Report on Weather App for Android

**Department: Computer Science Engineering**

**Member Roll**

**RESHMINA FARHIN 33**



Index

**Topics Page**

Introduction 3

System Design 5

Implementation 8

Conclusion 11

References 12

Introduction

SCOPE OF THE PROJECT

The weather app has been implemented on android studio version 3.4.

* The app is a light-weight application (consuming only about 7MB).
* It comes with features such as viewing the current weather conditions and forecasts of the entire day (every three hours using a line-chart).
* Forecasts of the upcoming four days are given.
* Temperature triggers change in the colour of the layout.
* Cloud forecasts are facilitated with small icons for easier understanding of the user.
* The weather conditions available for view include temperature, wind speed, and cloud.
* Timing of sunrise and sunset of the current day is also available.
* The app can be used to keep track of weather around supported areas, which include twenty-seven major cities around the world such as London, New York, Shanghai, Tokyo, Delhi to name a few.

TECHNOLOGIES USED

* ANDROID STUDIO **Android Studio** is the [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used for the weather application. It provided [Gradle](https://en.wikipedia.org/wiki/Gradle)-based build support and a [layout editor](https://en.wikipedia.org/wiki/Graphical_user_interface_builder) that allowed to drag-and-drop UI components and option to [preview layouts](https://en.wikipedia.org/wiki/WYSIWYG).
* THIRD PARTY LIBRARY HELLOCHARTS that was used to make the line-chart for the current day’s weather forecasts at intervals of three hours. Some salient features include line chart, column chart, pie chart and bubble chart.
* Volley is a tool for handling HTTP requests. The ‘getJsonObject()’ and ‘getJsonArray()’ method for handling the response received from the API. Volley operates with the help of a request queue where the exact details of HTTP request are stored. Each HTTP request is has 2 methods, one for
* handling the successful execution of the given http request and the other for handling of errors incase one occurs while making the HTTP request.
* OPEN WEATHER API Using a generated API key JSON files were retrieved. Forecasts for five days at an interval of three hours. The weather data could be fetched for over 3700+ cities. The information was available in metric and imperial system.

ANDROID WIDGETS USED

* Image View
* Line Chart
* Card View
* Navigation Drawer
* Progress Bar

System Design

The weather app is designed keeping in mind the ease of use for everyone. The colour of the layout changes as per the temperature of that region. Higher temperature increases the red component and lower temperature increases the blue component.

The first activity displays the temperature of the current location. The temperature at that time is displayed first. A linear layout immediately beneath the current temperature contains a line-chart that shows the forecast of the same day at intervals of three hours. Then there are four cards which are swipe-able from left to right. Each card represents forecast of an upcoming day, the values of which are given directly on the cards. Swiping over the will bring up a new activity containing the forecast details of that day for the selected city. The forecasts include average temperature, average clouds, pressure, wind speed and humidity percentage.

All the layouts have drawer layout as their parent layout. Nav-drawer has menu options to start the temperature, cloud and wind activities. The timings of sunrise and sunset of the present day are given at the bottom of the nav-drawer.

In the Nav-drawer options for viewing the cloud activity and wind activity is available. The pattern of cards in cloud activity remains the same except on the cards values of average cloud percentage for that particular day is given. In the cloud forecast the line graph is represented with symbols and description instead of cloud percentage. The basic colour of cloud activity has been kept in shades of blue, higher cloud index will mean deeper shades of blue and lower cloud index will show lighter shades of blue. The wind layout has a fixed colour gradient and its cards show the average wind-speed of the mentioned day. All cards are swipe-able.

The timings of sunrise and sunset were in timestamp format. So it was coded to twelve hour human-date format.

The app provides locations of twenty seven major cities around the world. The user has to liberty to choose any one of the cities for viewing the forecast and present weather conditions. The option of choosing locations has been implemented use Alertdialog with single choice. The default city has been set to “Kolkata, IN”. The dates and timings were in accordance to “UTC”. Hence the date was parsed according local time zone of the city as selected by the user. When a user changes city in one activity then the city is changed in all other activities and the value is updated for user’s perusal. This exchange of data between activities was achieved using ‘getStringExtra()’ and ‘putStringExtra()’ after initialising the Intent.

The elevation of action bar has been set to zero and the colour has been set in accordance to the presently open layout to give the entire layout a smooth change of gradient.

Progress Bar has also been implemented in the application which is in effect till all the required data has been fetched from the API Server.

