
GitHub Username: nehas510

Apps don't Lie

Description

Smart devices have made our life easy as well as lazy. There are zillions of works out in the market to become fit and healthy, we follow it but not able to see the results and progress and easily get demotivated. We try to convince our own self and the people around us that "IT'S NOT WORKING". But who knows how seriously and religiously you were following the plan? Why not let your phone tell you and your friends about your progress. This simple app will basically count the steps you made throughout the day. And next morning you get to know where so you stand among your friends. And this way you can compare and motivate each other every morning. Remember you may lie to your friends and yourself but "APPS DON'T LIE"!!!

Intended User

Anyone who is always making an excuse of not working out because "I don't have a companion to motivate me."

Features

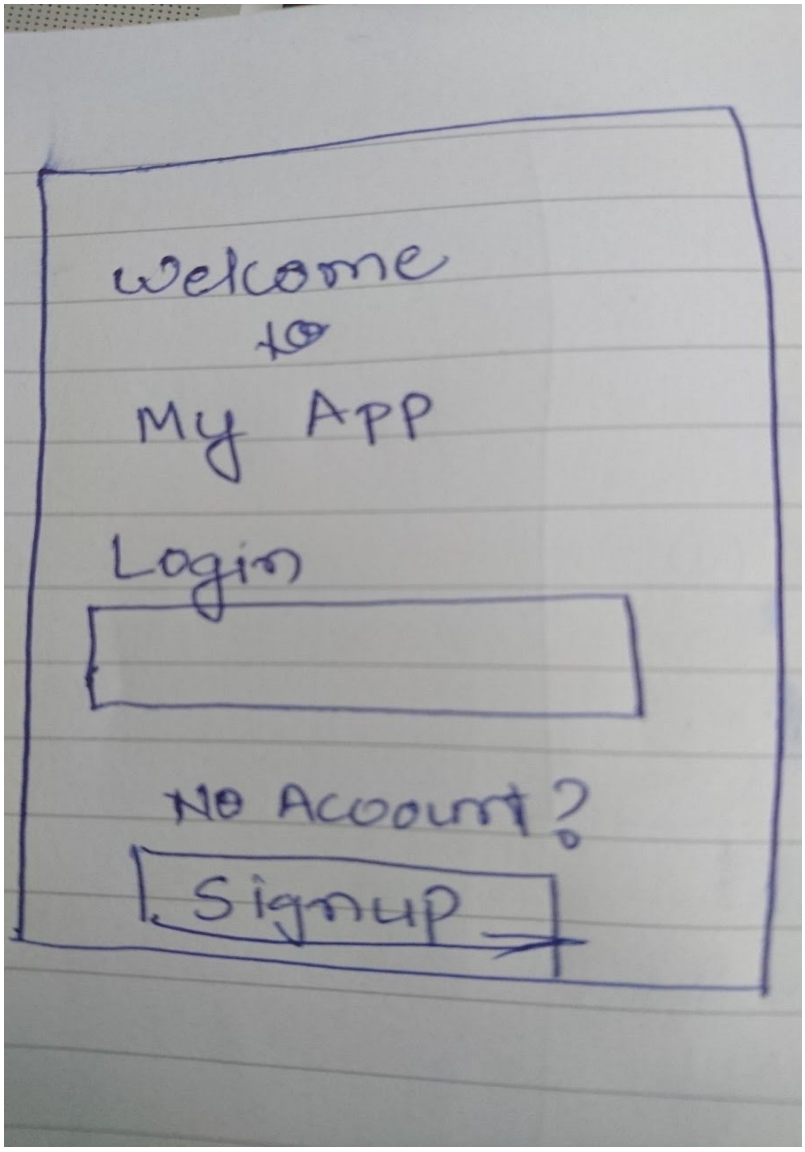
List of the main features of your app.

- Saves the steps user made throughout the day and sends this information to the firebase database. The server will send a push notification every morning to the user to show where he/she stands in the group of friends. The user can also share this data on the social networking sites.
- This app asks the user to take a new picture of his/her own self every month and show the before and after picture and the end of the month. The user can also share this on the social networking sites.

