Testing and Inspection Report

Prepared by : Dhaval Patel, Kruneet Patel,
Charvi Virani, and Michael Gallagher
for use in CS 440
at the University of Illinois Chicago
September 2018

Table of Contents

List of Figures

List of Tables

I Project Description

1 Project Overview

This project is intended to create a water quality reporting and flood forecasting app. The app targets two type of users : public users who log in from their homes and laboratory users that work at industrial plants or water quality labs. The public interface reports on the weather, the water quality in the immediate area, and the flood chance. The lab user is the same with a more in depth analysis and the possibility of generating reports on the current and historical data. Our design is intended to use publicly available weather data from weather APIs to display information and generate reports.

2 Project Domain

The system uses MVC framework architecture along with a reliable backend and some javascripts and json objects used to work with open source API. The sensor interface should correctly validate the mock sensor's input data for various metrics used for water quality report generation and flood forecasting. Also the system has two major use cases in terms of laboratory user and public user.The lab user can use the Map API to fetch the water quality data for a particular location he selects on the Map. The data is then validated and fed into the database which can be further edited/deleted by the lab user using the GUI. The public user interface interacts with OpenMap Weather API, which gives the attribute values for certain parameters that affect flood forecasting. The reports are generated to give visual representation using various plots which are dynamically generated from the database.

3 Relationship to Other Documents

This document outlines the testing and inspecting that was conducted to assure that the finished product met the requirements defined in the Project Requirements Document.

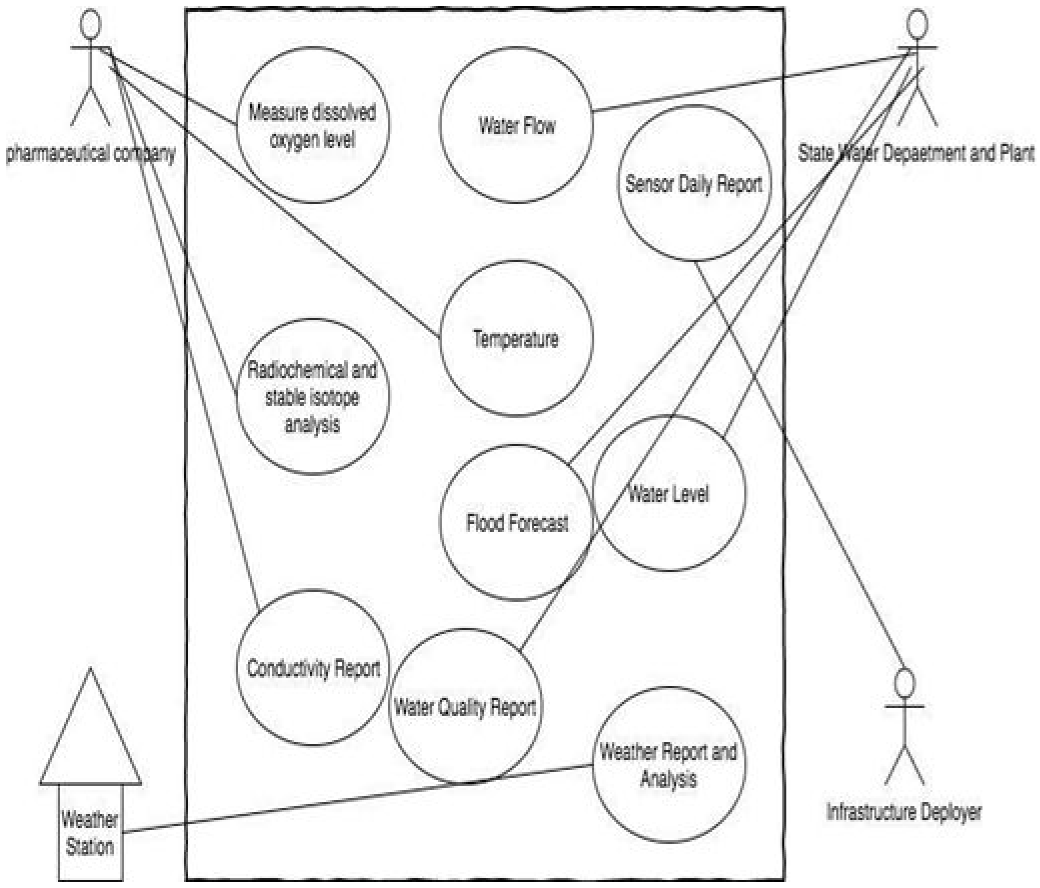
4 Naming Conventions and Definitions

The same definition and terms that are used from the previous coding/development report.

