GitHub Username: anklinuxboy

TeachSmile

Description

This app will teach different emotions like Smiling, angry, sad to people suffering from autism

Intended User

People suffering from Autism

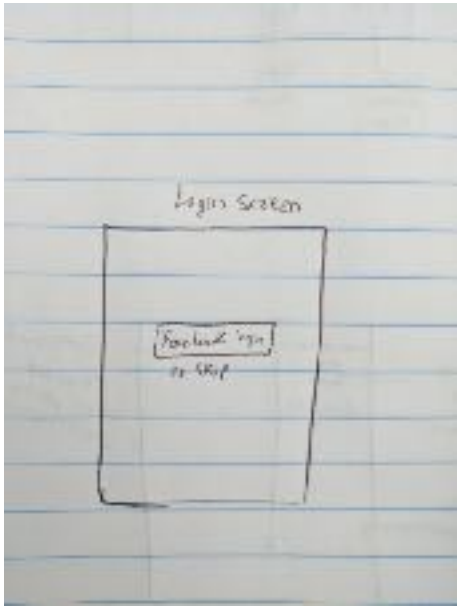
Features

1. Select different emotion modes like smiling, anger, sadness
2. Share photos on Facebook/Instagram
3. Saves the photo with the emotion value attached with the photo

UI Mocks

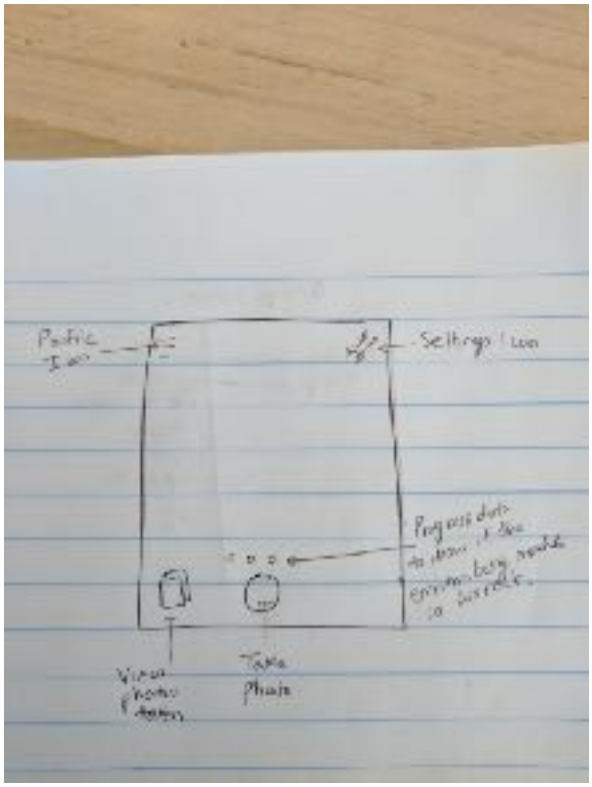
Login Screen

This screen gives the option to the user to login with Facebook or skip login. If he/she chooses to login with Facebook, then the user is authenticated using the Facebook login. In the profile screen, they will see their email and profile photo. If they skip login, then they won't see their photo and email in the profile screen.



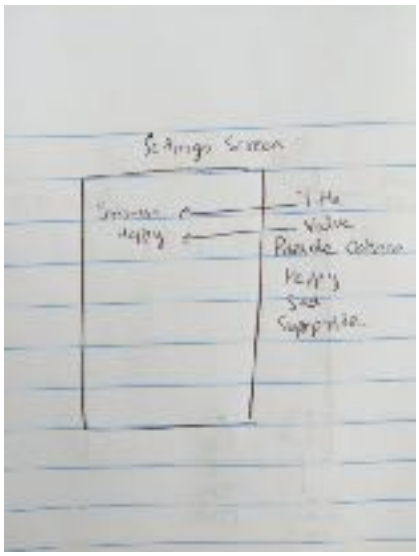
Camera Screen

This screen is the main screen for the app. User can take his/her photo based on the emotion they selected. Clicking on the settings icon will open the Settings screen. Clicking on the profile icon will open the profile screen. Clicking on View Photos icon will open the View Photos screen to view the photos with the emotion setting they were taken in. Clicking on the Take Photo button will take a selfie. The camera will also take a photo when all the lights turn green



Settings Screen

The settings screen will enable the user to change the emotion setting and select from a dropdown list of three emotions - Happy, Sad, Surprised (Happy is equivalent to smile/laughter)



Profile Navigation Screen

This is a side navigation screen which has the profile photo, Location and email of the user