User Interface Mocks

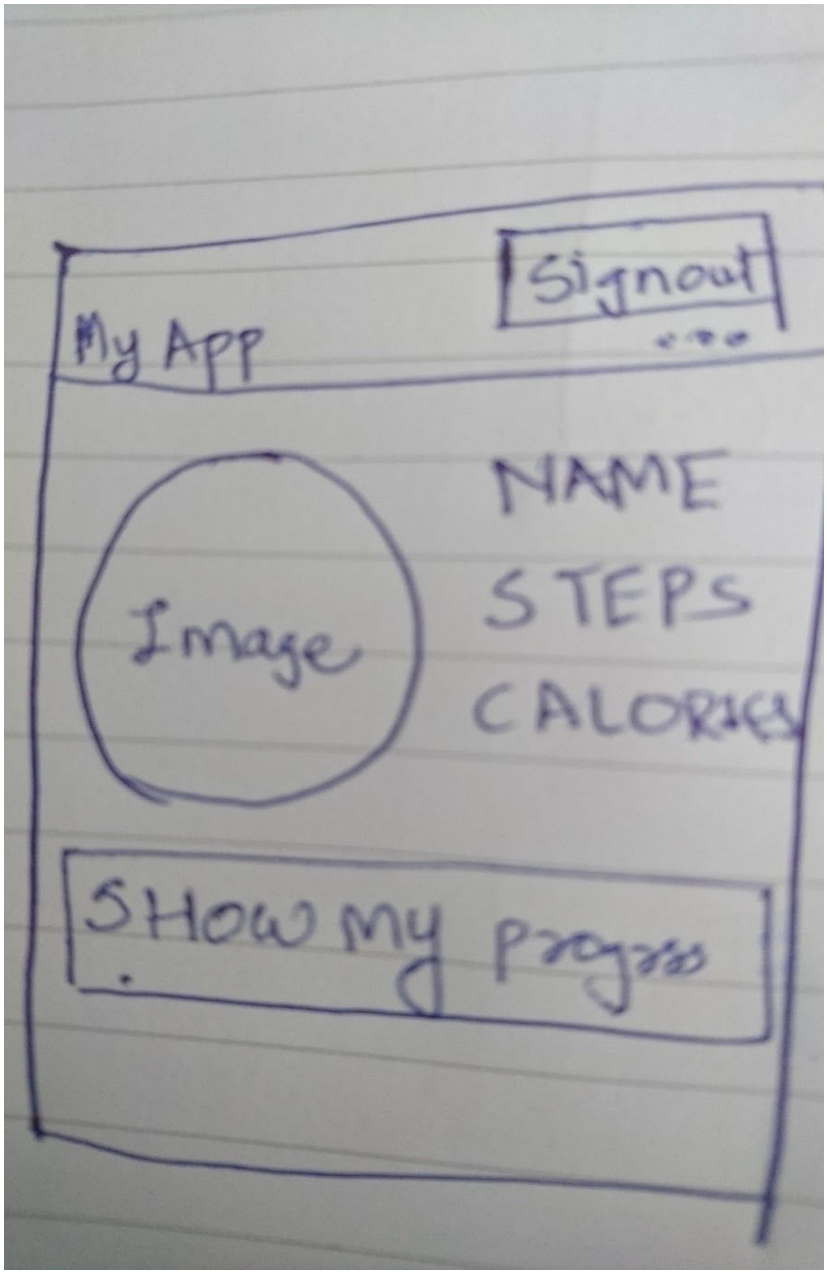
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



