Corona Reminder

Description

Corona Reminder is an application that reminds users of Covid-19 recommendations before going to a place. Beyond that, it is possible to see Corona numbers in all the world and track a specific country.

Intended User

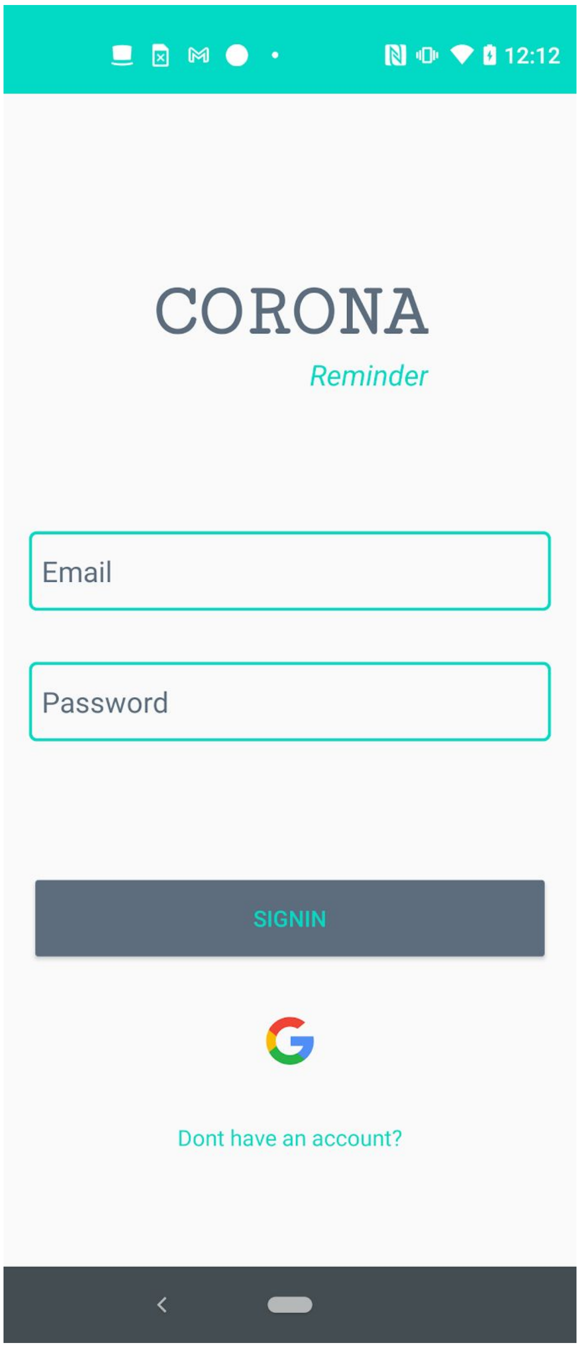
People that want to find a job in the United States.

Features

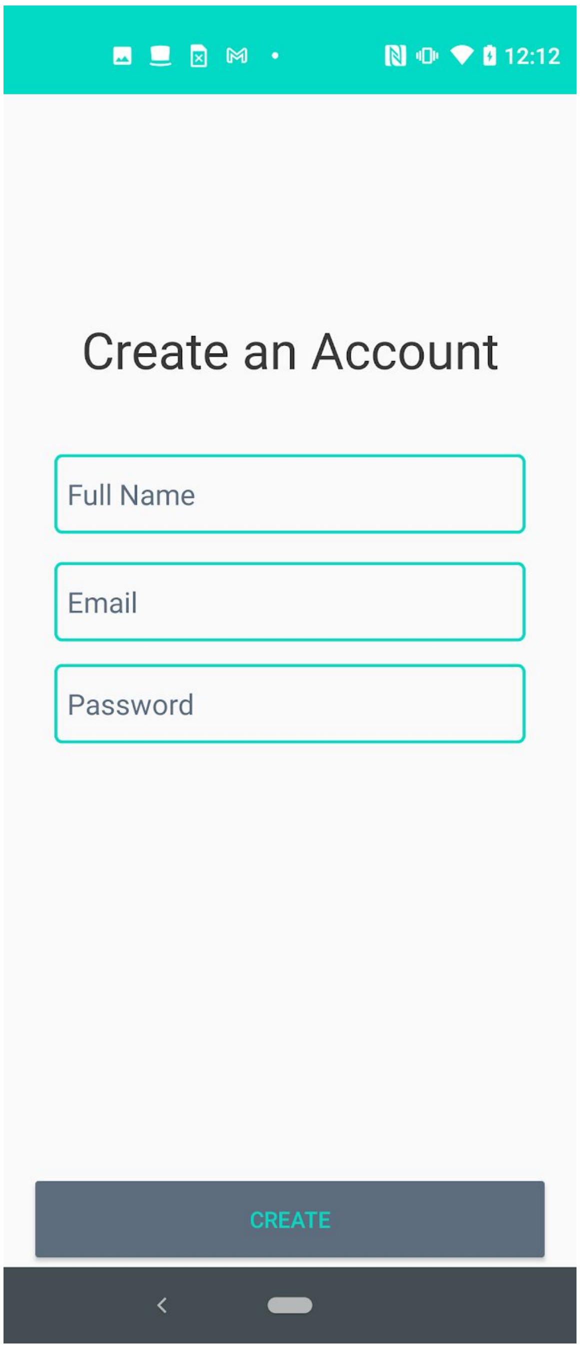
- Sign in with a Gmail or Email account;
- Sign up new account
- See Covid-19 informations about world and countries
- Create alarm to reminder user about recommendations
- List history of alarm

