



Green University of Bangladesh

*Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2023), B.Sc. in CSE (Day)*

Weather Forecasting App

*Course Title: Mobile Application Development Lab
Course Code: CSE 426
Section: PC 202*

Students Details

Name	ID
Md. Manzurul Alam	202902003

*Submission Date: 21/06/23
Course Teacher's Name: Md. Shihab Hossain*

[For teachers use only: **Don't write anything inside this box**]

<u>Lab Project Status</u>	
Marks:	Signature:
Comments:	Date:

Contents

1	Introduction	2
1.1	Overview	2
1.2	Motivation	2
1.3	Design Goals/Objectives	2
1.4	Application	3
2	Design/Development/Implementation of the Project	4
2.1	Introduction	4
2.2	Project Details	4
2.3	Front-End Implementation	4
2.4	Back-end Implementation	5
3	Performance Evaluation	6
3.1	User Interface	6
3.2	Project Simulation	6
4	Conclusion	8
4.1	Discussion	8
4.2	Limitations	8
4.3	Scope of Future Work	8

Chapter 1

Introduction

1.1 Overview

The Weather Forecasting App is an Android application developed using XML, Java, and API in Android Studio. It allows users to enter a city name and retrieves various weather-related information such as sunrise time, sunset time, humidity, pressure, wind speed, minimum temperature, and maximum temperature. The app utilizes an API to fetch real-time weather data for the specified location and presents it to the users in a user-friendly manner.

1.2 Motivation

The motivation behind choosing the Weather Forecasting App project is to provide users with a convenient and reliable tool for accessing weather information on their Android devices. Weather plays a crucial role in our daily lives, affecting our clothing choices, outdoor activities, and overall planning. By creating this app, we aim to empower users with accurate and up-to-date weather data, enabling them to make informed decisions based on weather conditions.

1.3 Design Goals/Objectives

The design goals and objectives of the Weather Forecasting App project include:

- Provide an intuitive and user-friendly interface for entering and retrieving weather information.
- Retrieve accurate and up-to-date weather data from a reliable API source.
- Display weather information in a visually appealing and easily understandable manner.
- Incorporate features like sunrise and sunset times, humidity, pressure, wind speed, and temperature range to provide comprehensive weather details.

- Optimize the app for efficient performance and responsiveness.
- Real-time tracking of vehicle locations within the GUB campus
- User-friendly mobile application for accessing tracking information
- Integration with existing university systems and infrastructure
- Reliable and secure data storage and communication
- Customizable features and settings for different user roles

1.4 Application

The design goals and objectives of the Weather Forecasting App project include:

- **Daily Weather Updates:** Users can use the app to stay updated with the current weather conditions in their location or any desired city.
- **Travel Planning:** The app can assist travelers in deciding what to pack and planning their activities based on the weather forecast of their travel destination.
- **Outdoor Events and Activities:** Event organizers and individuals planning outdoor activities can utilize the app to determine the weather conditions during the scheduled time, ensuring a smooth and enjoyable experience.
- **Agriculture and Farming:** Farmers can utilize the weather information provided by the app to make informed decisions about crop planting, irrigation, and pest control measures based on the weather patterns.
- **Safety Precautions:** The app can help users take necessary safety precautions during extreme weather conditions like storms, hurricanes, or heatwaves.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

In this section, we will discuss the design, development, and implementation aspects of the "Weather Forecasting App" project [2] [3] [4]. We will provide a general overview of the project's design considerations and highlight the key development and implementation decisions.

2.2 Project Details

The functional requirements outline the specific functionalities and features of the transport tracking system. These requirements define what the system should do to meet the needs of its users.

2.3 Front-End Implementation

The frontend implementation of the "Weather Forecasting App" project involves creating a user interface using XML layouts and implementing the necessary functionality using Java in Android Studio. The project's frontend consists of several key steps. Firstly, XML layout files are created to define the visual structure and components of the app's screens. These layouts include the main activity layout where users can enter the city name and view weather information, as well as additional screens or dialogs for displaying results or error messages. UI components such as TextViews, EditTexts, Buttons, and ImageViews are added to the XML layouts, representing different elements of the weather information interface. The appearance and positioning of these components are customized using XML attributes. Resource files, such as images for icons or backgrounds, are included in the project's resource directory. A Java class, typically named MainActivity, is created to handle the logic and functionality of the UI components. View binding is implemented to access the UI components defined in

the XML layouts from the activity class, simplifying the process of manipulating UI elements. User interactions, such as button clicks or text input changes, are captured through event listeners or callbacks, triggering actions such as retrieving weather data. API integration is essential, making use of appropriate libraries and methods to make HTTP requests to the weather API, passing the user's chosen city name. Error handling mechanisms are implemented to address API request failures or invalid data, displaying relevant error messages to the user. The retrieved weather information is then displayed by updating the appropriate UI components, ensuring clear formatting and user-friendly presentation. Through testing and refinement, the frontend implementation is optimized for a smooth user experience, incorporating user feedback and enhancing the UI design and functionality as needed.

