GitHub Username: rafaelanastacioalves

Speko

Description

Speko is a 100% language exchange App aimed at practicing conversation between students through chat. Speko allows the language student to achieve fluency in record time by talking to students already fluent in the language anywhere in the world.

Intended User

Our main audience are young people and adults interested in practicing and learning new languages.

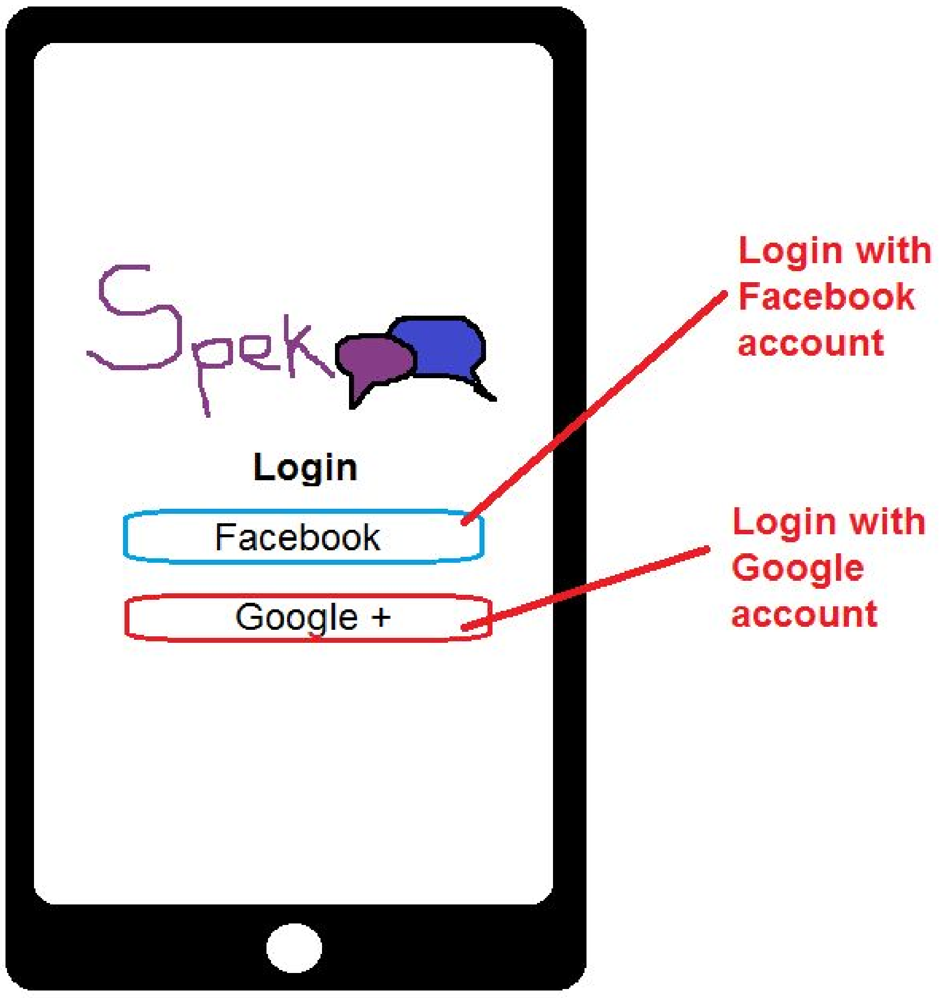
Features

- Find Users;
- Chat.

