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: srv-twry

Student Companion

Description

This is an all in one app that aims to serve as a companion for all the Undergraduate students and help make their day to day life easier with the help of certain automated features.

Intended User

Students

Features

The main features of the app are:

- Coding calendar
    1. Upcoming competitive coding contests
    2. Details of each contest
        - Platform
        - Date
        - Start time
        - End time
        - URL
        - Description
    3. Ability to mark favourites and set reminders
    4. Ability to share the contest link and description
- Flash cards
    1. Store flash cards with questions and answers
    2. Mark important to add to important's list
    3. Share a flash card
- Task reminder (Tentative i.e. if i complete the above features on time)
    1. Set reminders for a task
    2. Option to set weekly reminders
- PDF converter(Tentative i.e. if i complete the above features on time)
    1. Convert an image or set of images to a PDF
    2. Take images from the camera or SD card
    3. Save pdf file to sd card
    4. Share the pdf using e-mail or whatsapp.

NOTE: This will be an active project even after the project submission. The development process is divided into stages, I only wish to complete stage-1(Coding calendar, Flash cards) for the capstone project and hence i have only added the UI mockup of the first two modules plus one extra for Task Reminder(In case i am on schedule) . If i can implement the stage-1 features in time then i will go ahead and also implement PDF converter and Task Reminder(stage-2)

Stage-2 features :

- PDF converter
- Task Reminder

Stage-3 features:

- Monthly expenditure tracker
- Attendance tracker

Stage-4 features :

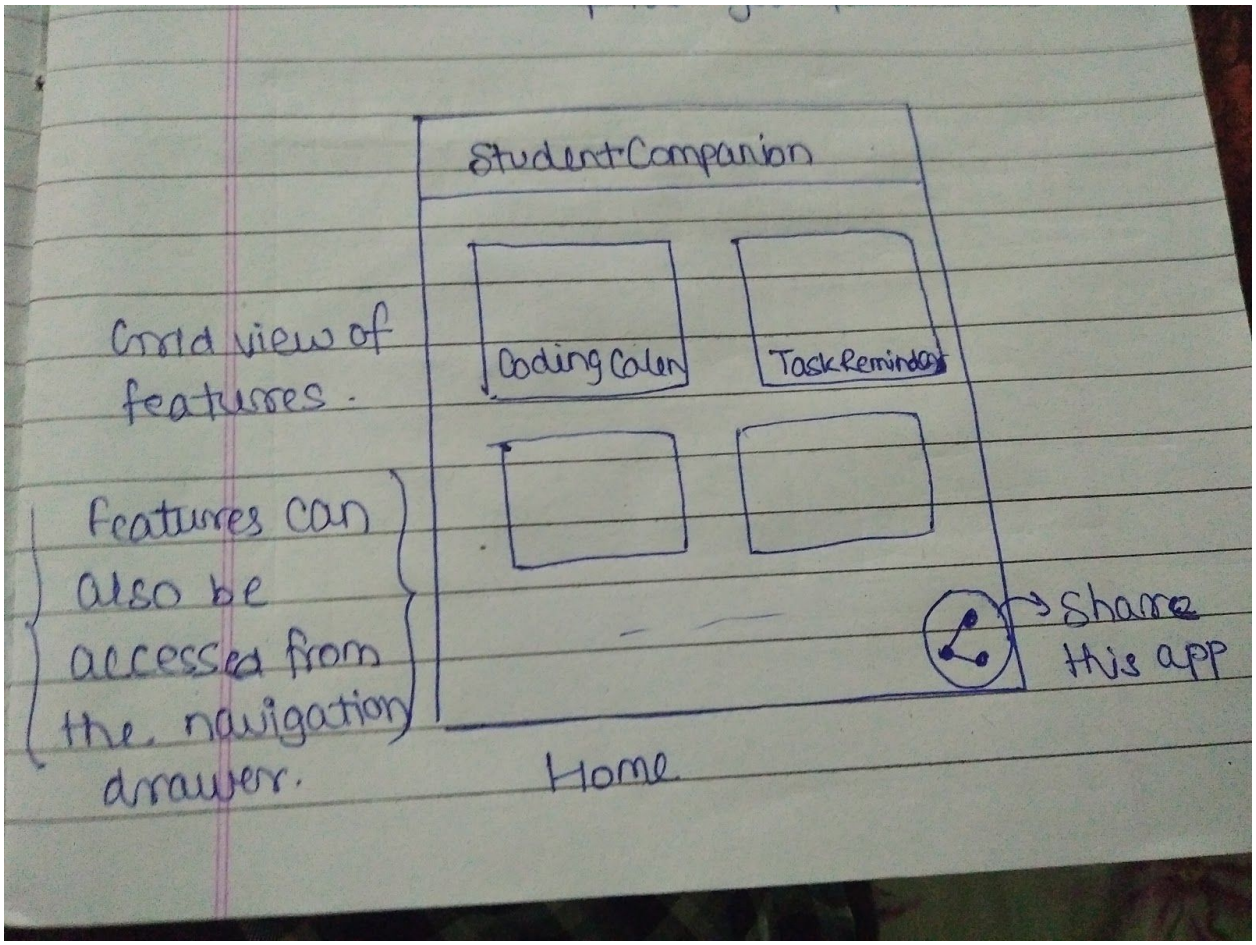
- Set the phone/tablet in silent/vibrate mode in the class hours.

