SDA Weather Project Deliverable 01.

Phase 01 – Design.

Table of Contents:

Group Members:

- Azfar Nayyan (22L-7886).
- Saim Imran (22L-7906).
- Talha Tofeeq (22L-6190).
- Abdullah Shafqat (22L-7905).
- Bilal Afzal (22L-7889).
- Anas Khan (22L-7963).

Purpose:

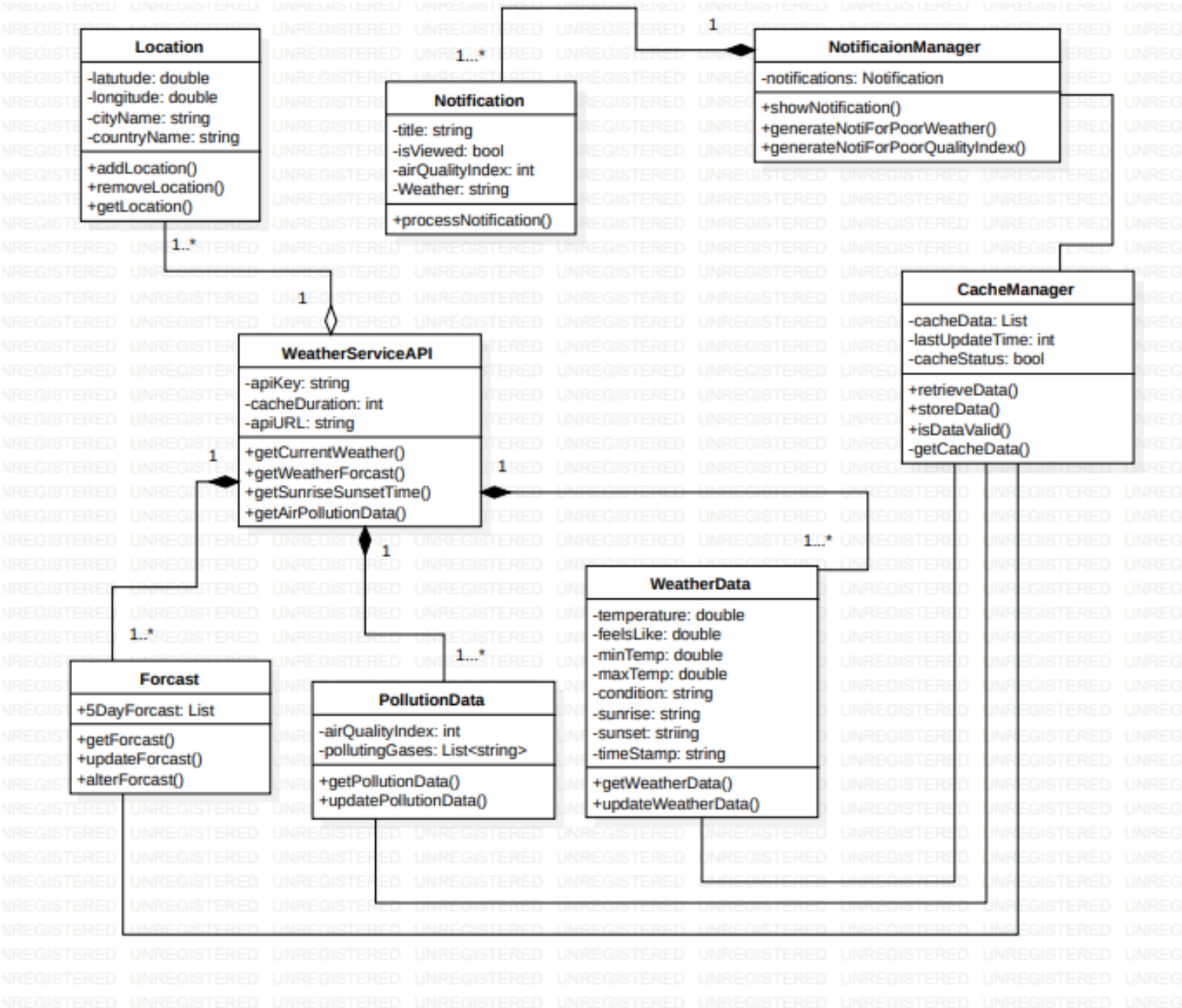
This document presents a comprehensive analysis and design of a software system. The deliverables include a Use Case Diagram, Class Diagram, and Sequence Diagrams for specific use cases. The deliverables aim to offer a clear and concise representation of the software system, facilitating understanding and communication among stakeholders. Each student is assigned specific responsibilities, contributing to the overall completeness of the document.

Introduction/Abstract:

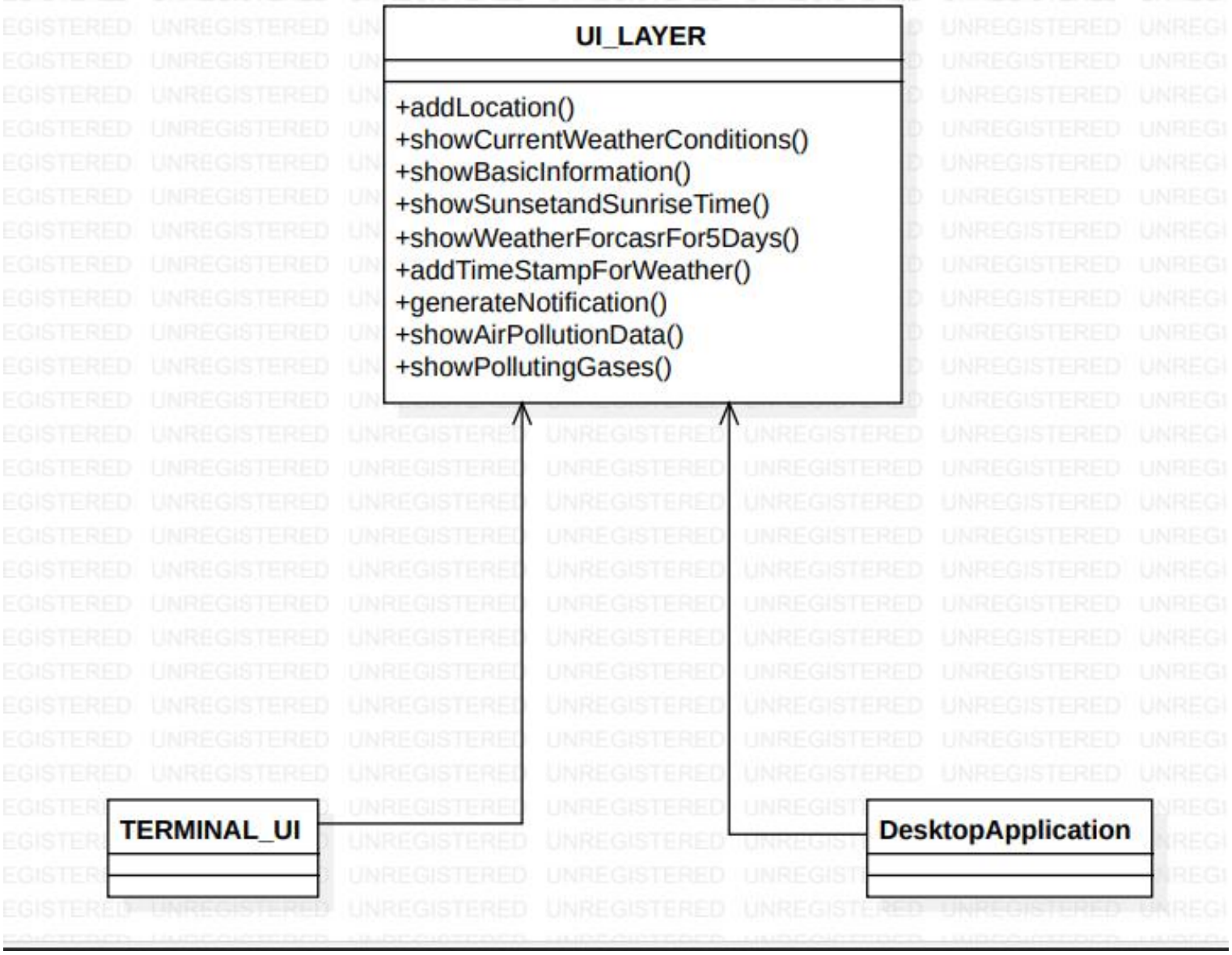
This document outlines the design and implementation details for a comprehensive weather application that allows users to access real-time weather data, forecasts, and air quality information for multiple locations. The application features two distinct user interfaces – a terminal-based console application and a Java desktop application. Additionally, two independent storage methods are integrated into the system, including SQL-based storage and a text-based storage mechanism. The application leverages OpenWeatherMap APIs to gather weather-related data, including current weather conditions, 5-day forecasts, and air pollution indices.

