Chapter 1

1. INTRODUCTION:

1.1 About Android Studio

The official Integrated Development Environment (IDE) for android application development. Android Studio provides more features that enhance our productivity while building Android apps.Android Studio was announced on 16th May 2013 at the Google I/O conference as an official IDE for Android app development. It started its early access preview from version 0.1 in May 2013. The first stable built version was released in December 2014, starts from version 1.0.Since 7th May 2019, Kotlin is Google39;s preferred language for Android application development. Besides this, other programming languages are supported by Android Studio. Android Studio is the official Integrated Development Environment (IDE) for Android app development,based on Interlay IDEA. On top of Intel liJ39;s powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradlebased build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

1.2 Features of Android

Android is a powerful open-source operating system that open-source provides immense features and some of these are listed below.

- Android Open Source Project so we can customize the OS based on our requirements.
- Android supports different types of connectivity for GSM, CDMA, Wi-Fi, Bluetooth, etc. for telephonic conversation or data transfer.
- Using wife technology we can pair with other devices while playing games or using other applications.
- It contains multiple APIs to support location-tracking services such as GPS.
- We can manage all data storage-related activities by using the file manager.
- It contains a wide range of media supports like AVI, MKV, FLV, MPEG4, etc. to play or record a variety of audio/video.
- It also supports different image formats like JPEG, PNG, GIF, BMP, MP3, etc.
- It supports multimedia hardware control to perform playback or recording using a camera and microphone.

1.3 Project Introduction

A weather app is a software application designed to provide users with up-to-date weather information for a specific location or multiple locations. It serves as a convenient tool for users to access accurate and real-time weather data, enabling them to plan their activities, make informed decisions, and stay prepared for weather changes. Weather apps typically offer a range of features to enhance the user experience and provide comprehensive weather information. They display the current weather conditions, including temperature, humidity, wind speed, precipitation, and visibility. Users can also access short-term and long-term forecasts, including hourly forecasts, daily forecasts, and extended forecasts, allowing them to plan their schedules accordingly. Additionally, weather apps often provide radar images, satellite imagery, and interactive maps, enabling users to visualize weather patterns, track storms, and monitor other weather-related phenomena.One of the essential functions of a weather app is to send notifications and alerts for severe weather conditions.

1.4 Objective

- Accessibility: The weather app should be easily accessible to users on various platforms, such as mobile devices, desktop computers, and web browsers.
- Real-time Data: The app should retrieve weather data from reliable sources and provide users with real-time updates on current weather conditions
- Forecasts: The app should provide short-term and long-term weather forecasts, including hourly, daily, and extended forecasts.
- Maps and Radar: The app should provide radar images, satellite imagery, and interactive maps to visualize weather patterns, track storms, and monitor other weather-related phenomena.
- Location-based Weather: The app should offer the option for users to set their preferred location(s) and receive weather updates specific to those locations.
- Reliability and Accuracy: The weather app should rely on reputable data sources to ensure the accuracy and reliability of the weather information provided.

1.5 Advantages

- Accurate and Real-Time Information.
- Convenience and Accessibility.
- Forecasting and Planning.
- Severe Weather Alerts.
- Radar and Maps.
- Customization.
- Multiple Location Support.
- Educational and Recreational Value.

Chapter 2

2. REQUIREMENT ANALYSIS

2.1 Hardware Requirements

System	PC
Processor	i5 0r i7
RAM	8Gb or more
Memory	512mb
Hard Disk	256Gb

2.2 Software Requirements

Operating System	windows 10 or Ubuntu
Programming Language	Java,XML
Emulator	AVD
Software	Android Studio

2.3 Functional Requirements

his section gives a high-level summary of the system requirements. The system will be able to implement a variety of functional modules, including:

User Module: Smartphone, Internet Connection, Installed App.

Mobile Module: Android version, Hardware requirements, Installed App.

Weather App Module: User Service, Touch Feature, Request Information, Verify Location.

API Module: Request Location, Search Location, Verify Location, Weather Data.