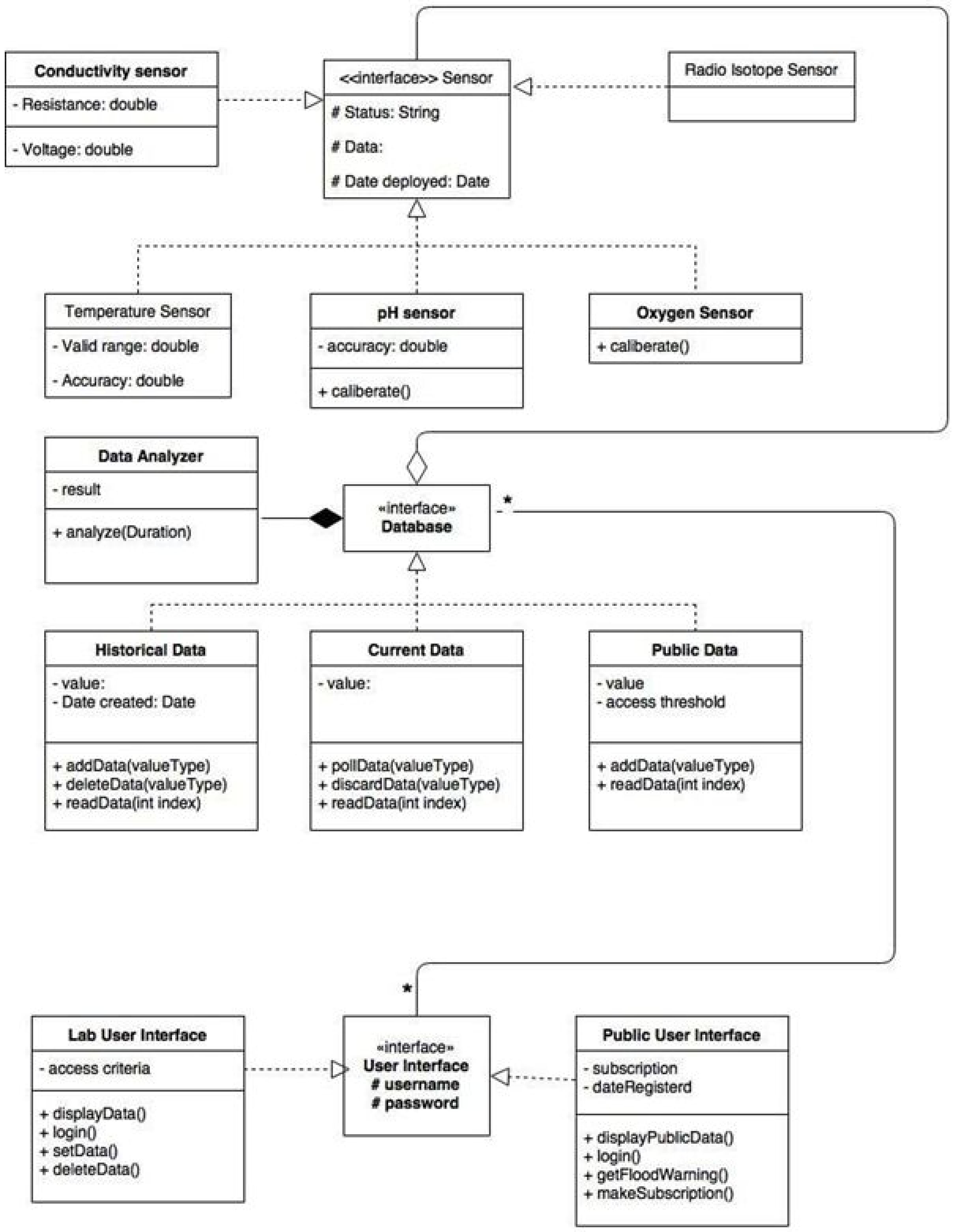
4a Definitions of Key Terms

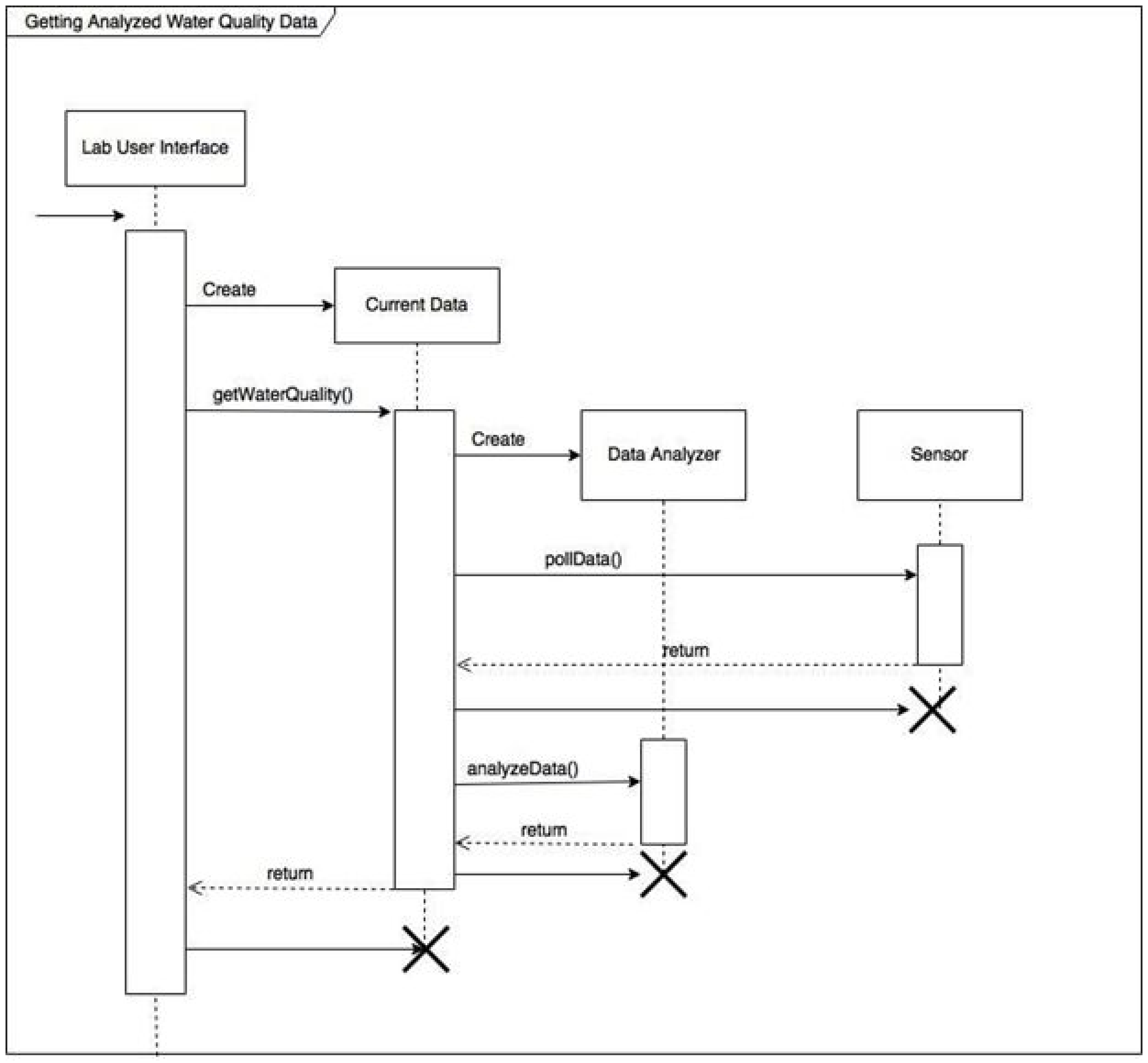
The same definition and terms that are used from the previous coding/development report.