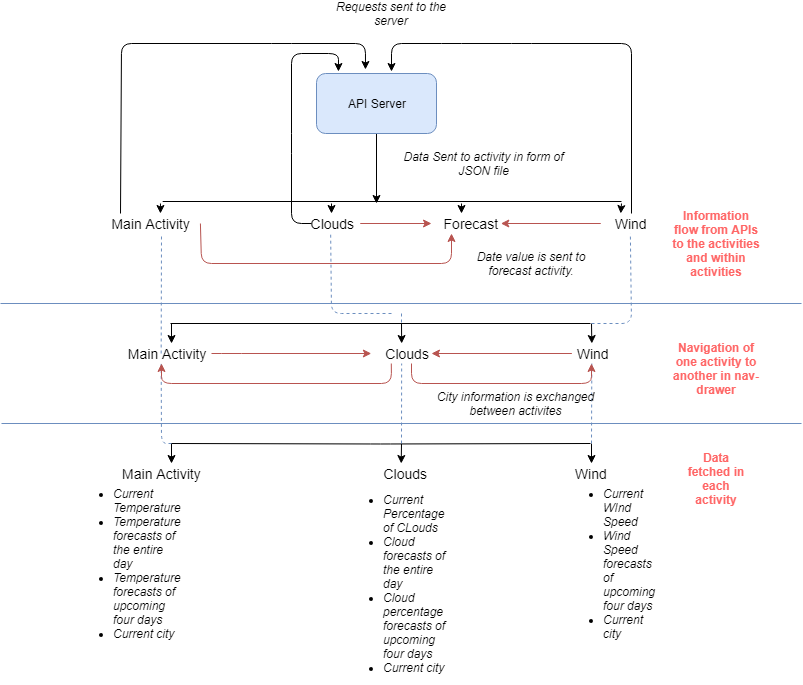


Fig 1. Block Diagram of the system

Implementation

The activity\_main.xml file has been implemented in the following manner.

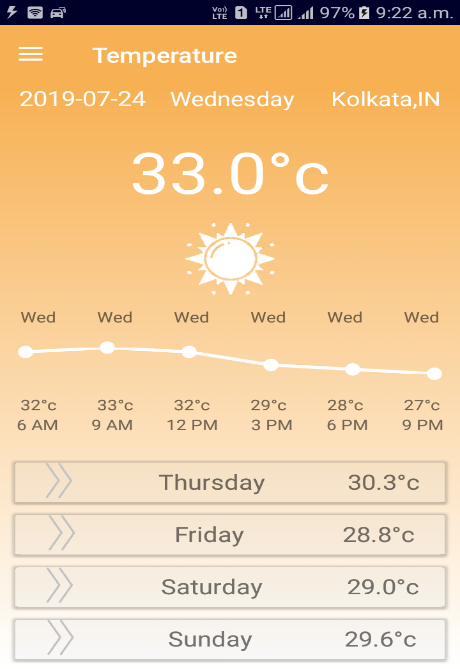
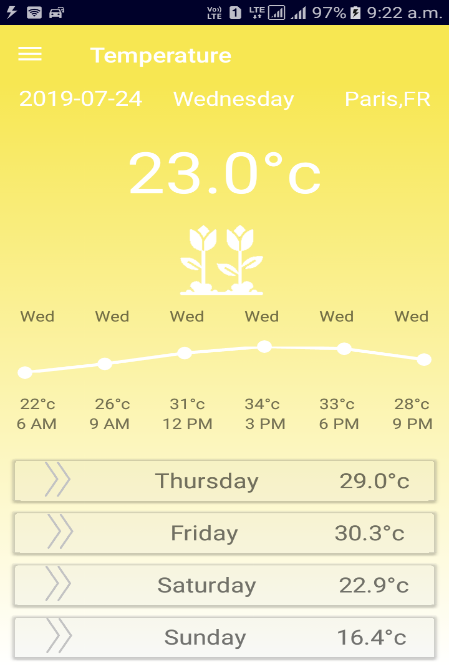
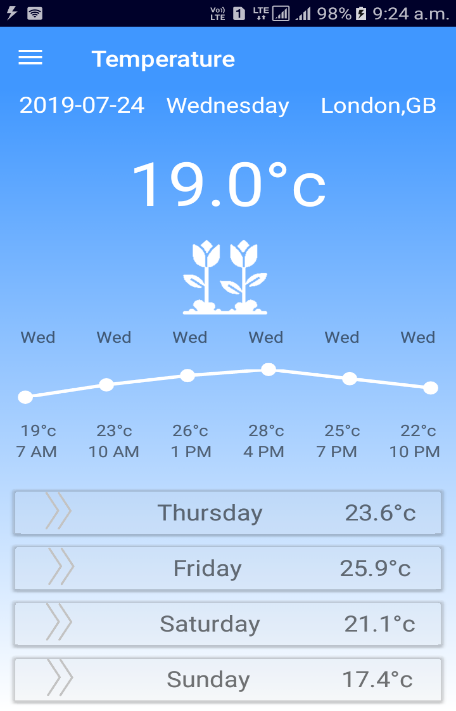
  