2.4 Non Functional Requirements

- **(i)** Proper application termination and usage of limited space in memory.
- **(ii)** Availability 24 hours a day, 7 days a week.
- **(iii)** Improved component design to improve peak performance.
- **(iv)** Live weather updates and refresh within a fraction of minutes.
- **(v)** For future expansion, a flexible service-based architecture will be particularly desired.
- **(vi)** System properties and limitations are defined by Non-Functional Requirements: Security, Reliability, Maintainability, Portability, Extensibility, Reusability, Compatibility, and Resource Utilization are some of the additional non-functional requirements.

2.5 XML

stands for Extensible Markup Language. XML is a markup language much like HTML used to describe data. It is derived from Standard Generalized Markup Language(SGML). Basically, the XML tags are not predefined in XML. We need to implement and define the tags in XML. XML tags define the data and used to store and organize data. It's easily scalable and simple to develop. In Android, the XML is used to implement UI-related data, and it's a lightweight markup language that doesn't make layout heavy. XML only contains tags, while implementing they need to be just invoked.

2.6 Java Programming Language

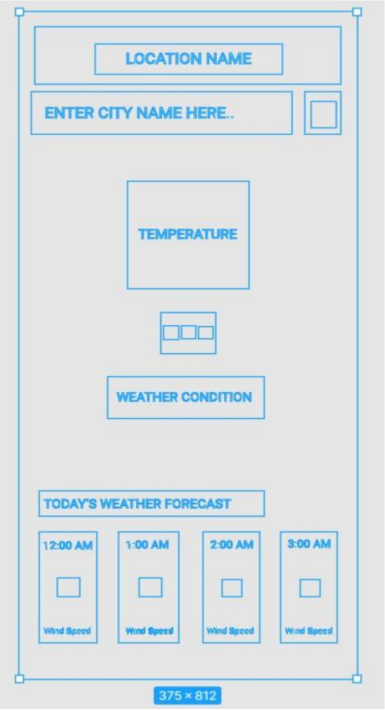
Android App are mostly developed in JAVA language using Android SDK (Software Development Kit). Other languages like C, C++, Scala etc. can also be used for developing Android App, but JAVA is most preferred and mostly used programming language for Android App Development. It is class based and object oriented programming whose syntax is influenced by C++. The primary goals of JAVA is to be simple, object-oriented, robust, secure and high level.JAVA application runs on JVM (JAVA Virtual Machine) but Android has it's own virtual machine called Dalvik Virtual Machine (DVM) optimized for mobile devices.

Chapter 3

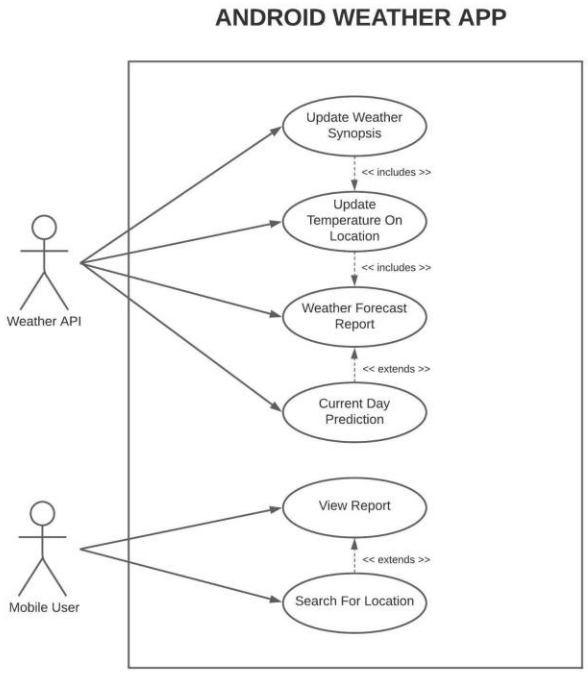
3. METHODOLOGY AND DESIGN

When you launch the weather app, it requests permission to access your current location. If access is denied, the application will display a toast message before terminating and closing. When a user grants current location access, the weather forecast, temperature, and climatic state of that current place are displayed. There is also a search box where you may look for any location in the world. A toast message displaying "Please Enter City Name" validates the empty entries. If an invalid city is submitted, a toast message with the text "Invalid City name entered" appears. If a valid city name is provided, it will locate the device's coordinates (longitude and latitude). The data will then be sent to the API using an API key. The Weather API (WeatherBit API Keys) will provide us a JSON file from which we will extract the necessary data, which is the weather data, temperature, and city of the location. It displays the current temperature, weather description, climatic conditions, and forecasts the entire day's weather.

