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 FUNCTIONAL REQUIREMENTS 4
3. RATIONALE FOR YOUR USE CASE MODEL 7
4. NON-FUNCTIONAL REQUIREMENTS 7
5. EVIDENCE OF CONFIGURATION MANAGEMENT 8
6. ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS 9
7. ADDITIONAL REFERENCES 9

**LIST OF FIGURES**

Figure 2.1 – CLAP Use Case Model 4

**LIST OF TABLES**

Table 6.1 – Configuration Management – File Revision Log 8

Table 6.2 – Configuration Management – Difference Link 8

**INTRODUCTION**

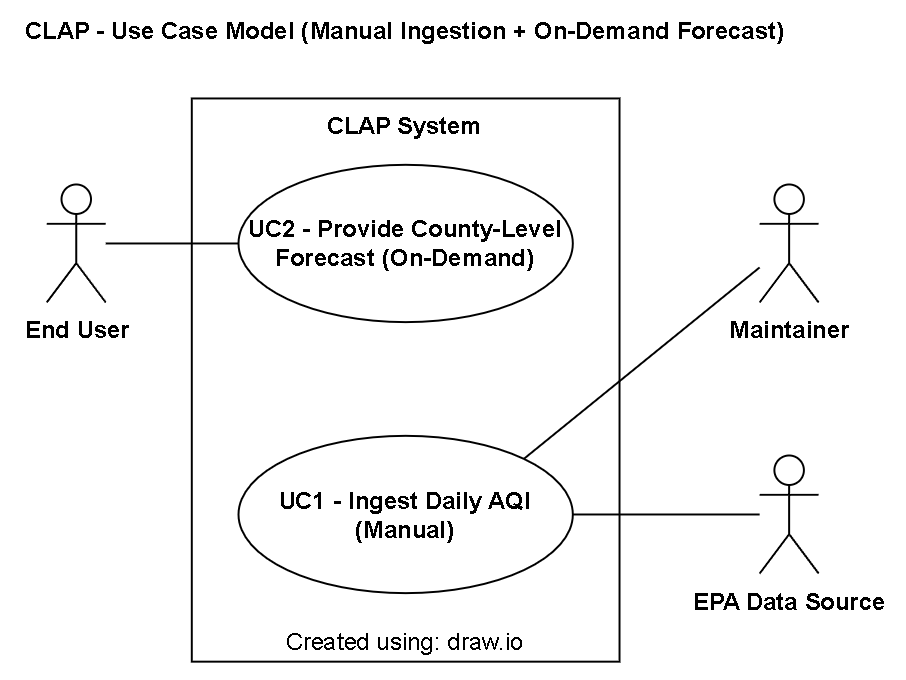
This document presents the Requirements Specification for the County Level Air Quality Prediction (CLAP) web application. Its primary purpose is to define the system’s functional and non-functional requirements, ensuring that the application developed aligns with the objectives set by the sponsor. The 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 end 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**

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

1. Data Ingestion:

* FR-1.1: The system shall utilize daily AQI data from the Environmental Protection Agency (EPA), ingested manually from provided datasets.

1. Data Processing:

* FR-2.1: The system shall store historical AQI data in a CSV.
* FR-2.2: The system shall generate lag features for use in prediction.

1. Predictive Analytics:

* FR-3.1: The system shall train and run a predictive model using historical AQI and county location data.
* FR-3.2: The system shall output a next-day AQI category for a selected county.
* FR-3.3: The system shall provide the prediction with an associated probability score for each AQI category.

1. Dashboard:

* FR-4.1: The system shall provide a graphical user interface (GUI) for visualizing AQI data and predictions.
* FR-4.2: The system shall display a chart or graph of the most recent 30 days of AQI for the selected county and the next-day predicted category with probabilities.
* FR-4.3: The system shall provide a Refresh control that updates the dashboard from locally stored dataset and regenerates the forecast for the selected county.

Textual Use Cases:

1. UC1 – Ingest Daily AQI (Manual):

* Participating Actors:
  + Primary Actor: Maintainer – the person who manually downloads the EPA dataset and places it where the system can ingest it.
  + Supporting Actor(s): EPA data source.
* Goal:
  + To keep the local county-level AQI dataset up to date by manually providing the latest EPA daily file for use by the forecasting system.
* Entry Conditions (Preconditions):
  + The CLAP application is running and available (e.g. ready to receive inputs, and required services are reachable).
  + Network access is available to obtain the file.
  + The system is configured to accept the EPA dataset format (e.g. CSV).
* Normal Flow of Events:
  + The maintainer downloads the daily AQI dataset from the EPA source.
  + The maintainer saves or uploads the file to the system’s designated input location.
  + The system loads the provided dataset into memory.
  + The system validates schema and value ranges.
  + The system cleans and normalizes the dataset, handling missing/outlier data.
  + The system generates lag features for each county/date.
  + The system persists the cleaned data and lag features in local storage (e.g. CSV).
  + The system records ingest metadata (e.g. timestamp, file name).
* Success Condition (Postconditions):
  + Cleaned AQI data and lag features are persisted locally for all tracked counties with provenance metadata.
* Exceptions:
  + EPA data unavailable (e.g. network error):
    - The system logs an error and retries ingestion at least once.
* Traceability to Functional Requirements (FRs):
  + FR-1.1: Utilize daily AQI data from the EPA (manual ingestion).
  + FR-2.1: Store historical AQI data (locally).
  + FR-2.2: Generate lag features for use in prediction.