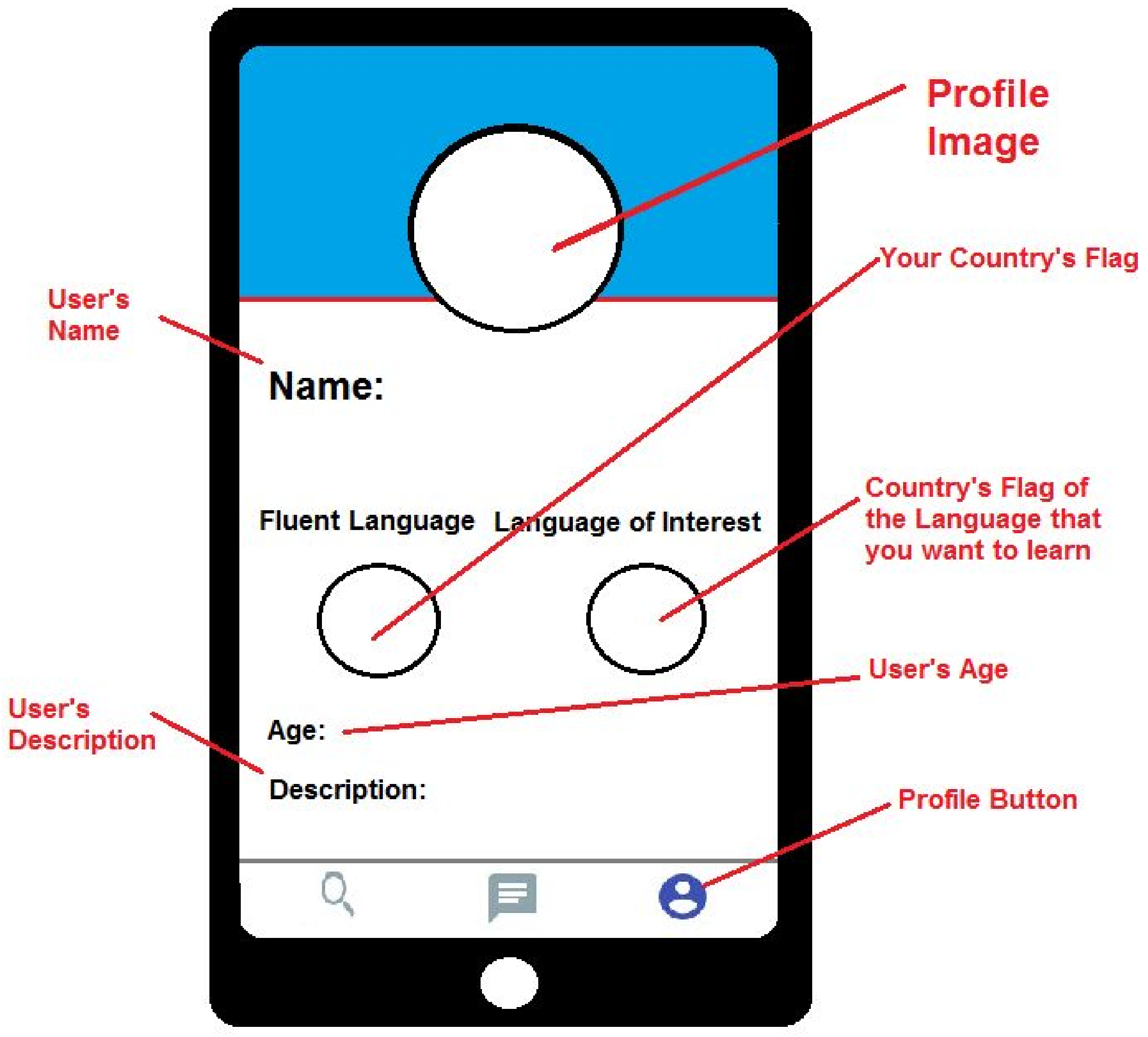
User Interface Mocks

Screen 1 (Login)

*The **Screen 1** is the Login Screen, the user will choose one of the options of signing (google or facebook) that he will use in Speko, the data of this account will be used in the automatic creation of his profile in the App.*