Will keep on adding features as they come to my mind.

User Interface Mocks

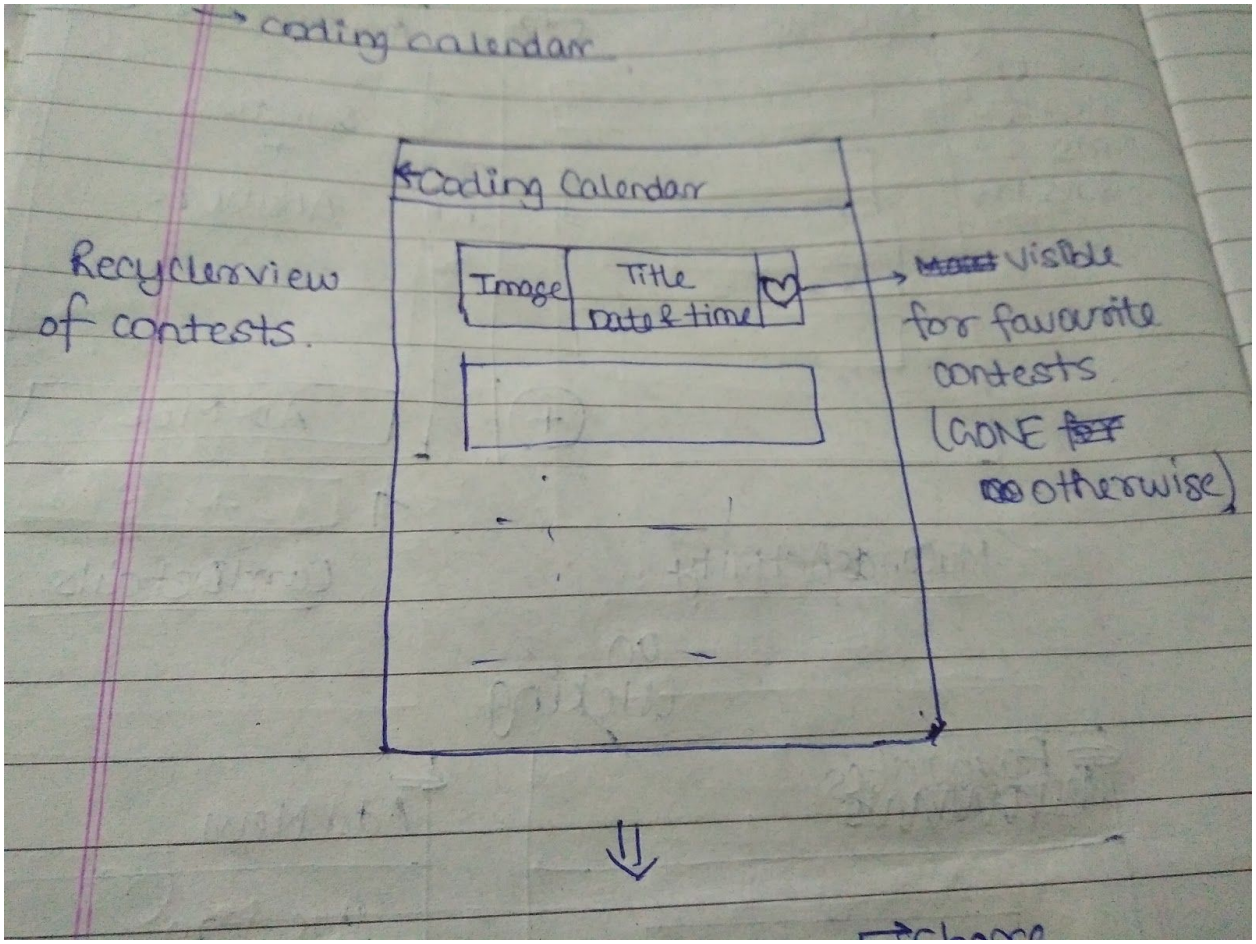
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Home Screen

