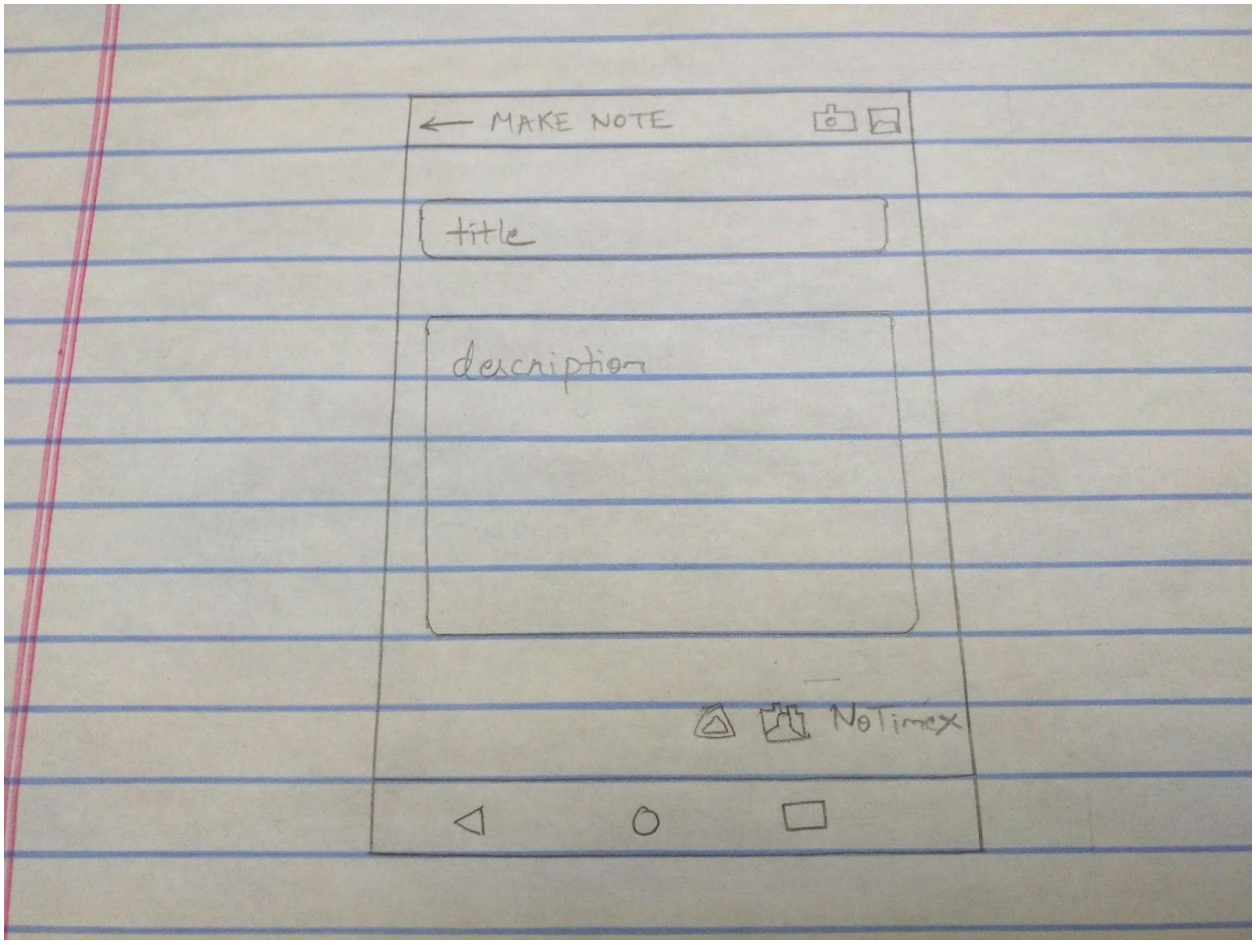
This screen allows users to navigate to a notes screen for taking notes. It also allows users to view their reminders, archived notes and notes the user moved to a trash.