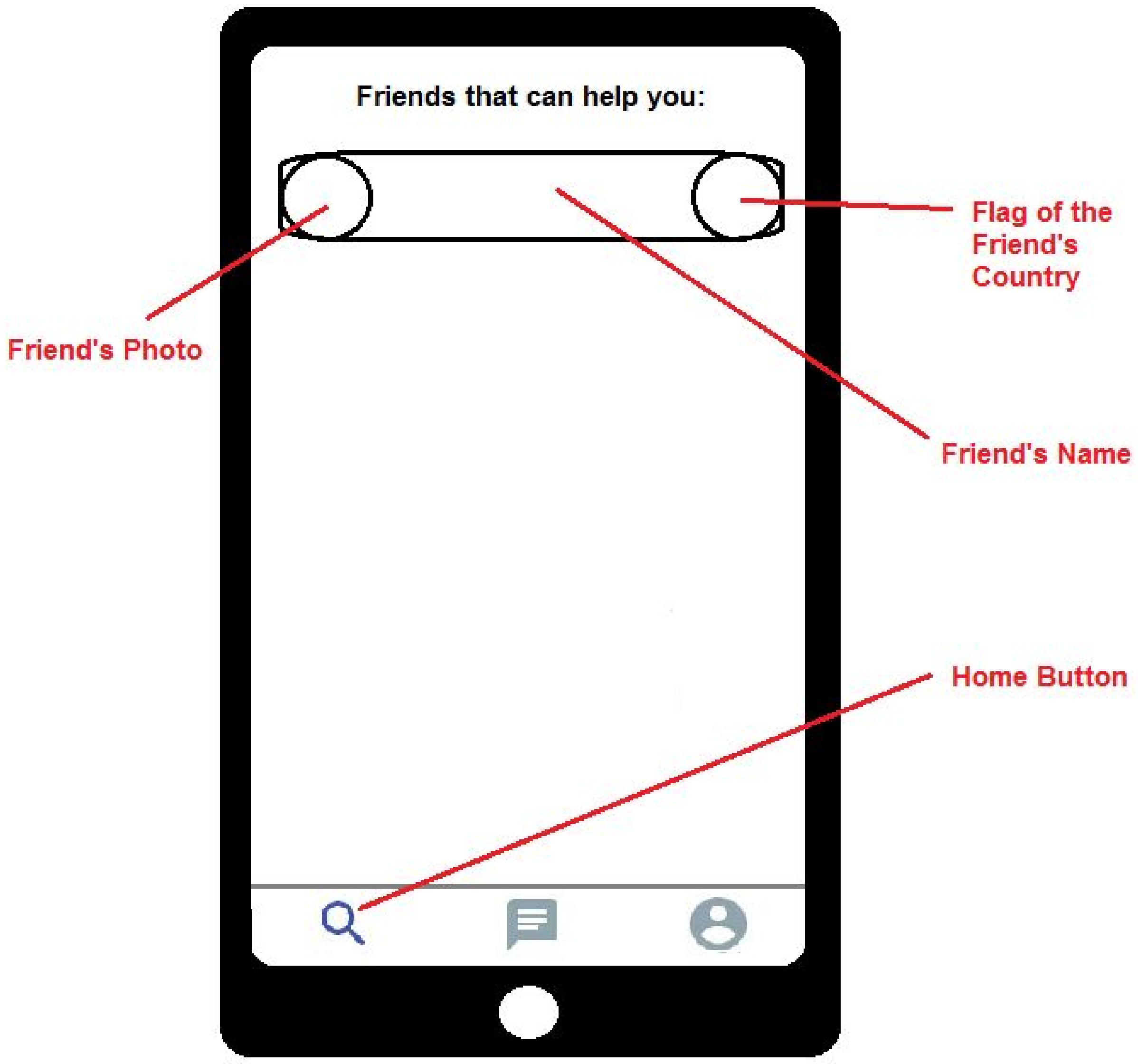
Screen 2 (Languages Selection)



*On the next screen (**Screen 2**), the user will complete his registration by selecting the language he speaks natively/fluently and the language he wishes to practice/learn using Speko.*

There'll be no landscape screen for this app. It'll be locked in portrait mode.