Class Diagram:

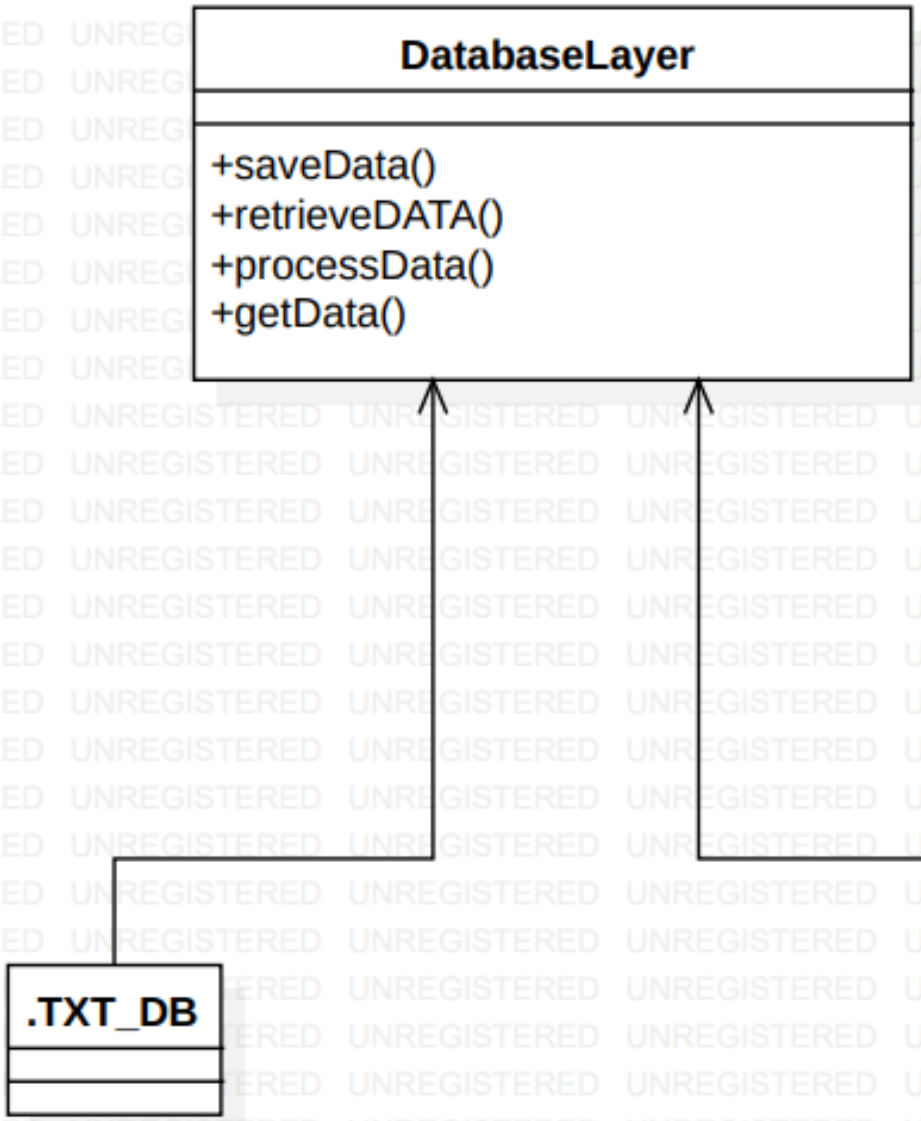
- This shows the main business logic.



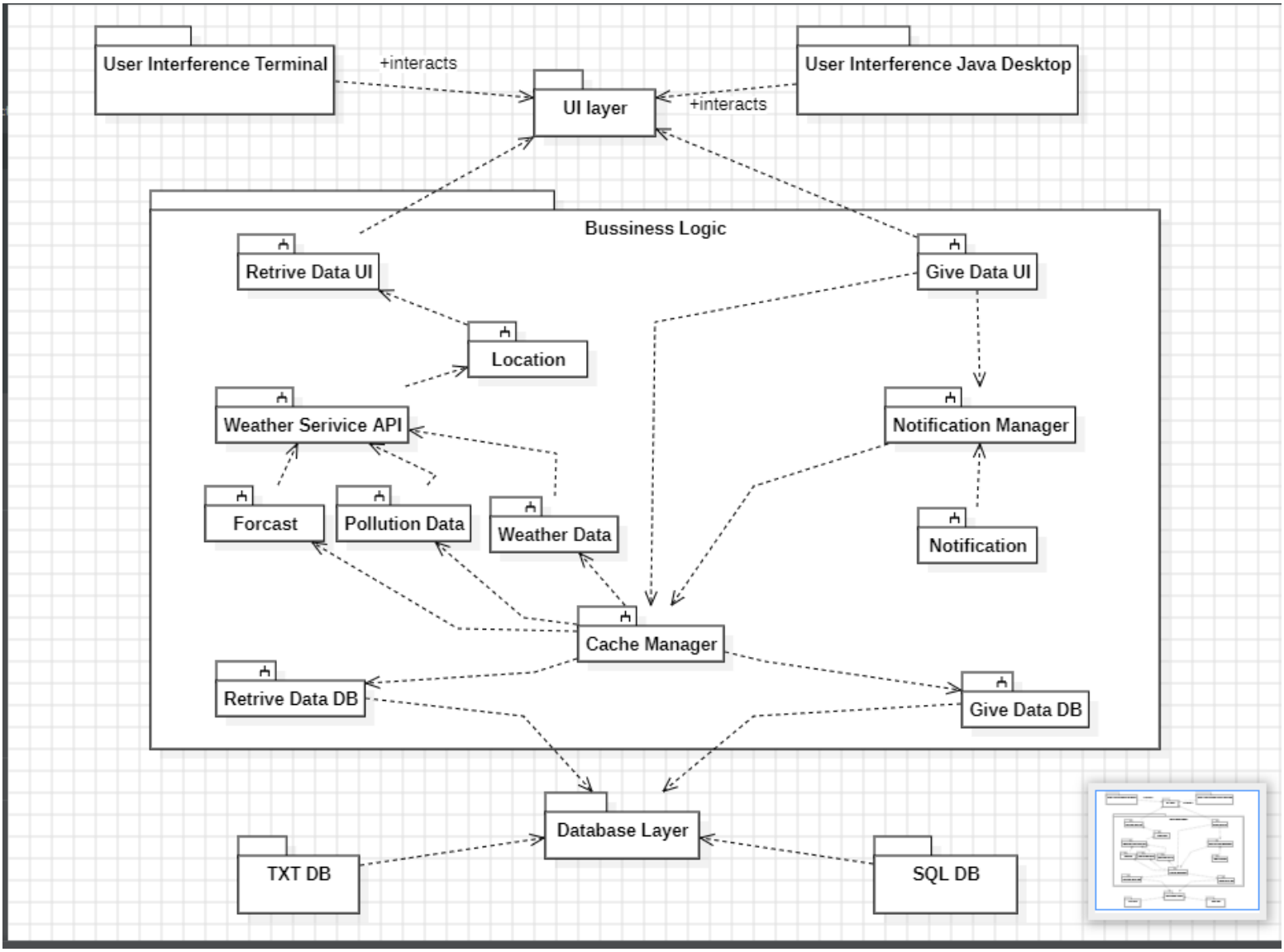
UI Layer:


UI_LAYER
+addLocation()
+showCurrentWeatherConditions()
+showBasicInformation()
+showSunsetandSunriseTime()
+showWeatherForcasrFor5Days()
+addTimeStampForWeather()
+generateNotification()
+showAirPollutionData()
+showPollutingGases()
TERMINAL_UI
DesktopApplication