Screen 2



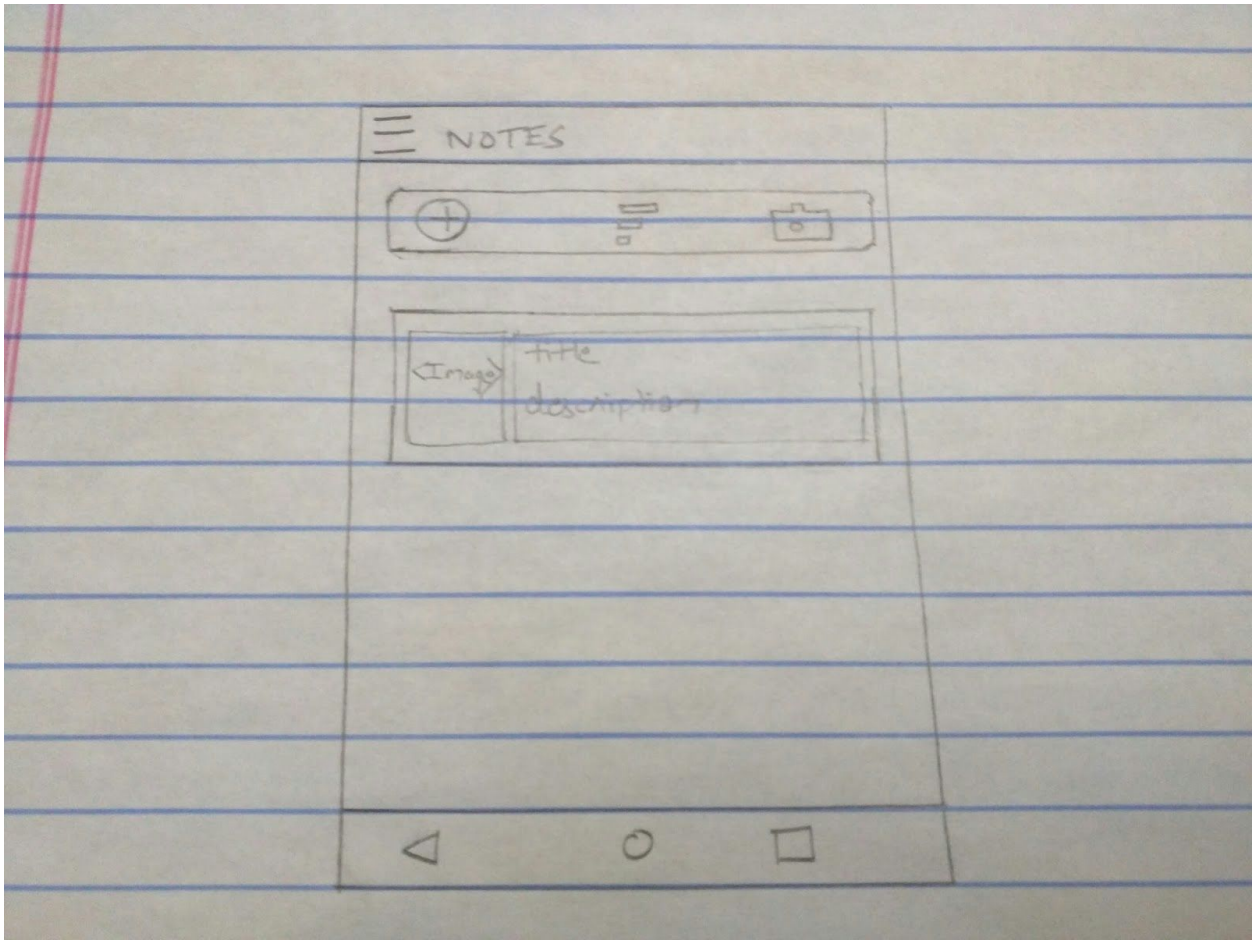
This screen allows users to create new notes. It also allows users to view archived notes. The user can use the phone's camera to save a photo along with the note.

Screen 3



This screen allows users to add contents to a note. The user can use the phone's camera to store a photo along with the note. The user can also store a photo from this photos gallery. Users can also store their note in Google Drive and Dropbox. Users can also set reminders on their note for a given date and time.

Screen 4



This screen displays all the notes along with their title and a brief description. Images (if any) attached with the notes are also displayed.

Key Considerations

How will your app handle data persistence?

The app will be using a Content Provider to store and access locally stored notes, archived notes and trashed notes. The app will also be using SharedPreferences so that the data remains in a consistent state. Also, the app will be interacting with the Google Drive API and Dropbox API for storing notes in the cloud.

Describe any corner cases in the UX.

The reminders for a note would appear as a notification using a receiver.
The user would return to the main screen containing the list of notes already stored whenever the user hits the back button.
The app will also have a navigation drawer which slides from the left of the screen.

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

Libraries used:

- Design Library- Material Design Ripple Transition, NavigationDrawer
- Google Play Services, namely, Google Drive API, to store notes, select and get the name of directories in Google Drive.
- Google Play Services, namely, AdMob for displaying test ads.
- Other libraries depending on feature requirements and user experience enhancements.

Next Steps: Required Tasks

Task 1: Project Setup

The project requires the following setup steps:

- Configure libraries
    - Google Play Services (Drive API and AdMob)
    - Google Design Library
- Configure Android Manifest file depending on which Android versions the app would run on. Currently, the app runs on Android 5.0 (Lollipop) or higher supported devices.

Note: This section might be updated as I go along developing the app.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Screen 1
- Build UI for Screen 2
- Build UI for Screen 3
- Build UI for Screen 4
- Building Other UI's as needed.

Task 3: Content Providers, Async Tasks and Loaders, Notifications

- For storing and retrieving notes.
- For archiving notes
- For deleted notes.
- Implement notifications for notes with reminders.

Task 4: API Interaction and Implement Google Play Services

- Add functionality to interact with Google Drive API.
- Add functionality to interact with Dropbox API.
- Add AdMob to the app.

Task 5: Handle Corner Cases

- Example: When internet connection is not available and the user wants to store an image on Google Drive or Dropbox.
- Other cases as discovered during development.

Task 6: Tablet Compatibility

- Add support for making the app compatible with Tablet UI
- Test on 7” and 10” tablets.

Task 7: Material Design

- Add Ripple Effect Transition.
- If needed, use other material design features that will enhance the user experience.

Task 8: Testing and Debugging

- Use JUnit to write basic unit tests.
- Debug and fix issues uncovered while testing.
- Run the app on different screen size devices.

Other tasks as needed.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"