MicroForecast

Software Architecture

March 7, 2024

Brae Webb & Richard Thomas

1 Brief

CSIRO has successfully developed a tailored weather forecasting system called *Weather Together*[1]. You are tasked to design a hyper-local weather application, MicroForecast, for a broader market. The hyper-local weather application uses weather information from traditional sources, such as weather stations, and other devices, including mobile phones. The application will provide users with up-to-date information about the weather conditions in their immediate vicinity.

2 Requirements

1. The team is currently planning to collect and process weather data from:
   - weather stations;
   - street cameras;
   - cell phone towers;
   - drones;
   - airplanes; and
   - mobile phones.

   Your design must be able to support these interfaces and be easily extendible to support new sources of data.
2. Metrics including temperature, humidity, wind speed, and amount of precipitation over a time period should be supported as types of data. However, your design should be easily extended to support more types of data in the future (e.g. barometric pressure).
3. The application should display current weather conditions for the user's current location.
4. The application should allow users to view short and long-term weather forecasts for their current location.
5. The application should provide weather alerts for severe weather conditions such as thunderstorms or flooding.
6. The application relies on users sharing their data. It is a high priority that this data remains secure and protects user privacy.

3 Outline

Introduction (5 minutes)

Introduction to the brief and hyper-local weather applications.

[1] https://www.csiro.au/en/research/natural-environment/ecosystems/weather-together

Design (10 minutes)

In teams, discuss and sketch out a potential design for the system. You can use any tools you like, but you should be able to explain your design to the class. If you are using digital tools, excalidraw[2] is useful for collaborative sketching. Your design does not need to be complete nor perfect, try to be creative so that we can discuss the pros and cons of various design options.

Discussion (10 minutes)

With the class, present a few of the designs and discuss the pros and cons of each. Consider the following questions:

- Which quality attributes are prioritised in this design?
- How would you extend this design to support more weather data sources?
- How would you extend this design to support more types of data?
- How could updates to the data sources be handled?
- Is user data treated appropriately?
- Are there trade-offs in this design?

Sketching (20 minutes)

Individually sketch out a basic implementation of your preferred design from the discussion. Your design sketch should include:

- A high-level overview of the architecture, including any architectural patterns used.
- A description of the data sources and the format of the data they provide.
- A sketch of the required interfaces that data source devices need to implement.
- A free-form diagram that illustrates the communication between components of the system.

You may include any other details including pseudocode, class diagrams, etc. that you think are relevant.

Optimisation (10 minutes)

Discuss how you would optimise your design to improve extensibility, performance, scalability, etc. in a real system. Consider the following questions:

- What is the burden to implement a new data source?
- How would this system scale if it was deployed across Australia, with millions of devices providing data?
- Are there any security concerns?

[2] https://excalidraw.com/

4 Design Challenges

Challenge 1: Data Consistency

Different weather data sources may provide conflicting data. For example, one source may report that it is currently raining while another source reports that it is not. How would you modify your design to ensure data consistency across multiple sources?

Challenge 2: Mobile Data Usage

Users have limited data on their mobile plans and would like to minimise data usage. How would you design the application to minimise data usage while still providing the desired functionality?

Challenge 3: Outdated Weather Information

Weather data is constantly changing and can become outdated quickly. How would you ensure that users receive up-to-date weather information?