Database Layer:

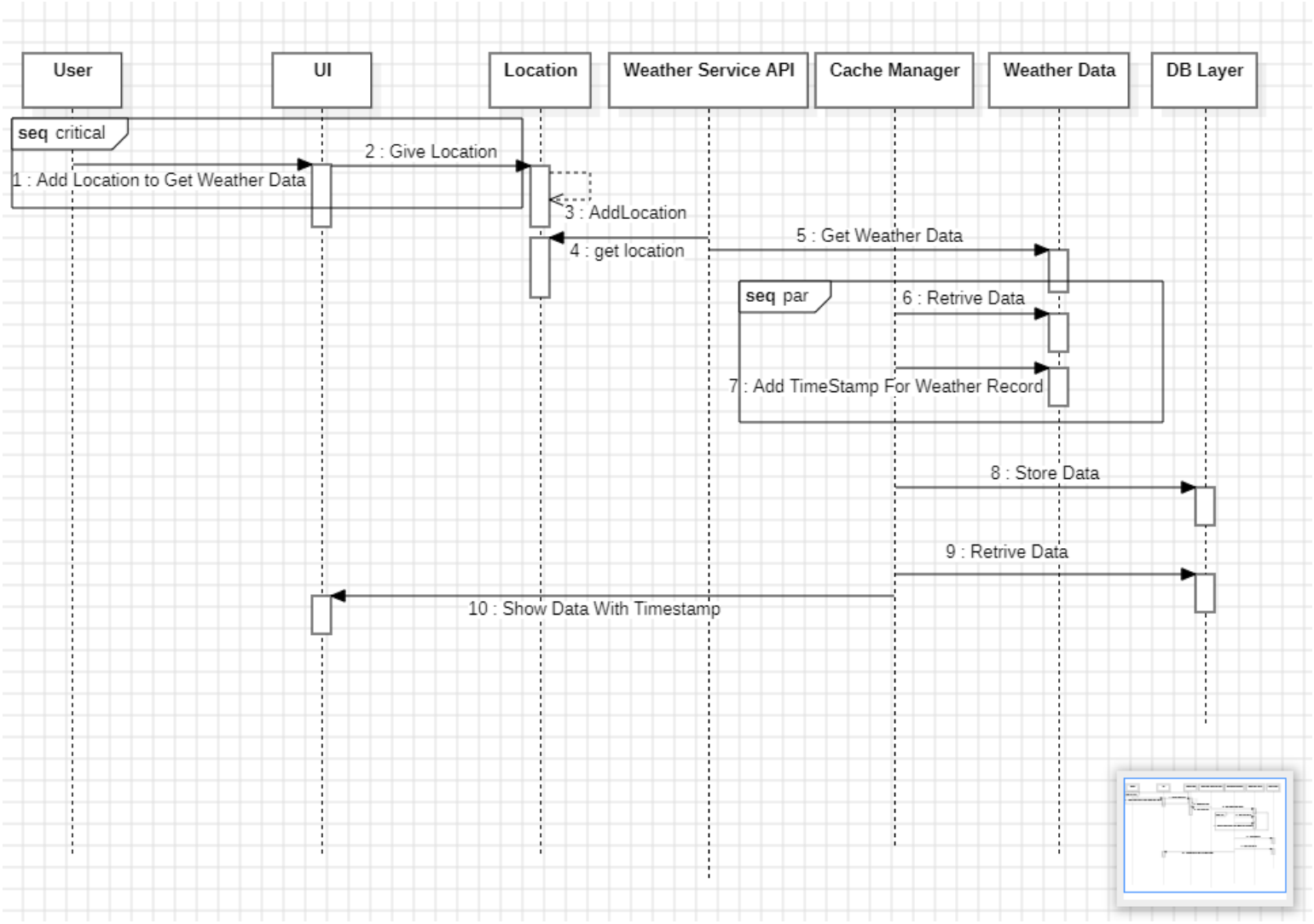


Package Diagram:

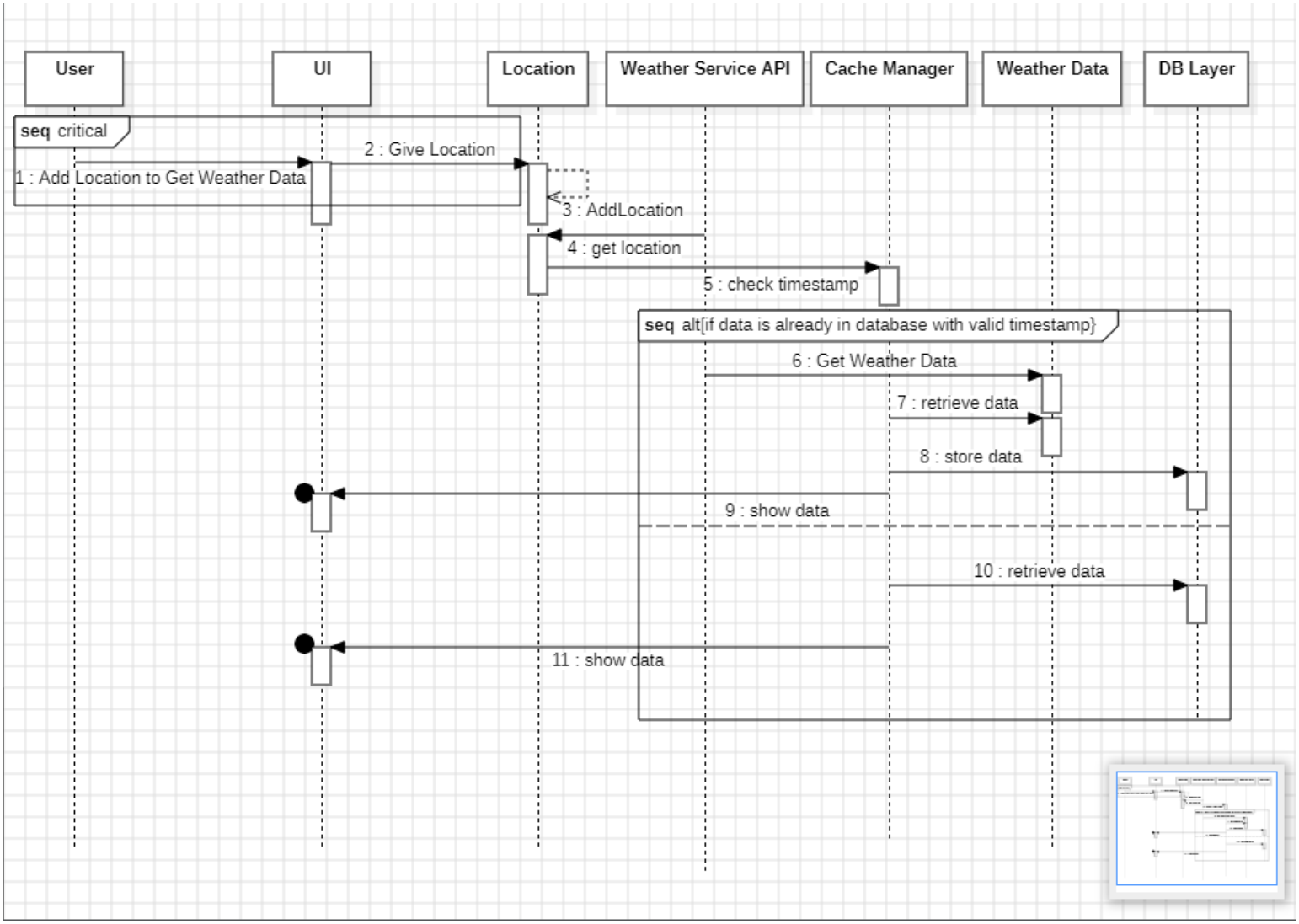


Sequence Diagrams:

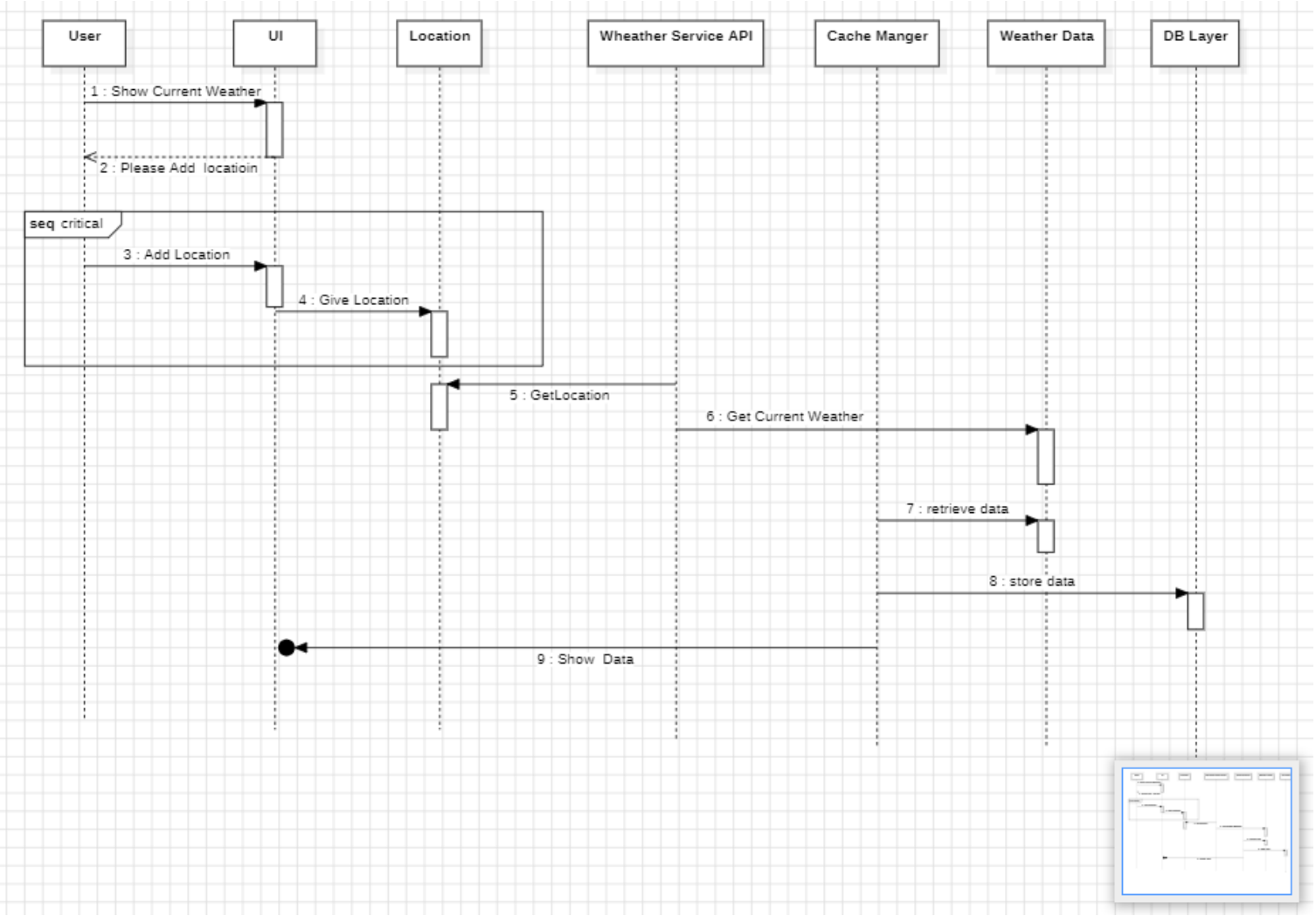
1. Add timestamp for weather records. (22l-7886)



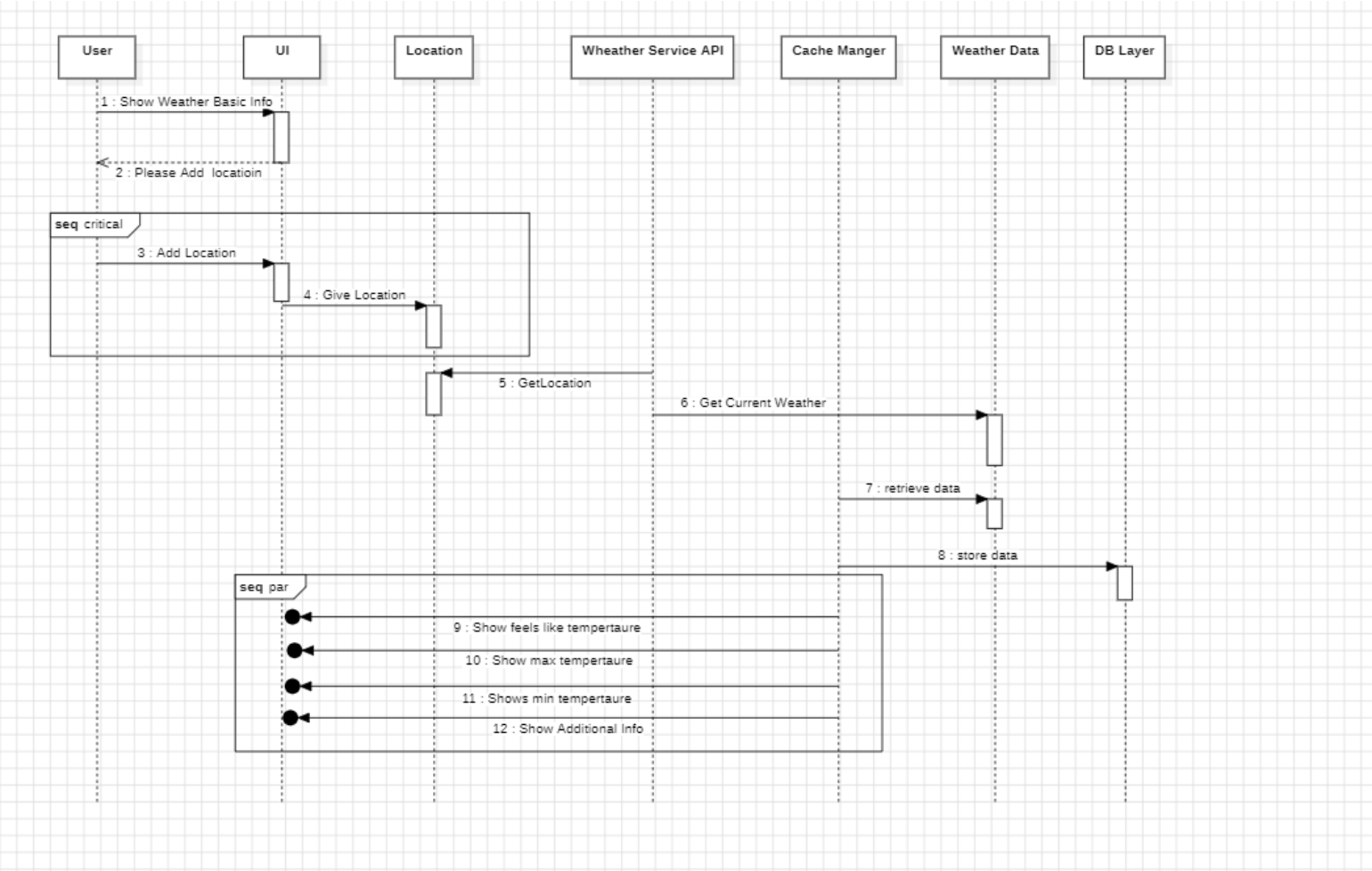
2. Implement Cache Management: (In order to reduce the number of API calls and improve application performance, you can use a database as a cache to store frequently accessed weather data. This can help minimize the load on the OpenWeatherMap API and improve response times for users. For example, if weather data for Lahore is requested, make an API call to get the data, store it in DB and then show the result. If on the same day, the weather for same location is accessed, instead of making a call to API, fetch the data from DB.) (22I-7886)