User Interface Prototype

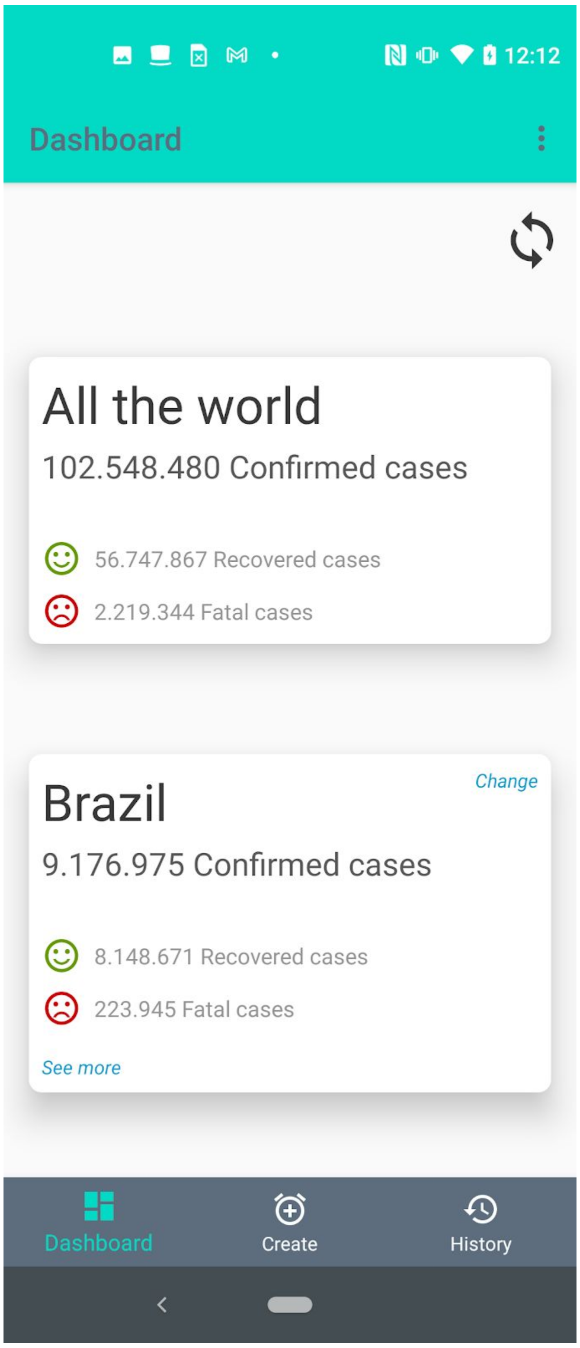
Sign in

 12:12 CORONA Reminder Email Password SIGNIN Dont have an account?	The first screen of the app. User can sign in using email or gmail