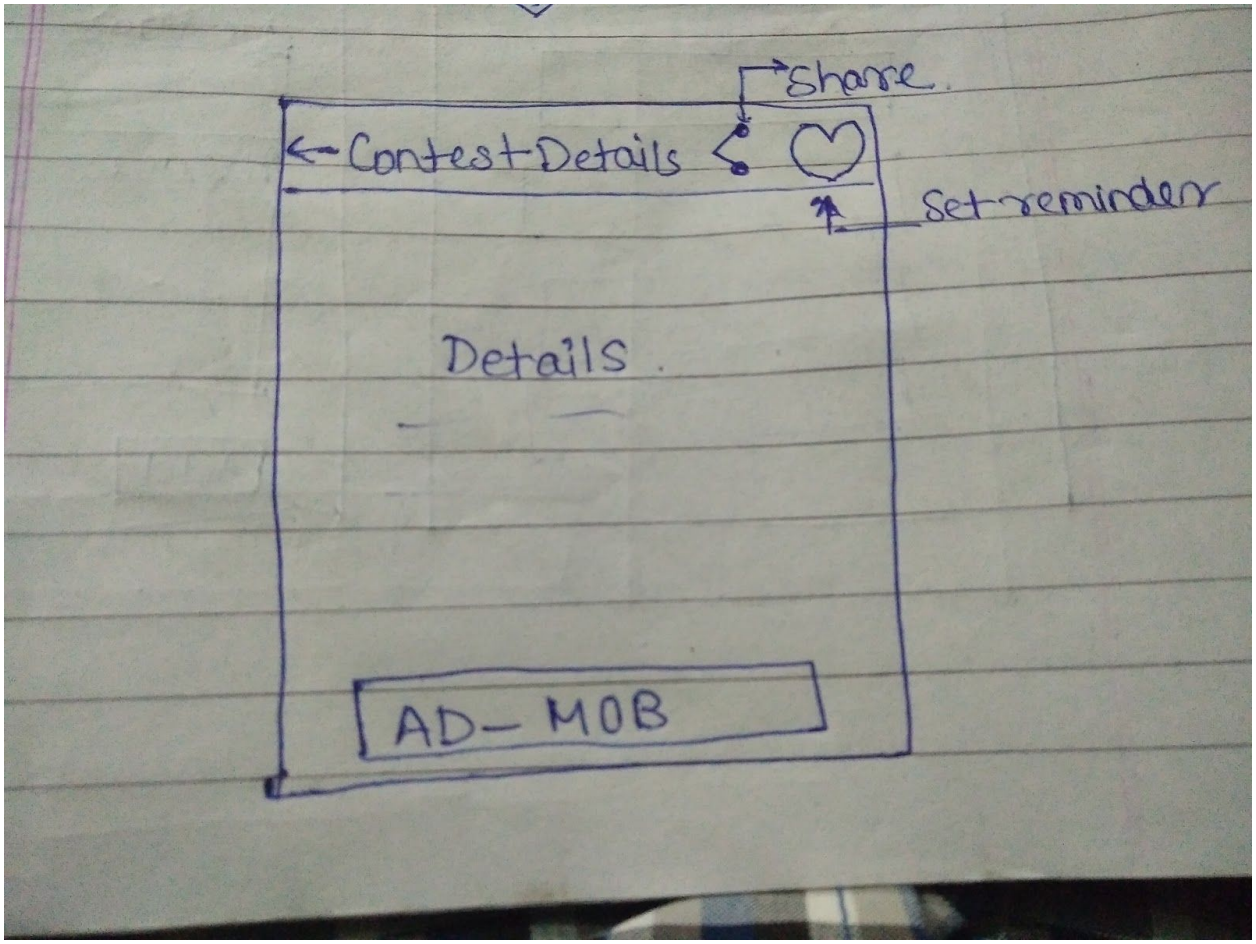


The Home screen for the App containing grid view of all the features available, Features can also be accessed using the navigation drawer.