3.1 App Layout/Design :

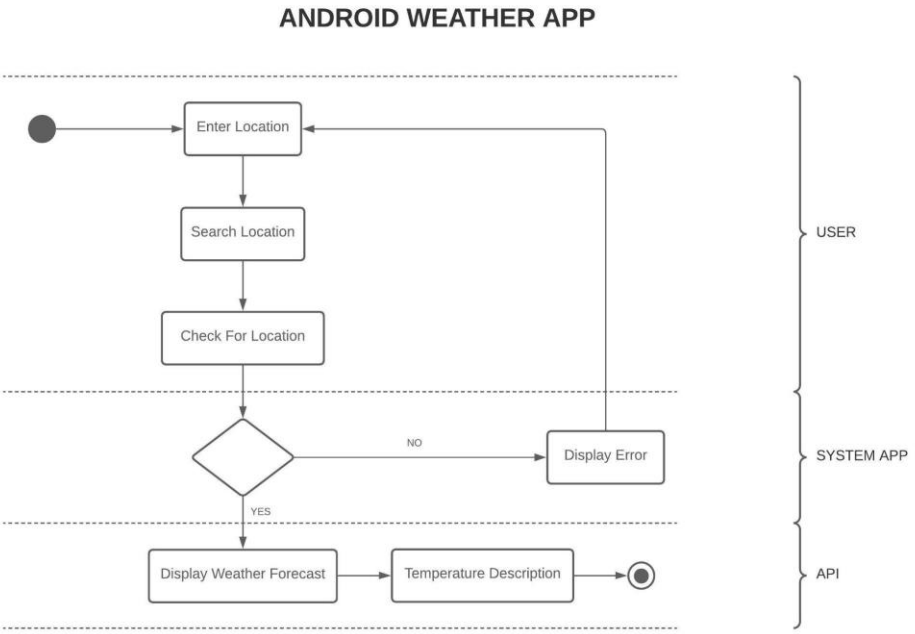


3.2 Use caseDiagram :



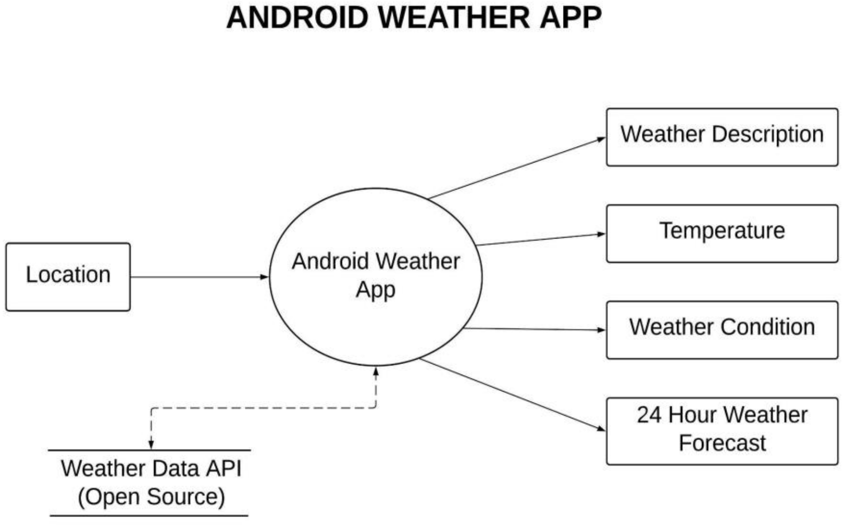
USE CASE DIAGRAM

3.3 Activity Daigram :



ACTIVITY DIAGRAM

3.4 Data Flow Daigram :

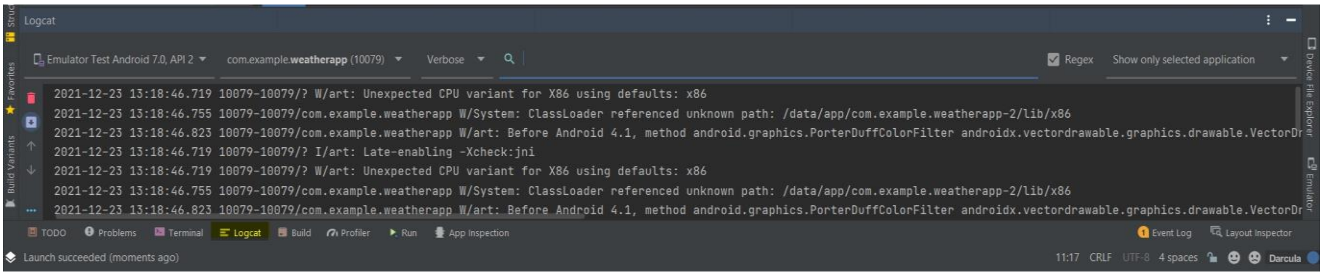


Data Flow Diagram

Chapter 4

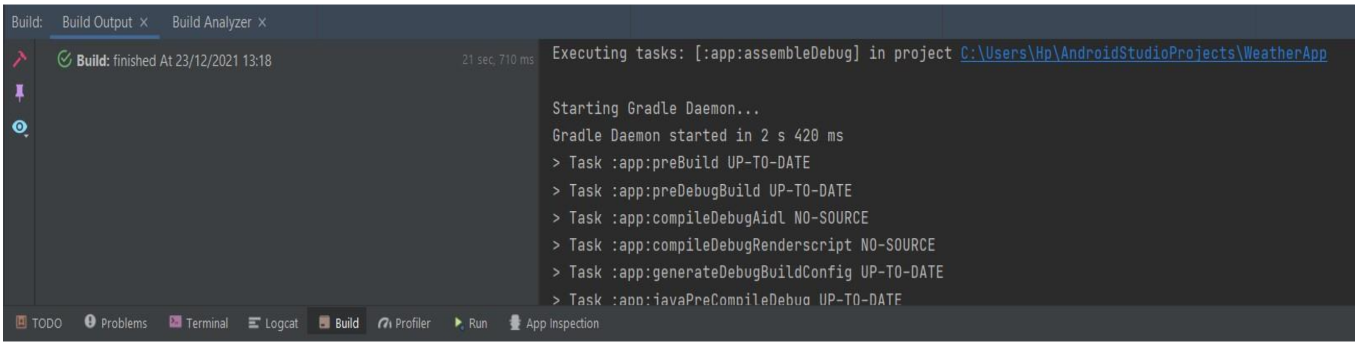
4. APP TESTING AND DEBUGGING

Since we are outlined to use the Incremental Model of SoftwareEngineering, Testing and Development of the modules is done in parallel.After coding the Project, testing of the software begins with the Unit Testing method.



```
2021-12-23 13:18:46.719 10079-10079/? W/art: Unexpected CPU variant for X86 using defaults: x86
2021-12-23 13:18:46.755 10079-10079/com.example.weatherapp W/System: ClassLoader referenced unknown path: /data/app/com.example.weatherapp-2/lib/x86
2021-12-23 13:18:46.823 10079-10079/com.example.weatherapp W/art: Before Android 4.1, method android.graphics.PorterDuffColorFilter androidx.vectordrawable.graphics.drawable.VectorDr
2021-12-23 13:18:46.719 10079-10079/? I/art: Late-enabling -Xcheck:jni
2021-12-23 13:18:46.719 10079-10079/? W/art: Unexpected CPU variant for X86 using defaults: x86
2021-12-23 13:18:46.755 10079-10079/com.example.weatherapp W/System: ClassLoader referenced unknown path: /data/app/com.example.weatherapp-2/lib/x86
2021-12-23 13:18:46.823 10079-10079/com.example.weatherapp W/art: Before Android 4.1, method android.graphics.PorterDuffColorFilter androidx.vectordrawable.graphics.drawable.VectorDr
```