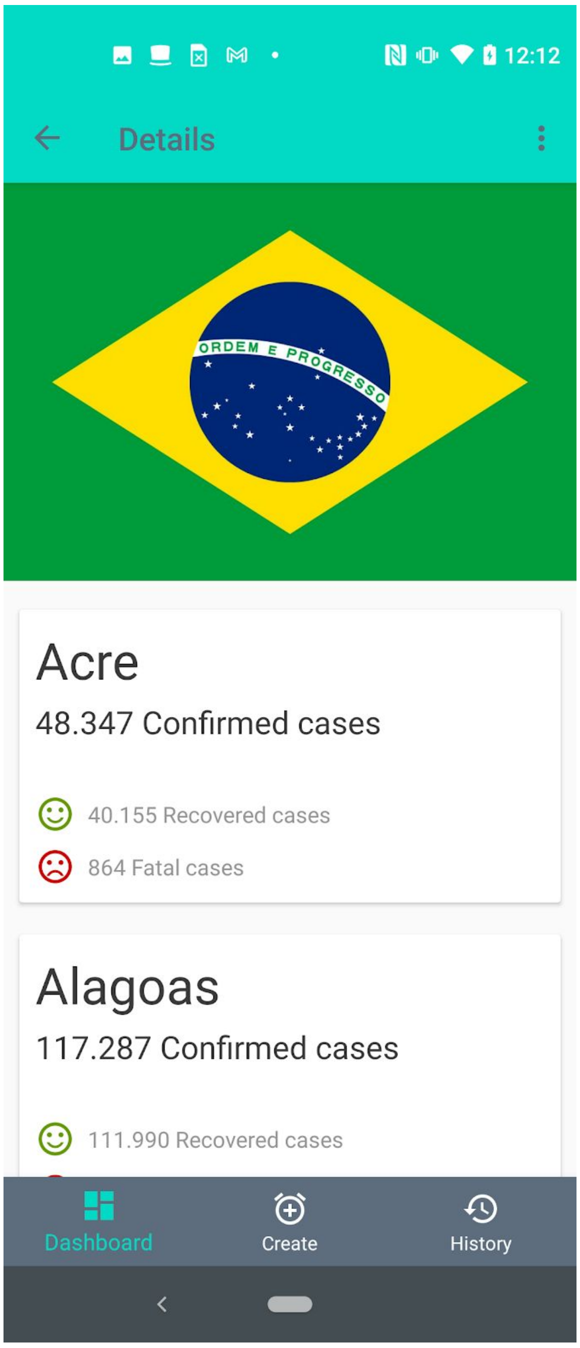
Sign Up

 12:12 Create an Account Full Name Email Password CREATE	Screen responsible to register a new user.