2.4 Back-end Implementation

The backend implementation of the "Weather Forecasting App" project involves handling data retrieval and processing tasks using Java and API integration. The backend functionality is responsible for making HTTP requests to the weather API, retrieving weather data based on the user's chosen city name. The API key, necessary for authentication and access to the weather API, is securely stored and used in the backend code. To handle the API request and response, a background task or asynchronous process is typically implemented to avoid blocking the main UI thread. This allows the app to remain responsive while waiting for the API response. The backend code retrieves the JSON response from the API request and parses it to extract the required weather information, such as sunrise and sunset times, humidity, pressure, wind speed, minimum and maximum temperatures, etc. JSON parsing libraries or built-in methods are utilized to efficiently extract the desired data from the response. Error handling mechanisms are implemented to handle cases of failed API requests or unexpected response formats, providing appropriate error messages to the user. The retrieved weather data is then passed back to the frontend for display. Additionally, the backend code may include data validation and conversion logic to ensure the received data is in the expected format and units. Throughout the development process, testing and debugging are performed to ensure the backend code functions correctly and reliably retrieves and processes weather data. Continuous improvement and maintenance of the backend implementation may involve monitoring API updates, addressing any changes in response formats, and incorporating new features or enhancements based on user feedback and requirements.

Chapter 3

Performance Evaluation

3.1 User Interface

The user interface (UI) of the "Weather Forecasting App" project allows users to easily retrieve and view weather information. By entering the city name in the search input field and tapping the search button, users can initiate the retrieval process. The UI provides visual cues, such as loading indicators, to indicate the progress of data retrieval. Once the weather data is obtained, it is displayed on the screen in a structured format, including details like sunrise and sunset times, humidity, pressure, wind speed, minimum and maximum temperatures. The UI is designed to be responsive and adaptable to different screen sizes and orientations. Users can seamlessly enter new city names to check weather information for different locations. Overall, the UI provides a straightforward and user-friendly experience for accessing and interpreting weather data.

3.2 Project Simulation

The simulation of the "Weather Forecasting App" project involves setting up a realistic testing environment to evaluate the application's performance and functionality. This includes installing the necessary software components, configuring dependencies, and creating test scenarios. The simulation focuses on testing various inputs, verifying data accuracy, assessing responsiveness, and analyzing error handling. Performance metrics are measured, including response time and compatibility across different devices. Issues and bugs are identified through logs and debug outputs, leading to optimizations and bug fixes. Screenshots and videos are captured to document the app's behavior. By conducting a thorough simulation, the project's performance and reliability can be assessed, ensuring the delivery of a high-quality weather application.

