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: NikhilBhutani

Utilio

Description

As the number of apps are growing on our handheld devices every day, the data consumption is increasing. It becomes very tedious to take note of data used by apps, phone usage and all.

Utilio comes to rescue, which will do the heavy lifting by rendering insights about phone usage, app usage and all the data used by apps.
App displays the demographics with help of Graphs with the immersive UI and UX.

This app also keep track of your Activities like Walking time, Running time, Travelling time and even counting your steps by connecting to Google Fit.

Intended User

As it is general productivity hack hence can be used by anyone who uses an Android handheld device.

Features

Main features of my app:

- Monitors application data usage for every app in your device
- Displays phone usage and app usage, defining time spent on app.
- Renders the demographics in graphical manner, e.g Bar charts or Pie charts
- Keep note of your activities like running, walking.
- Saving the data in your app for data persistence.
- Users can share their usage data and insights about activities.
- App provides a Widget to display maximum data usage by an app and total time spent on handheld device.

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.

ONBOARDING SCREEN

- OnBoarding Slides
- Google SignIn

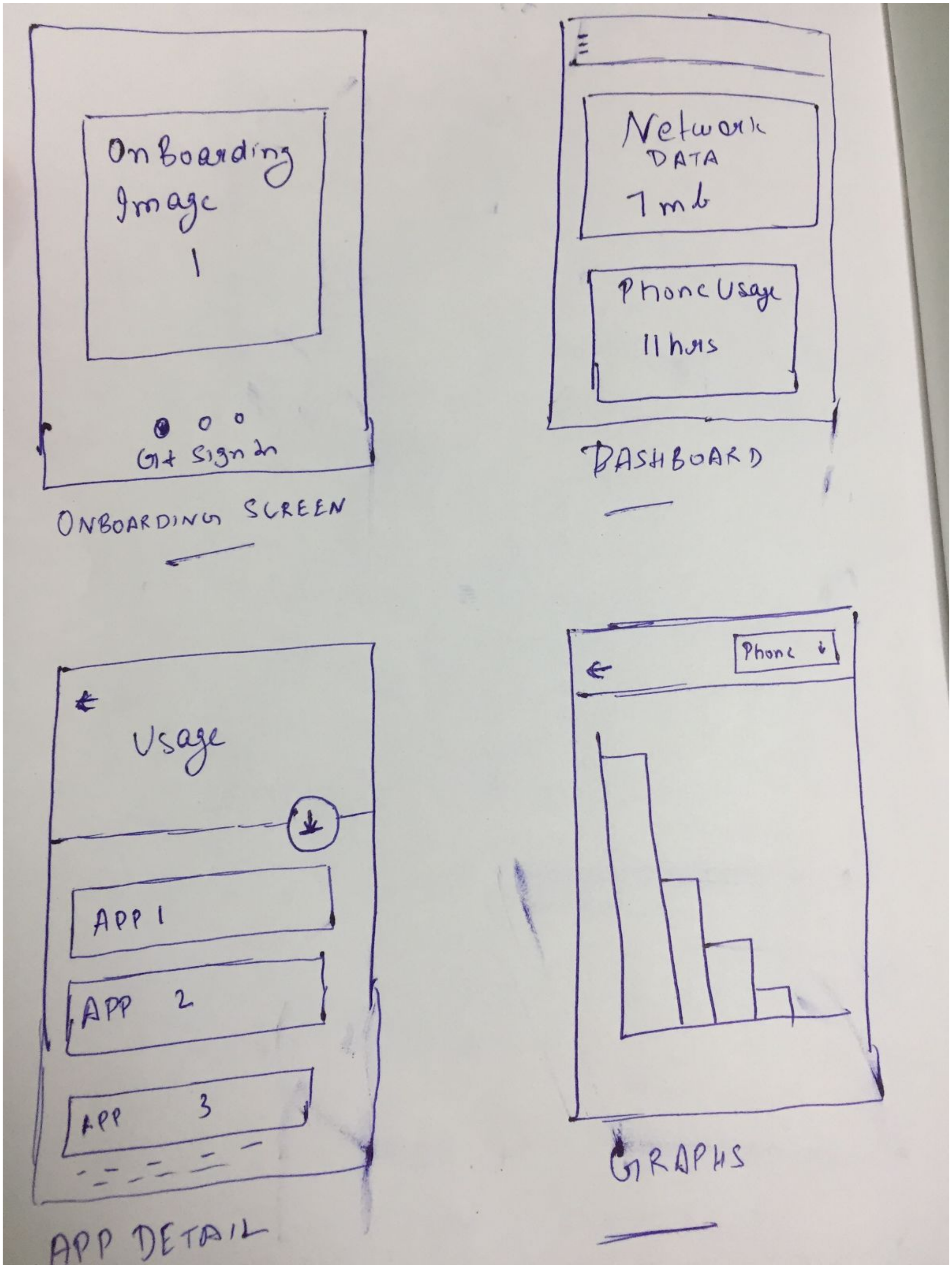
DASHBOARD SCREEN

- Network data summary on cardview.
- Phone usage data summary on cardview.
- Activities options to let user connect to google fit.
- Demographic summary using graphs.

DETAIL SCREEN

- Specific apps with the amount of data used.
- Specific apps with the timestamp, displaying the time spent on particular app by the user.
- Details of tracked activities performed by the user..

GRAPH SCREEN Displaying the phone data demographics in graphs.



Key Considerations

How will your app handle data persistence?

I will build my own Content Provider to persist the mobile data used by the app, the time spent on user on a specific app.

Describe any corner cases in the UX.

The back button for detail screen and graphs will be on the left top in the toolbar of an app.

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

I will be using libraries like ButterKnife, Glide, and Intent Services for Asychronous calls to the remote server. Firebase libraries, Google Analytics and Gson to parse the json.

Describe how you will implement Google Play Services.

Cannot think of it as of now, I will handle this during the development phase.

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

- Project setup
- Initialize the build.gradle with required dependencies.
- Sync all repositories

Task 2: Implement UI for Each Activity and Fragment

- Build UI for OnBoarding screens
- Build UI for Dashboard
- Build UI for Detail screens
- Build UI for Graphs screen

Task 3: Implementing OnBoarding screen

- Layout for OnBoarding screen
- Implementing Google SignIn

Task 4: Creating Dashboard screen

- Creating layout for dashboard screen
- Using cardview to implement three main functionalities
- Cardview for App usage, Data Usage, and Integration of google fit
- IntentService to get the data from a background thread and Loaders to display the data.

Task 5: Details Screen

- Create layout for displaying app
- Displaying listview of app icons, with data accordingly
- IntentService to get the data from a background thread and using Loader to display the data in listview.

Task 6: Graph Screen

- Layout for rendering demographics using graphs.

Task 7: Adding Widget

- App provides a widget to provide relevant information to the user on the home screen.