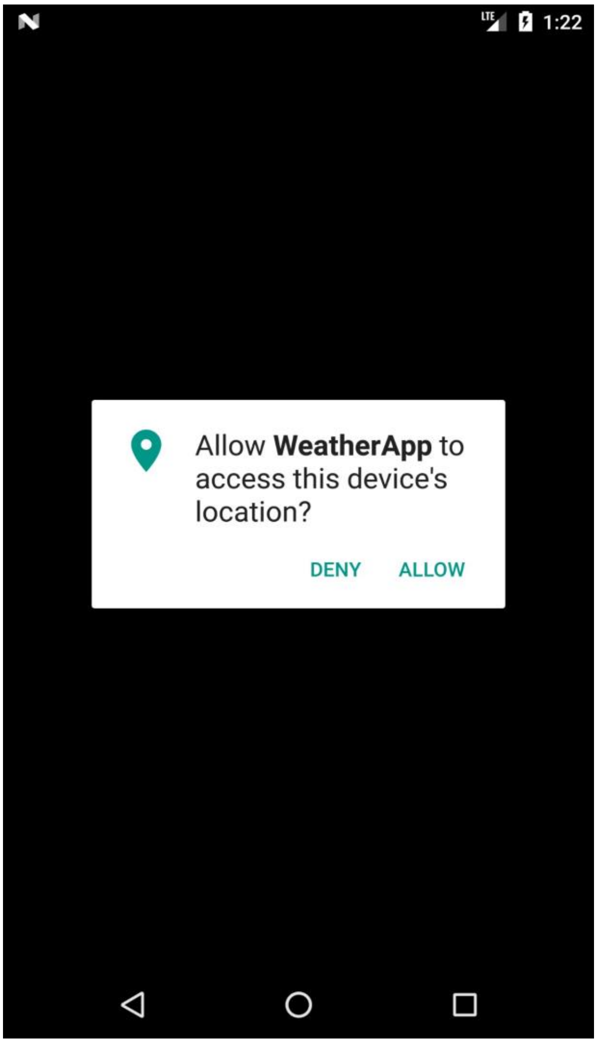
Fig : Debugging and checking for errors in Logcat window in Android Studio



Build: finished At 23/12/2021 13:18 21 sec, 710 ms

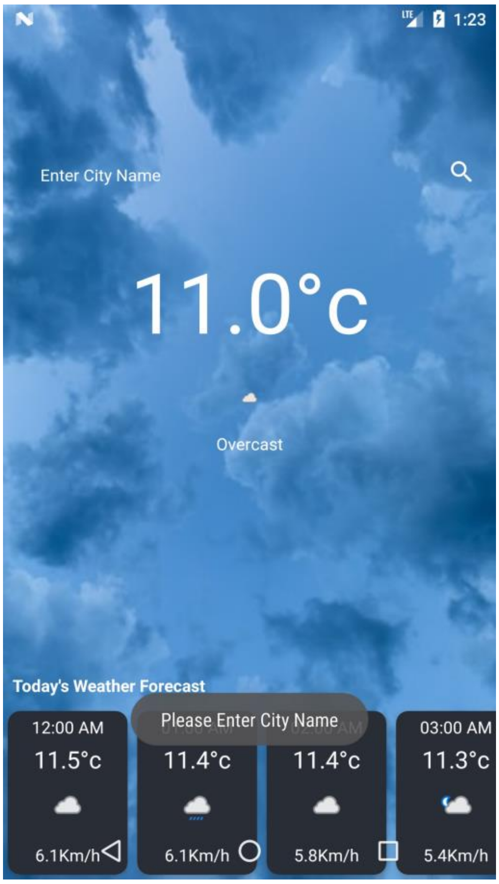
```
Executing tasks: [:app:assembleDebug] in project C:\Users\Hp\AndroidStudioProjects\WeatherApp

Starting Gradle Daemon...
Gradle Daemon started in 2 s 420 ms
> Task :app:preBuild UP-TO-DATE
> Task :app:preDebugBuild UP-TO-DATE
> Task :app:compileDebugAidl NO-SOURCE
> Task :app:compileDebugRenderscript NO-SOURCE
> Task :app:generateDebugBuildConfig UP-TO-DATE
> Task :app:javaPreCompileDebug UP-TO-DATE
```