Fig 2. Temperature triggers the change of colour of layout and the icon.

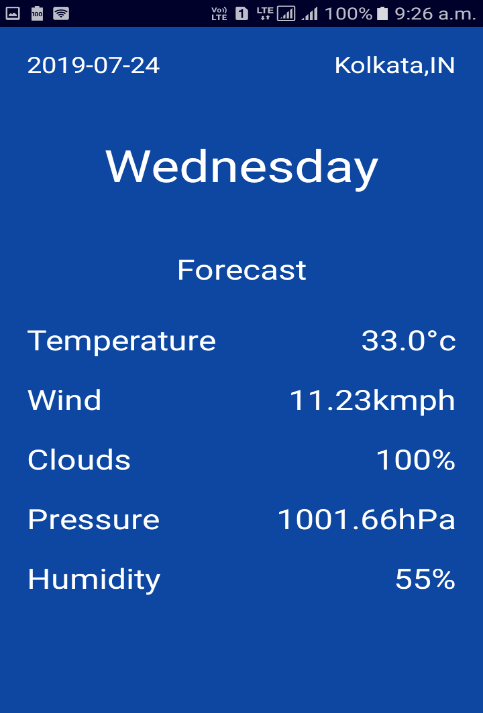


Fig 3. The forecast.xml file that appears after swiping the cards



Fig 4. Cloud.xml file also exhibits change of gradient which is triggered by the cloud percentage