4b UML and Other Notation Used in This Document

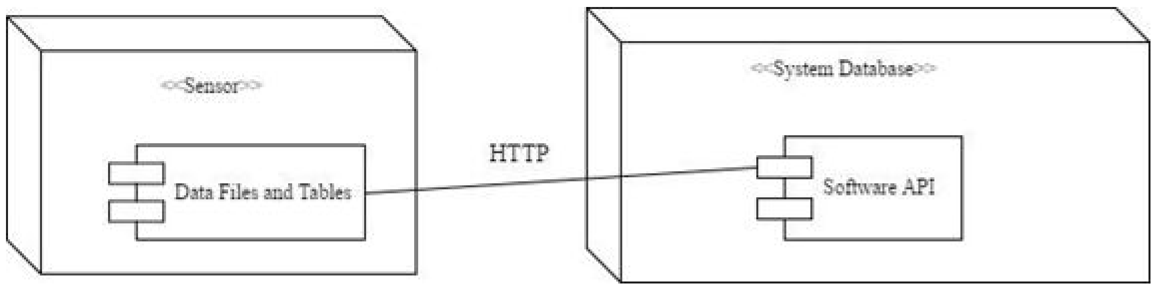


Use case Diagram


Conductivity sensor
- Resistance: double
- Voltage: double
<<interface>> Sensor
# Status: String
# Data:
# Date deployed: Date
Radio Isotope Sensor
Temperature Sensor
- Valid range: double
- Accuracy: double
pH sensor
- accuracy: double
+ caliberate()
Oxygen Sensor
+ caliberate()
Data Analyzer
- result
+ analyze(Duration)
«interface»
Database
*
Historical Data
- value:
- Date created: Date
+ addData(valueType)
+ deleteData(valueType)
+ readData(int index)
Current Data
- value:
+ pollData(valueType)
+ discardData(valueType)
+ readData(int index)
Public Data
- value
- access threshold
+ addData(valueType)
+ readData(int index)
*
Lab User Interface
- access criteria
+ displayData()
+ login()
+ setData()
+ deleteData()
«interface»
User Interface
# username
# password
Public User Interface
- subscription
- dateRegisterd
+ displayPublicData()
+ login()
+ getFloodWarning()
+ makeSubscription()



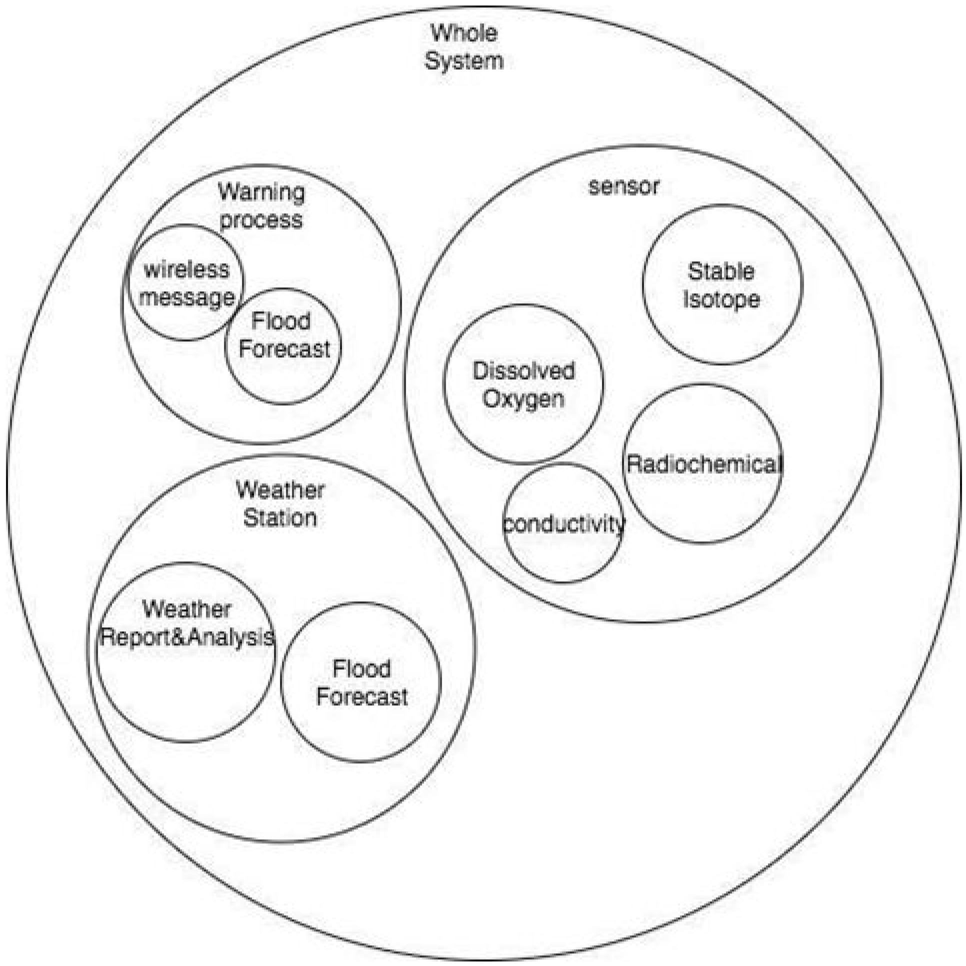
Getting water quality data analyzed



Mapping of Hardware/Software

4c Data Dictionary for Any Included Models

The system shares data internally.



