SE 4485: Software Engineering Projects

Fall 2025

Requirement Documentation

|  |  |
| --- | --- |
| Group Number | Group 1 |
| Project Title | County Level Air Quality Prediction Application |
| Sponsoring Company | Raytheon (Team A) |
| Sponsor(s) | Ryan Havens <Ryan.Havens@rtx.com>,  Marc Perna <marc.perna@rtx.com>,  Trey Williams <trey.williams@rtx.com>,  Trevor Lang <trevor.a.lang@rtx.com> |
| Students | 1. Jay Chung <cwc130330@utdallas.edu>  2. Amelia Quinn <qcb220000>  3. Kevin Melo <ksm220005>  4. AJ Kimbrough <ank210005>  5. David Santos <des210001>  6. Andrew Enright <ame210008> |

**ABSTRACT**

This document defines the use case model, textual use case, supporting rationale, functional requirements non-functional requirements, and configuration management details for the County Level Air Quality Prediction (CLAP) web application. These requirements provide the foundation for system design, implementation, and validation while ensuring alignment with project objectives and stakeholder expectations.

**TABLE OF CONTENTS**

1. INTRODUCTION 3
2. USE CASE MODEL FOR FUNTIONAL REQUIREMENTS 4
3. RATIONALE FOR YOUR USE CASE MODEL 6
4. NON-FUNCTIONAL REQUIREMENTS 6
5. EVIDENCE OF CONFIGURATION MANAGEMENT 7
6. ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS 8
7. ADDITIONAL REFERENCES 8

**LIST OF FIGURES**

Figure 2.1 – Use Case Model – CLAP Application 4

**LIST OF TABLES**

Table 6.1 – Configuration Management – File Revision Log 7

Table 6.2 – Configuration Management – Difference Link 7

**INTRODUCTION**

This document presents the Requirements Specification for the County Level Air Quality Prediction (CLAP) web application. Its primary purpose is to define the system’s functional and non-functional requirements, ensuring that the application developed aligns with the objectives set by the sponsor. The scope of this document includes the identification of system capabilities, constraints, and use cases that will guide the design, implementation, and validation of the CLAP system.

The CLAP web application is a predictive analytics tool designed to forecast next-day Air Quality Index (AQI) categories at the county level. The system leverages historical AQI data provided by the Environmental Protection Agency (EPA) and applies machine learning models to generate predictions. This approach serves both as a proof of concept for county-level forecasting and an educational framework for developing and testing predictive models in environmental domains.

The CLAP system will provide an interactive dashboard to visualize recent AQI trends and predicted categories. These features are intended to support users in understanding air quality risks and making informed decisions about outdoor activities. Furthermore, the project aims to demonstrate the feasibility of portable, data-driven forecast classification solutions that can run locally on student hardware with minimal setup.

The remainder of this document is organized as follows: use case model for functional requirements, rationale for the use case model, non-functional requirements, and configuration management details.

**USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS**

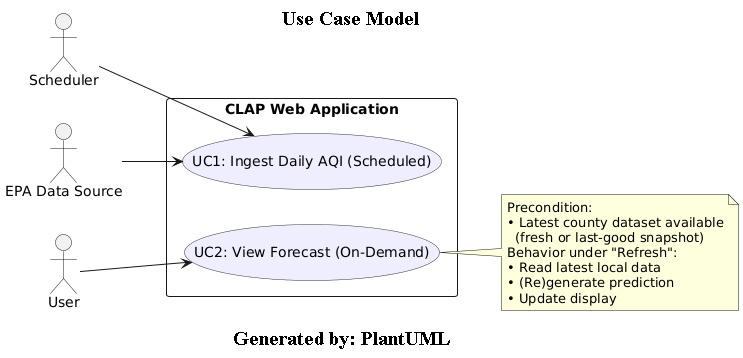


Figure 2.1 – Use Case Model – CLAP Application

The following functional requirements (FRs) define the system’s expected capabilities and core functionality:

1. Data Ingestion:

* The system shall utilize daily AQI data from the Environmental Protection Agency (EPA).

1. Data Processing:

* The system shall store historical AQI data in a database.
* The system shall generate lag features for use in prediction.

1. Predictive Analytics:

* The system shall train and run a predictive model using historical AQI and county location data.
* The system shall output a next-day AQI category for a selected county.
* The system shall provide the prediction with an associated probability score for each AQI category.

1. Dashboard:

* The system shall provide an interactive dashboard for visualizing AQI data.
* The system shall provide a Refresh button to trigger new ingestion, processing, and prediction.

Textual Use Cases:

1. UC1 – Ingest Daily AQI (Scheduled):

* Participating Actors:
  + Primary Actor: Scheduler forecasts through the dashboard.
  + Supporting Actor(s): EPA data source.
* Goal:
  + To keep the local dataset for all tracked counties.
* Entry Conditions (Preconditions):
  + EPA data sources are available.
  + Network access to EPA is available.
* Normal Flow of Events:
  + The system cleans AQI data and stores the result in the database.
  + The system generates lag features and stores the results in the database.
* Success Condition (Postconditions):
  + The AQI and lag features are persisted and time stamped.
* Exceptions:
  + The EPA data is unavailable 🡪 System logs error and retries.

