Project Plan

GitHub Username: doilio

CulArte

Description

Application used to view, filter and contact all types of artists in my country (In the future I will scale), from singers, dancers, songwriters, models(artistic, photographic, runway), painters, and others.

The end user will be able to know exactly what artists we have in this country, who they are, their story, and how to get in touch with every single one of them, either for a show in case of a singer, or a service in case of a painter.

The artist will be able to showcase his portfolio increasing his awareness to everyone who has access to the app.

PS: In the future I want to have 3 Flavors:

- Admin: where I will be able to manage artists and categories directly from the app, without going to the firebase console
- Artist: where the artist will be able to alter his own portfolio's details in real time.
- User: Where the end user will not be able to perform a hard CRUD to the cloud. He/She will only be able to view the data.

Intended User

Everyone who loves Culture and Art.

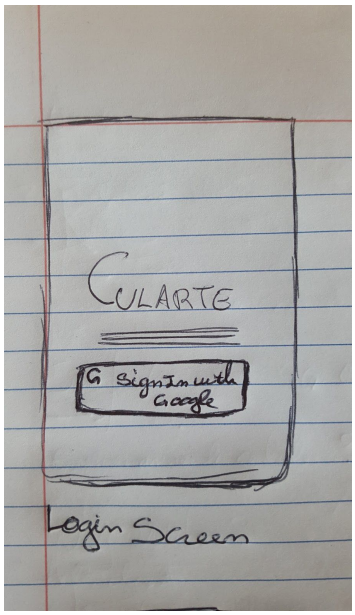
Features

Main features are

- Allows end-users to contact artists either for show or service.
- Shows a list of artists segmented by categories
- Shows a list of art categories
- Allows the user to mark artists as favorites, so that their portfolio may be accessible offline.

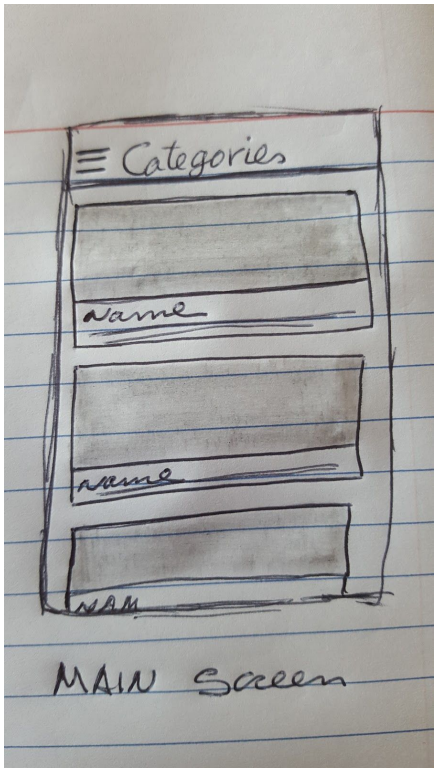
User Interface Mocks

Screen 1



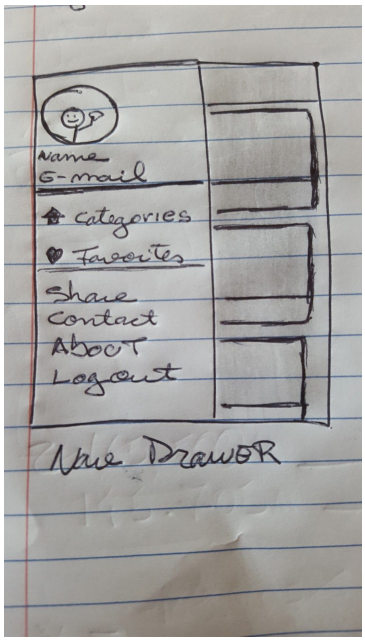
Login Screen with the name of the App, a Google Sign In Button and an App Slogan

Screen 2



Main Screen with a list of art categories showing category name and description, sorted by Name.

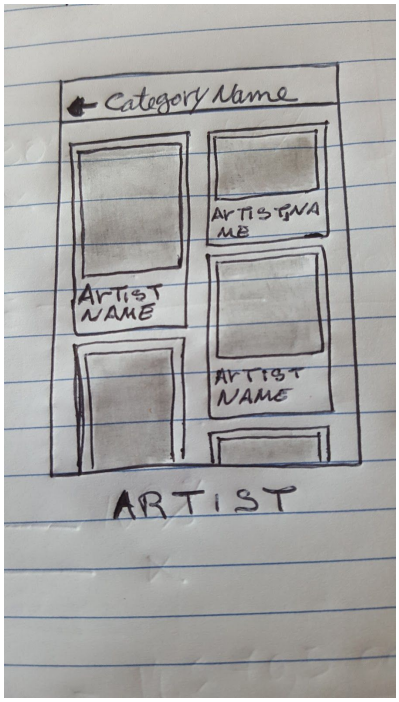
Screen 3



Nav Drawer with options to:

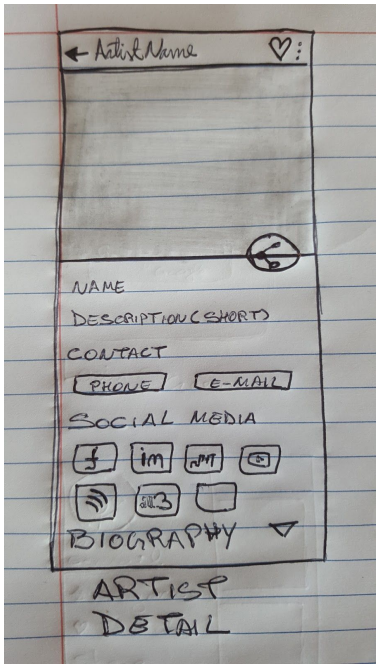
- Open Favorites
- Share App
- Contact Developer
- Open About
- Logout

Screen 4



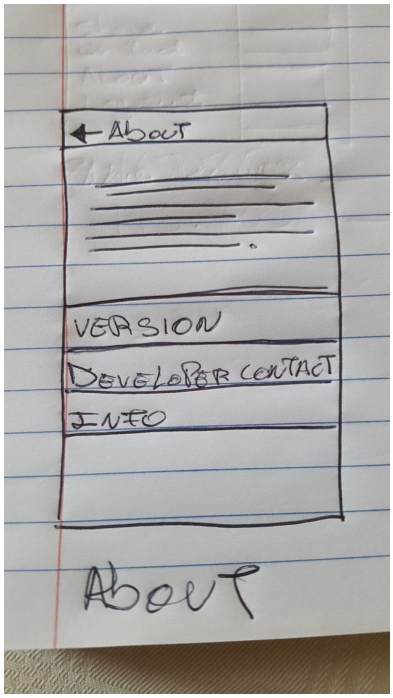
Artist UI which will show a list of artists in a given category.

Screen 5



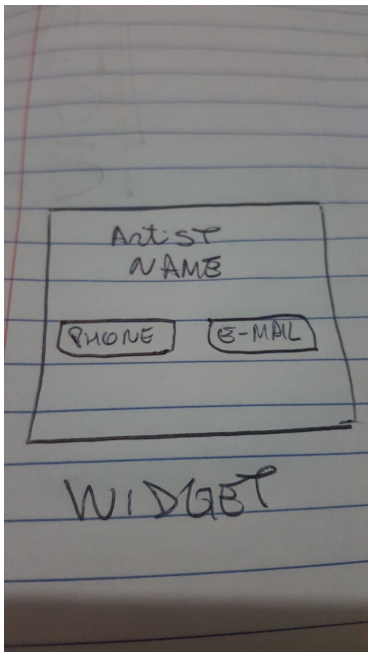
Artist Detail Screen to show the Artists contact Information, and other information such as biography, and several social media Icons which will open the App if it is installed or a web page otherwise.

Screen 6



About Page to show description of the App, version number, and information about the developer.

Screen 7



App Widget to show contact information of the last artist he clicked

Key Considerations

How will your app handle data persistence?

My App will have 2 sources of data, Firebase(Firestore and Cloud Storage) and Room.
On Firebase I will store information about the Art category and the artist's details so that it will be available to everyone.
When the user marks an artist as his favorite, details of this artist will be stored locally on a room database.
In case of Rich Media like Photos, these will be stored on the File System.

Describe any edge or corner cases in the UX.

If the data is not available at a certain screen I will use Images or SVG's to provide feedback to the user.

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

Firebase:

- Analytics - Required when setting up firebase, and it's used to provide insight on app usage and user engagement.
- Firestore - It's a scalable NoSQL database to store and sync data on the cloud in real time. I prefer Firestore rather than the Realtime database due to the way it is organized to scale and its ability to write expressive queries.
- Auth - A sign in method to Identify users uniquely and offer them a personalized experience.
- Ui-Auth - The recommended way to add Firebase Sign in to an app.
- Storage - Used to store large files such as photos.

Timber - A better logging then the regular one provided by Android.
ViewModel - Class that will be responsible to manage and prepare the data to be presented on my UI.
LiveData - An observable data holder, so that I won't have to be doing excessive reads(Polling) on the database.
FlexboxLayout - To align one of the details of the artists on their detail page
Picasso - To download and cache images from firebase on the app
CircleImageView - There will be parts where I will need to display the pictures in circular form.
DataBinding - To automatically bind views and avoid writing so many findViewById.

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

I will be using Firebase Authentication, to Identify the users uniquely and provide them with a personal experience as the app scales, I will set it up so that the user has to login with Google Sign In.
Firebase Firestore will be used to store the artists details such as name, biography, contact information. Only Authenticated users will be able to view this information.
Cloud Storage will be used to store Images of the artists.

Next Steps: Required Tasks

Task 1: Project Setup

Setup Firebase on the console by:

- Configuring permissions
- Inserting the data, setting up filters and permissions
- Getting the google-services.json file

Task 2: Implement UI for Each Activity and Fragment

Build the UI for:

- Main, Favorites, Artist, ArtistDetails, About, SignIn Screens
- Create the UI for the Widget

Task 3: Setup the Android App

- Fetch the categories from Firebase to populate the required screens, handle missing data gracefully.
- Fetch the artists and their details and populate them according to the categories where they belong to.

Task 4: Implement Offline Storage

After all of the connection with firebase is already set up, I will start implementing the adding to favorites functionality and offline storage with room.

Task 5: Implement Additional Menus

When offline storage is done I will implement additional menus options to:

- Share app, share artist.

Task 6: Polish the App

Check that everything is in place before submitting:

- Extracted resources(strings, dimens, styles)
- Remove unused code
- Polish layouts and implement tests where necessary.