Environmental Monitoring and Prediction System

SWENG 837 – Garrett Ruths

Git Repo - https://github.com/gdr5075/SWENG836_Final_Ruths

Video Presentation - https://youtu.be/f2AGIONQiSM

1. Problem Statement and Requirements

1.1. Business Requirements:

An Environmental Monitoring and Prediction System would provide a platform for the collection, analysis, and dissemination of data and trends related to key environmental factors, such as Air Quality, Water Quality, CO2 Levels, and Deforestation. It would provide tooling and methods for monitoring trends over time, determining the state of the environment at specific points the past, and would help to guide policy makers to decisions that can have the greatest impact on the wellbeing of our environment.

The system would have to be able to record many kinds of data and be able to geocode the data to a specific location or region. Once the data has been recorded, analysis can begin, and the system must support many kinds of analysis. In terms of extensibility, analysis will be a key area where the system must be open to extension, whether as input from a 3rd party or as new functionality programmed into the system. To reach the most customers, the system should be able to provide information in the form of raw data, summaries, analyzed reports, and, critically, as a health monitoring platform which can provide notifications when measurements fall outside expected bounds.

While reaching as many customers as possible is always a desire, the primary focus of the platform will be on researchers, analysists, and policy makers who can take the information the platform provides and act on it in one way or another. The functionality of the system can also be opened as a Business-to-Business model, where the data and analysis can be provided as a service. Users of our system will be looking for solid authoritative data, which can be used in their own analysis and research. Additionally, the monitoring and tracking features of the platform will provide automatic monitoring and analysis of data to produce trends and to notify customers when measurements fall outside the expected.

As an environmentally focused system, its business goals should include energy efficiency and scalability to support the greatest number of customers without adversely impacting the environment we hope to protect. As such, the software will eschew technologies that can be expected to require large amounts of energy, such as machine learning, and blockchain. Advances in machine learning may cause this decision to change, and the system should be open to extension in this field, but as currently is, the technology should not be included for environmental reasons.

1.2. Non-Functional Requirements:

Being an internet-based platform focused on correctness and accuracy, the main bulk of the system needs to support scalability, mostly horizontally, so that as customer demands grow, the system can support the addition of more customers without impacting responsiveness. The parts of the system that are customer focused, such as monitoring and analysis need to be reasonably performant, so that customers are not sitting around waiting for their requested information to load, but services such as raw data dumps and large downloads do not necessarily require exceptional response time or throughput. Given the nature of the data that is being recorded, maintaining multiple copies of the data is likely to be expensive, so vertical scaling will be preferred. In general, any user interface should be expected to initially load in a second or two and have usable data in front of the customer within 10 to 15 seconds.

Given the nature of the data, there will not be stringent security requirements past the standard expected security. Technologies such as HTTPS encryption and simple account authorization will be sufficient. Since there should not be any personal data, there would not be any great expectation of safeguarding of data, or for advanced encryption on communication. Security should be focused on many of the pitfalls of databases and web services, such as Cross-Site-Scripting and SQL Injection.

As a system dedicated to recording and maintaining information over long periods of time, documentation and code maintainability will be important. The system should be extensible and support the addition of new kinds of data and new ways to analyze data without breaking backwards compatibility. Additionally, any Business-To-Business capabilities must be well documented and maintained so that services and APIs we provide can be reliable and highly available.

2. Domain Modeling

2.1. Case Diagrams

2.1.1. Actors

Type	Actor	Goal Description
Primary	Researcher / Analyst	Analysts and Researchers, whether from industry or government, will want a platform that allows them to view important environmental data, how it has changed over time. They will want the ability to directly access raw data and download it for further analysis.
Primary	Policy Makers	Policy Makers, whether from industry or government, will want data that is easy to access, clearly shows trends over time, and empowers their decision making. Policy makers may also want to be notified if there are changes in predictions for a particular system or measurement.
Primary	Average Citizen	Average Citizens want a system that is easy to use and allows them to quickly and easily see the big picture trends of the environment in their area. They may only have a curiosity, or may be trying to research data
Supporting	Database System	One of the key actors that the system would rely on when providing services to customers would be a database, which should provide analysis and monitoring in a highly available format.
Supporting	Analysis Engine	Another important actor the system as a whole would rely on would be some sort of analysis engine. Just in time analysis would not likely be sufficient for our customers, and thus, some computational power and software must be dedicated directly to the analysis of data.

Supporting	Sensors and Data Reporters	The system is useless without data to analyze, so 3rd party sensors and systems that generate data will be an important, but largely passive actor for the system.
Offstage	Federal Agencies	Much of the environmental monitoring that occurs is overseen by federal agencies. Accessing their data and making decisions based on their plans and upcoming system will be an important aspect of the long-term viability of the system.
Offstage	Federal Regulators	Federal Regulators will be interested in ensuring all federal regulations that the system may be beholden to are upheld.

2.1.2. Use Case Scenarios

2.1.2.1. Scenario 1

Use Case Scenario	Comment
Name	Access Environmental Data
Scope	Data System
Level	User Goal
Primary Actor	Researcher / Analyst
Stakeholders and Interests	Researchers and Analysts want a system that allows them to easily access and download important historical data for their work.
Preconditions	Historical Data exists in the system
Sucess Guarantee	System provides the data in a standard format, such as zip or pdf.
Main Sucess Scenario	1 – User access Historical Data System 2 – User provides search criteria such as location and timeframe 3 – System retrieves the data based on the query 4 – System packages the data and provides the user with a download link or file.
Extensions	A – If there is no data that fits the criteria the user selected, the user is prompted to change their criteria.
Special Requirements	Communication must follow standard encryption standards to protect information.

