Lambton College

Proposal Structure and Content

Student Name: Unni Krishna Prasad Endla

Student Number: C0930172

Course Name: Python Programming Project

Title: Weather Forecast Application

Course Code: CSD - 4523

Date: 07-18-2024

1. Introduction

Project Overview: The proposed project aims to develop a command-line application that retrieves weather data from an API (e.g., OpenWeatherMap) based on user input (city or zip code). This application will provide users with quick access to current and extended weather information, making it an essential tool for planning activities and understanding current conditions

Motivation: Weather forecasting is crucial for various aspects of daily life, such as planning outdoor activities, scheduling events, and understanding weather-related risks. With the increasing availability of weather data through APIs, it is now possible to create a user-friendly application that provides instant access to weather information. This application will cater to the needs of general users who require quick and reliable weather updates for specific locations.

2. Project Scope

Features:

Input Validation: The application will validate user input to ensure that it is a valid city or zip code.

Current Weather Display: The application will display the current weather conditions, including temperature, humidity, wind speed, and other relevant details.

Extended Forecast: The application will provide an extended forecast for the next few days, including temperature, precipitation, and other weather conditions.

Error Handling: The application will handle errors and exceptions, such as API downtime, invalid user input, and network connectivity issues.

Limitations: The project will focus on retrieving weather data from a single API source (e.g., OpenWeatherMap) and handling basic user interactions through the console. The application will not provide advanced features, such as weather alerts, radar imagery, or detailed weather analysis.

Target Audience: The target audience for this application is general users who need instant weather updates for specific locations. This includes individuals, travelers, and professionals who require quick access to weather information for planning and decision-making.

3. Technical Details

Technology Stack:

Python: The application will be built using Python as the primary programming language.

requests library: The requests library will be used for API interaction and data retrieval.

JSON parsing: JSON parsing will be used for handling and processing weather data.

Architecture: The application will follow a single-tier architecture, with a console-based interface for user interaction.

Modules and Components:

User Input Handling Module: This module will handle user input, validate it, and pass it to the API interaction module.

API Interaction Module: This module will interact with the OpenWeatherMap API, retrieve weather data, and pass it to the data parsing module.

Data Parsing and Formatting Module: This module will parse and format the retrieved weather data into a user-friendly format.

4. Implementation Plan

Development Phases:

Phase 1: Setup Environment and Project Structure

Set up Python environment and dependencies

Create project structure and organize files

Phase 2: Implement User Input Handling and Basic Error Checking

Implement user input handling module

Validate user input and handle basic errors

Phase 3: Integrate API Interaction and Data Parsing

Implement API interaction module

Integrate data parsing and formatting module

Phase 4: Implement Current Weather Display Functionality

Implement current weather display module

Display current weather conditions

Phase 5: Add Extended Forecast Feature

Implement extended forecast module

Display extended forecast for the next few days

Phase 6: Implement Robust Error Handling and Edge Case Testing

Implement robust error handling for API downtime, invalid user input, and network connectivity issues

Conduct edge case testing for various user inputs and error scenarios

Phase 7: Documentation and Final Testing .Prepare documentation for the application

Conduct final testing and debugging

Task Allocation:

Task 1: Setting up Python environment and dependencies (Day 1)

Task 2: Implementing user input handling (Day 2 – 3)

Task 3: Integrating API interaction (Day 4 – 5)

Task 4: Developing current weather display (Day 6 – 7)

Task 5: Adding extended forecast feature (Day 8-9)

Task 6: Error handling and testing (Day 10 – 11)

Task 7: Documentation preparation (Day 12)

Risk Assessment:

Risk 1: API changes or downtime affecting data retrieval * Impact: High * Likelihood: Medium * Mitigation strategy: Regularly check API status, implement robust error handling, and conduct thorough testing

Risk 2: Complexity in handling various user inputs and errors * Impact: Medium * Likelihood: High * Mitigation strategy: Implement robust error handling, conduct thorough testing, and provide clear user interface prompts and instructions

5. Additional Considerations

User Interface Design: The application will feature a console-based interface with clear prompts and structured output for weather data. The interface will be designed to be user-friendly, with minimal complexity and easy-to-understand output.

Deployment Strategy: The application will be deployed locally on the user's machine, requiring a Python environment and internet access for API interaction. This will ensure that the application is easily accessible and can be run on a variety of devices.

Testing Strategy: The testing strategy will consist of three components:

Unit Tests: Critical functions, such as API interaction and data parsing, will be unit tested to ensure that they are working correctly.

Integration Testing: API interaction will be integration tested to ensure that it is working correctly with the API.

Manual Testing: User interaction flows will be manually tested to ensure that the application is working as expected and that the user interface is intuitive and easy to use.

Conclusion

The proposed weather forecast application aims to provide a straightforward yet effective tool for users to obtain current and extended weather information. By following the outlined development phases and leveraging Python's capabilities, this project will deliver a reliable solution for weather data retrieval.