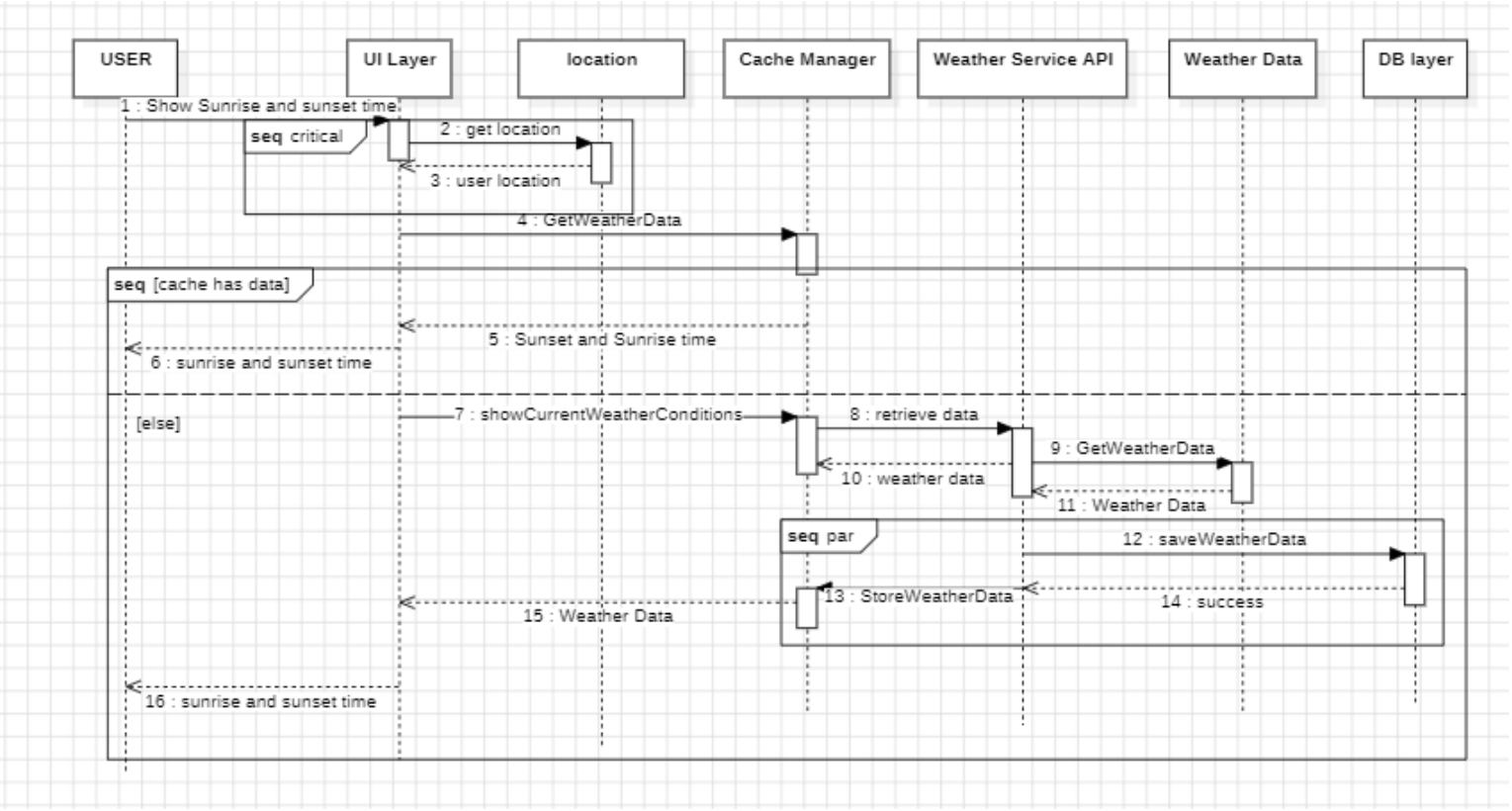
3. Show current weather conditions. (22l-6190)



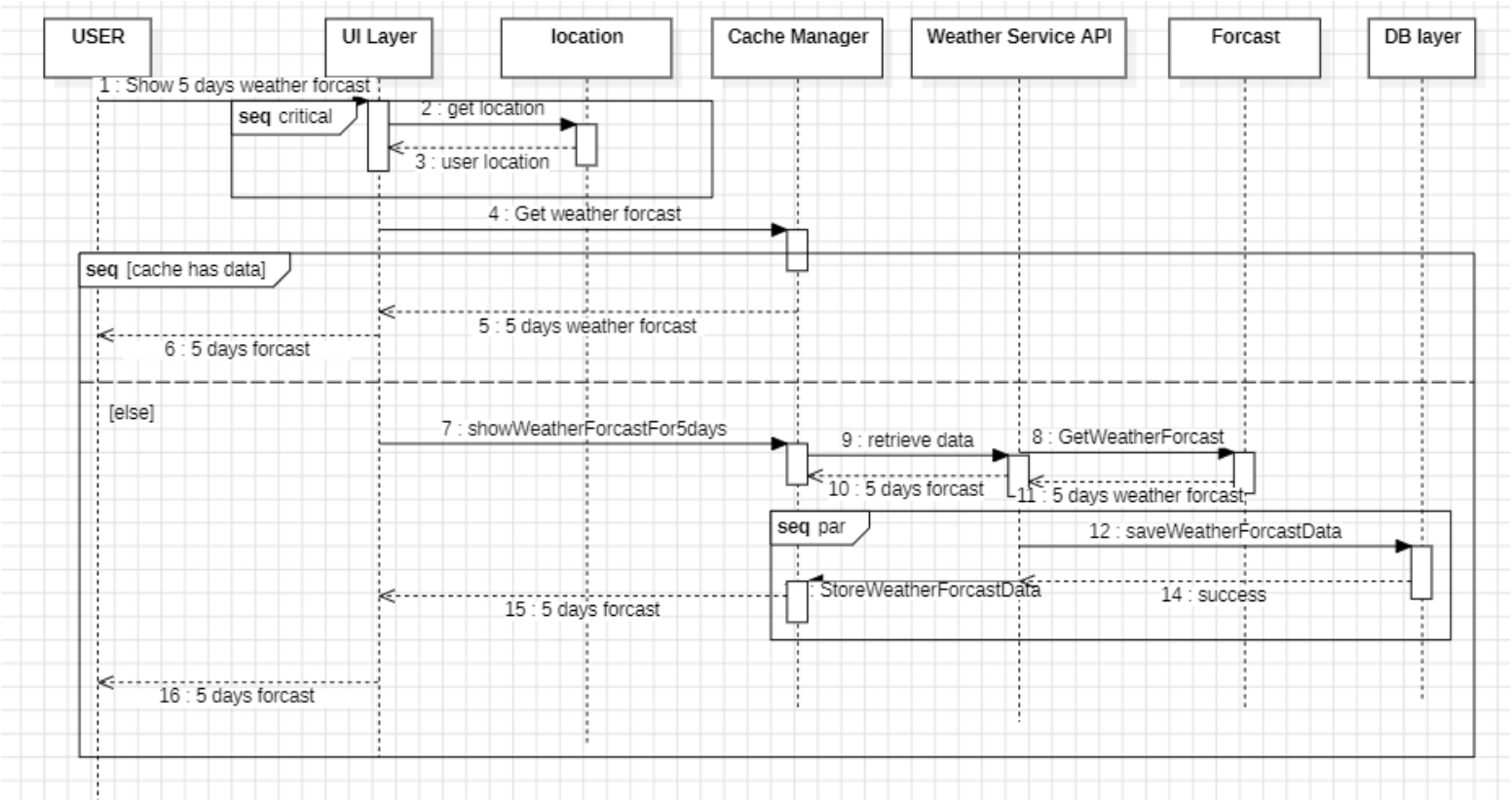
4. Show basic information like “Feels like, minimum and maximum temperature” etc. (22l-6190)



