GitHub Username: ronanlima

Visiva App

Description

My app has the objective to expose the projects made by an architect and this way, capture clients in potential through technology. Today, that architect do not have a platform to expose your work.

Intended User

This app is destined to an specific architect, that I hope to be helping.

Features

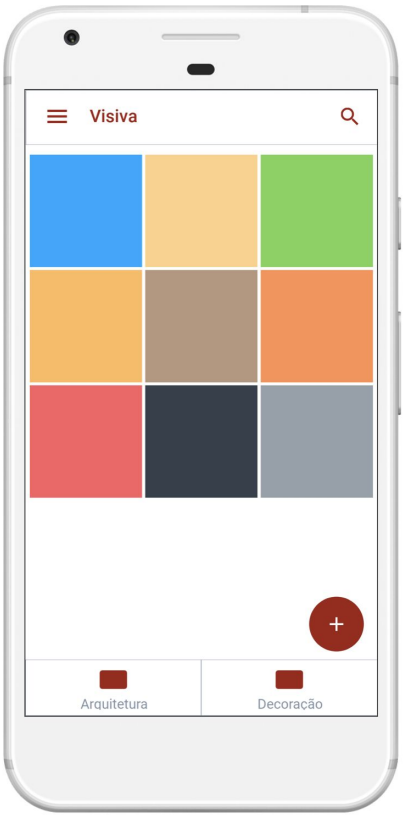
- List of projects done.
- Saves information
- Integration with email apps.

User Interface Mocks

Screen 1

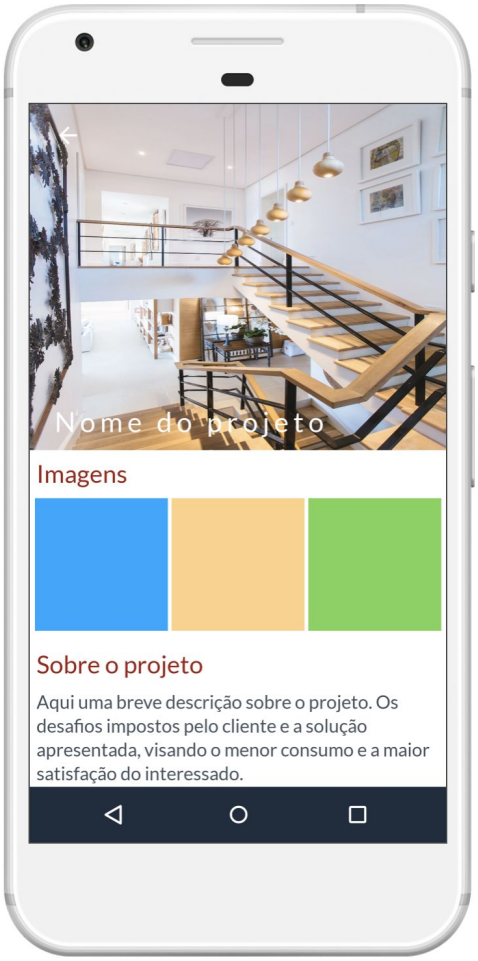
Splash screen

Screen 2



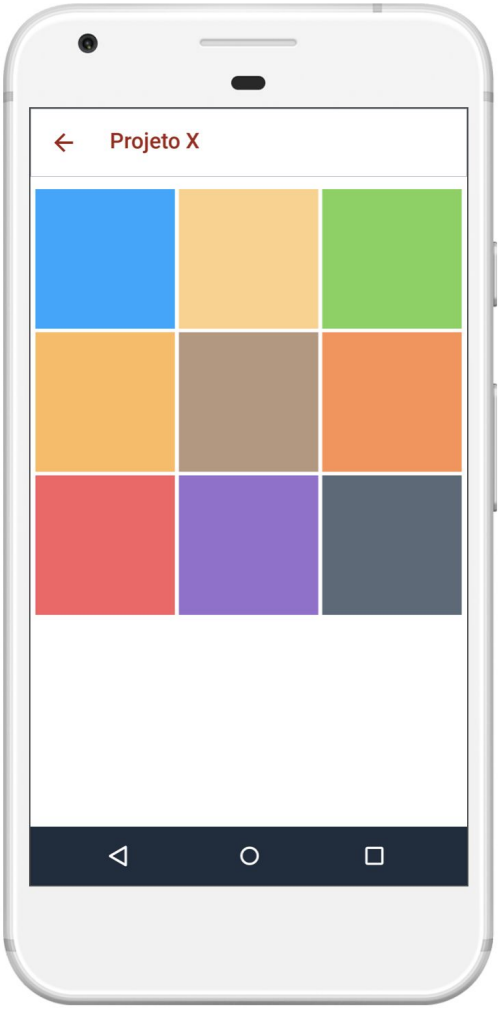
Home screen. This screen shows the projects done by the architect. Allows the user do swipe the screen between two types of projects (architectural and decoration).