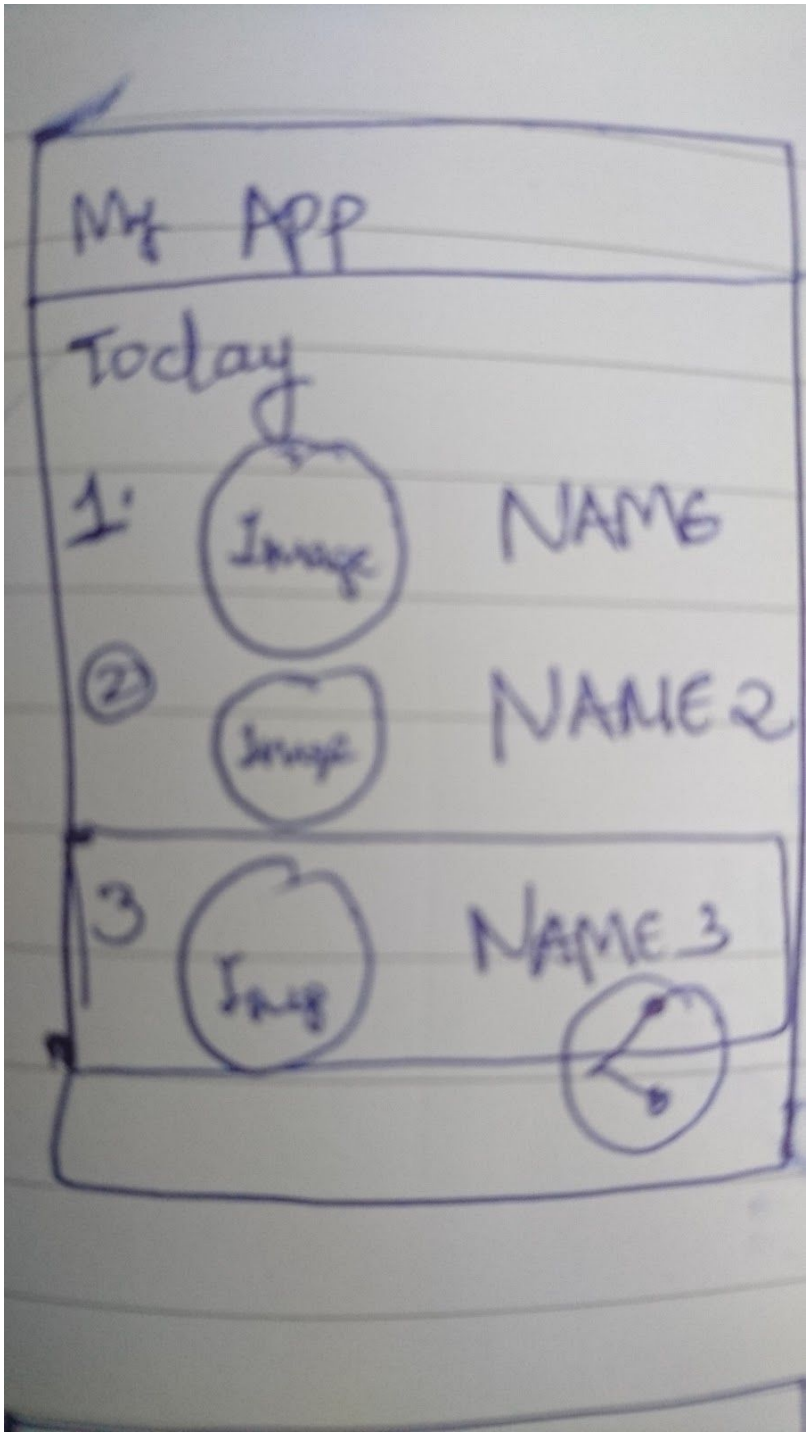
Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
This is the startup screen, where the user is asked to login or if he/she doesn't have an account he can signup.

Screen 2



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
This is the second screen, once the user logs into the app, he/she will be able to see the step count and calories. This is a dashboard which provides the basic info of the user.