Coding calendar contest list

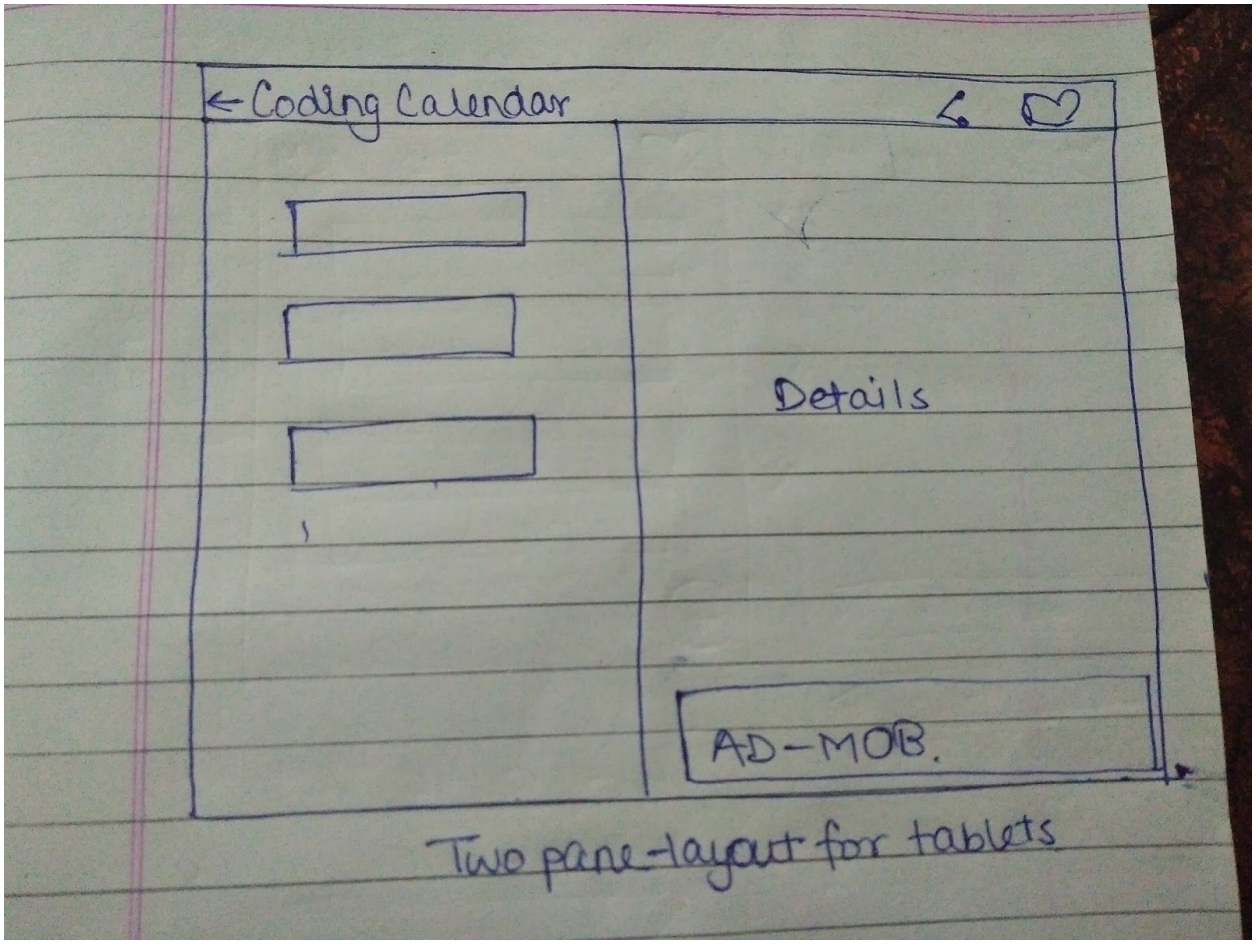


The coding calendar list activity that contains a fragment containing a recycler view of contests that are active or are upcoming. Clicking on one of them shows the details screen.



