GitHub Username: DhruvamSharma

Goals

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 fulfil their short-term and long-term goals.

Features

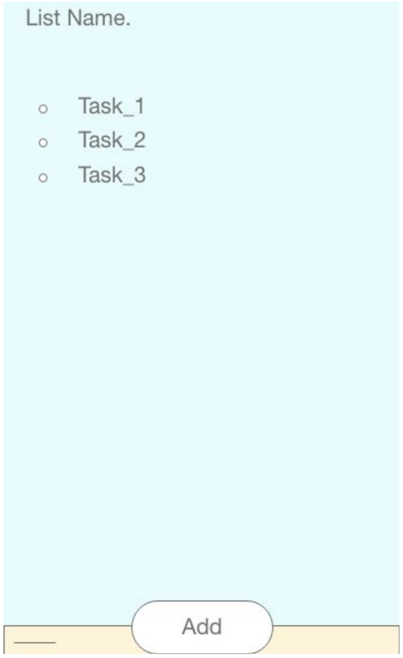
The mainly focusses on these priorities:

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

User Interface Mocks

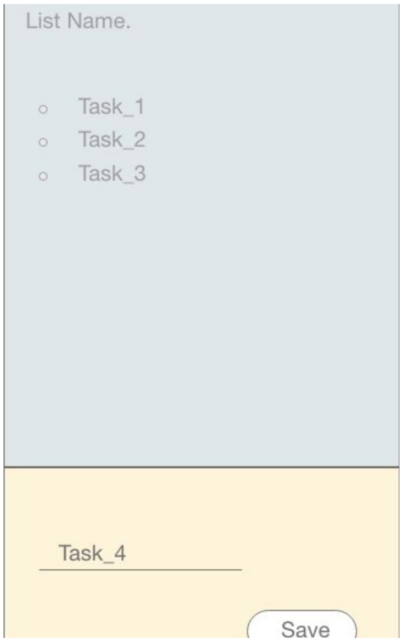
The application has many screens but these 5 will be the most important ones needed in the application.

Screen 1



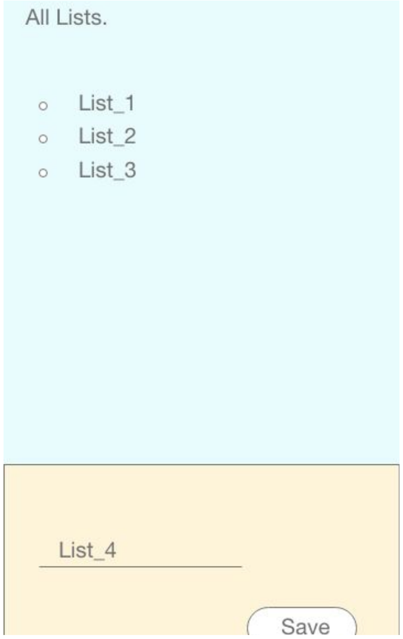
This is the main screen that shows all the tasks within a list created by the user.

Screen 2



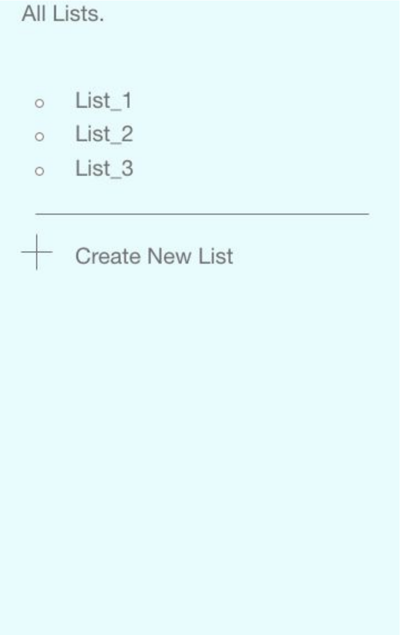
This is the main screen that allows the user to create a new task.

Screen 3



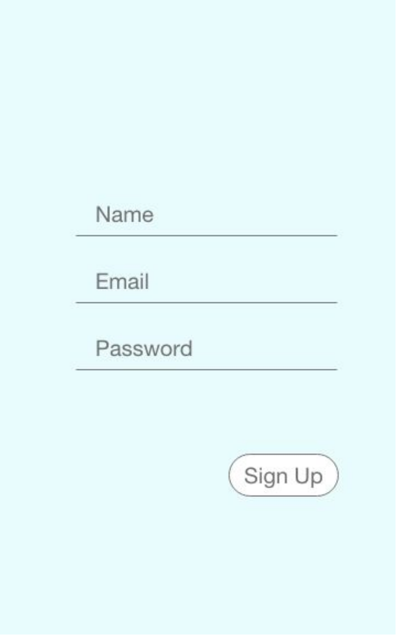
This is the screen to create a new list.

Screen 4



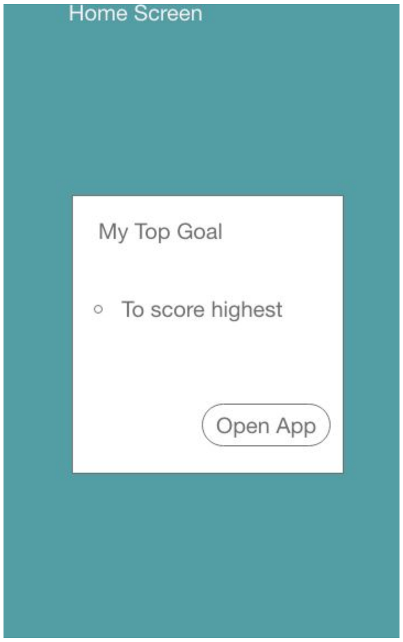
This activity shows the all the list available to the user and an option to create more.

Screen 5



This is the sign-up activity that allows the user to sign in into the application

Screen 6



This is the widget that stores the latest goal created.

Key Considerations

- I will be using the stable releases for all the dependencies and for Android Studio and Gradle.
- The application will be solely written in Java.
- The application will support RTL layout switching.
- All the resources that will be used in the application like strings will be in the res/value/strings folder, all the drawable resources will be in the res/drawable folder, all the dimensions used will be in the res/value/dimens folder.
  There will be no hardcoded value.
- For app accessibility, all the images will have a content description and all talk-back feature [Optional].
- The application will be using an Async Task for retrieving any network calls required in this application.

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.

The application is fairly simple and does not involve many complications. The only case would be while editing a task or a list, when the user presses the back button, will the data be saved automatically or there should be a save 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

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 splash screen. [Optional]

Task 3: Building Data Models for the database

This task includes 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: Adding a widget to the application

After the database and the corresponding tables have been set up and wired into the application, I will work on adding a widget so that users can save the goals on to the screen so that they can refer to the goal as and when required and without opening application.

Task 6: Implementing Google Services

This task will involve adding Google services to the application.

The google services I will be adding are:

- Admob,

- Crashlytics.

Task 7: Modifying the UI

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

- Creating beautiful UI
- Creating Backgrounds
- Creating icons
- Adding animations.