II Testing

1 Items to be Tested

1. Home Page
2. User Interface
3. Public User Interface Page
4. Laboratory User Interface
5. Sensor Interface
6. Database

7. Map API Functionality
8. OpenWeatherMap Functionality
9. Graphical Report Generation
10. Admin Pages

2 Test Specifications

ID 1 - Home Page Test

Description: Test the home page of the webapp.

Items covered by this test: Home Page

Requirements addressed by this test: Not covered by requirements but necessary for the use of the webapp.

Environmental needs: Must support MySQL and VisualStudio solutions

Intercase Dependencies: . None

Test Procedures:

1. Run program
2. Enter the URL for the home page into the browser
3. Inspect interface to ensure it meets requirements

Input Specification: URL: localhost:54950/HomeTrial/Home/Index

Output Specifications: A web page with no errors on loading.

Pass/Fail Criteria:

Pass: All specified outputs are present.

Fail: Either of the outputs are missing or have some other issue.

ID 2 - User Interface Page Test

Description: Test of the select user type page to make sure it loads, displays and redirects all users properly.

Items covered by this test: User interface

Requirements addressed by this test: Functional requirements 1 and 2.

Environmental needs: Must support MySQL and VisualStudio solutions

Intercase Dependencies: None

Test Procedures:

1. Run program
2. Enter the URL for the user interface into the browser
3. Inspect interface to ensure it meets requirements

Input Specification: URL: localhost:54950/HomeTrial/UserInterface

Output Specifications: A web page with no errors on loading. Properly formatted data. Redirects to appropriate user page.

Pass/Fail Criteria:

Pass: All specified outputs are present.

Fail: Either of the outputs are missing or have some other issue.

ID 3 - Public User Interface Page Test

Description: Test of the public user page to make sure it loads and displays all information properly.

Items covered by this test: Public user interface, OpenWeatherMap API

Requirements addressed by this test: Functional requirements 1 and 2.

Environmental needs: Must support MySQL and VisualStudio solutions

Intercase Dependencies: OpenWeatherMap API, Database

Test Procedures:

1. Run program
2. Enter the URL for the public user interface into the browser
3. Inspect interface to ensure it meets requirements

Input Specification: URL: localhost:54950/HomeTrial/PublicUser

Output Specifications: A web page with no errors on loading. Properly formatted data.

Pass/Fail Criteria:

Pass: All specified outputs are present.

Fail: Either of the outputs are missing or have some other issue.

ID 4 - Laboratory User Interface Page Test

Description: Test of the public user page to make sure it loads, displays, validates and passes all information properly.

Items covered by this test: Laboratory user interface, Database

Requirements addressed by this test: Functional requirements 1 and 2.

Environmental needs: Must support MySQL and VisualStudio solutions

Intercase Dependencies: Database

Test Procedures:

1. Run program
2. Enter the URL for the public user interface into the browser
3. Inspect interface to ensure it meets requirements

Input Specification: URL: localhost:54950/HomeTrial/LaboratoryUser

Output Specifications: A web page with no errors on loading. Properly formatted data. Validates input information. Passes on validated information to the database.

Pass/Fail Criteria:

Pass: All specified outputs are present.

Fail: Either of the outputs are missing or have some other issue.

ID 5 - Sensor Interface Page Test

Description: Test of the sensor page to make sure it loads, validates and passes on all information properly.

Items covered by this test: Sensor Interface page, Database

Requirements addressed by this test: Functional requirements 1 and 2.

Environmental needs: Must support MySQL and VisualStudio solutions

Intercase Dependencies: Database, Admin View

Test Procedures:

1. Run program
2. Enter the URL for the public user interface into the browser
3. Inspect interface to ensure it meets requirements

Input Specification: URL: localhost:54950/HomeTrial/PublicUser

Output Specifications: A web page with no errors on loading. Properly formatted data. Validates input information. Passes on validated information to the database.

Pass/Fail Criteria:

Pass: All specified outputs are present.

Fail: Either of the outputs are missing or have some other issue.

<u>ID 6 - Database Test</u>

Description: Test of the database to make sure it loads, displays, connects and queries all information properly.