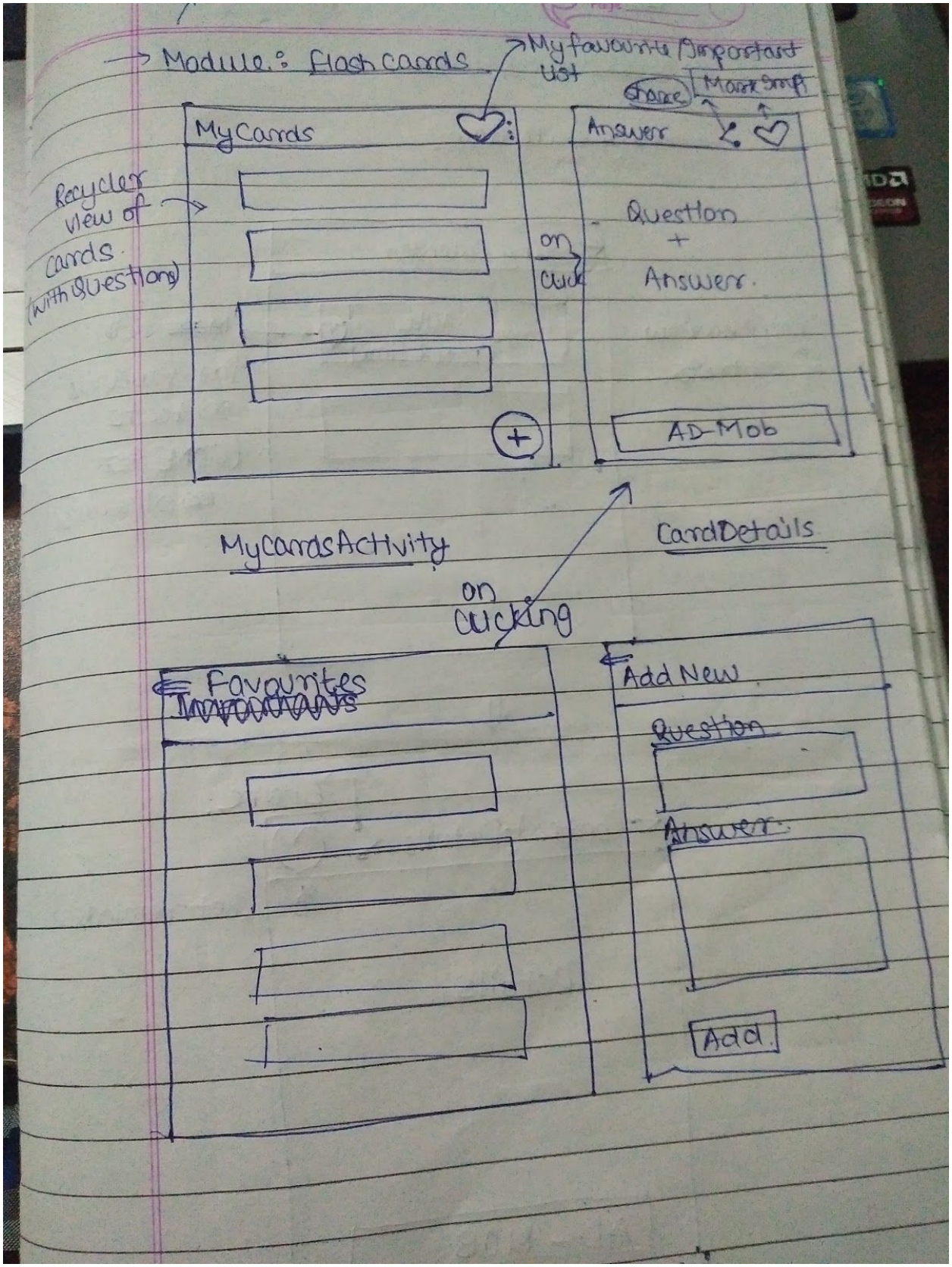
Coding calendar contest details

Activity showing the details screen for the selected contest with features to share and set reminders.