Screen 3



Detail screen. This screen shows the user, the images about the selected project and display a brief description of the architect, about the curiosities of work done. Also, allows the user select one image on the horizontal scroll and shows all images of project in another screen gallery.

Screen 4



Especific Project gallery screen. This screen shows all the images of the project selected by user and allow the user to select an image and visualize then in a full screen.

Screen 5

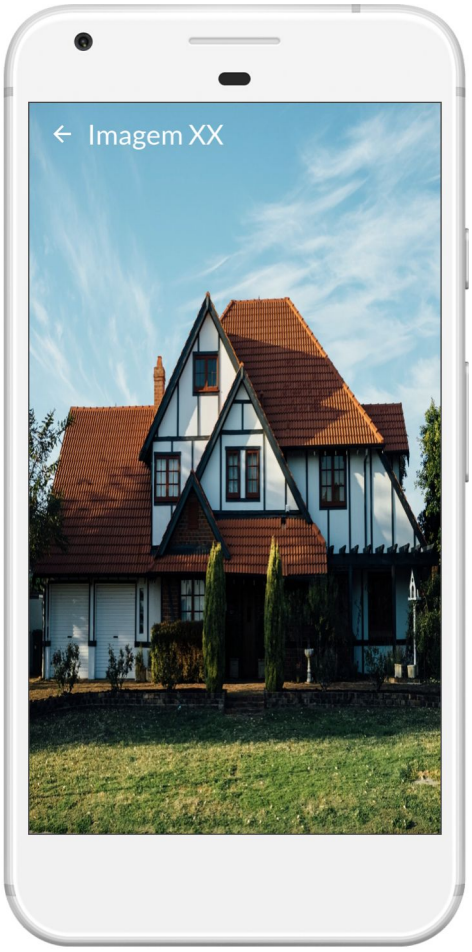
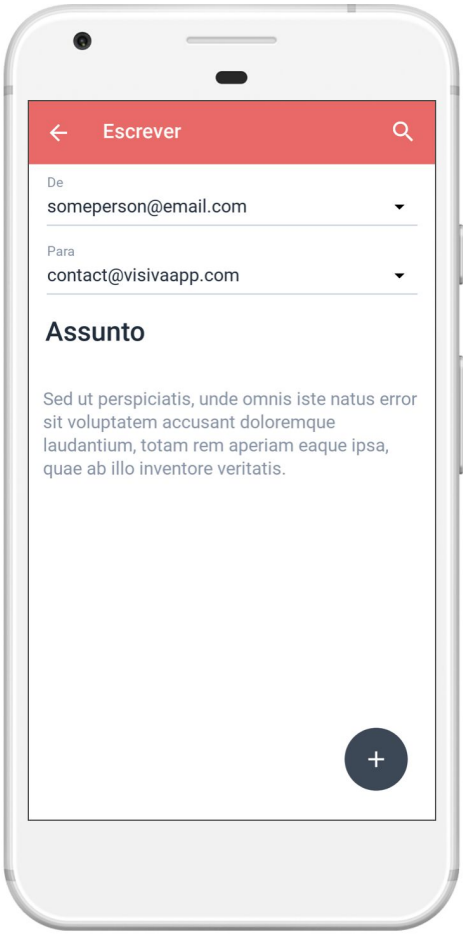