1. UC2 – Provide County-Level Forecast (On-Demand):

* Participating Actors:
  + Primary Actor: End User
* Goal:
  + To allow the end user to view recent county-level AQI trends and obtain a next-day AQI forecast on demand, using data from the last successful ingest.
* Entry Conditions (Preconditions):
  + The CLAP application is running and available
  + The dashboard is accessible.
  + A county is selected.
  + A dataset exists from a successful ingest.
* Normal Flow of Events:
  + The end user clicks Refresh to request an update for the selected county.
  + The system reads the latest stored AQI dataset and lag features for that county.
  + The system runs the predictive model using historical AQI & location data.
  + The system generates a next-day AQI category and associated probability distribution.
  + The system displays:
    - A 30-day AQI chart for the county with EPA color bands/labels.
    - The predicted category and probabilities.
    - Timestamps for the dataset and prediction.
* Success Condition (Postconditions):
  + The dashboard shows updated history and forecast with timestamps for the selected county.
* Exceptions:
  + Ingested dataset unavailable:
    - The system logs error.
    - A warning banner forms the end user that data is not available.
  + Prediction failure:
    - The system logs error.
    - A warning banner informs end user that the prediction failed and displays the last valid prediction with timestamp.
* Traceability to Functional Requirements (FRs):
  + FR-3.1: Train and run a predictive model.
  + FR-3.2: Output next-day AQI category for a county.
  + FR-3.3: Provide probability scores for each category.
  + FR-4.1: Provide a GUI for AQI data and predictions.
  + FR-4.2: Display 30-day chart + predicted category with probabilities.
  + FR-4.3: Provide a Refresh control to regenerate forecast.

**RATIONALE FOR YOUR USE CASE MODEL**

The case model separates manual data preparation from on-demand end user interaction to keep the system simple and testable in its current state.

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 end user flow focused 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:

* NFR-1.1: The system shall complete ingestion, validation, and feature generation for one county in ≤ 60 seconds.
* NFR-1.2: The system shall render the dashboard in ≤ 5 seconds (p95) after end user clicks Refresh.

1. Usability:

* NFR-2.1: The dashboard shall present AQI categories using standard EPA labels (e.g. Good, Moderate, Unhealthy for Sensitive Groups, Unhealthy, Very Unhealthy, Hazardous).
* NFR-2.2: The dashboard shall present each EPA category with a unique, distinguishable color code consistent with EPA/AirNow guidance (e.g. green, yellow, orange, red, purple, maroon).

1. Accessibility:

* NFR-3.1: The system shall conform to WCAG 2.1 Level AA accessibility requirements for visual content (e.g. SC 1.4.1 “Use of Color”, SC 1.4.3 “Contract (Minimum)”, SC 1.4.11 “Non-text Contrast”).

1. Reliability:

* NFR-4.1: Daily ingest shall succeed on ≥ 90% of runs.

1. Availability:

* NFR-5.1: The CLAP dashboard shall be available for use by end users at least 99% of the time over any rolling 30-day period once deployed.

1. Maintainability:

* NFR-6.1: The system shall implement structured logging for key operations (ingestion, validation, feature generation, and prediction).
* NFR-6.2: Log entries shall include timestamps and error details.

**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 6.1 – 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), Amelia Quinn (qcb220000) |
| 2 | 9/22/25 | v0.2 (bcf15f4) | v0.3 (f542477) | Jay Chung (cwc130330) | Revised all sections. | All Team Members |
| 3 | TBD | v0.3 (f542477) | v0.4 (3f36a14) | Jay Chung (cwc130330) | Revised Use case and Requirements. | All Team Members |
| 4 | TBD | v0.4 (3f36a14) | v0.5 () | Jay Chung (cwc130330) | Added Traceability (UC & FR), Revised NFRs. | All Team Members |

Table 6.2 – 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..f542477 |
| 3 | https://github.com/cchung7/rtx\_team1/compare/f542477..3f36a14 |
| 4 | https://github.com/cchung7/rtx\_team1/compare/3f36a14.. |

**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