2.1.3. Scenario 2

Use Case Scenario	Comment
Name	Upload new Environmental Data
Scope	Data System
Level	System Goal
Primary Actor	Sensors and Data Reporters Database System
Stakeholders and Interests	A primary goal of the system is the collection and recording of environmental data for future analysis.
Preconditions	The environmental data is in a format that is compatible with the system
Sucess Guarantee	System records the environmental data to its database and any replication that must occur is carried out.
Main Sucess Scenario	1 – Another system or sensor provides environmental data to the system. 2 – System converts data where necessary for compatibility with the database. 3 – System automatically checks data for threshold exceptions
Extensions	A – If the data exceeds set limits, a notification will be generated (Scenario 5). B – System checks data for use in its analysis engine and generates a new analysis report if necessary (Scenario 3). C – If replication has been configured, the system replicates data to ensure data integrity.
Special Requirements	Communication must follow standard encryption standards to protect data.

2.1.3.1. Scenario 3

Use Case Scenario	Comment
Name	Generate Analysis
Scope	Data System
Level	System Goal
Primary Actor	Analysis Engine
Stakeholders and Interests	A primary goal of the system is the analysis of historical environmental data.

Preconditions	The environmental data is in a format that is compatible with the system
Sucess Guarantee	The system analyses the data for changes in historical trends and produces summary information.
Main Sucess Scenario	1 – System event triggers analysis 2 – System loads up historical environmental data related to the location. 3 – System automatically calculates values for standard statical analysis numbers, such as mean, standard deviation, standard deviation as a percent of the mean, and additional calculations as deemed necessary. 4 – System stores analysis alongside the data for future retrieval.
Extensions	A – If limits on analyzed data are exceeded, the Analysis Engine generates a report.
Special Requirements	Communication must follow standard encryption standards to protect data.

2.1.3.2. Scenario 4

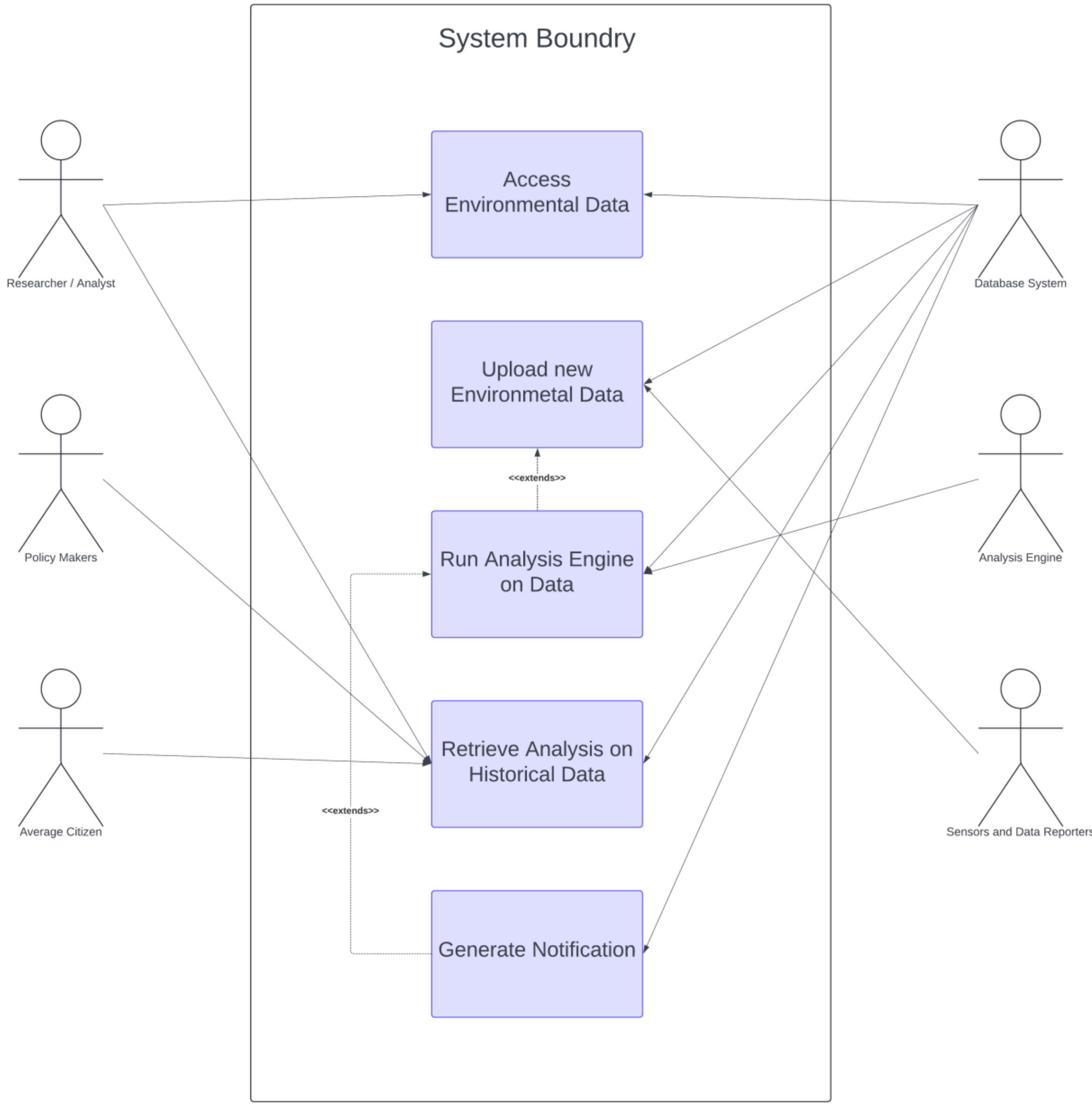
Use Case Scenario	Comment
Name	Retrieve Analysis on Historical Data
Scope	Data System
Level	User Goal
Primary Actor	Policy Makers
Stakeholders and Interests	A policy maker wishes to decide on some topic and to know more about the historical environmental data related to that decision.
Preconditions	The environmental data already exists for the given location.
Sucess Guarantee	System provides analysis of historical data to the user and provides links to retrieve more granular data.
Main Sucess Scenario	1 – User requests analysis of historical environmental data. 2 – System retrieves historical data. 3 – System provides the historical data to the user and shows graphics for some standard analysis techniques. 4 - System gives the user the option to be notified of changes to historical trends related to that query.
Extensions	A – If the historical data has not been analyzed, the system automatically analysis the data (Scenario 3).