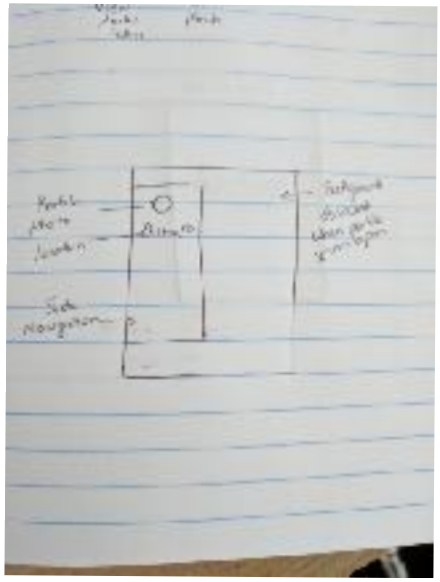
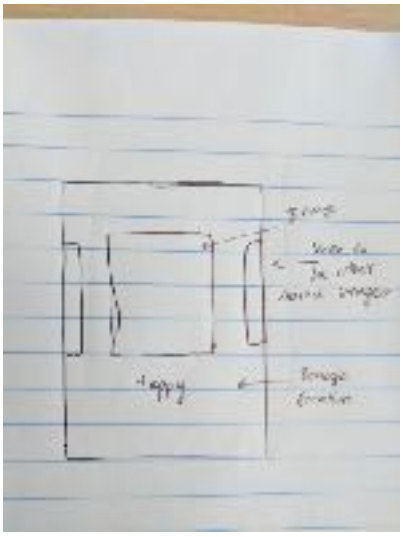


Image Screen

The image screen shows the image taken from the app with the emotion setting with the image. The user can swipe the images right and left to see other images they saved. The user will also be able to share the image.



KEY CONSIDERATIONS

How will your app handle data persistence?

I'll build a content provider which will save the URI of the image and the emotion setting in a database and each time the user clicks on the See Images button, a loader will load the images in the view

Describe any corner cases in the UX.

Hitting back will enable user to navigate the UX between different screens

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

Picasso to load the images.
Facebook to authenticate the user and load their information in the profile screen
Affectiva SDK to detect the emotion level of the user
http://developer.affectiva.com/
RxJava and Retrofit for network requests

Describe how you will implement Google Play Services.

Location to show the location of the user.
Analytics to see how the users are using the app and how many users are using it.

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

Setup up all the libraries and the SDK in the build.gradle file and make sure there are no errors

Task 2: Implement UI for Each Activity and Fragment

* Build the UI for login screen
* Build the camera screen UI
* Build the gallery UI

* Build the Settings UI
* Build the profile navigation screen

Task 3:

Implement facebook authentication and load the user details in the profile UI

Task 4:

Create the content provider and database and save the image details every time the user takes a photo.

Task 5:

When the user hits the gallery icon, load all the images from the database with the emotion setting and implement the swiping of the images

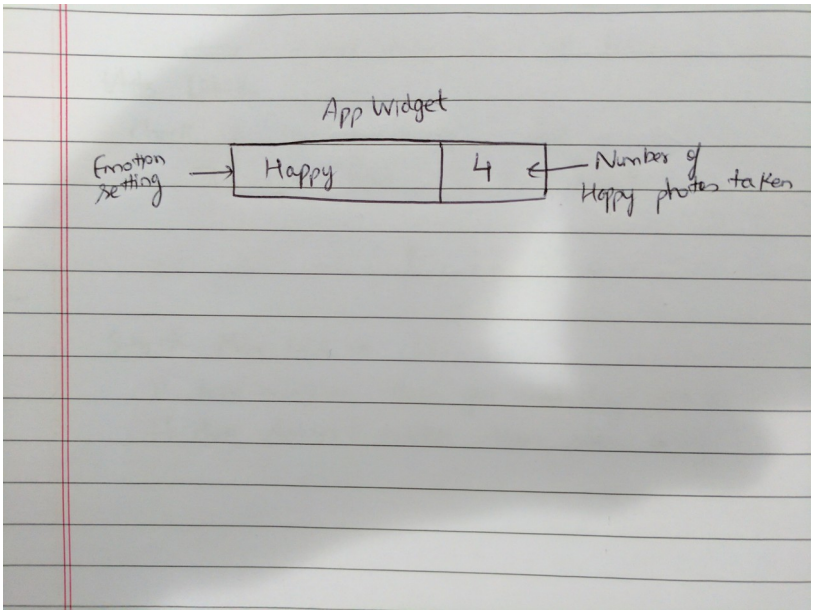
Task 6:

When the user hits the settings screen, open the settings screen and allow the user to change the emotion setting. When the setting is changed, use the emotion SDK to take photos according to the new setting.

Added after the first review -

App Widget -

The app widget screen will enable the user to select between different emotion setting like – Happy, Sad, Surprise and tapping on the setting will start the app in that setting. The widget also displays the number of relevant emotion photos taken. The user can also change the emotion setting after the app has opened from the setting screen of the app if they want.



This app doesn't really need an app widget to improve the functionality of the app because it's highly dependent on the camera to function.

Async Task – The data will be saved in the database using Async Task. For network calls I'd prefer using RxJava/Retrofit