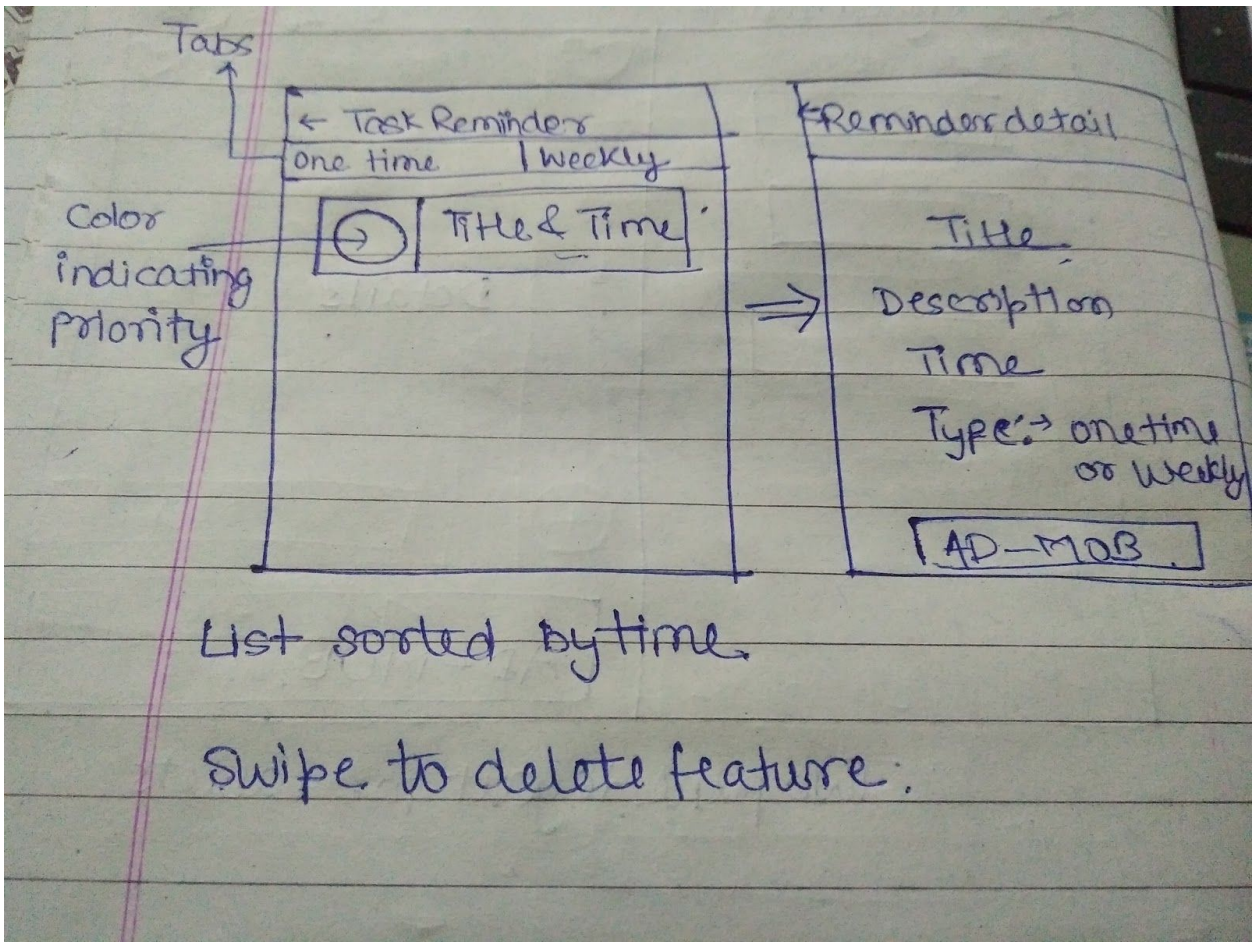
Coding calendar two pane

The two pane layout for the coding calendar for tablets.


Module: Flash cards
My favourite / Important list
Share
Mark Imp
MyCards
Answer
Recycler view of cards (with Question)
on click
Question + Answer.
AD-Mob
MyCardsActivity
CardDetails
on clicking
Favourites
Add New
Question
Answer
Add