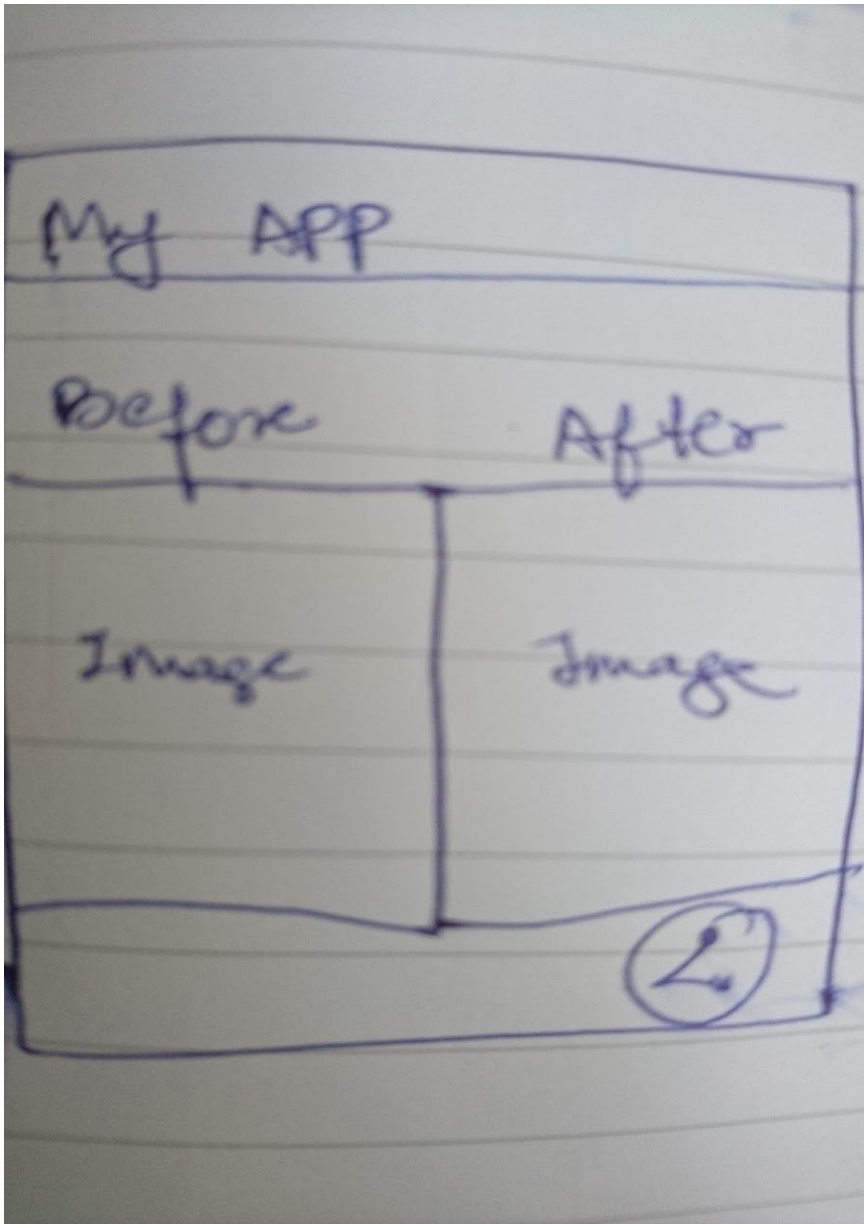
Screen 3



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]

As soon as the user clicks on the show my progress, this activity will launch showing the user what is his/her rank among all the participants. User can also share this details with a click of the button.

Screen 4



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
As User click on his/her name in the list, this activity launches and it will show the before and after picture of the user. He can share this detail with a click of the button.

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

Key Considerations

How will your app handle data persistence?

Describe how your app with handle data. (For example, will you build a Content Provider or connect to an existing one?)

Describe any 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.

For example, Picasso or Glide to handle the loading and caching of images.

Describe how you will implement Google Play Services.

Describe which Google Play Services you will use and how.

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.

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:

- Will start with creating a firebase database using firebase console.
- Next will start creating the login screen.
- Implement the logic to count the steps of a user using Motion control

- Do some math calculating the calories by analysing the number of steps.
- Show the data to the Dashboard of the user (screen 2)
- Reading the steps count of each user and setting it in the Descending order with the user details using the recycler view. User will get a push notification every morning about the previous day rank.(Screen 3)
- Requesting the user via a push notification every month to add the current picture, Showing the user the old and new pictures with Before and After caption. Also the user will be able to share this to the social media sites. (Screen 4).

Task 2: Implement UI for Each Activity and Fragment

Creating all the xmls.

- Build UI for Login Screen
- Build UI for Main activity (Dashboard)
- Build UI for the list of the participants ordered in highest steps count in Descending order, will use the recycler view for this.
- Build a simple Framelayout showing the before and After images

Task 3: Creating POJO

Describe the next task. List the subtasks. For example:

- Creating POJO classes in the project which will have the detail of individual user.

Task 4: Implement the step count logic

- Create logic to count steps of the motion sensor etc.
- Calculate the calories burnt on the basis of steps taken a day.
- At the end of the day say at 12:00 AM the data (step counts) will be pushed to Firebase.
- Next morning the user will receive the push notification about the progress.

Task 5: Implement the before and After photo logic

- Assuming that user has uploaded the current picture and in firebase database there will be 2 keys “photo_old” and “photo_new”. Every start of the month the user will get a push notification to upload the current picture.
- The image in photo_new will move to photo_old and the current uploaded picture will be added to “photo_new”. And both the pictures will be fetched in the screen 4 (will try to merge both and make it one so that the user can easily share only this image

Add as many tasks as you need to complete your app.

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**"