Special Requirements	Communication must follow standard encryption standards to protect data.

2.1.3.3. Scenario 5

Use Case Scenario	Comment
Name	Generate Notification
Scope	Data System
Level	System Goal
Primary Actor	Analysis Engine
Stakeholders and Interests	The system wants to notify important personnel when historical trends change, or data exceeds certain thresholds.
Preconditions	The environmental data already exists for the given location.
Sucess Guarantee	System produces a notification and sends it out to users who have requested notifications for that data set.
Main Sucess Scenario	1 – During analysis, the system determines that data exceeds set bounds. 2 – System retrieves historical data and analysis to put into a report. 3 – System places analysis and historical data into a report and stores it in the database. 4 - System generates a notification for users and sends out the notification with a summary of the report and a link to retrieve the full report.
Special Requirements	Communication must follow standard encryption standards to protect data.

2.1.4. Use Case Diagram



2.2. Domain Model

2.2.1. Conceptual Classes

Conceptual Class Category	Example from Unified Patient Manager
Physical or Tangible Objects	-
Specifications, Designs or Descriptions of Things	Environemtal Regulations, Environmental Standards
Places	Locations, States, Countries
Transactions	Query
Roles of People	Analyst, Researcher, Average Citizen, Policy Makers, User
Containers of Other Things	Environemtal Data, ZIP, PDF
Things in a Container	Air Quality, Water Quality, Deforestation Measurements
Other Computers/Systems (external)	Sensor Data
Abstract Noun Concepts	Database, Historical Data, File, Criteria, Timeframe, Packages, Link, Threshold, Limits
Organizations	NASA, EPA, AEC
Events	Upload, Download, Generate Notification, Search, Format, Replication
Processes	Encryption, Analysis
Rules and Policies	Clear Water Act
Manuals, Books, Documents, Reference Papers	Report

2.2.2. Pruned Classes

Too Vague - Environemtal Regulations, Environmental Standards, Locations, Packages, Link, Upload, Download, Search, Format, Replication, Encryption, Analysis

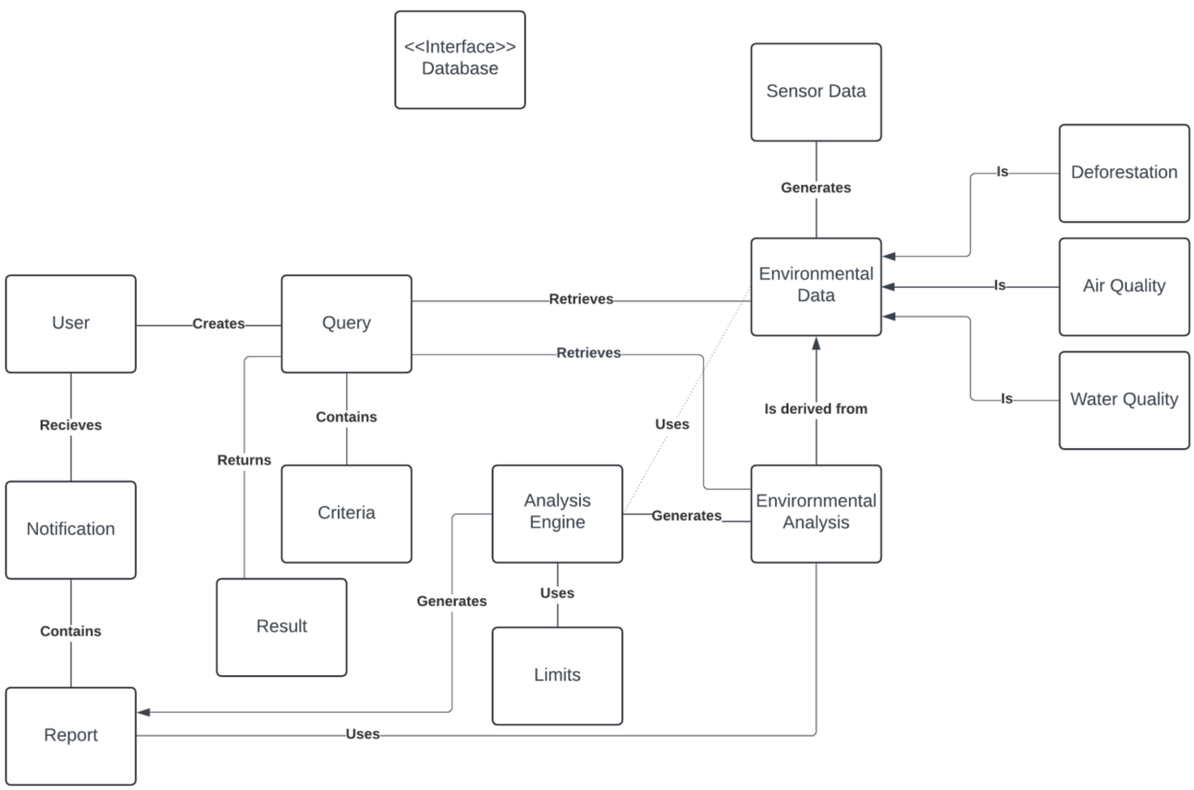
Irrelevant Classes – Query, Analyst, Researcher, Average Citizen, Policy Maker, NASA, EPA, AEC