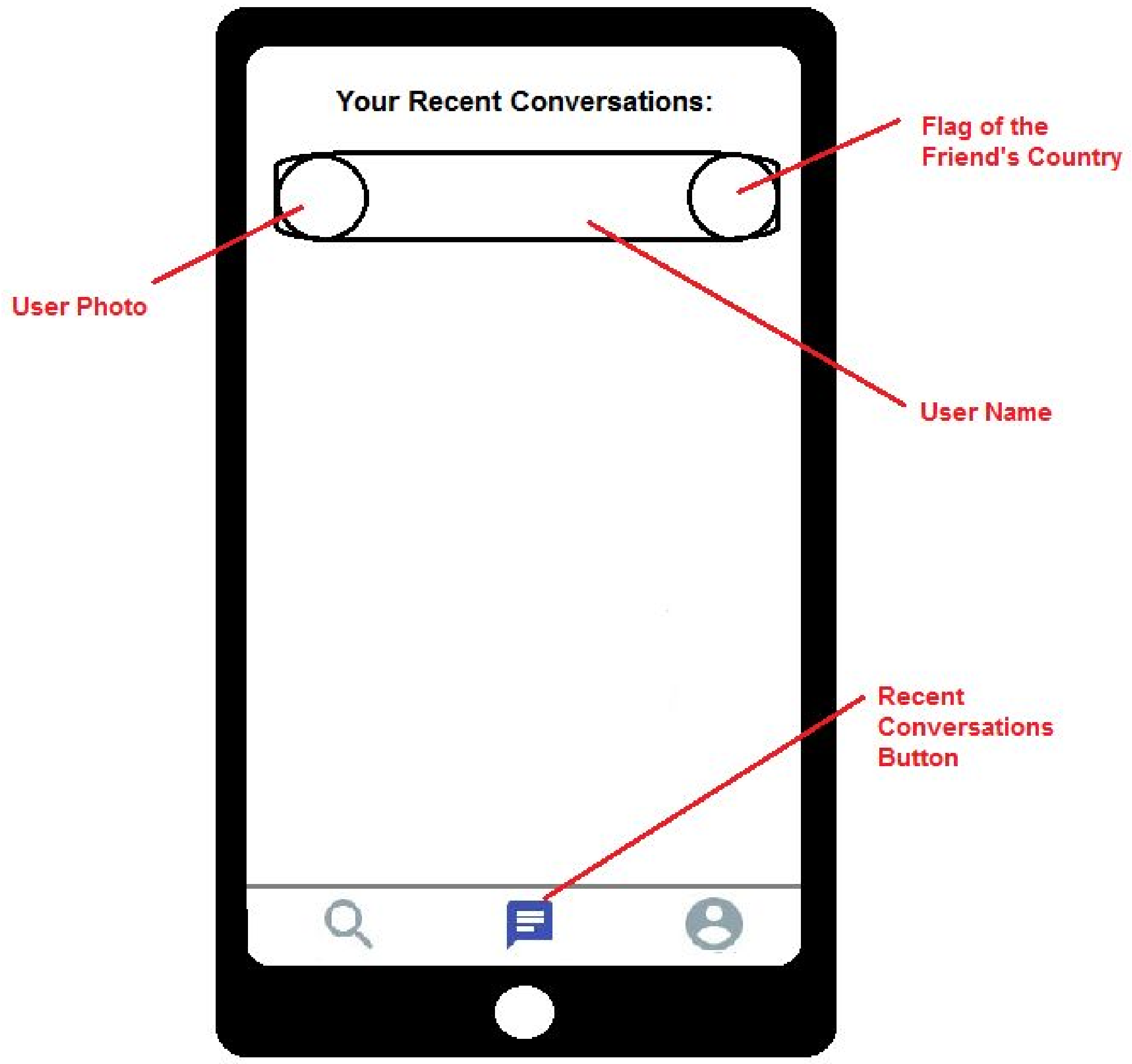
Screen 3 (Home)



Home screen: Displays the All Users in the App.