1. UC2 – View Forecast (On-Demand):

* Participating Actors:
  + Primary Actor: User (Both technical and non-technical)
* Goal:
  + To view historical AQI and the next-day AQI category for a selected county.
* Entry Conditions (Preconditions):
  + Dashboard is accessible.
  + County is selected.
  + Dataset is available.
* Normal Flow of Events:
  + The system reads the latest county dataset from the database.
  + The system generates next-day AQI prediction using historical AQI and county data.
  + The system displays historical AQI and predicted category with probabilities using EPA categories.
* Success Condition (Postconditions):
  + The dashboard shows updated history and forecast with timestamps.
* Exceptions:
  + The UC1 dataset is unavailable 🡪 System shows most recent available forecast with warning.
  + A prediction failure occurs 🡪 Shows last valid prediction and log error.

**RATIONALE FOR YOUR USE CASE MODEL**

The case model separates scheduled data preparation from on-demand user interaction to keep the system simple and testable.

* UC1 – Ingest Daily AQI (Scheduled): A non-human actor accepts a CSV containing clean county-level AQI from the EPA, validates it, generates lag features, and persists a fresh snapshot.
* UC2 – View Forecast (On-Demand): A user selects a county and views historical AQI with the next-day predicted categories produced from the most recent snapshot.

This decomposition follows a black-box perspective (actors ↔ system), improves traceability from requirements to design and tests, enables incremental demonstrations, and supports performance goals by keeping the user flow fast and predictable.

**NON-FUNCTIONAL REQUIREMENTS**

The non-functional requirements (NFRs) define the quality attributes and operational constraints of the system, describing how the system performs rather than what it does. Each requirement is ranked by priority to indicate its relative importance for successful implementation and stakeholder satisfaction.

1. Performance:

* UC1 (Ingest): The system shall complete ingestion, validation, and feature generation for one county in ≤ 60 seconds.
* UC2 (View): The system shall render the dashboard in ≤ 5 seconds p95 after a user clicks Refresh.

1. Usability:

* The dashboard shall present AQI categories using standard EPA labels and color coding.

1. Reliability:

* UC1 daily ingest shall succeed on ≥ 90% of scheduled runs.

1. Availability:

* The CLAP dashboard shall be available for use by users at least 99% of the time over any rolling 30-day period.

**EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT**

The team has selected GitHub as the configuration tool for this project. The tables below provide evidence of configuration management by recording version history, authorship, and reviews of document changes. The *ID* column identifies each entry. The *date of change* column indicates when a modification was made to an existing file, and the v*ersion (before & after)* columns include the associated Git commit hash for distinction. The *author* column refers to the author of the new version. The *difference link* column provides a URL to the GitHub comparison view between two consecutive commits. The format of the difference link is as follows:

“https://github.com/cchung7/rtx\_team1/compare/<ver-before-hash>..<ver-after-hash>”.

Table 1.1 – Each entry (or row) tracks a single file revision.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID:** | **Date of Change:** | **Version Before:** | **Version After:** | **Author:** | **Review -Change Summary:** | **Reviewers:** |
| 1 | 9/10/25 | v0.1 (bb4d945) | v0.2 (bfc15f4) | AJ Kimbrough (ank210005) | Added refresh workflow pipeline. Updated NFRs and use case. | Jay Chung (cwc130330) |
| 2 | 9/22/25 | v0.2 (bcf15f4) | v0.3 () | Jay Chung (cwc130330) | Revised all sections. | All Team Members |
| 3 | TBD | TBD | TBD | TBD | TBD | TBD |
| 4 | TBD | TBD | TBD | TBD | TBD | TBD |

Table 1.2 – Each entry (or row) lists a difference link.

|  |  |
| --- | --- |
| **ID:** | **Difference Link:** |
| 1 | https://github.com/cchung7/rtx\_team1/compare/bb4d945..bfc15f4 |
| 2 | https://github.com/cchung7/rtx\_team1/compare/bfc15f4.. |
| 3 | TBD |
| 4 | TBD |

**ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS**

Engineering Standards:

* IEEE Std 1058-1998: Software Project Management Plans [[pdf](https://course.techconf.org/se4485/IEEE/IEEE-Std-1058-1998-Software-Project-Management-Plans.pdf)]
* PMBOK® Guide: Project Management Body of Knowledge [[pdf](https://course.techconf.org/se4485/IEEE/PMBOKR.pdf)]
* IEEE Std 12207: Software Life Cycle Processes [[pdf](https://course.techconf.org/se4485/IEEE/IEEE%2012207%20(2017)%20-%20Software%20Life%20Cycle%20Processes.pdf)]
* IEEE Std 15939: Measurement Process [[pdf](https://course.techconf.org/se4485/IEEE/IEEE%2015939%20(2017)%20-%20Measurement%20Process.pdf)]
* ISO/IEC/IEEE Std 29148-2018: Systems and Software Engineering

§ Life Cycle Processes

§ Requirements Engineering [[pdf](https://course.techconf.org/se4485/IEEE/ISO-IEC-IEEE-29148-2018.pdf)]

Multiple Constraints:

* Project may utilize one data set as long as multiple fields are used to train the predictive analytics model.

**ADDITIONAL REFERENCES**

* + Lamsweerde, A.V., 2009. *Requirements Engineering: From System Goals to UML Models to Software Specifications.* John Wiley