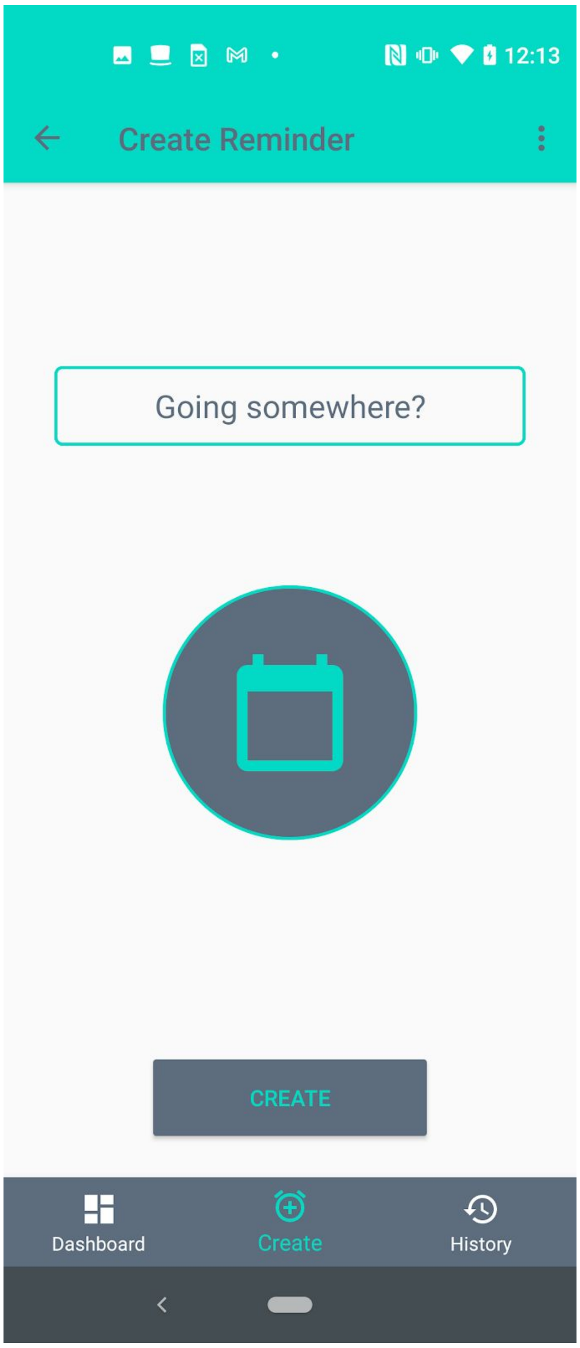
Dashboard

 Dashboard All the world 102.548.480 Confirmed cases 56.747.867 Recovered cases 2.219.344 Fatal cases Change Brazil 9.176.975 Confirmed cases 8.148.671 Recovered cases 223.945 Fatal cases See more	Dashboard is the first screen after user is logged. In it, user will could see covid-19 informations about all the world and your favorite country. The user has a option change country and see more. The top right icon refresh is used to refresh data base with new numbers The three dots menu is a logout option

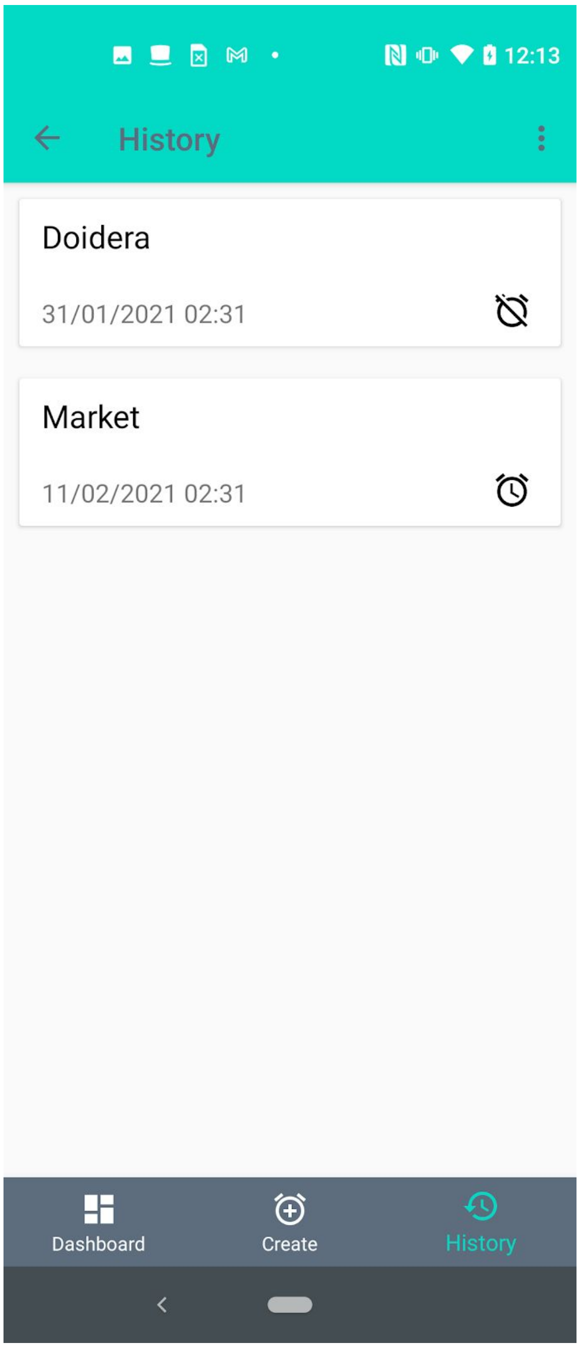
Country detail

 12:12 Details Acre 48.347 Confirmed cases 40.155 Recovered cases 864 Fatal cases Alagoas 117.287 Confirmed cases 111.990 Recovered cases Dashboard Create History	This screen show country flag and informations about states/regions of country if exists in API