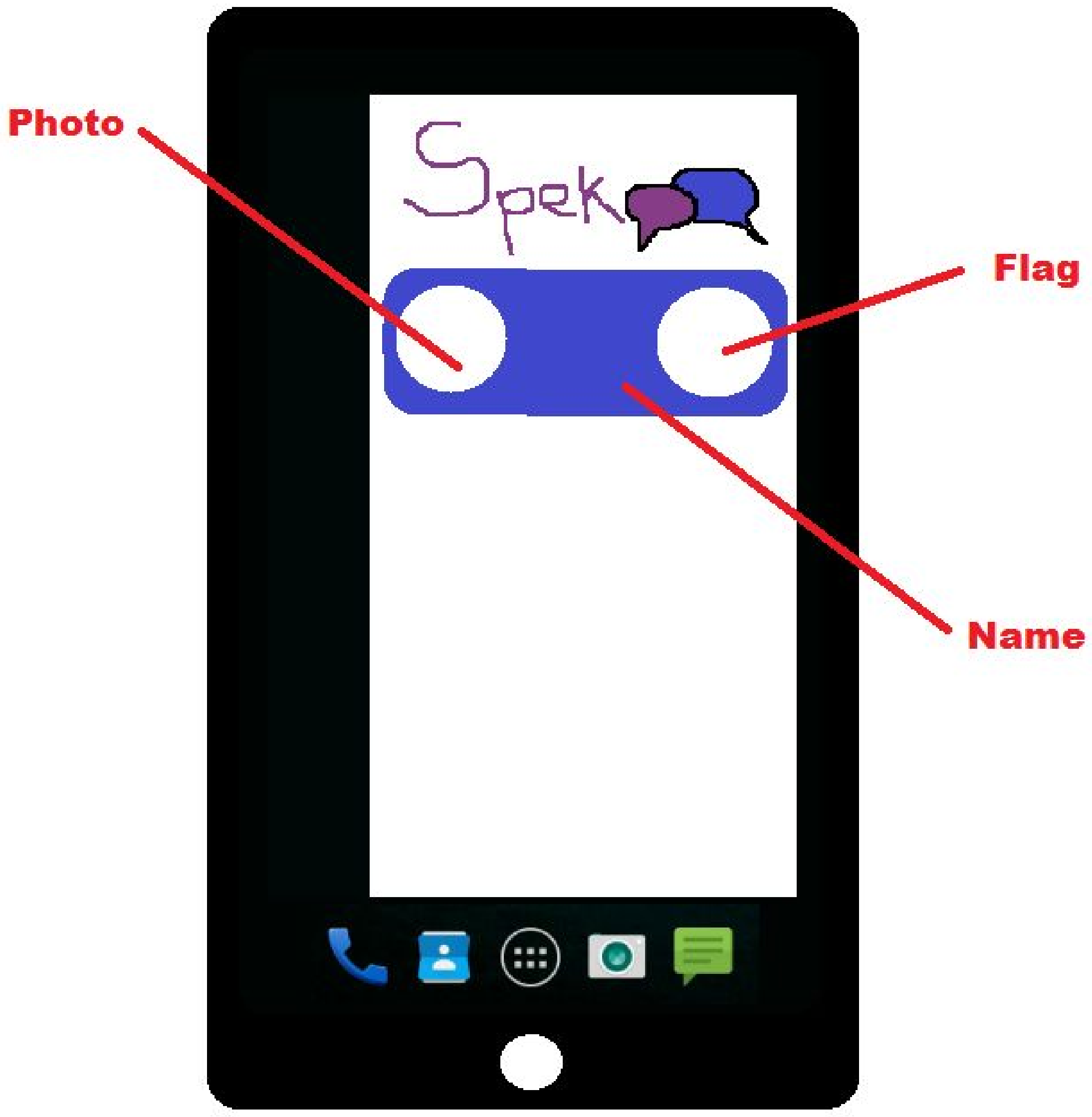
There'll be no landscape screen for this app. It'll be locked in portrait mode.

Screen 4 (Recent Conversations)


Your Recent Conversations:
Flag of the Friend's Country
User Photo
User Name
Recent Conversations Button