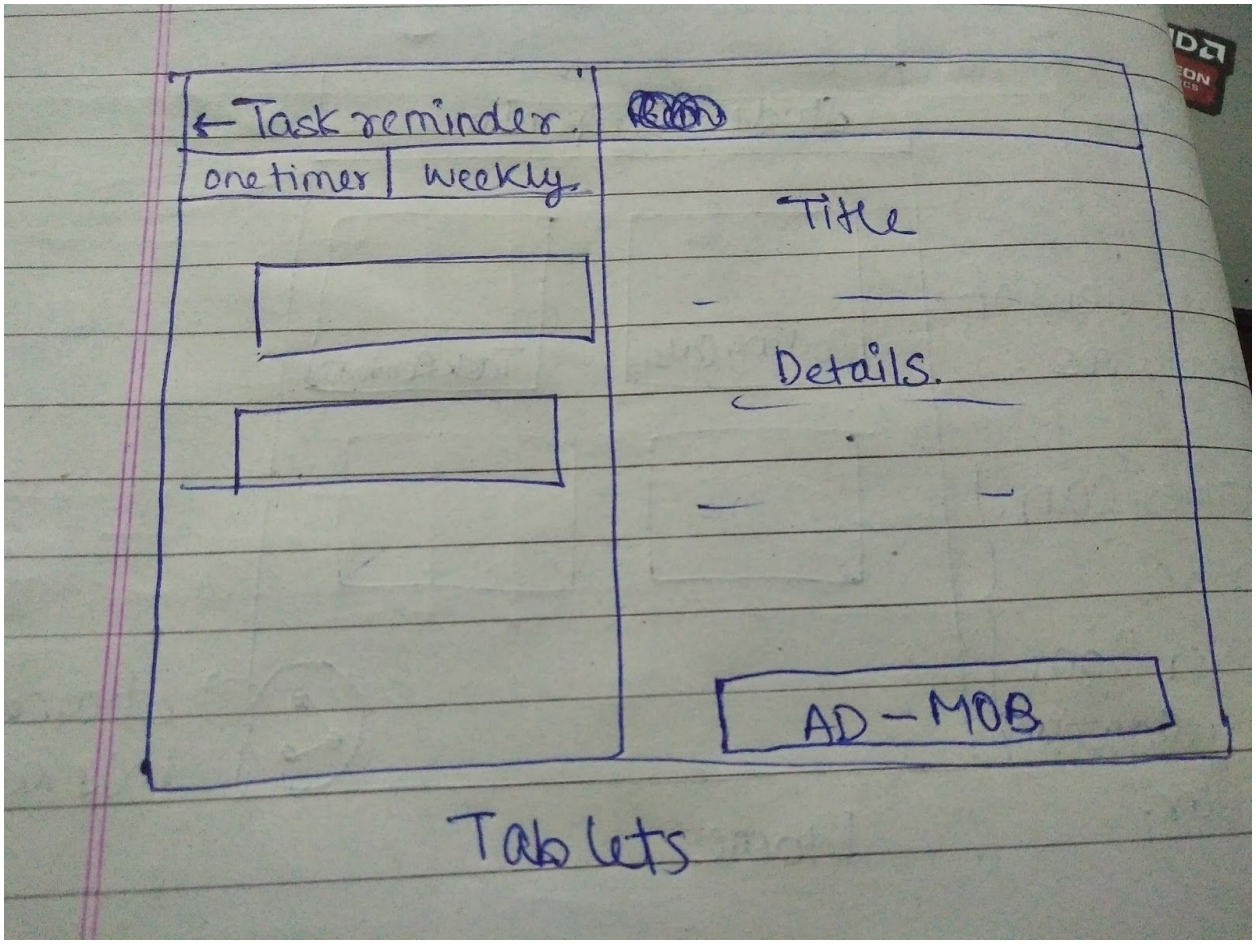
Flash cards layout

The layout for the flash cards screens, note that there is no two pane layout as it would show both answers and questions which would defeat the purpose of revision.