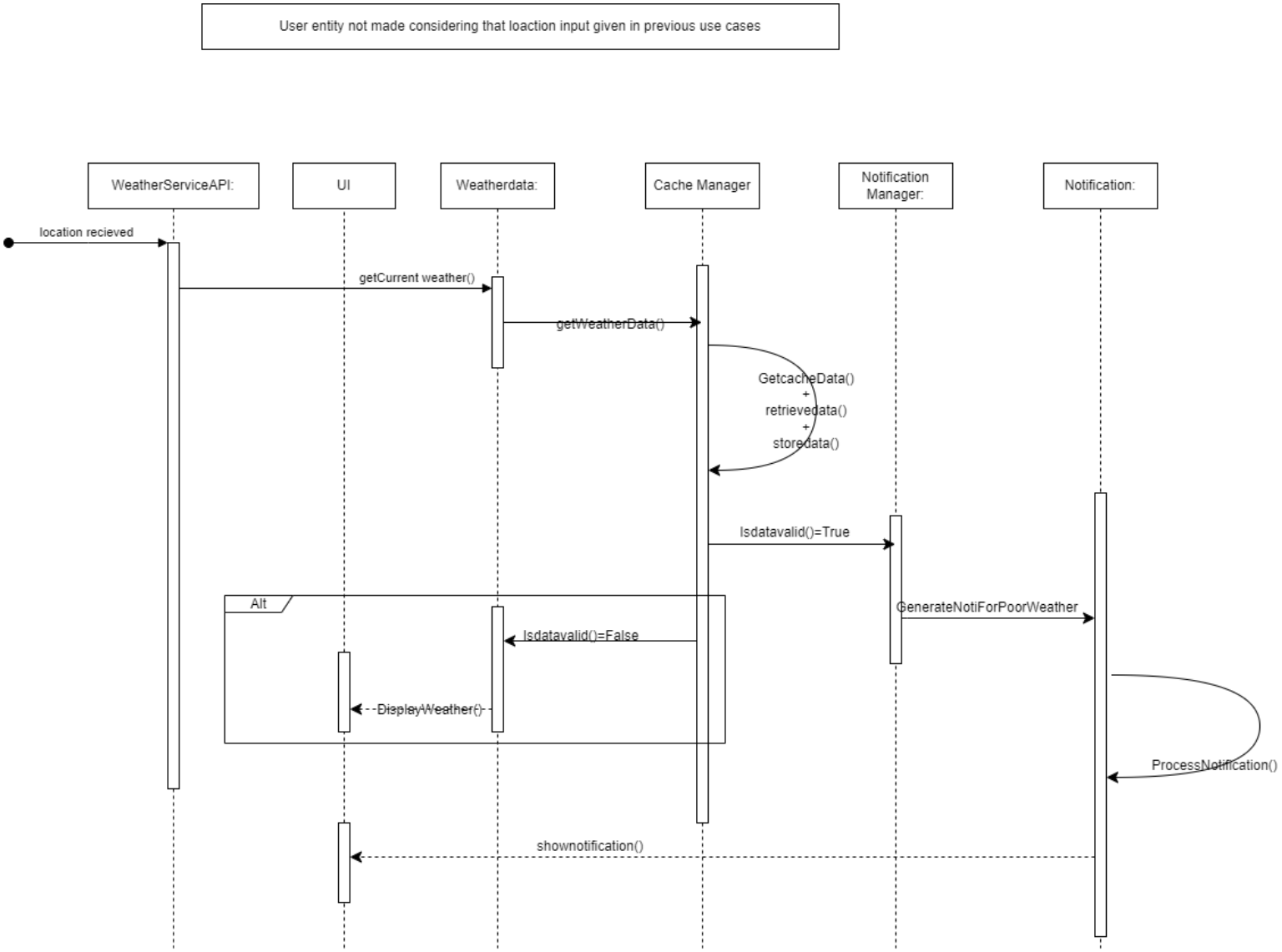
5. Show sunrise and sunset time. (22l-7906)



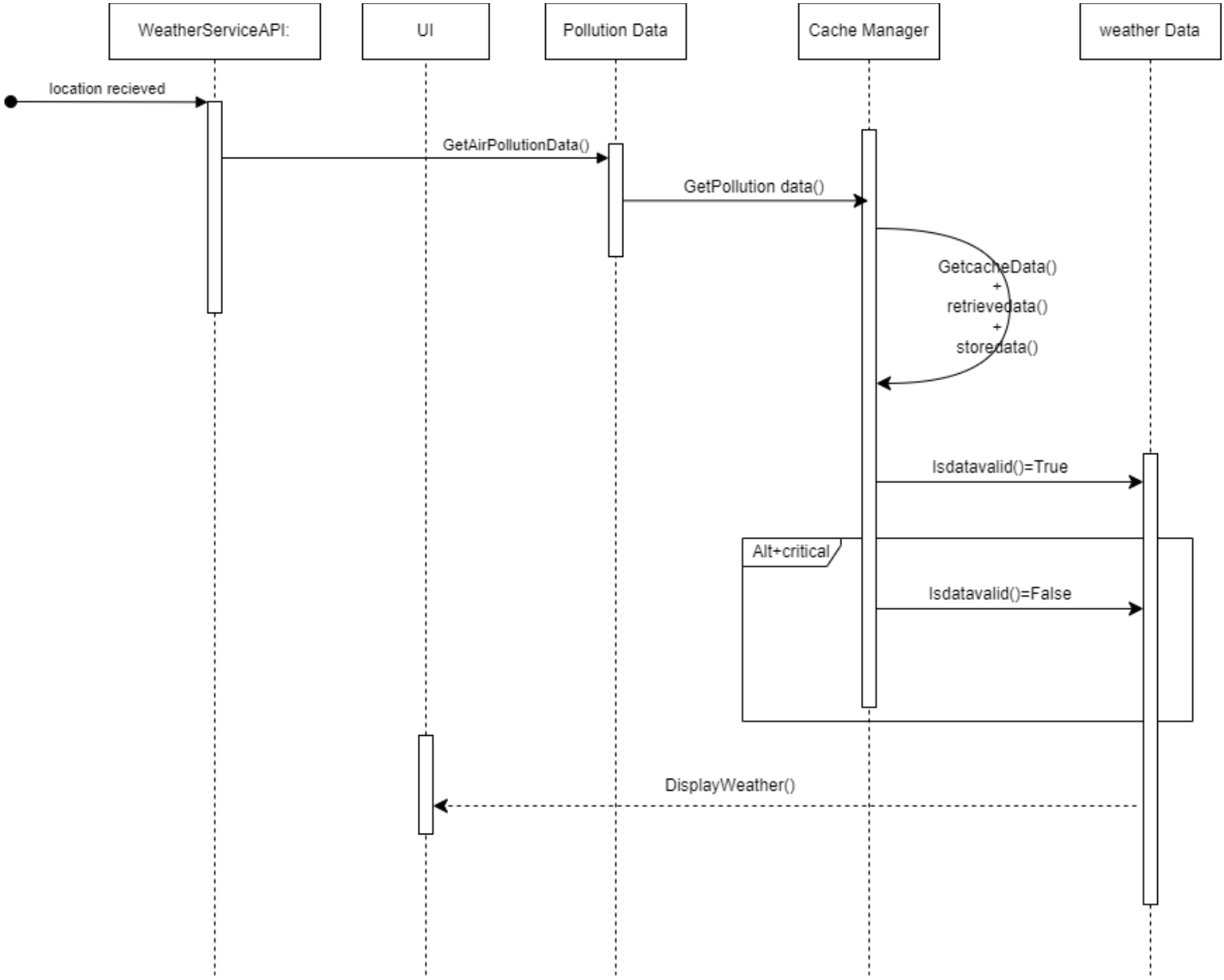
6. Show the weather forecast for 5 days. (22l-7906)



