

Capstone project

App name: Map-Weather-App

GitHub username: MohamedAssemAli

Description

Do you ever wonder what is the weather in somewhere you want to visit or know about it more? Or do you wonder what's the weather in your foreign friends now? Okay, the answer is here in this app. Map-Weather app let you surf the world and know what's the weather in each place in the world. It's easy, you just have to open it, view google maps, then drop any marker on the map. After that go to bookmarks page to see your marks with weather details about all of them.

Intended User

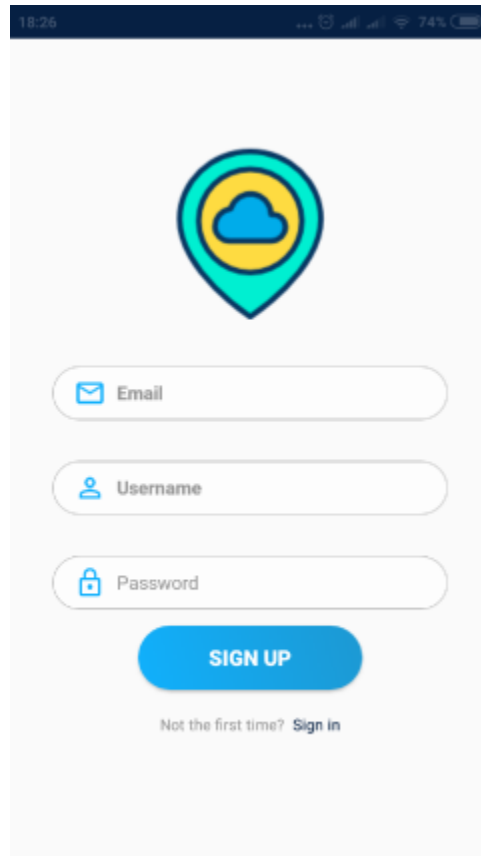
This app for travelers specially and for all the people who has friends abroad.

Features


- Surf the whole world from app map.
- Know weather all over the world.
- Bookmark locations.


User Interface Mocks


- Screen #1




18:26 74%



 Email

 Username

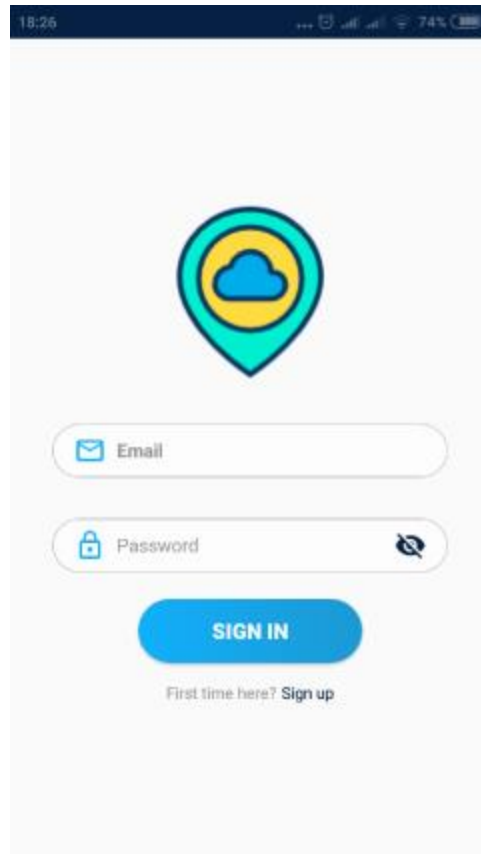
 Password

SIGN UP

Not the first time? [Sign in](#)

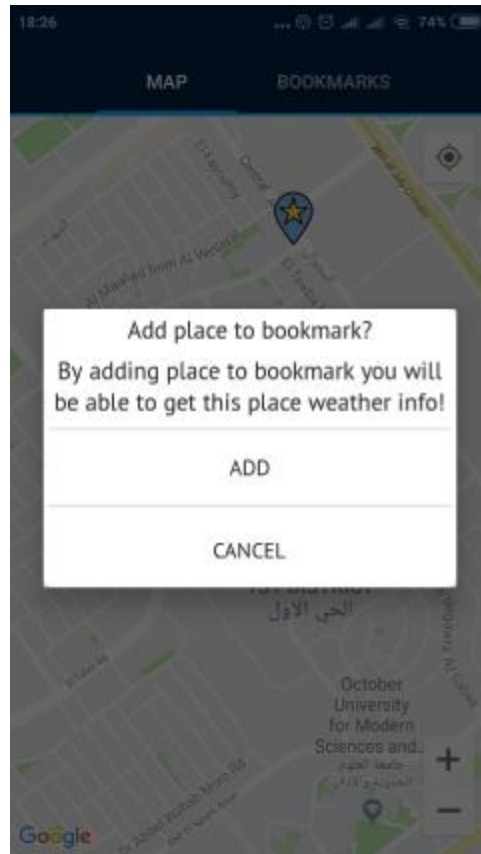
Sign-up screen in which user can sign-up and create an account. Also user can navigate to sign-in page if he has an account.