Fig : Debugging through Build Output Panel in Android Studio



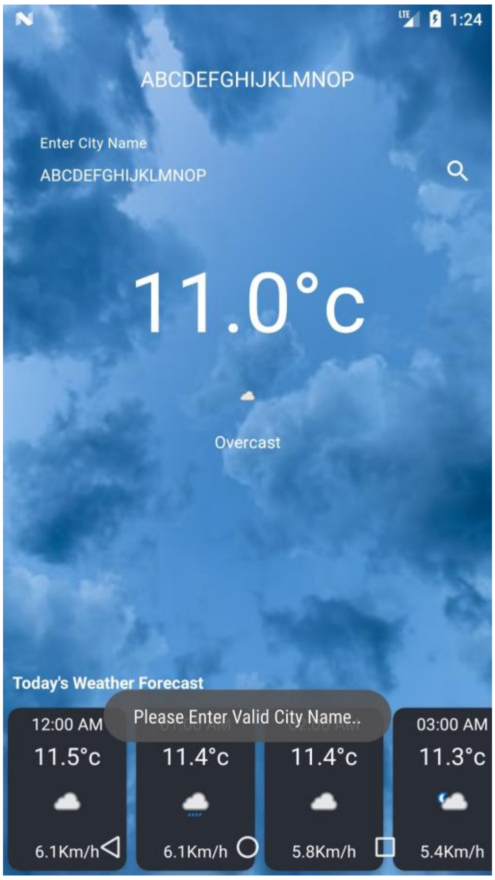
Allow **WeatherApp** to access this device's location?

Fig : App requesting for permissions of user location



Please Enter City Name

Fig : Empty User input field validation



Enter City Name: ABCDEFGHIJKLMNOP

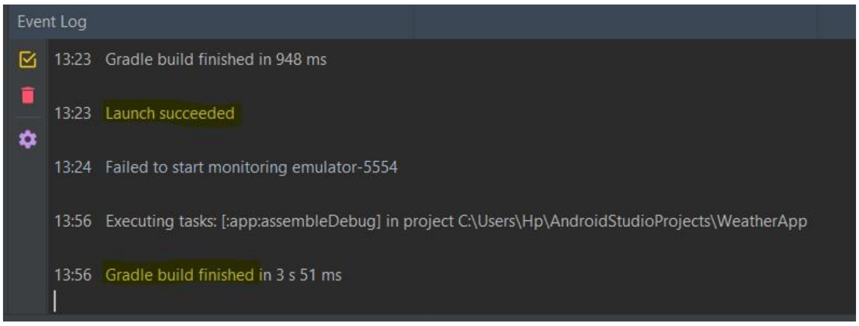
Please Enter Valid City Name..

Fig : Validating user input search on invalid location

Chapter 5

5. RESULTS DISCUSSION:

Below are the output screenshots showcasing working of the app based on user controls.



```
Event Log
13:23  Gradle build finished in 948 ms

13:23  Launch succeeded

13:24  Failed to start monitoring emulator-5554

13:56  Executing tasks: [:app:assembleDebug] in project C:\Users\Hp\AndroidStudioProjects\WeatherApp

13:56  Gradle build finished in 3 s 51 ms
```