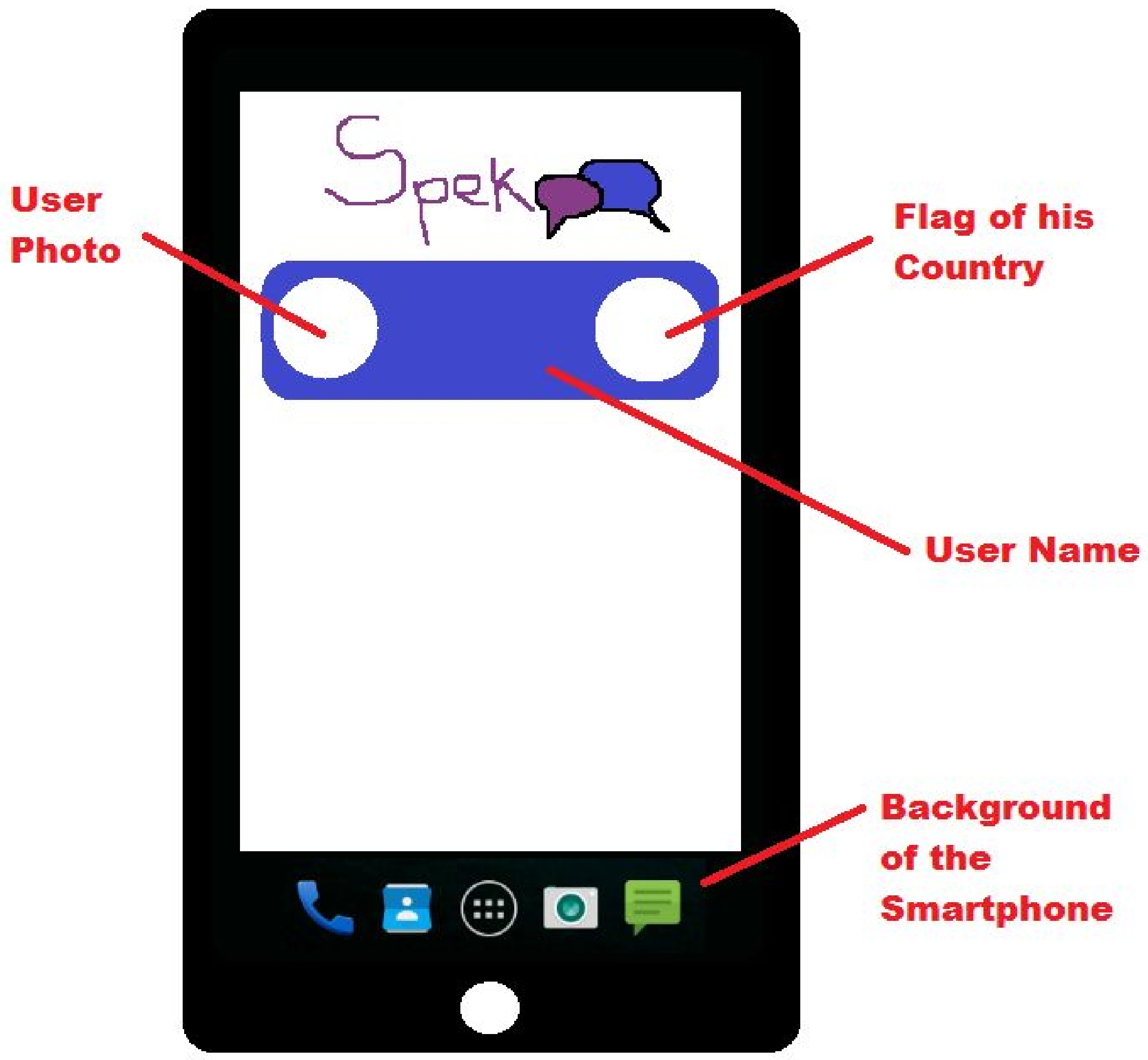
Recent Conversations Screen: Shows all user conversations.
There'll be no landscape screen for this app. It'll be locked in portrait mode.

Screen 6 (3x4 Widget)