Attribute Classes – State, Country, Historical Data, File, Timeframe, Threshold, Clean Water Act

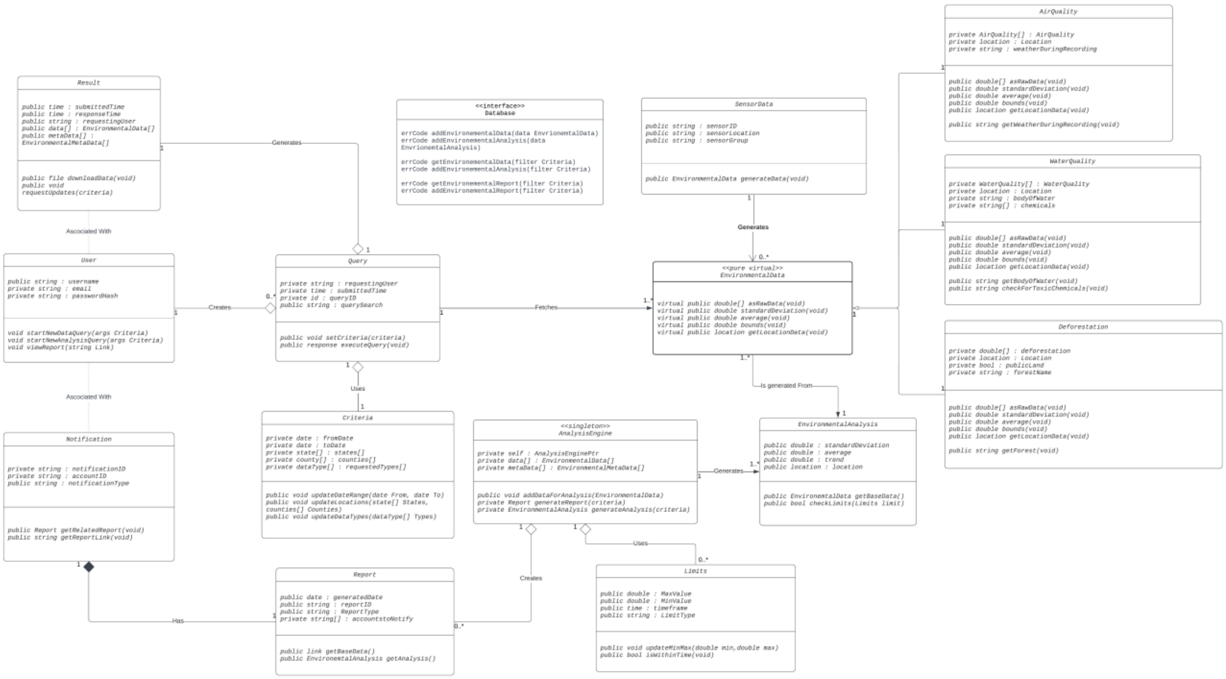
---

Remaining Classes – Environmental Data, Air Quality, Water Quality, Deforestation, Sensor Data, Database, Query, Limits, Notification, Report, User, Analysis Engine, Criteria

