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: DhruvamSharma

MyCheckList

Description

This application helps you to create a list of todos, short-term goals and long-term goals in an android application in a nice and intuitive manner.

Intended User

This application is for a general user who has needs and wants to fulfill their short-term and long-term goals.

Features

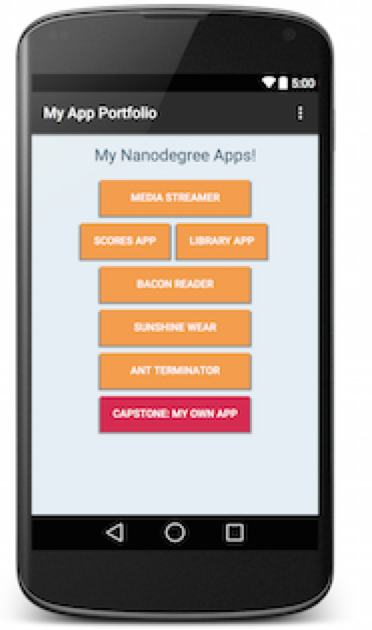
The mainly focusses on these priorities:

- Saves goals and todos.
- Persists the goals and todos.
- Maintains separate list for multiple tasks.

User Interface Mocks

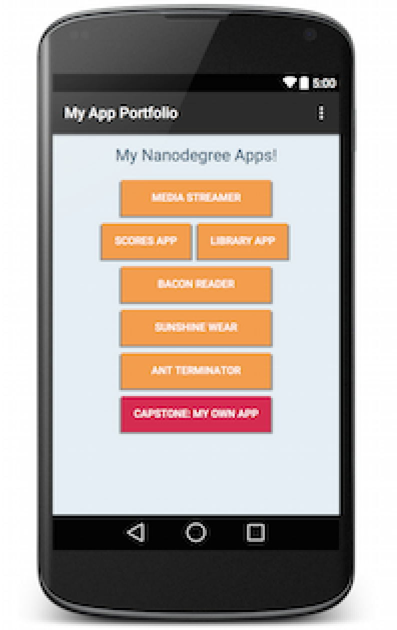
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
Provide descriptive text for each screen

Screen 2



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
Provide descriptive text for each screen

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

This application will handle data persistence through Room. I will use this dependency :

```
def room_version = "1.1.1"

implementation "android.arch.persistence.room:runtime:$room_version"

annotationProcessor "android.arch.persistence.room:compiler:$room_version”
```

Describe any edge or corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

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

In this project, I will be using various libraries,

1. Glide: For image loading and caching.
2. Room: For Data persistence.
3. ViewModel: For separating data from the controller and for maintaining the state.
4. LiveData: For getting the data from the Room in real time.
5. Google Crashlytics: For proper crash analytics of the app after it has been released.
6. Admob: For monetizing the application on the play store.
7. RecyclerView: To view the list efficiently
8. AppCompat: For adding backward compatibility while building activities.
9. ConstraintLayout: For building efficient layouts.
10. CardView: For building nice UI.

Library Used	**Version**
Glide	4.8.0
Room	2.1.0
ViewModel	2.0.0
LiveData	2.0.0
CrashLytics	2.9.5
Firebase Admob	17.0.0
RecyclerView	28.0.0
AppCompat	28.0.0
ConstraintLayout	1.1.2
CardView	28.0.0

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

I will be using Firebase Admob and Crashlytics.

- I will be using Admob for monetizing the application.
- And I will be using Crashlytics to see and analyze the crashes in the application.

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

The first task is to set up the project. The steps would be:

- Create a project using “New Project” Wizard in the Android Studio.
- Configure the required libraries
- Create a Github Repository.
- Add a Readme.md file.

Task 2: Implement UI for Each Activity and Fragment

This task includes building a minimal UI required for making a functional application:

- Build UI for Main Activity. Which will present the tasks for a corresponding list.
- Build a BottomSheet for the Main Activity that will help toggle the lists.
- Constructing a View Model and hooking it up to the main activity.
- Building a login and signup screen.
- Building a splashscreen. [Optional]

Task 3: Building Data Models for the database

This task include building the various tables required for the application.

This application will include 3 tables, which are:

- User Table (It will store the list of users and their corresponding details).
- List Table (This table will store the various lists created by all the users).
- Tasks Table (This table will store all the tasks of the user corresponding to a list).

Task 4: Adding database to the application

After the database and the corresponding tables have been created, I will wire the database into the application through the Room persistence library.

This task will include:

- Creating a list.
- Editing a list.
- Deleting a list.
- Creating a task.
- Editing a task.
- Deleting a task.
- Querying all the tasks.
- Querying all the lists.

Task 5: Implementing Google Services

This task will involve adding the Google services to the application.

The google services I will be adding are:

- Admob,
- Crashlytics.

Task 6: Modifying the UI

This task will involve updating the UI to meet the material design specifications and making the application feel more beautiful.