Items covered by this test: Sensor Interface page, Laboratory User Interface page, Graphical Report Generation page, Map API

Requirements addressed by this test: Functional requirements 1 and 2.

Environmental needs: Must support MySQL and VisualStudio solutions

Intercase Dependencies: Admin View Page Test

Test Procedures:

1. Run program
2. Enter the URL for the public user interface into the browser
3. Inspect interface to ensure it meets requirements

Input Specification: Through Admin Pages

Output Specifications: Able to create, update and delete records from the database tables

Pass/Fail Criteria:

Pass: All specified outputs are present.

Fail: Either of the outputs are missing or have some other issue.

ID 7 - Map Test

Description: Test of the Map API to make sure it loads, displays and passes all information properly.

Items covered by this test: Public user and laboratory user landing page

Requirements addressed by this test: Functional requirements 1 and 2.

Environmental needs: Must support MySQL and VisualStudio solutions

Intercase Dependencies: NA

Test Procedures:

1. Run program
2. Enter the URL for the public user interface into the browser
3. Inspect interface to ensure it meets requirements

Input Specification: URL: localhost:54950/HomeTrial/PublicUserLandingPage and URL: localhost:54950/HomeTrial/LaboratoryUserLandingPage

Output Specifications: A web page with no errors on loading. Displaying area map. Onclick functionality of getting location coordinates and passing them onto the main interface. Properly formatted data.

Pass/Fail Criteria:

Pass: All specified outputs are present.

Fail: Either of the outputs are missing or have some other issue.

ID 8 - OpenWeatherMap API Page Test

Description: Test of the OpenWeatherMap API to make sure it loads and displays all information properly.

Items covered by this test: Public user interface

Requirements addressed by this test: Functional requirements 1 and 2.

Environmental needs: Must support MySQL and VisualStudio solutions

Intercase Dependencies: NA

Test Procedures:

1. Run program
2. Enter the URL for the public user interface into the browser
3. Inspect interface to ensure it meets requirements

Input Specification: URL: localhost:54950/HomeTrial/PublicUser

Output Specifications: A web page with no errors on loading. Properly formatted data. The web page contains weather information as per the online database source OpenWeatherMap

Pass/Fail Criteria:

Pass: All specified outputs are present.

Fail: Either of the outputs are missing or have some other issue.

<u>ID 9 - Graphical Report Generation Page Test</u>

Description: Test of the graphical report generation tool to make sure it loads and displays all graphs properly and dynamically changes according to changes in database.

Items covered by this test: Graphical Report Generation tool, view page, Sensor Data table in database and Laboratory user interface page

Requirements addressed by this test: Functional requirements 1 and 2.

Environmental needs: Must support MySQL and VisualStudio solutions

Intercase Dependencies: Database, Sensor Interface Page

Test Procedures:

1. Run program
2. Enter the URL for the public user interface into the browser
3. Inspect interface to ensure it meets requirements

Input Specification: URL: localhost:54950/HomeTrial/GraphReportGen

Output Specifications: A web page with no errors on loading. Properly formatted graphs.

Pass/Fail Criteria:

Pass: All specified outputs are present.

Fail: Either of the outputs are missing or have some other issue.

ID 10 - Admin View Page Test

Description: Test of the admin user page to make sure it loads and displays all information properly.

Items covered by this test: Admin View table, Database

Requirements addressed by this test: Functional requirements 1 and 2.

Environmental needs: Must support MySQL and VisualStudio solutions

Intercase Dependencies: NA

Test Procedures:

1. Run program
2. Enter the URL for the public user interface into the browser
3. Inspect interface to ensure it meets requirements

Input Specification: URL: localhost:54950/HomeTrial/AdminView

Output Specifications: A group of web pages with no errors on loading. Properly formatted data. Properly connects to database. Able to create, update and view database records.

Pass/Fail Criteria:

Pass: All specified outputs are present.

Fail: Either of the outputs are missing or have some other issue.

3 Test Results

ID 1 - Home Page Test

Date(s) of Execution: 11/20/2018

Staff conducting tests: Michael Gallagher, Dhaval Patel

Expected Results: A web page with no errors on loading.

Actual Results: A web page with no errors on loading.

Test Status: PASS

ID 2 - User Interface Page Test

Date(s) of Execution: 11/20/2018

Staff conducting tests: Michael Gallagher, Dhaval Patel

Expected Results: A web page with no errors on loading. Properly formatted data. Redirects to appropriate user page.