2.2.3. Domain Model

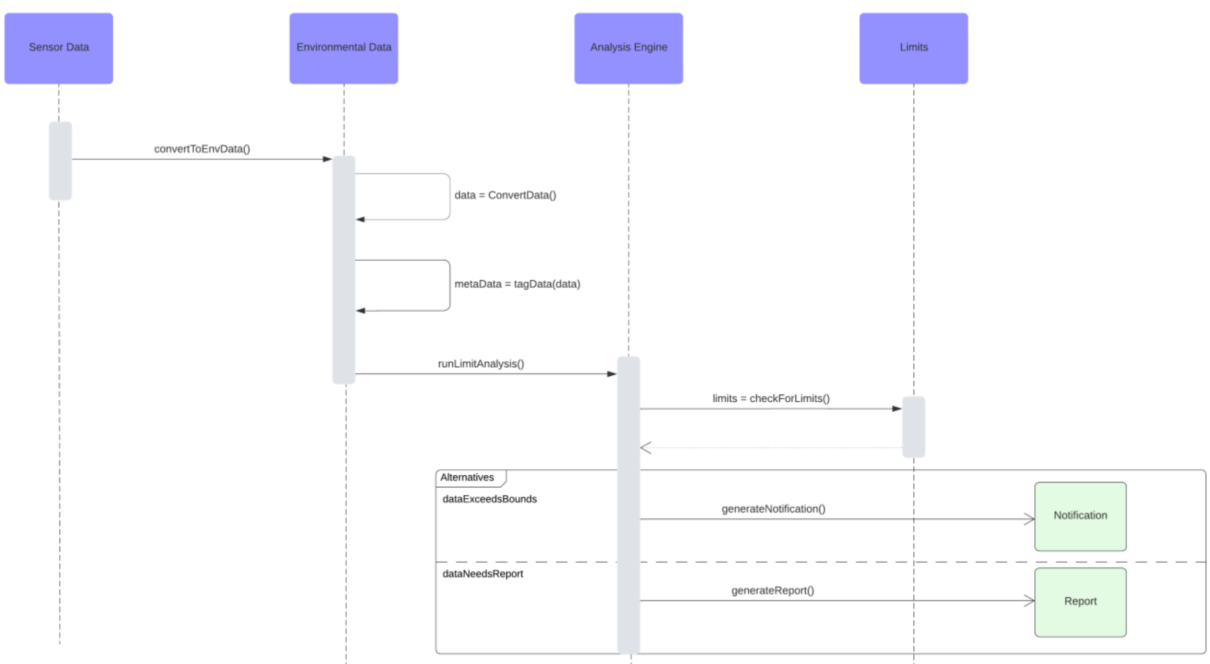


2.3. Class Diagram

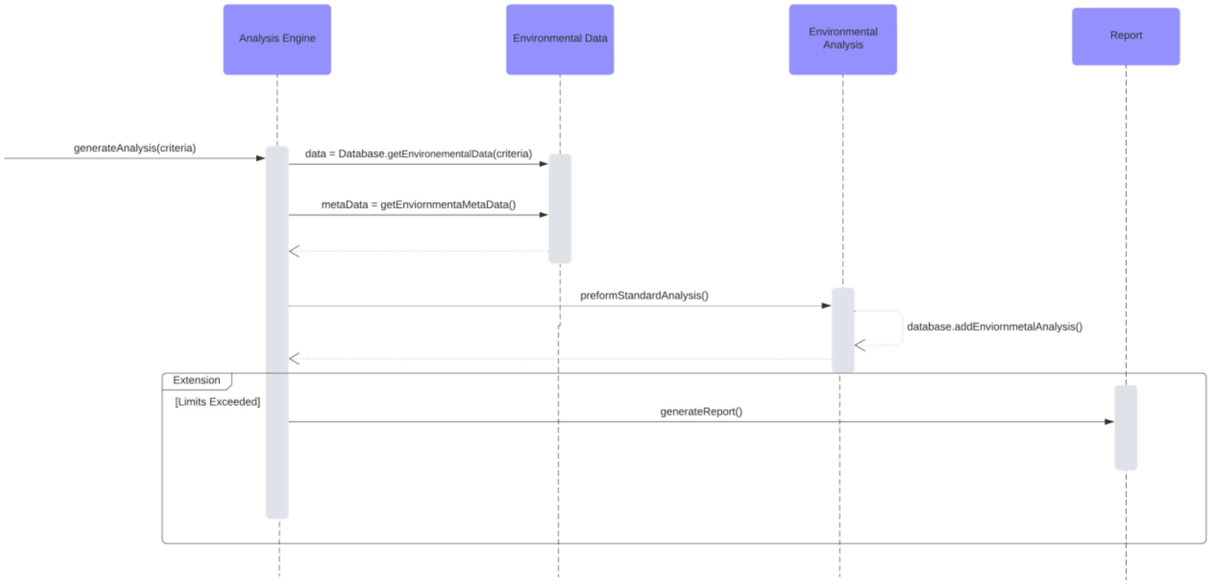


2.4. Sequence Diagram

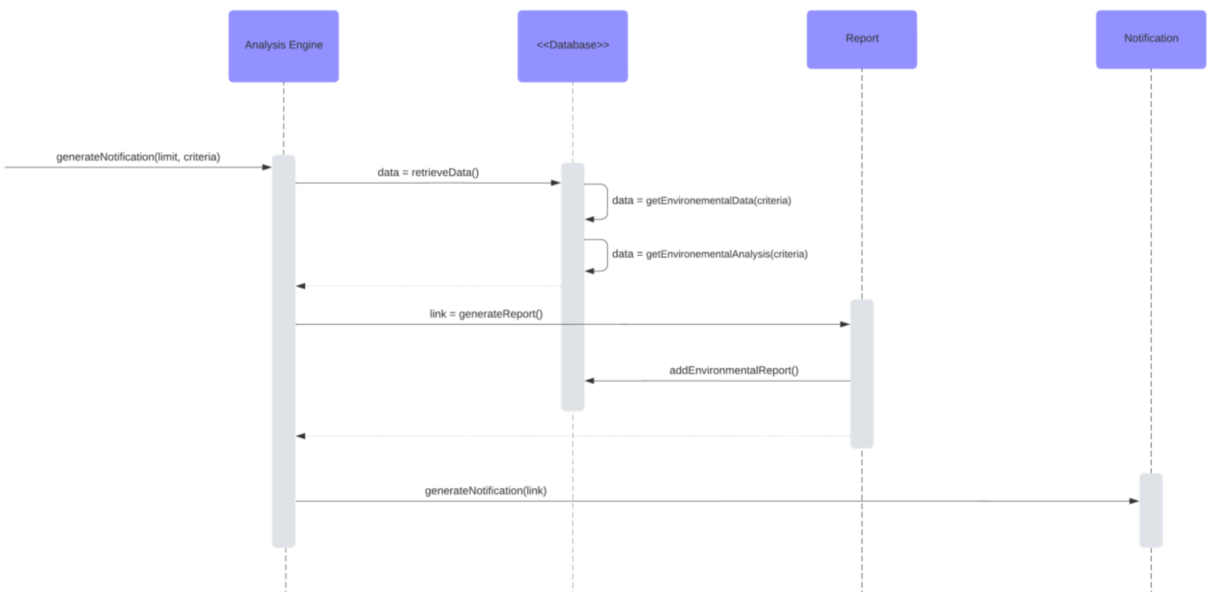
2.4.1. Sequence Diagram Scenario 2



2.4.2. Sequence Diagram Scenario 3

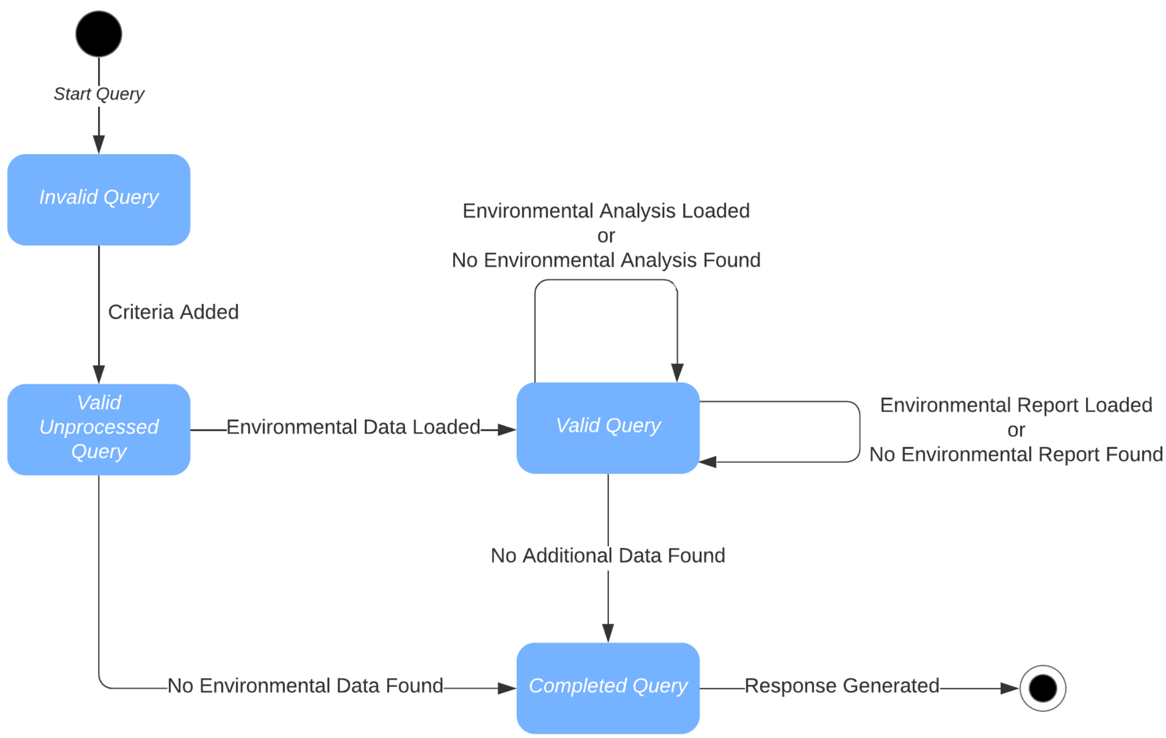


2.4.3. Sequence Diagram Scenario 5

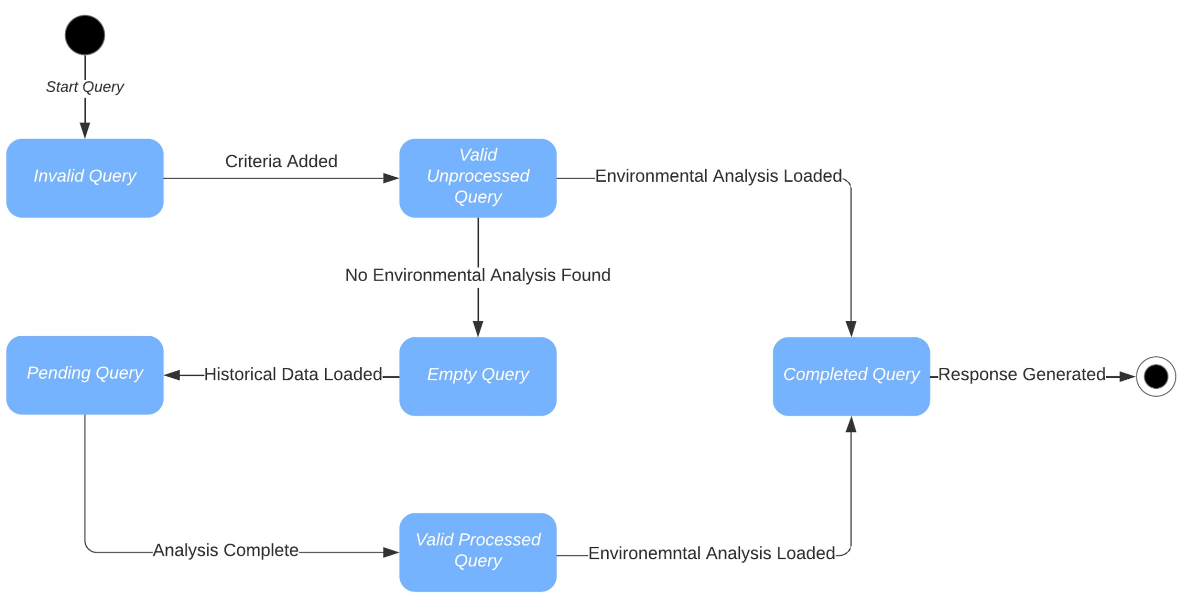


2.5. State Diagram

2.5.1. *State Diagram Scenario 1*

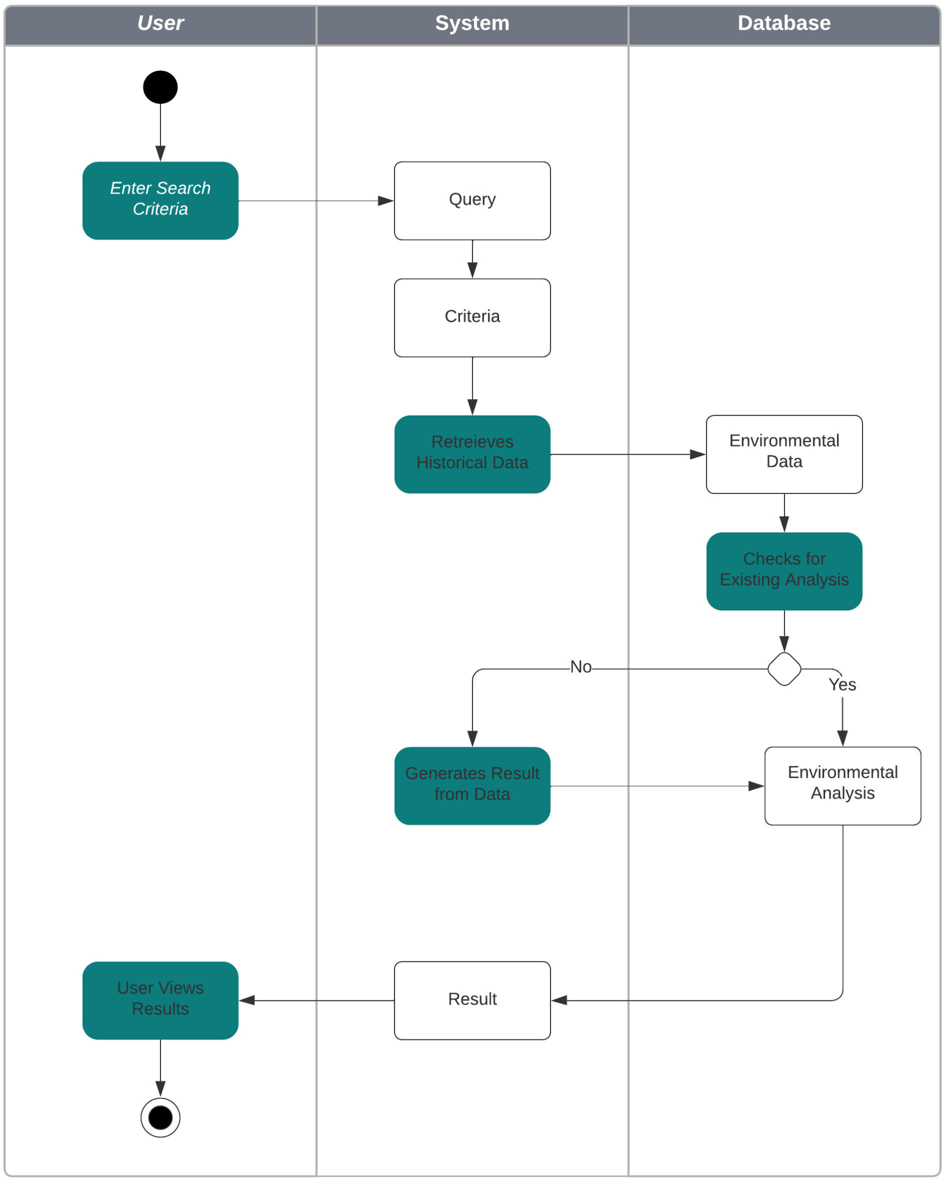


2.5.2. *State Diagram Scenario 4*

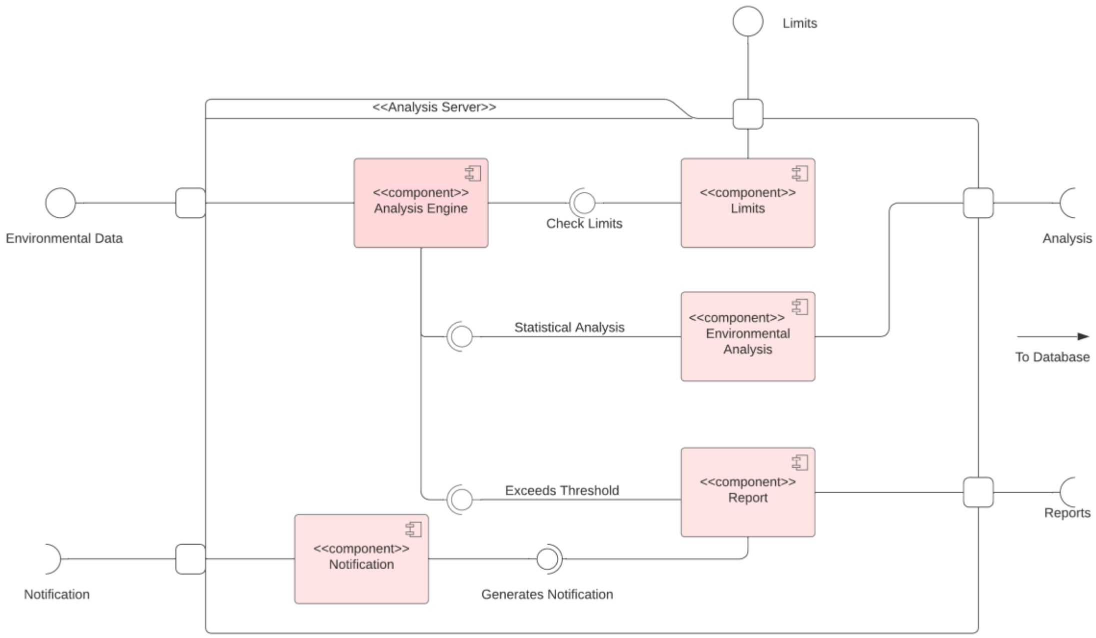


2.6. Activity Diagram

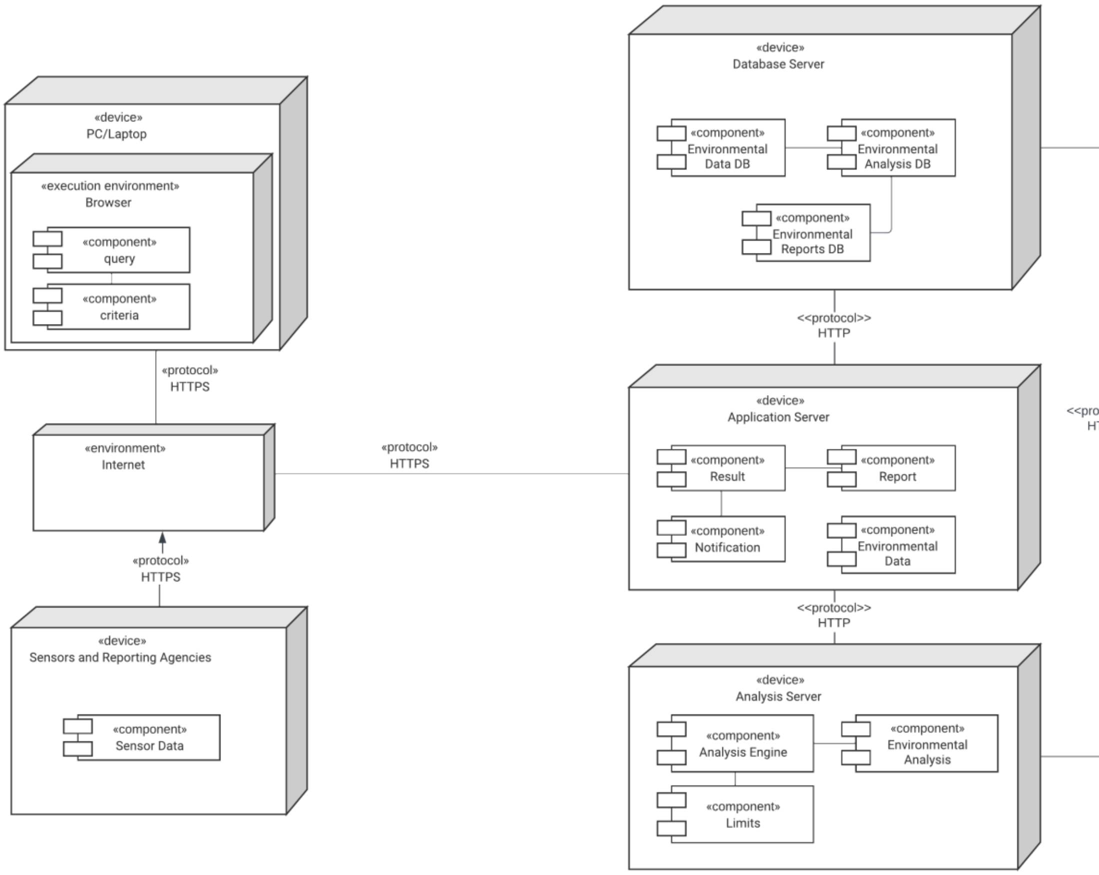
2.6.1. *Activity Diagram Scenario 1*



2.7. Component Diagram



2.8. Deployment Diagram



2.9. Skeleton Classes

2.10. Design Patterns