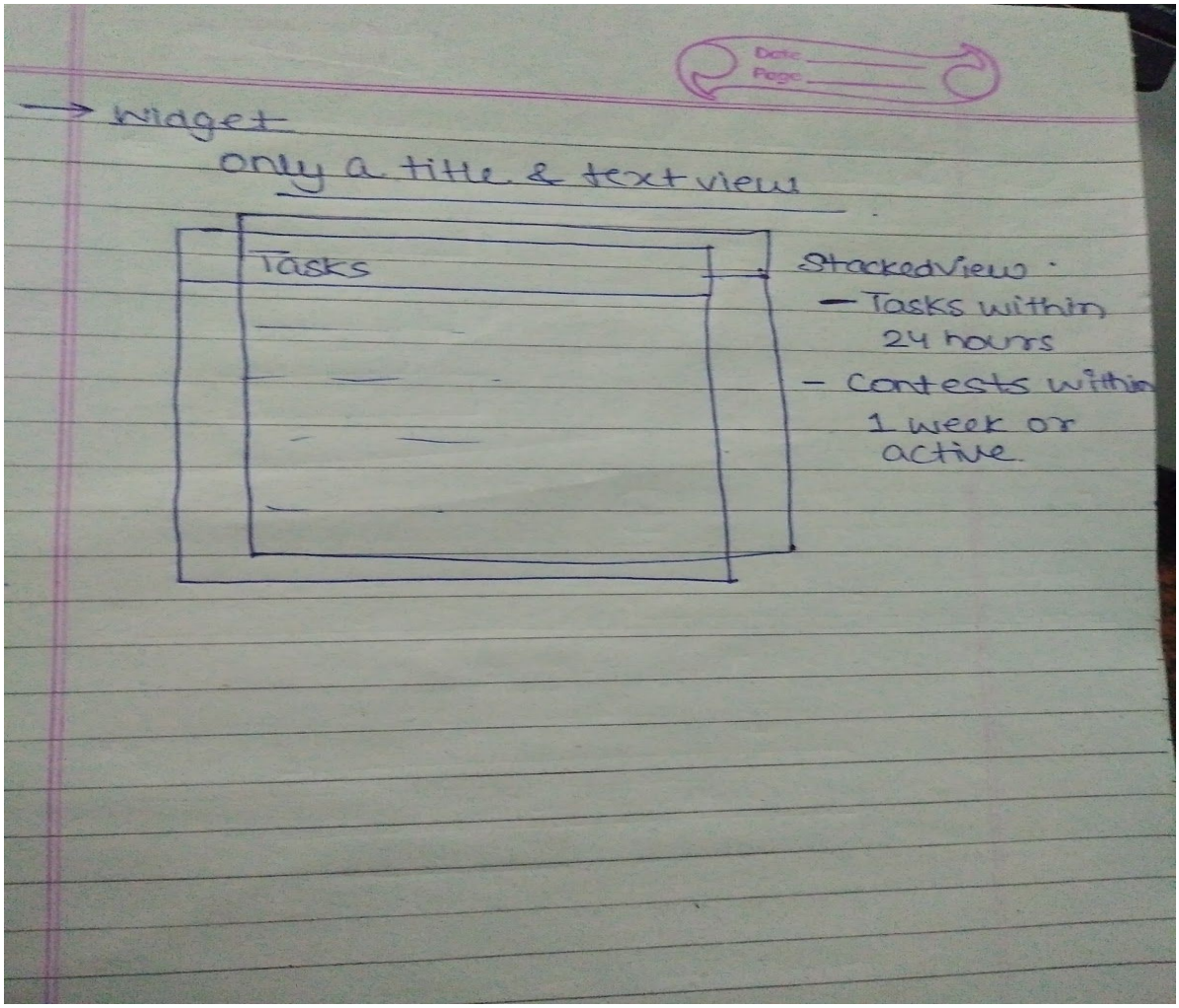
Task reminder

The layout for the task reminder, nothing fancy in here.
NOTE: I forgot the fab button at the right bottom to add a new task reminder ! It's ui will also be similar to the add flash card activity with minor changes.



Task reminder two pane

The two pane layout for the task reminder.



Widget layout
Stack view of two screens , one containing the tasks to do within the 24 hours and other containing contests that are active or upcoming in next 1 week.
The layout is simple and contains a title and a text view showing all the necessary data as a single string.

Key Considerations

How will your app handle data persistence?

I will build a content provider to handle database operations as their will be multiple tables in the database.

Describe any corner cases in the UX.

None

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

Butterknife - Bind the views
Volley for the background data loading
Firebase job dispatcher for background tasks
Hackerrank's api for the raw json data

Describe how you will implement Google Play Services.

Ad-Mob service to show ads
Analytics to get an idea about the modules of the app that are popular.

Next Steps: Required Tasks

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

I will work module by module.

Task 1: Project Setup

I will setup my project , its local git repository and library imports for the coding calendar module

Task 2: Implement UI for Each Activity and Fragment of Coding Calendar and set up HomeActivity(2-3 days)

These are the steps in detail:

- Build UI for HomeActivity
- Build Functionality of the HomeActivity i.e. the gridview and its adapter.
- Implement small features like share intent for the App in the HomeActivity.
- Build UI for all the fragments and activities associated with the Coding calendar.

Task 3: Set up the Coding Calendar Functionality(10-12 days)

I will setup the basic functionality of the coding calendar in the following order:

- Setup the POJO for contest.
- Setup contest recyclerview with fake data
- Clicking on a contest should open its details, set up its details activity.
- Use volley to get the actual data from the Hackerrank API and use it for the recyclerview.
- Write the SQlite database table for contest recyclerview and also write its content providers.
- Modify the recyclerview to get the data from the database using Loaders and content providers rather than the network call.
- Setup sync adapter to refresh content for the database every hour if network is available.
- Implement the add reminder facility using firebase job dispatcher.
- Implement minor features like sharing features and google admob, analytics.

Task 4: Add two pane layout for Coding Calendar(1-2 days)

The aim of this task will be to implement the two pane layout for the tablet.

Task 5: Set up Flash Cards UI(1-2 days)

I will implement the UI for the flash-cards module and design layouts for the following:

- My cards activity
- Answer activity
- Favourites activity
- AddNewCardActivity

Task 6: Set up Flash Cards functionality(7-8 days)

- Set up the recyclerview with fake cards
- Add the table in the database for the flash cards
- Modify the content provider to implement the CRUD operations for the same.
- Modify the recyclerview to fetch data from the database now.
- Set up the AddNewCardActivity
- Implement the favourites, sharing , admob and analytics.

Task 7: Set up Task Reminder UI(2-3 days)

I will implement the UI for the task reminder module:

- Create layout for activities and the fragments.
- Implement basic viewpager tabs with fake data for now.

Task 8: Set up Task Reminder functionality(7-8 days)

- Set up the two recycler views with fake data in the viewpager tabs
- Set up the reminder detail screen
- Set up the database and content provider
- Set up the AddReminderActivity and schedule the reminder using firebase job dispatcher.
- Set up the notification and its functionality for weekly repeating tasks.
- Implement admob and analytics.

Task 9: Set up two pane layout for the task reminder(1-2 days)

Title is enough to describe the action.

Task 10: Implement Material design (1-2 days)

- Choose color scheme
- Choose suitable font
- Implement shared element transitions if any.

Task 11: Add widget (1-2 days)

- Implement the widget as proposed in the UI mockup

Task 12:Final touch

Giving final touch before submission. For example:

- Removing hardcoded string
- Optimize with Lint , testing on variety of devices etc.

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**"