Actual Results: A web page with no errors on loading. Properly formatted data. Redirects to appropriate user page.

Test Status: PASS

ID 3 - Public User Interface Page Test

Date(s) of Execution: 11/22/2018

Staff conducting tests: Kruneet Patel, Charvi Virani

Expected Results: A web page with no errors on loading. Properly formatted data.

Actual Results: A web page with no errors on loading. Properly formatted data.

Test Status: PASS

ID 4 - Laboratory User Interface Page Test

Date(s) of Execution: 11/22/2018

Staff conducting tests: Kruneet Patel, Charvi Virani

Expected Results: A web page with no errors on loading. Properly formatted data. Validates input information. Passes on validated information to the database.

Actual Results: A web page with no errors on loading. Properly formatted data. Validates input information. Passes on validated information to the database.

Test Status: PASS

ID 5 - Sensor Interface Page Test

Date(s) of Execution: 11/24/2018

Staff conducting tests: Michael Gallagher, Kruneet Patel

Expected Results: A web page with no errors on loading. Properly formatted data. Validates input information. Passes on validated information to the database.

Actual Results: A web page with no errors on loading. Properly formatted data. Validates input information. Passes on validated information to the database.

Test Status: PASS

ID 6 - Database Test

Date(s) of Execution: 11/25/2018

Staff conducting tests: Charvi Virani, Dhaval Patel

Expected Results: Able to create, update and delete records from the database tables

Actual Results: Able to create, update and delete records from the database tables

Test Status: PASS

ID 7 - Map API Test

Date(s) of Execution: 11/26/2018

Staff conducting tests: Michael Gallagher, Charvi Virani

Expected Results: A web page with no errors on loading. Displaying area map. Onclick functionality of getting location coordinates and passing them onto the main interface. Properly formatted data.

Actual Results: A web page with no errors on loading. Displaying area map. Onclick functionality of getting location coordinates and passing them onto the main interface. Properly formatted data.

Test Status: PASS

ID 8 - OpenWeatherMap API Test

Date(s) of Execution: 11/27/2018

Staff conducting tests: Kruneet Patel, Dhaval Patel

Expected Results: A web page with no errors on loading. Properly formatted data. The web page contains weather information as per the online database source OpenWeatherMap

Actual Results: A web page with no errors on loading. Properly formatted data. The web page contains weather information as per the online database source OpenWeatherMap

Test Status: PASS

ID 9 - Graphical Report Generation Page Test

Date(s) of Execution: 11/28/2018

Staff conducting tests: Michael Gallahager, Kruneet Patel

Expected Results: A web page with no errors on loading. Properly formatted graphs.

Actual Results: A web page with no errors on loading. Properly formatted graphs.

Test Status: PASS

ID 10 - Admin View Page Test

Date(s) of Execution: 11/29/2018

Staff conducting tests: Charvi Virani, Dhaval Patel

Expected Results: A group of web pages with no errors on loading. Properly formatted data. Properly connects to database. Able to create, update and view database records.

Actual Results: A group of web pages with no errors on loading. Properly formatted data. Properly connects to database. Able to create, update and view database records.

Test Status: PASS

4 Regression Testing

ID 3 - Public User Interface Page Test

Description: Test of the public user page to make sure it loads and displays all information properly.

Test Procedures: To be repeated in case of change in role based access for public user and/or any changes in OpenWeatherMap API

ID 4 - Laboratory User Interface Page Test

Description: Test of the public user page to make sure it loads, displays, validates and passes all information properly.

Test Procedures: To be repeated in case of change in role based access for public user and/or any changes in Database constraints

ID 5 - Sensor Interface Page Test

Description: Test of the sensor page to make sure it loads, validates and passes on all information properly.

Test Procedures: To be repeated in case of any changes in Database constraints

ID 6 - Database Test

Description: Test of the database to make sure it loads, displays, connects and queries all information properly.

Test Procedures: To be repeated in case of any changes in Database constraints

ID 7 - Map Test

Description: Test of the public user page to make sure it loads and displays all information properly.

Test Procedures: To be repeated in case of changes in information passed between Map API and database and users

ID 8 - OpenWeatherMap API Page Test

Description: Test of the OpenWeatherMap API to make sure it loads and displays all information properly.