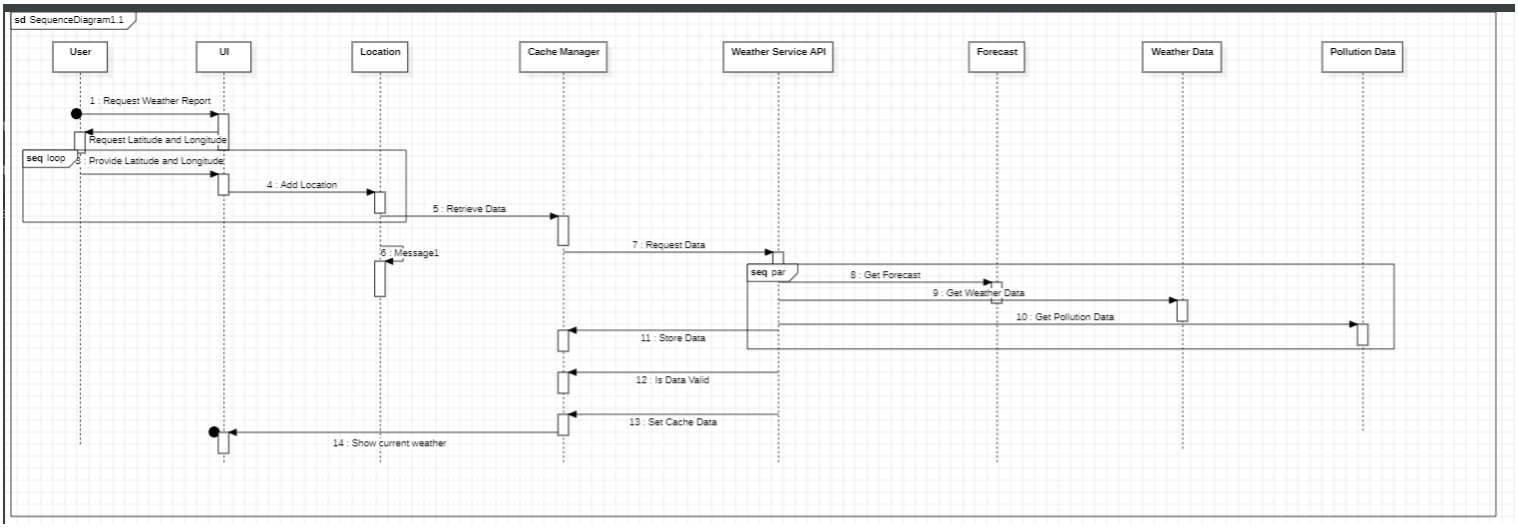
7. Generate Notification for poor weather conditions. (22l-7889)



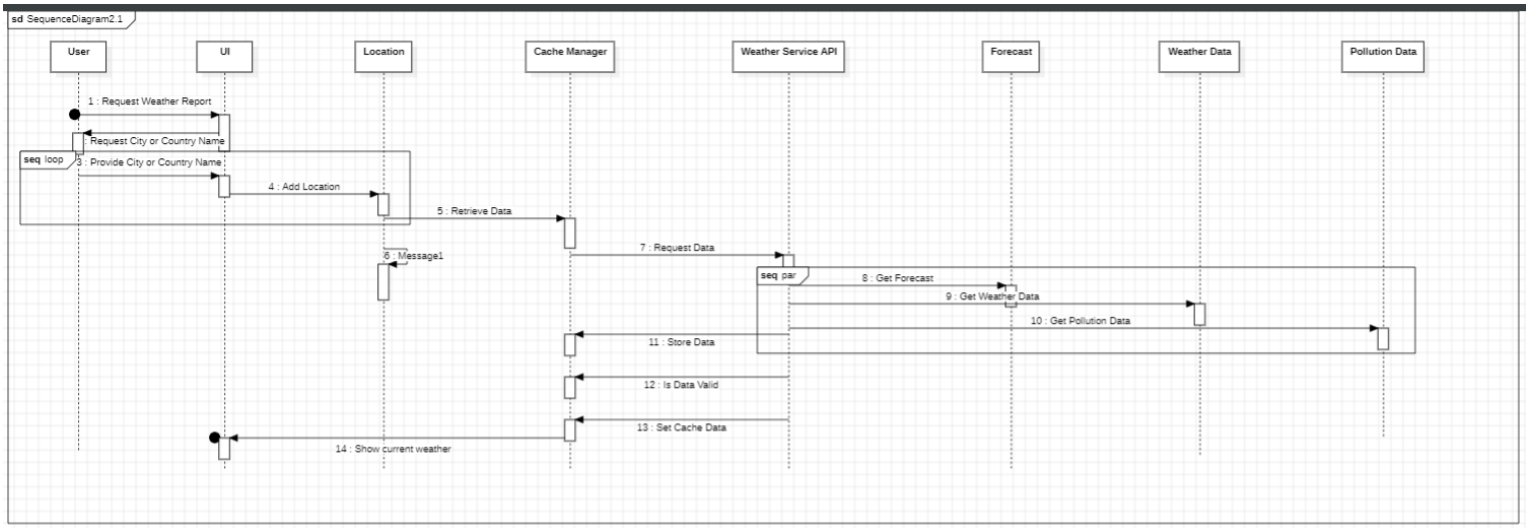
8. Show Air Pollution data. (22l-7889)


WeatherServiceAPI:
UI
Pollution Data
Cache Manager
weather Data
location recieved
GetAirPollutionData()
GetPollution data()
GetcacheData()
+
retrievedata()
+
storedata()
Isdatavalid()=True
Alt+critical
Isdatavalid()=False
DisplayWeather()

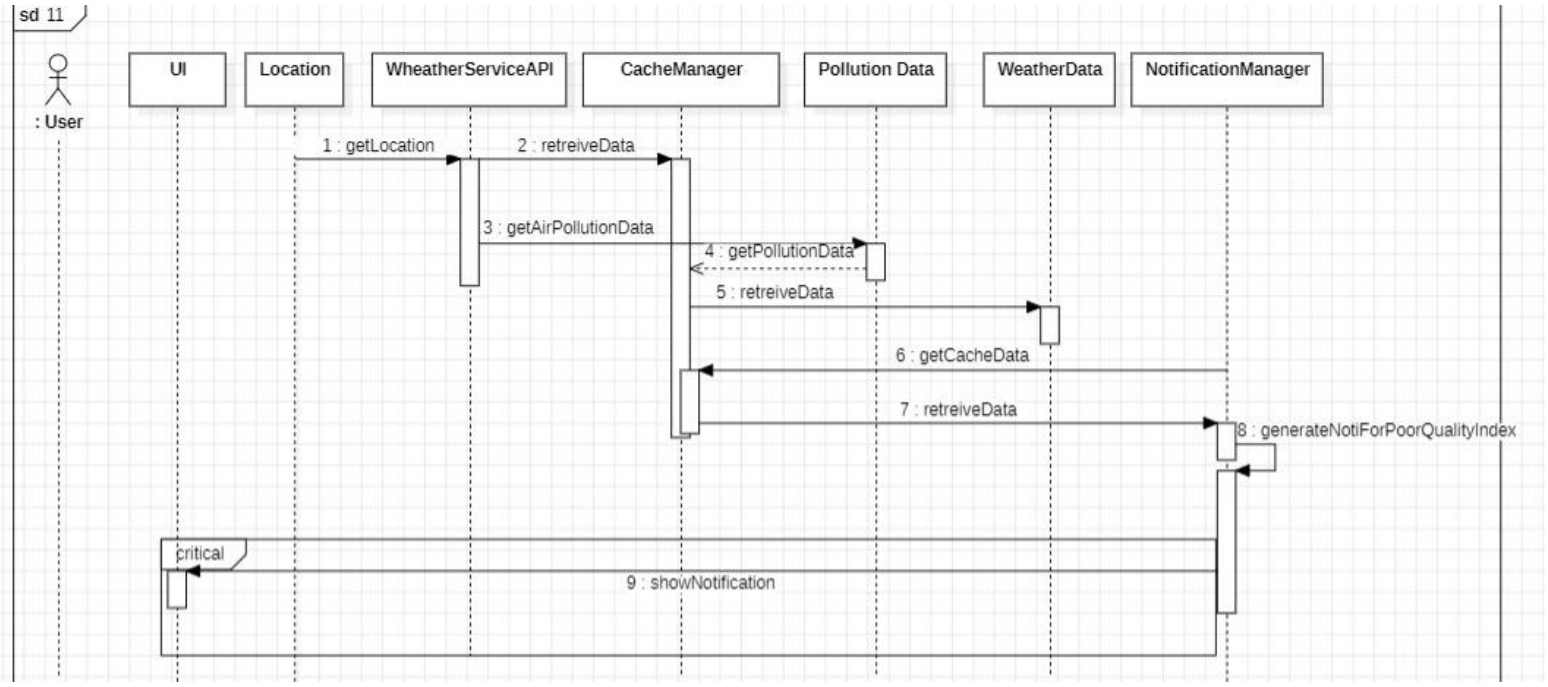
9. Add multiple locations to check weather with longitude and latitude. (22l-7905)



