GitHub Username: mohamed-foly

WeatherApp

Description

Weather App gets the current weather status for your chosen areas and save it on your phone for later offline check.

Weather App is written solely in the Java Programming Language.
Weather App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.

Intended User

Users who follow weathers status

Features

- Gets the weather of chosen area
- Saves weather information
- Display weather information (online/offline)
- Display weather information in Widget

User Interface Mocks

Screen 1

MainActivity which contains Recyclerview for areas and short temperature description for each one
And a button for adding new area
Another button to refresh weather data

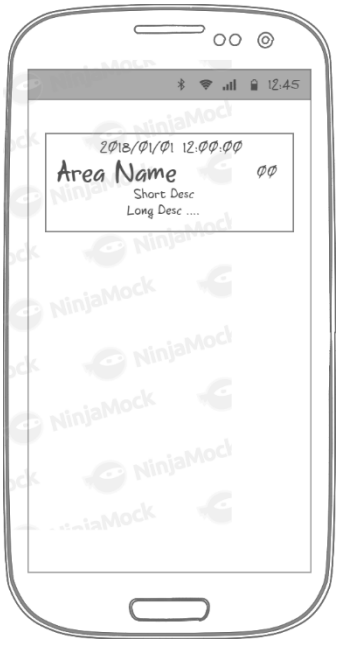
Screen 2

AreaAddActivity which contains map which helps user to choose an area to add to his list and also contains two labels to displays indicated position name and user's location name

Screen 3

AreaDisplayActivity which contains Clicked Area weather information

Screen 4



Weather App widget which summarized data for area like last update date, temperature, name, short description and long weather description

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

Room Persistence Library

Describe any edge or corner cases in the UX.

Users can return to the back activity from any activity using back button or using top left back button

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

Location: will be used to help user to get his current location
Maps: will display user's location and indicator for choosing an area to add to his weather list
OpenWeatherMap API: app will use this to get current weather information while being online
By user request to refresh data .

Next Steps: Required Tasks

Task 1: Project Setup

- Create a developer account in openweathermap.org
- Waiting for confirmation mail of activation status on mail
- Get the Default API key for the created account
- Get necessary google API Keys

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for AreaAddActivity
- Build UI for AreaDisplayActivity
- Build UI for AreasRecyclerview
- Build UI for AreasRow
- Create strings.xml file

Task 3: Implement Google Play Services

- Implement Maps
- Implement Location

Task 3: Preparing Permission

- Get Location Permission
- Implement Location

Task 4: Create Classes

- Create model for Area
- Create data access object for Area
- Create database class extends RoomDatabase
- Create adapter for Areas
- Create asynctask for update weather
- Create interface for on weather updated
- Create interface for on area click
- Create Area view model extends AndroidViewModel
- Implement LiveData in Area view model

Task 5: Saving Data

- Save chosen position

Task 6: Refresh Weather

Task 7: Build Widget

Libraries & versions

Library Name	Version
Android studio	3.1.4
Gradle	4.4
Google Location	15.0.1
Google Maps	15.0.1
Room	1.1.1
Lifecycle	1.1.1

Schema


12:45
WeatherApp
Area Name
Area last Temperature
Area Name
Area last Temperature
Area Name
Area last Temperature
Area Name
Area last Temperature
12:45
Add New Area
Current Location Name
Current chosen Area name
12:45
Area Name
Weather
Description :
Short Desc
Long Desc ....
Temperature : ØØ
Humidity : ØØ
Pressure : ØØ
Wind
Speed : ØØ
Degrees : ØØ
Cloud ØØ %
Rainy