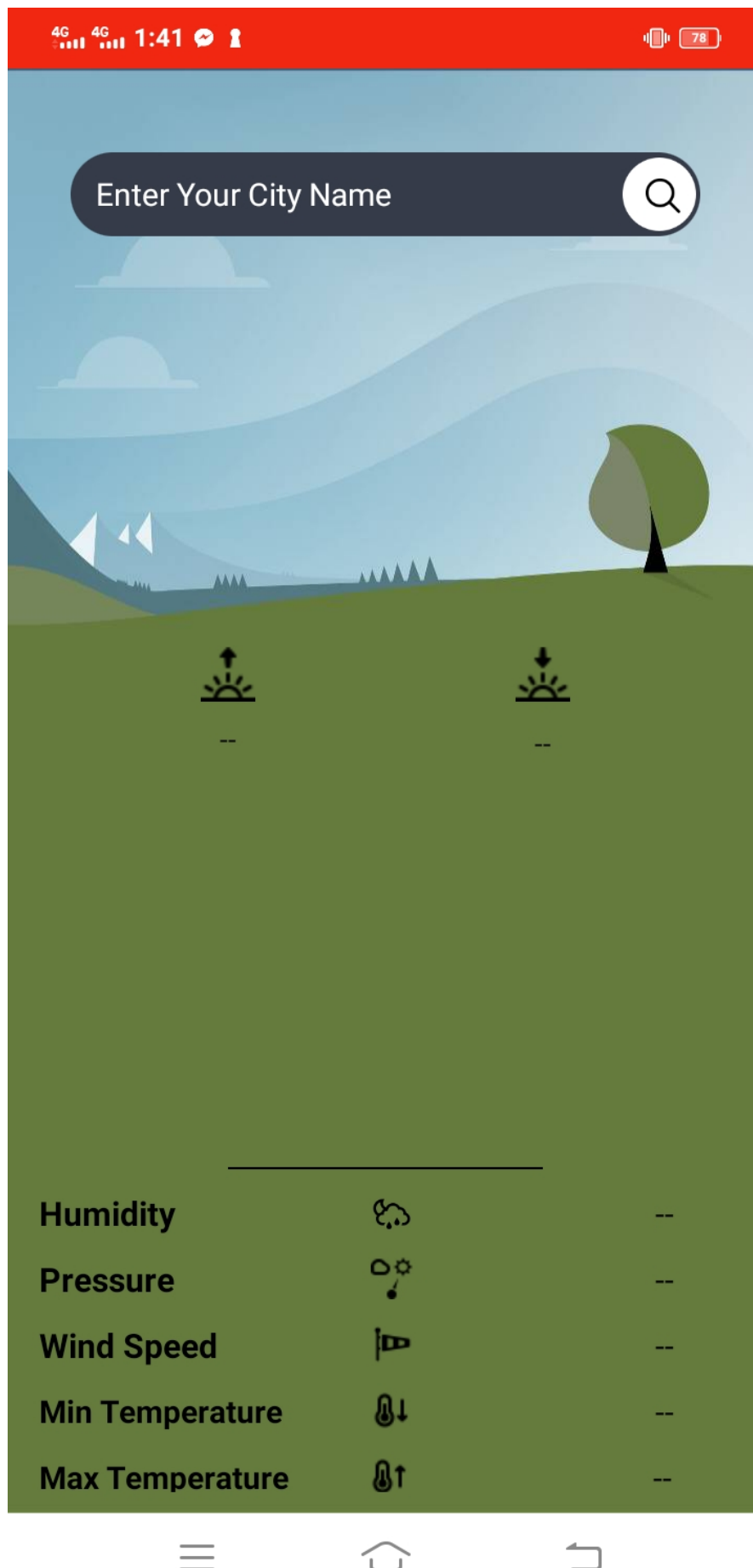


Figure 3.1: User interface

Chapter 4

Conclusion

4.1 Discussion

In this chapter, we provide a summary of the work done in the "Weather Forecasting App" project and discuss the results and observations. The project successfully implemented an Android application using XML, Java, and API integration in Android Studio. The application allows users to enter a city name and retrieves important weather information such as sunrise and sunset times, humidity, pressure, wind speed, minimum temperature, and maximum temperature. The retrieved data is displayed in a user-friendly format, providing users with valuable weather insights.

4.2 Limitations

Despite the successful implementation of the project, there are certain limitations that need to be acknowledged. Firstly, the accuracy and reliability of the weather data depend on the accuracy of the API used. Any discrepancies or inaccuracies in the API's data may affect the reliability of the information presented to the users. Additionally, the project's scope is limited to providing weather information for a single city at a time. Scaling the application to handle multiple cities or incorporating advanced forecasting algorithms could be considered as future enhancement.

4.3 Scope of Future Work

There is significant scope for future work and extensions to enhance the capabilities of the "Weather Forecasting App" project. Some possible areas for future work include:

- **User Personalization:** Implementing user accounts and preferences to allow users to save their favorite cities, set custom notifications for weather updates, and personalize the application according to their preferences.
- **Extended Forecasting:** Integrating advanced forecasting algorithms and models to provide more accurate and detailed weather predictions for longer time periods.

This could include hourly forecasts, extended day forecasts, or even long-term climate predictions.

- **Real-Time Updates:** Enhancing the application to provide real-time weather updates by implementing push notifications or automatic data refresh at regular intervals.
- **Location-Based Services:** Leveraging GPS or location-based services to automatically detect the user's current location and provide weather information without the need for manual city input.
- **Data Visualization:** Developing interactive and visually appealing data visualizations, such as charts, graphs, and maps, to present weather trends and patterns in a more intuitive and informative manner.

By addressing these future work areas, the "Weather Forecasting App" can be further improved and expanded to offer enhanced user experiences and more comprehensive weather-related features. In conclusion, the "Weather Forecasting App" project has successfully developed an Android application that allows users to retrieve and display important weather information based on the city name input. While limitations exist, there are ample opportunities for future work and extensions to improve the application's functionality, accuracy, and user experience.