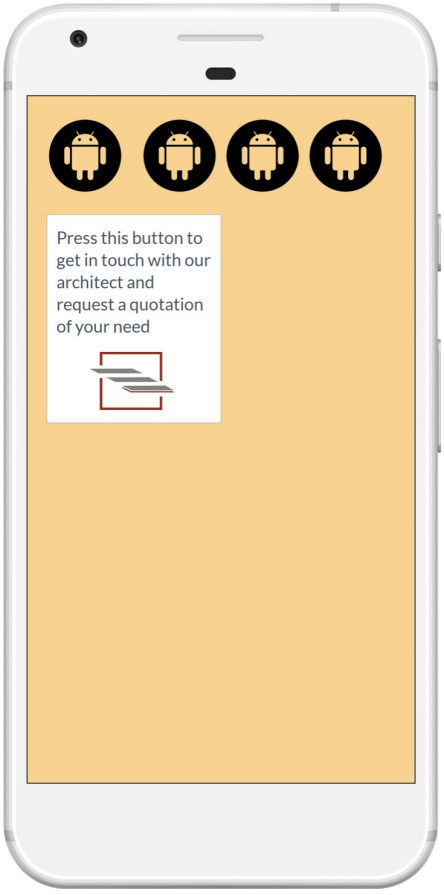
Image detail screen. This screen only shows the imagem in a full screen for better visualization.

Screen 6



Contact screen. This screen allows the user to send an email to the architect, requesting a budget.

Screen 7



Widget screen. This screen is a shortcut to the user send an email to the architect, requesting a budget.

Key Considerations

How will your app handle data persistence?

The app will consume the data using Firebase Realtime database, to get description of project and title. Also, will use Firebase Storage, to get the images of projects. In the first time the user open the app, the data will come from Firebase. After, we will persist the data locally (using Room) to avoid new requisitions to get same data. If the user want to search for new projects, he have to make the swipe gest.

Describe any edge or corner cases in the UX.

If the user does not have internet access in the first time he has open the app, then will be displayed informing this screen.
To the user enter in contact with the architect, he have to tap the FAB on the home screen and then write an email.

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

- Firebase Database to request data.
- Firebase Storage to save/request images.
- Glide to handle the loading and caching of images.
- Room to handle data locally.
- AAC to better build de project.

Describe how you will implement Google Play Services or other external services.

The idea is to use Firebase Database to persist the information about each project and also, use Firebase Storage to save the image attached of each project.
The use of the libraries described above will be done at the first access and / or when the swipe gesture is made, because every request made using Room will save the information of the projects locally, to avoid unnecessary requests..

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

To setup this project, the first thing is put the libraries of Firebase, AAC and Glide into app/build.gradle file. Also, put the butterknife library to avoid boilerplate code.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for SplashActivity
- Build UI for MainActivity
- Build UI for ArchitectureFragment and DecorationFragment to be attached to MainActivity
- Build UI for DetailFragment
- Build UI for GalleryFragment
- Build UI for DetailGalleryFragment

Task 3: Implement functionalities for MainActivity

In this task, we will create:

- HomeScreenPageAdapter class to host a fragment with the architectural projects and another with the decoration projects.
- Create bottom sheet view to indicate to the user that has a way to filter the projects displays

Task 4: Build fragments to show projects

In this task, we will create:

- ArchitectureFragment to display projects of this category
- DecorationFragment to display projects of this category

Task 5: Build detail fragment

In this task, we will create a fragment to show details of selected project, like name, images and curiosities.

Task 6: Build gallery fragment

In this task, we will create a fragment to show the project images, in a gallery format, allowing clicking another fragment to be displayed showing the image in full screen.

Task 7: Build detail gallery fragment

In this task, we will create a fragment to show an image in full screen mode, allowing the user to see details of image.

Task 8: Build a float action button to contact

In this task, we will create a FAB into MainActivity, to allow the user to contact architect.

Task 9: Build a widget

In this task, let's build a widget that allows the user to enter send an email to the architect without having to open the app

Necessary Tasks

Task 1: Make it easy to use the app

The visual components should contain the contentDescription tag, to facilitate the use of users with some kind of visual impairment.

Task 2: Map all the texts in a single point

The app will use the strings.xml file to map all the texts in one place.

Considerations

The app will be write in Java language.