Fig : Clicking Make Project in Android Build options

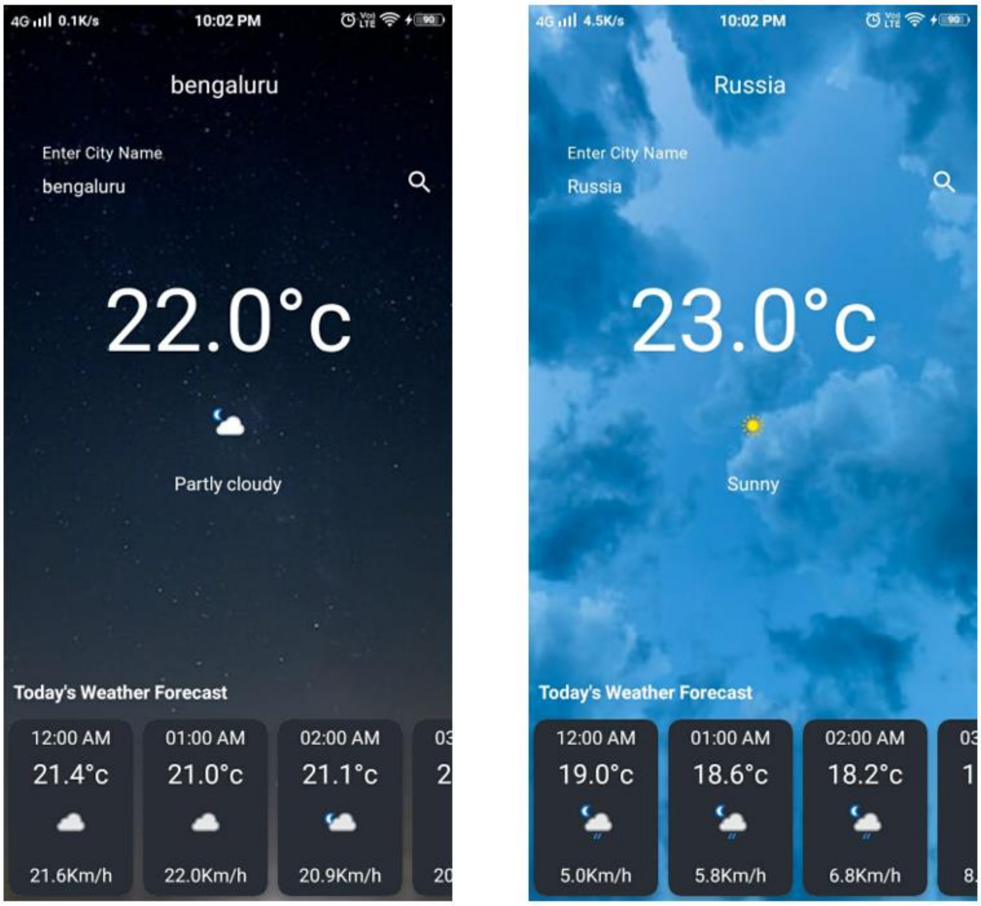
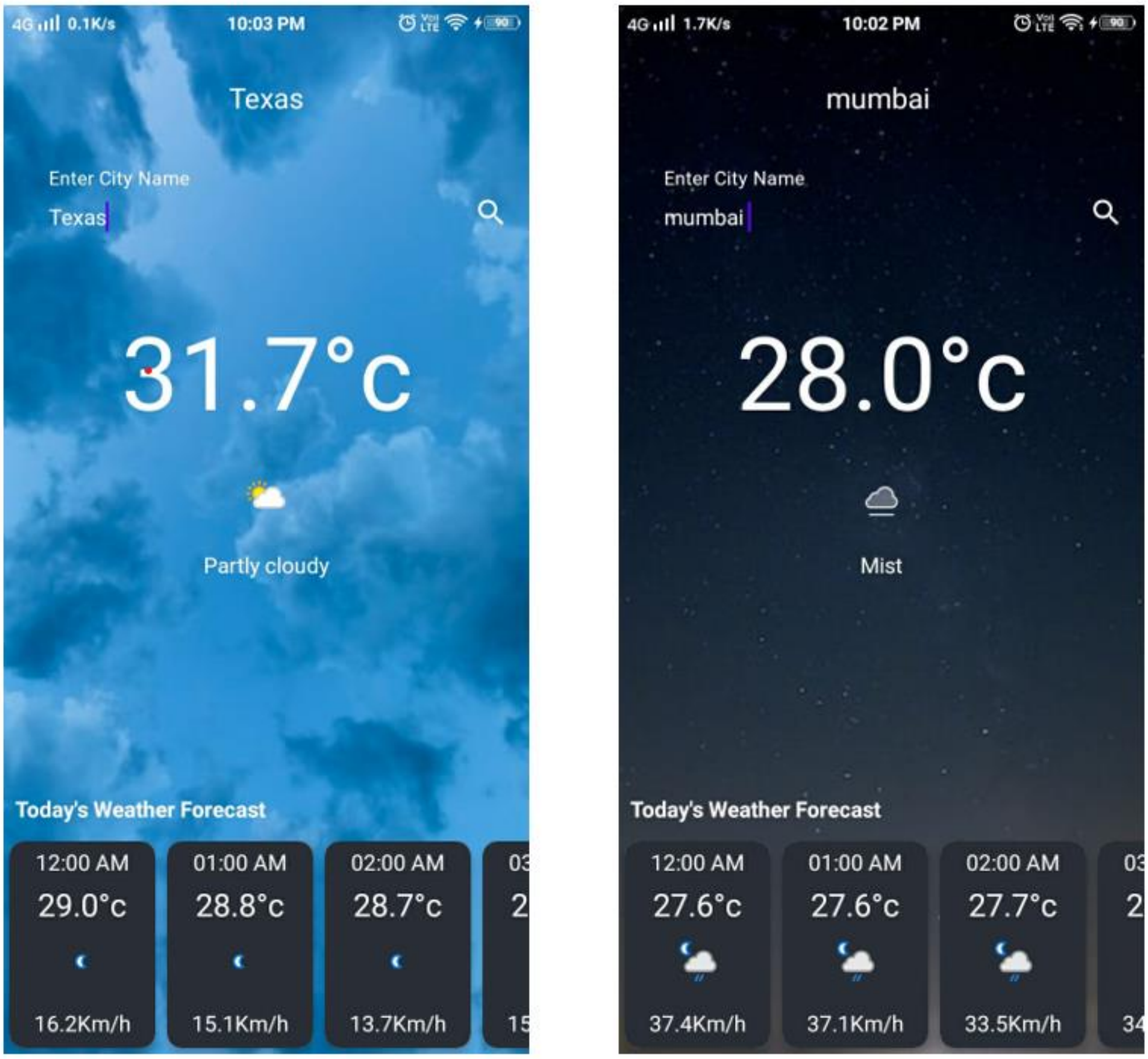