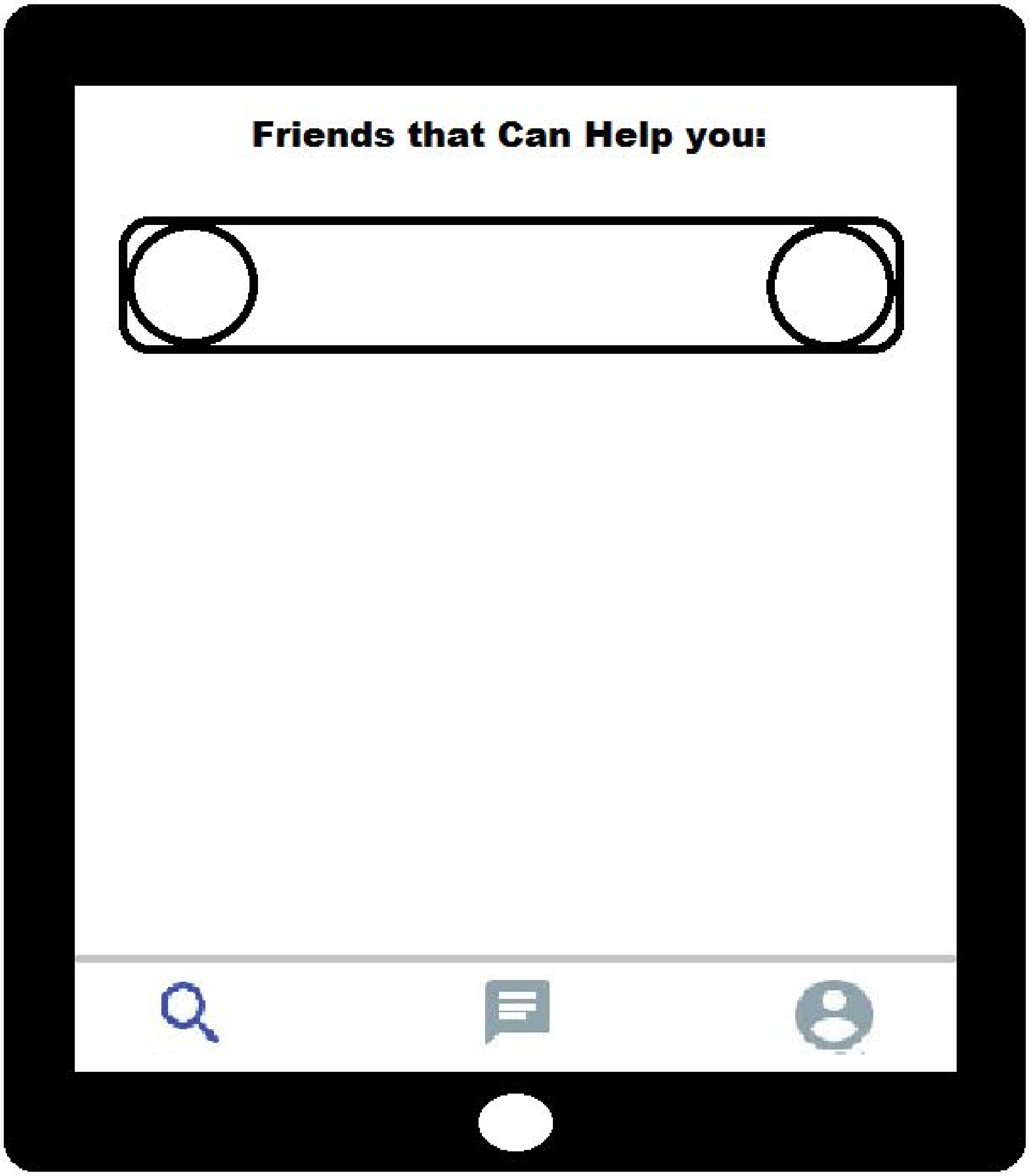


3x4 and 4x4 widgets in order to show the user's friends
There'll be no landscape screen for this app. It'll be locked in portrait mode.

Screen 7 (4x4 widget)



Screen 8 (Large Screen)



On large screens, the app will expand and fit the widthness of the screen properly
There'll be no landscape screen for this app. It'll be locked in portrait mode.

Key Considerations

How will your app handle data persistence?

It'll combine AbstractThreadedSyncAdapter class with a Service class in order to sync information from the web service.
All information will be stored in locally by using our own Content Provider associated with SQLite.
The data will be stored in Firebase database.
The chat will use the Real Time Firebase Database feature.

Describe any corner cases in the UX.

The user is already logged in but he is offline: The user will be able to see his friend list with the information from when he was last logged.Additionally, he'll see a snackbar notification on the bottom of the screen telling him to verify his connectivity. And he'll not be able to click on any friend listed while his connection is not restaured.

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

I'll be using:
Picasso for showing the user's image.
chatmessageview (a chat message view that will be imported and edit to achieve our UI purposes)
Retrofit 2 for http calls
Firebase UI auth - for a simple and faster signup and signin UI
De.hdodenhof:circleimageview - for circular imageview
Com.facebook.shimmer:shimmer - for image loading effects

Describe how you will implement Google Play Services.

I'll use "Google Sign In" (and "Facebook Log In" as well) framework in order to allow user to sign up and sign in in the app. I'll integrate this with our backend in order to manage user's information and logged status.
Additionally, I'll use Firebase for managing user's account info.

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

Create a new Android Studio project.

- Configure a Google API Console project and set up Android Studio project.
- Configure Facebook Login.
- Setup a backend service for persisting user information remotely.
- Setup a backend service for signalling protocol messages.
- Add accessibility features.

Task 2: Implement Signing Process

- Build UI for Signing.
- Integrate Google Sign In;
- Integrate Facebook Log In;
- Build UI for completing Signup process.
- Implement completing Singnup process.
- Add accessibility features.

Task 3: Implement Listing User's Friends

- Build UI For List of User's Friends
- Implementing retrieving list of User's Friends
- Add accessibility features.

Task 4: Implement Listing Recent Conversations

- Build UI for listing user's recent conversations
- Implement adding user's recent conversations
- Add accessibility features.

Task 5: Implement Text Chat Conversation

- Build UI for video call between users
- Implement video call between users
- Add accessibility features.

Task 7: See main user's info

- Build UI for visualizing own user's info
- Implement visualizing own user's info
- Add accessibility features.