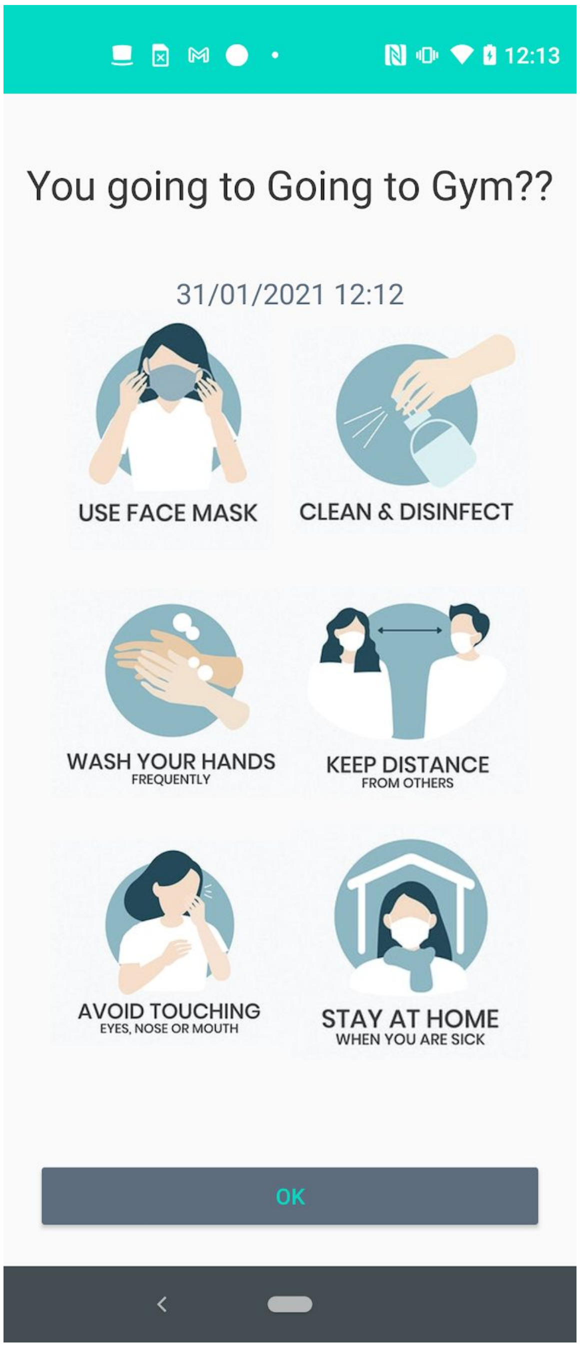
Create Reminder

 12:13 Create Reminder Going somewhere? CREATE Dashboard Create History	Create reminder screen is responsible to create an alarm that will remind users about covid recommendations. Has two parameters: Title: Place going to When: Date and time that going to

History

 Doidera 31/01/2021 02:31 Market 11/02/2021 02:31	Screen that show a list of all alarms created. If the alarm off icon is showing, mean that already passed.

Reminder Detail

12:13 You going to Going to Gym?? 31/01/2021 12:12 USE FACE MASK CLEAN & DISINFECT WASH YOUR HANDS FREQUENTLY KEEP DISTANCE FROM OTHERS AVOID TOUCHING EYES, NOSE OR MOUTH STAY AT HOME WHEN YOU ARE SICK OK	Screen that show recommendations about Covid-19

Data persistence

The app will persist data using Room Database and Shared Preferences. Room will be used to save information about covid-19 (countries and states) provided by this API and reminders created. Shared Preferences will be used to store a user's favorite country. We will Room integrated with Kotlin Flow to communicate with ViewModel layer and LiveData to communicate with the UI layer (Activities and Fragments). For images we will use Picasso to handle and store images

Libraries

- Navigation Component: Handle navigation between activities and fragments
- Retrofit 2: Handle http/https calls
- Room: Persist data locally
- Picasso: Fetch country flags images and store locally
- Koin: Service Locator work as Dependency injection
- Coroutines: Handle asynchronous calls (network and local)

Key considerations

- App will be write on Kotlin
- The app will hold all strings on strings.xml
- The app will RTL support
- All asynchronous calls will be used with Coroutines
- The hardware integration will be provided from Notifications and Alarms

Project Configuration

Step 1 - Build the project

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

- Create the base structure MVVM based;
- Import the libraries;
- Configure Firebase on console and enable auth with email and gmail;

Step 2 - Create layouts

In this task we will create all layouts and setup with ViewModels and setup with Koin

- Sign in Activity
- Sign up Activity
- Dashboard Fragment
- Country Detail Fragment
- Create Reminder Fragment
- History Fragment
- Reminder Detail Activity

Step 3 - Implement data persistence

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

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

After this, we will create a class to hold Shared Preference logic

Step 4 - API Response

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

- Create a call to the API domain;
- Parse the result to save in Room;
- After data stored and user is logged, the user will be able to navigate offline

Step 5 - Show cached data

In this task we load the data from the database and return from ViewModel layer handle

- Populate dashboard with corona infos
- Populate list of country states if exists
- Populate list of reminders