Fig : Weather Description at Places - Bengaluru and Russia



4G 0.1K/s 10:03 PM 90

Texas

Enter City Name

Texas

31.7°c

Partly cloudy

Today's Weather Forecast

12:00 AM 29.0°c 16.2Km/h

01:00 AM 28.8°c 15.1Km/h

02:00 AM 28.7°c 13.7Km/h

03 2 15

4G 1.7K/s 10:02 PM 90

mumbai

Enter City Name

mumbai

28.0°c

Mist

Today's Weather Forecast

12:00 AM 27.6°c 37.4Km/h

01:00 AM 27.6°c 37.1Km/h

02:00 AM 27.7°c 33.5Km/h

03 2 34

Fig : Weather Report at Places - Texas and Mumbai

Above screenshots displays weather conditions like weather description, temperature of a location, weather forecast of various cities across the globe.

Chapter 6

6. FUTURE SCOPE:

1. Enhanced Forecast Accuracy: Weather apps will benefit from ongoing advancements in forecasting models and data analysis techniques. With more sophisticated algorithms, machine learning, and access to vast amounts of weather data, future weather apps will deliver increasingly accurate and reliable forecasts.

2. Hyperlocal Weather Data: Weather apps will focus on delivering hyperlocal weather information, providing forecasts and real-time conditions for smaller regions.

3. Advanced Visualization: Future weather apps will leverage advanced visualization techniques to present weather information in more immersive and intuitive ways.

4. Personalized Insights and Recommendations: Weather apps will become more personalized, tailoring weather information to individual preferences, interests, and activities.

5. Integration with Smart Devices: As smart devices and IoT technologies continue to proliferate, weather apps will seamlessly integrate with these platforms.

6. Climate Change Monitoring: Weather apps will increasingly incorporate data and information related to climate change.

7. Social and Community Features: Weather apps will foster a sense of community by integrating social features.

8. Environmental and Health Integration: Weather apps will incorporate environmental and health-related data to provide a comprehensive understanding of weather impacts.

9. Energy Optimization: Weather apps will play a role in optimizing energy consumption and efficiency.

CONCLUSION

In conclusion, a weather app serves as a valuable tool for individuals seeking accurate and up-to-date information about weather conditions. It provides users with various features, including current temperature, forecast, radar maps, and alerts, enabling them to plan their activities and make informed decisions based on the weather forecast.

Weather apps offer convenience and accessibility, as they can be easily downloaded and installed on smartphones, tablets, and other devices. Users can personalize their settings, receive notifications, and access weather information from anywhere at any time.

Moreover, weather apps often use advanced meteorological models and data sources to provide reliable forecasts, increasing their accuracy and helping users anticipate weather changes. Historical weather data and trends are also available, allowing users to track patterns and plan accordingly.

Weather apps often offer additional features such as air quality index, UV index, pollen count, and sunrise/sunset times, catering to users' diverse needs and interests. These supplementary details provide valuable information for outdoor enthusiasts, travelers, and individuals with specific health concerns.

REFERENCES

[1] Textbook - Android for programmers, an app-driven approach developer series by Paul Deitel,Harvey Deitel, Abbey Deitel,Michael.

[2] Abhi Android Tutorials.

[3] Weather Apps UML Diagram.

[4] Open Weather Map - for free Weather data.

[5] GeeksforGeeks - Android Applications to APK.