Fig 5. Wind.xml file

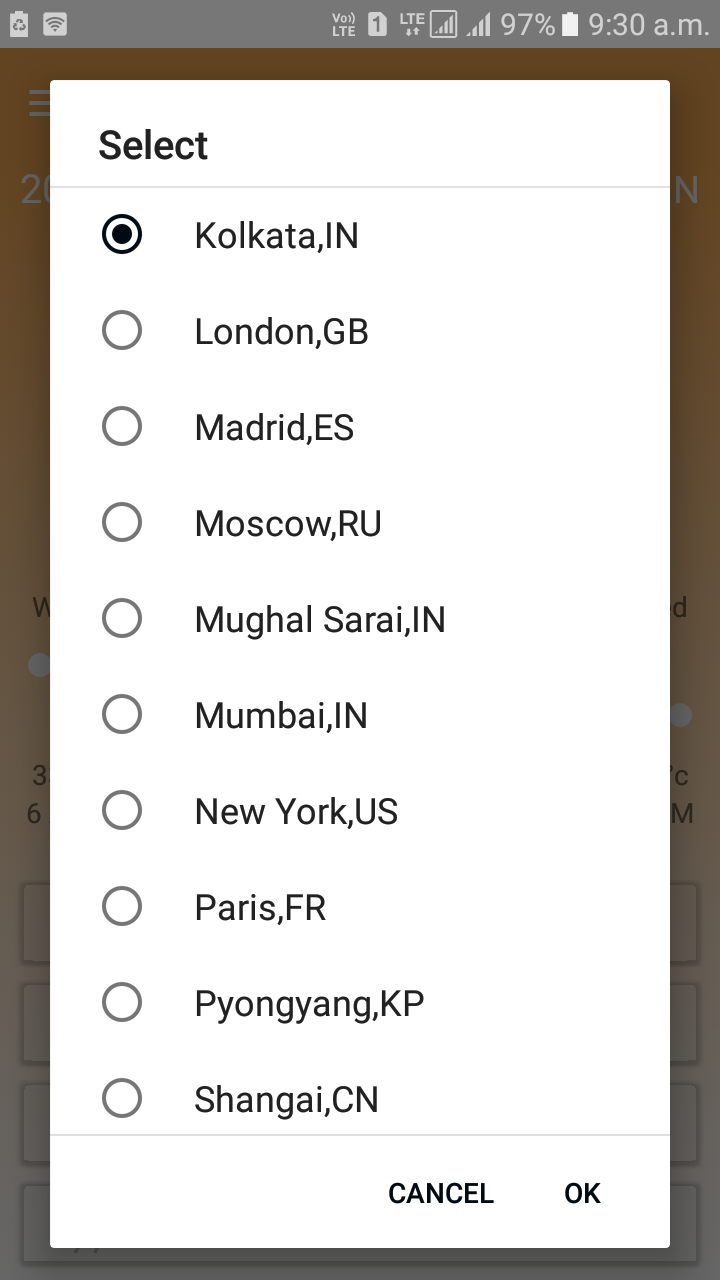
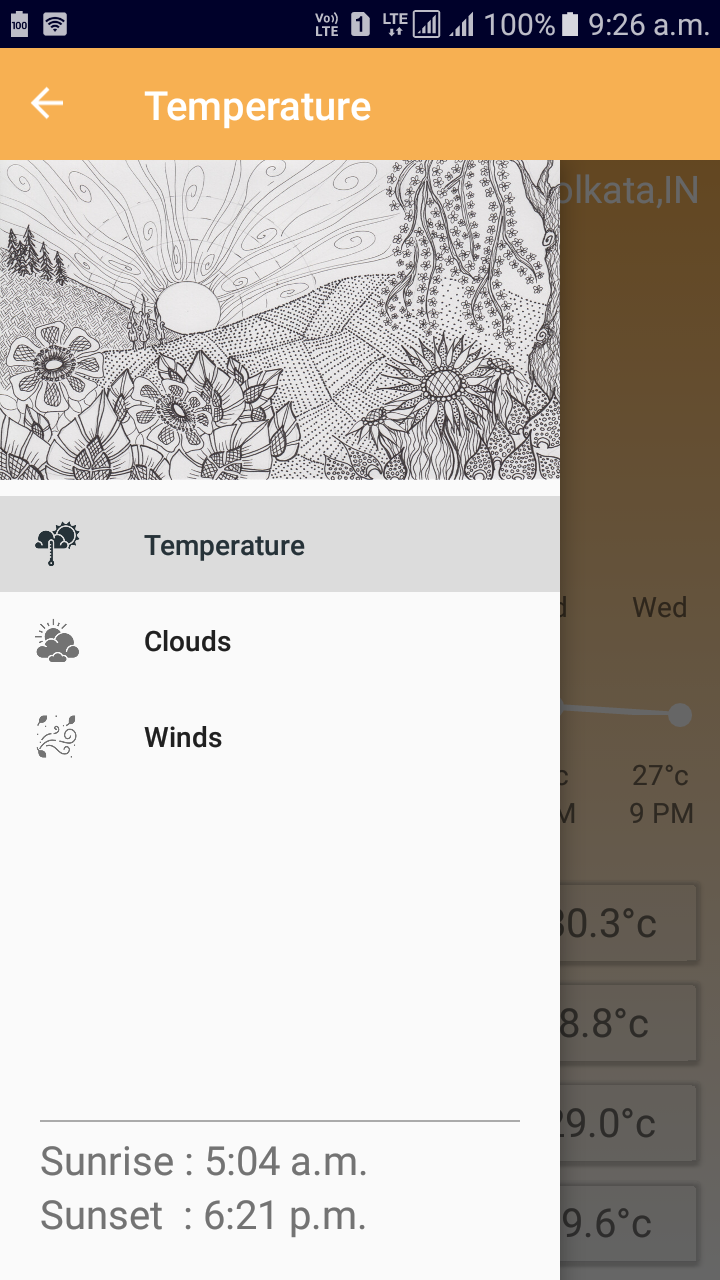


Fig 6. Nav-drawer view Fig 7. Single choice AlertDialog

Conclusion

The application has been constructed keeping in mind a layman’s understanding of the weather, without delving into much complexities. The app meets all of the said requirements. While a lot has been developed, it is still in its beta stage. Firstly the application needs to implement GPS detected located picker that will automatically update the city as per the user’s current location. Moreover the app also needs to provide multilingual support. But most importantly the app needs to develop on the feedback that will be provided by the users.

References

1. The code for AlertDialog was referred from <https://www.androidcode.ninja/android-alertdialog-example/>
2. The code of Line-Chart was referred from <https://www.codingdemos.com/draw-android-line-chart/>
3. The code for converting timings from timestamp to human readable format was referred from the article <https://stackoverflow.com/questions/3371326/java-date-from-unix-timestamp>