Singleton / Pure Fabrication- A design pattern I find myself constantly returning to, and one that I include in this project as well is the Singleton pattern. Somewhat controversial, I find that the Singleton pattern goes a long way towards simplifying code when it is used correctly. As with any globally accessible interface, its overuse increases code coupling, makes refactoring difficult, and can cause difficulties in testing. For this reason, there is only one singleton used in this design, and that is for the implementation of the interface for the database. Having a class own the database connection does increase code coupling, and will mean that the rest of the codebase will rely on this class quite heavily, but if the public functions on the interface are kept to a minimum, the logic under the hood can be refactored and improved without impacting the rest of the codebase. Additionally, as needs grow, new behaviors, such as batch processing, async await, and multithreaded (or even multiple databases) can be hidden from the rest of the code. This class allows follows the “Pure Fabrication” Grasp pattern, and should be written in a way as to allow for new behaviors and implementations without effecting the rest of the code.

DRY / Polymophism– One of the easiest ways I find to reduce code duplication is to properly implement generic functions when possible, or make data classes robust enough to make reuse an advantage, even when code coupling in increased. With that in mind, there are two strategies in this design which attempt to keep code duplication down. One is the use of the “Criteria” class. This class is intended to be the main way that data is pruned down in searches. In this case, it is extremely important that all classes in the code base agree on what a filter means in terms of what data is returned from a search. For data integrity and analysis purposes, a failure in filtering would mean invalid analysis and improper data set inclusion or exclusion. The second attempt to reduce code duplication lies in the pure virtual class “EnvironmentlData”. This class is meant to be the primary way the rest of the code base reads and processes data, and it provides the common interface for this classes polymophism. This generic implementation allows code to run analysis without needing to know exactly what data it is being ran on. This reduces the need to reimplement common behavior and make the code simpler overall.