- Screen #2



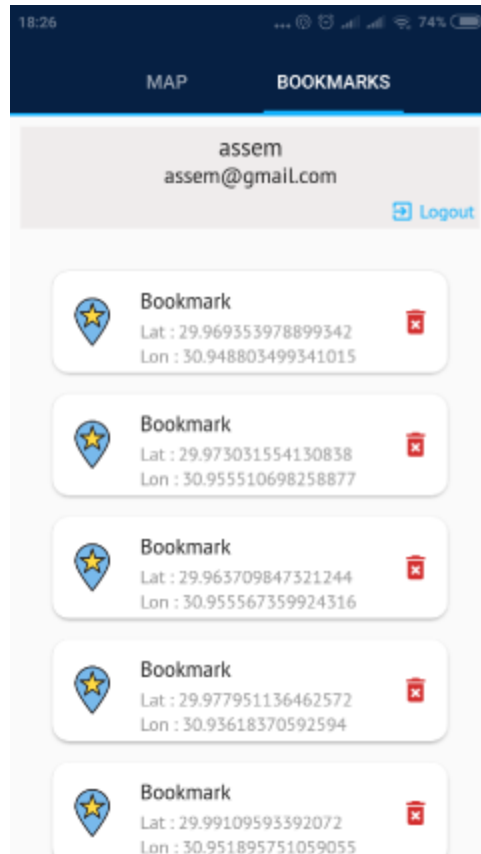
Sign-in screen where user can login into his account.

- Screen #3



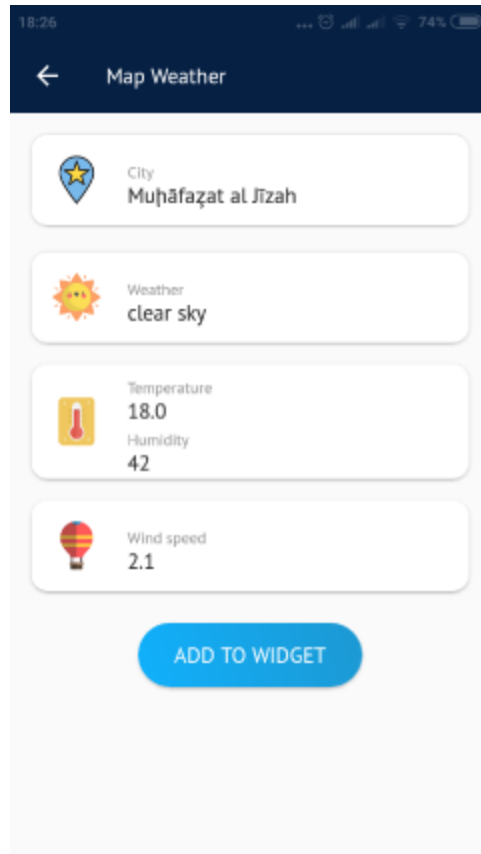
Here's dialog to confirm adding location to bookmarks.

- Screen #5



Bookmark page where user can see his bookmarks and logout.

- Screen #6



Location weather details page where user can know the weather in each location he saved and he can add his favorite location to widget.

- Screen #7



App widget.

Key Considerations

How will your app handle data persistence?

This app will use firebase real-time for data persistence.

Describe any edge or corner cases in the UX.

I use tab layout for easy navigation between maps and bookmarks page which make things faster and easier.

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

- Glide to handle the loading and caching of images.
- Retrofit to handle API calls.
- Calligraphy to handle text fonts in the whole app.
- Gson to handle json serialization.
- ButterKnife to handle views binding.
- Shape of view to handle circle image-views.
- Google maps to create google maps view.

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

- Google maps – this will be used to provide map view for the user
- Firebase Auth - This will be used to login user and provide necessary authentication.
- Firebase Real time Database - Real time database will be used to store the scores of the user.

Next Steps: Required tasks

➤ Task 1: Project setup

- You have to connect app to your firebase account and add google-services.json file.
- Add google maps key so maps can work probably.
- Add you openweathermap api key to AppConfig class in App package in project java classes.

➤ Task 2: Implement UI for start activity and its sub fragment

- Implement UI for start activity which holds login and signup fragments.

➤ **Task 3: Implement login and registration module**

- Implement methods regarding login and registration using firebase-auth, then navigate user to home screen.

➤ **Task 4: Implement UI for main activity**

- Implement UI for main activity and build tab layout that holds two main fragments (map, bookmarks).

➤ **Task 5: Build google maps**

- Build maps fragment by adding google maps key as the first step in strings file.
- Add maps controls to google maps like zoom and my location.
- Add dropping marks functionality.
- Add marks coordinates to firebase database.
- Get saved marks from firebase database.

➤ **Task 6: Build Bookmarks page**

- Get user credentials using firebase-auth.
- Get user bookmarks and user data from firebase database by user ID that is retrieved from firebase-auth.
- Add removing bookmarks functionality to bookmarks page.

➤ **Task 7: Build Bookmark details page**

- Using bookmark coordinates we can retrieve location weather details from weather map api.
- Using Retrofit as a networking library to parse location details from API with coordinates as parameters.

- Adding add to widget functionality where user can add location weather details to shared preferences so, we can get it in app widget.