

[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: santhoshkumar2794

Jarvis

Description

Were you ever in need of an assistant, which automatically does few things based on the instruction you give? You can control your ringtone volume based on your location. If at office, your phone's volume will be at low and high when at home.

Similarly, you can add many applets and configure based on your requirements. The context can be combination of weather, headphone, time, location, and phone's state and user activity.

Intended User

Professionals, Developers and students can use it.

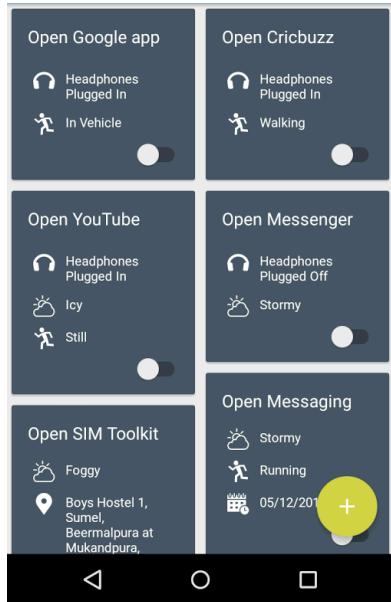
Features

List the main features of your app. For example:

- Location-based information
- Phone based activity

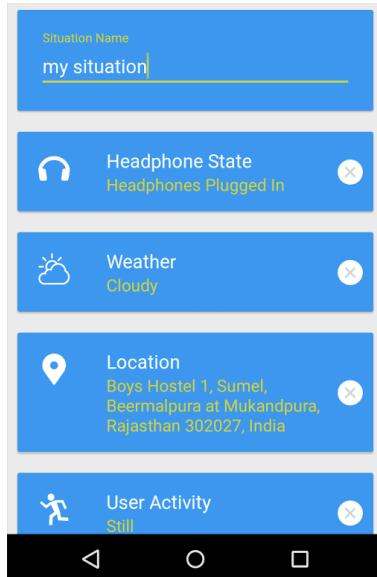
User Interface Mocks

Screen 1



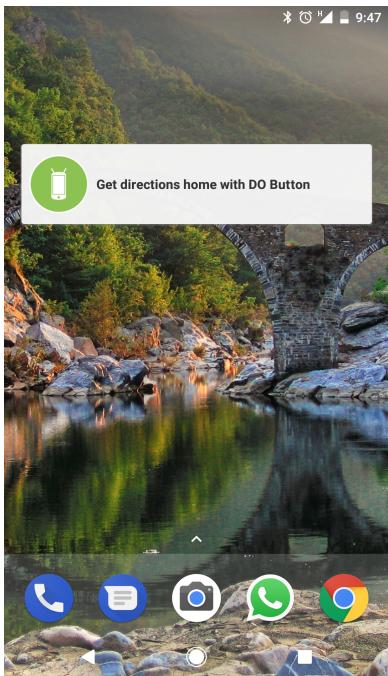
The Main Screen where all the applets will be listed. Also the user can create new applets.

Screen 2



This screen shows a sample structure of how to create the applet and specify the conditions.

Widget Screen



PS: The above screens are just reference. Final UI may differ

Key Considerations

How will your app handle data persistence?

Planning to use Firebase to store the user's data and Firebase cloud storage to share the data among users. Once the user creates an applet, the applet information are written to the storage using Firebase and the applet data is also stored in cloud so the other users can use it if needed.

Describe any corner cases in the UX.

The user can create an applet by clicking on the "+" button and then they can define the condition from the list of predefined conditions available. Once the condition is selected, they can choose the action to be performed.

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

As far as now Picasso or Glide to handle the loading and caching of images.

Describe how you will implement Google Play Services.

Awareness API will be included to handle location, time, and headphone's states

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

1. Configuring Libraries
2. Adding Dependencies
3. Generate API key from developer console

Task 2: Implement UI for Each Activity and Fragment

- Build UI for listing user created applets (Main Activity)
- Build UI for creating applet
- Build UI for widget and implement the services

Task 3: Engineering Implementation

- Implement Awareness API
- Handle different callback for each applet conditions.
- Functional UI implementation
- Use JobDispatcher to do the work based on the results from awareness API.
- Intent Service to upload the applet data to the cloud.
- LoaderTask used for fetching the data and updating it to views.

Task 4: UI Implementation

- Final UI Implementation

Task 5: Testing

- Write test cases
- UI Testing