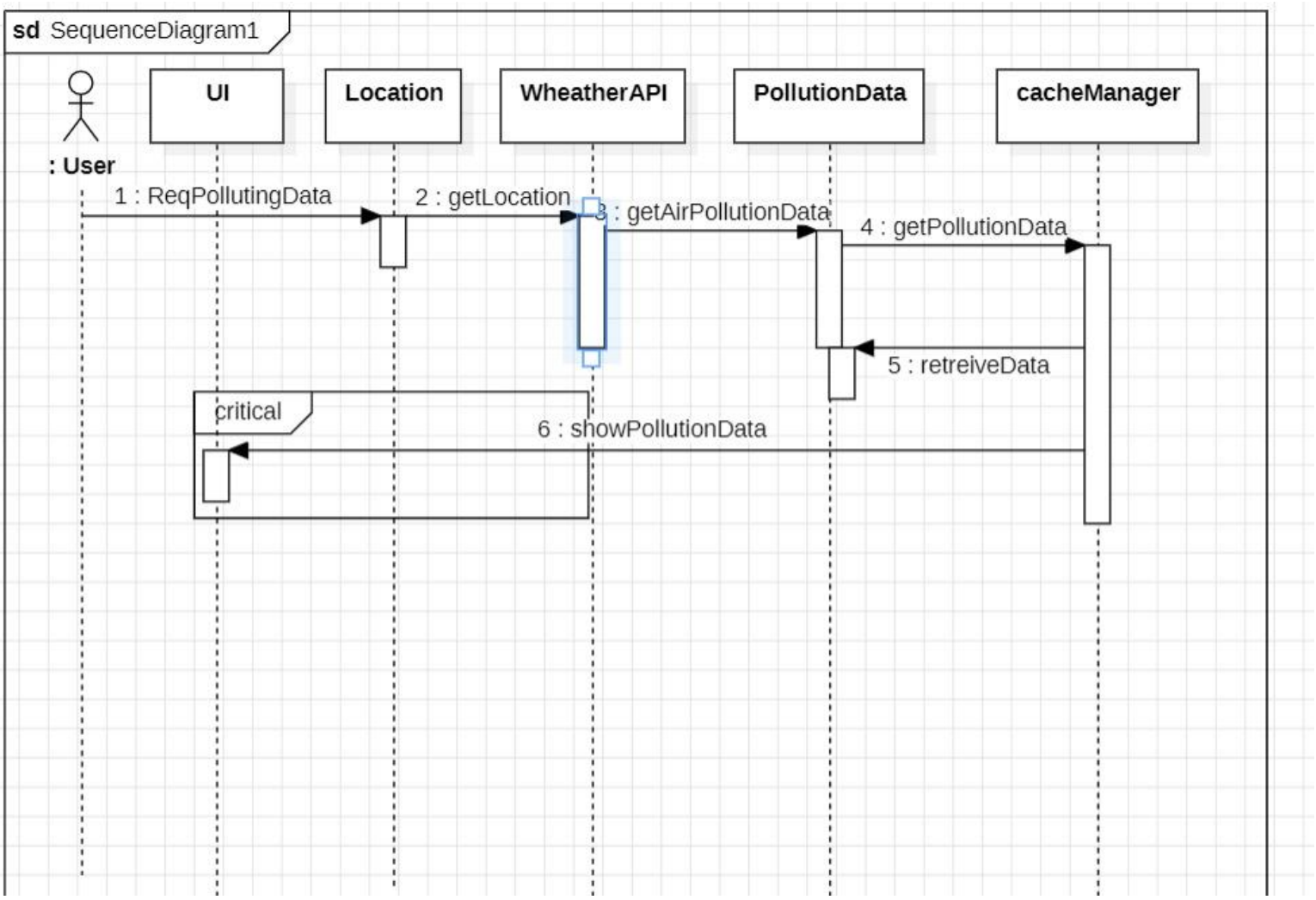
10.Add multiple locations to check weather with city/country name. (22I-7905)



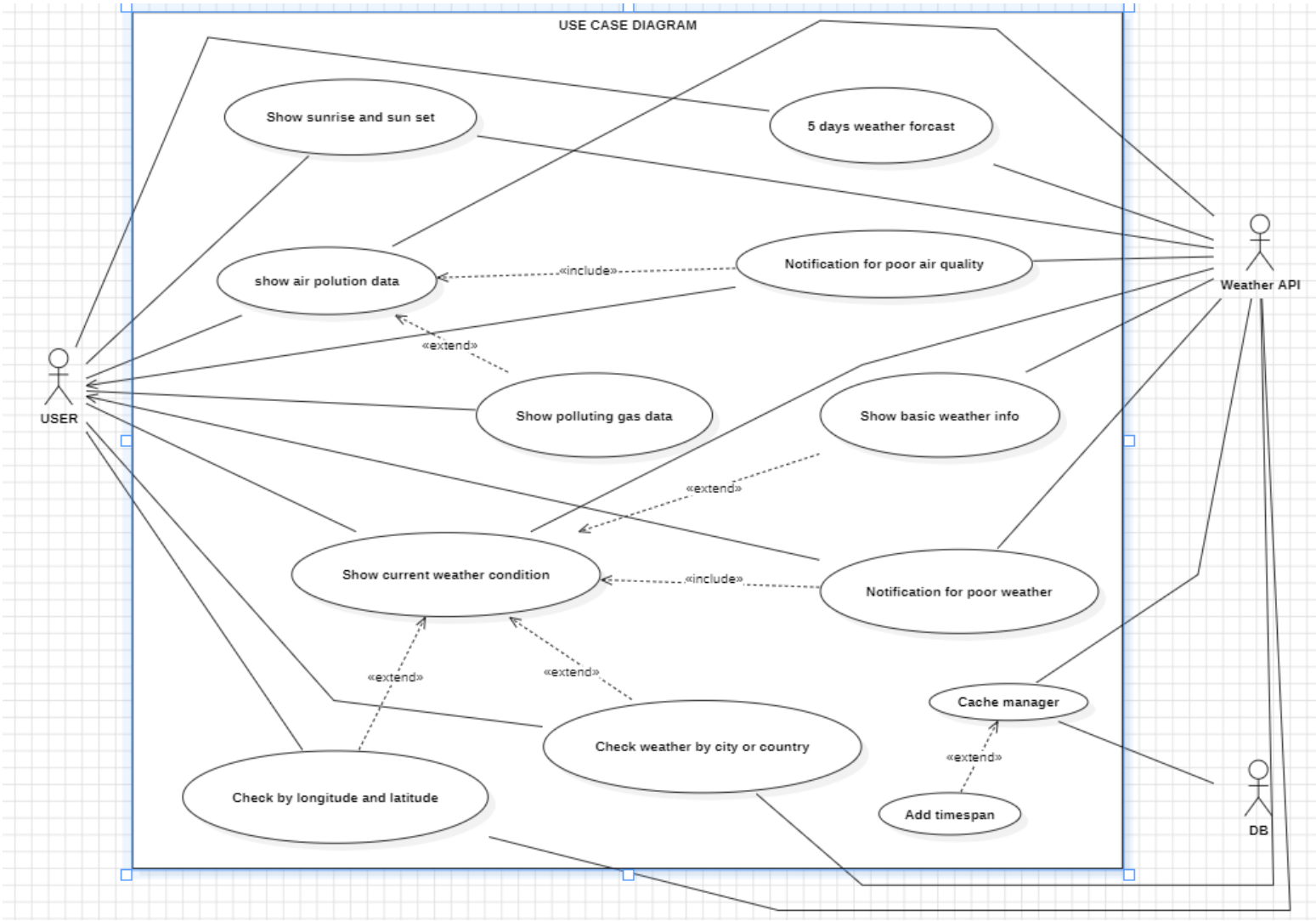
11.Generate Notification for poor air quality. (22l- 7963)



12.Show data about polluting gases. (22l-7963)



USE CASE Diagram:


USE CASE DIAGRAM
Show sunrise and sun set
5 days weather forcast
show air polution data
«include»
Notification for poor air quality
«extend»
Show polluting gas data
Show basic weather info
«extend»
Show current weather condition
«include»
Notification for poor weather
«extend»
«extend»
Cache manager
Check weather by city or country
«extend»
Check by longitude and latitude
Add timespan
USER
Weather API
DB