USA Jobs

Description

The USA Jobs is an application that users can search for jobs. The users can filter by subdivision for more precise results.

Intended User

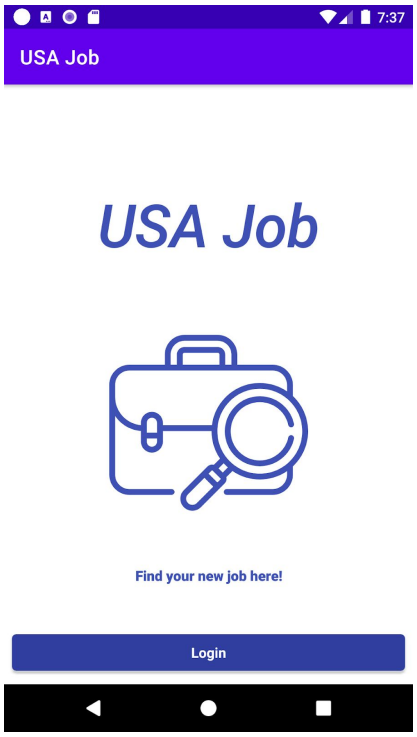
People that want to find a job in the United States.

Features

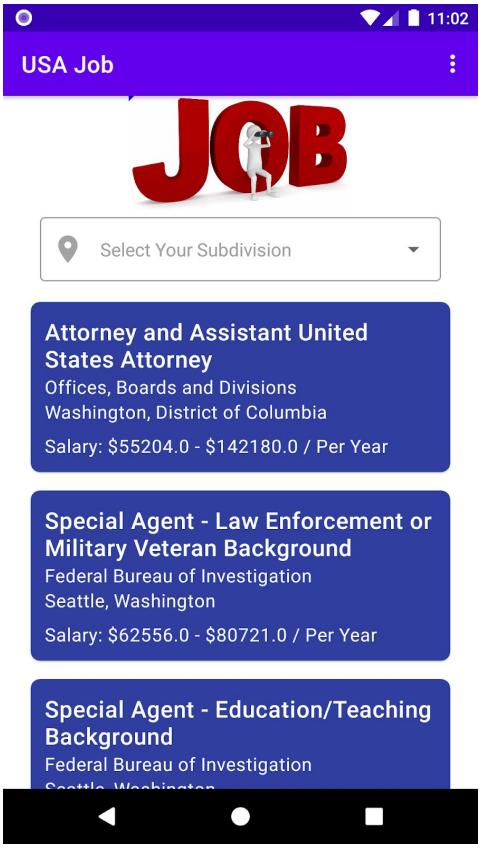
- Log In with a Gmail or Email account;
- Search the jobs by subdivision;
- Show in the Google Maps the route to the job location;
- Apply to the Job being redirected to the official Usa Jobs website;

User Interface Prototype

Authentication Fragment

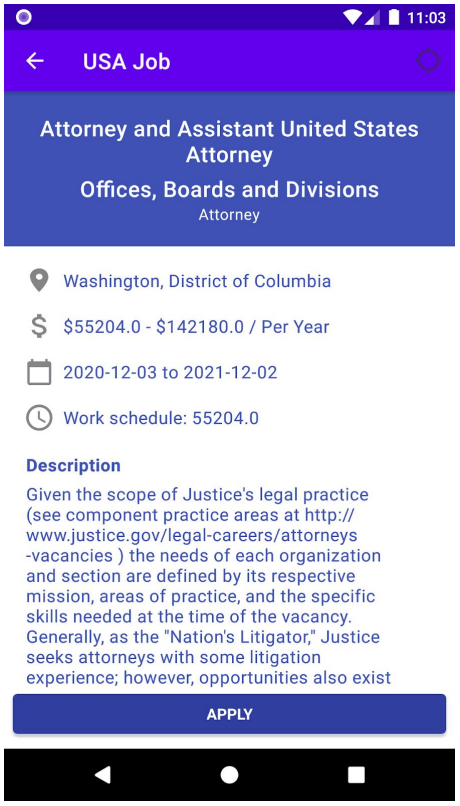
	This is the first screen, the user will be able to log in with a Gmail or Email account.

Job list Fragment



This will be the main screen, here the user will be able to search jobs filtering by subdivision. When the user chooses a job, he'll be redirected to the details screen.

Details Fragment

	In this screen the user has more information about the selected Job, like a location, salary, work schedule and a Job description. When the user clicks on the APPLY button he will be redirected to the usajobs main domain, where he can apply to the job. If the user click on map button, the Google Maps will be open setting a route to the job location with latitude and longitude sented by the API

Data persistence

The application persist the data that is received from the API in the Room database, when a new API call is made we clear the database before receive new datas

Libs

- Timber: Control the log system.
- Navigation Component: Control the navigation lifecycle
- Retrofi2: Control the API call
- Okhttp3: Logs the API calls
- Room: Control the local database
- Picasso: Control the images loaded from API

Project Configuration

Step 1 - Build the project

In this step the project will be created, the base structure of the MVVM and import the libs.

- Create the base structure MVVM based;
- Import the libs;
- Configure the firebase console for the log in;

Step 2 - Create the authentication screen

In this task we create the authentication fragment, his layout and integrate with Firebase Auth.

- Create the authentication fragment;
- Implement the login layout;
- Integration with Firebase log in libs;

Step 3 - Create the ROOM

In this task we create the ROOM and integrate with repository flow.

- Build the Repository Pattern;
- Create database and entities;
- Integrate with repository flow;

Step 4 - API Response

In this task we get the date from API, parse the result and save in local ROOM.

- Create a call to the API domain;
- Parse the result to save in ROOM;

Step 5 - Show the results

In this task we load the data from the database and populate in the list.

- Populate the list with the database data;

Step 6 - Show the details

In this task we set the click listener to the list and create the details fragment to show more details about the selected job.

- Set the listener to the list item;
- Create the details fragment to show;