Test Procedures: To be repeated in case of software updates for OpenWeatherMap API

ID 9 - Graphical Report Generation Page Test

Description: Test of the graphical report generation tool to make sure it loads and displays all graphs properly and dynamically changes according to changes in database.

Test Procedures: To be repeated in case of adding new types of maps to be generated or changes in database table structure

ID 10 - Admin View Page Test

Description: Test of the admin user page to make sure it loads and displays all information properly.

Test Procedures: To be repeated in case of changes in database table structure

III Inspection

1 Items to be Inspected

The following are the items to be inspected for the Water Quality & Flood Forecast Software Simulator

1) MVC .Net Frameworks setup.
2) Creating a Front-end User Input GUI for the Sensor Interface
3) Creating a Front-end User Input GUI for the Laboratory User Interface.
4) Leaflet Map API with front-end user input for latitude and longitude.
5) Manage Sensor and Laboratory Database from lab user input. on MySQL Workbench
6) Data Validation is done for Sensor and Laboratory Database and Laboratory User to give correct result for the Public User Interface and generate graphical Report.
7) Graphical Report from Laboratory Interface.

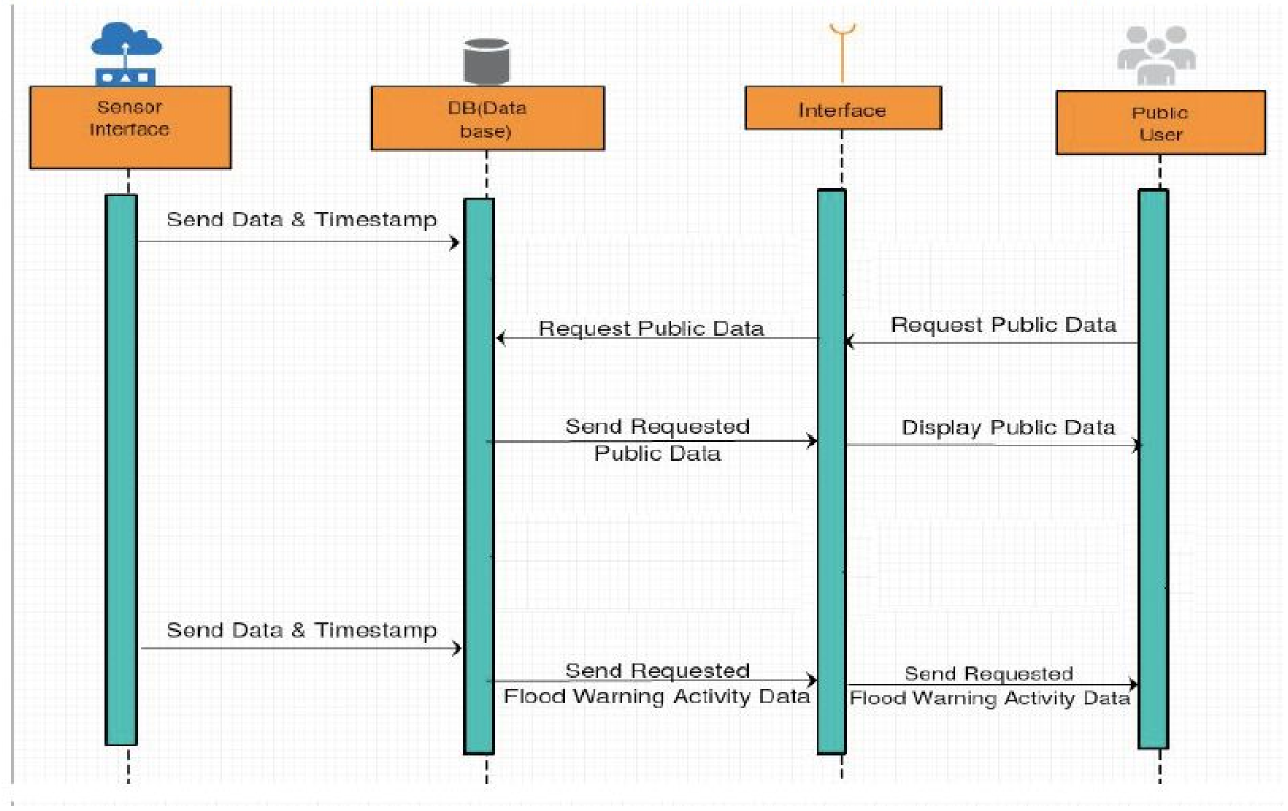
2 Inspection Procedures

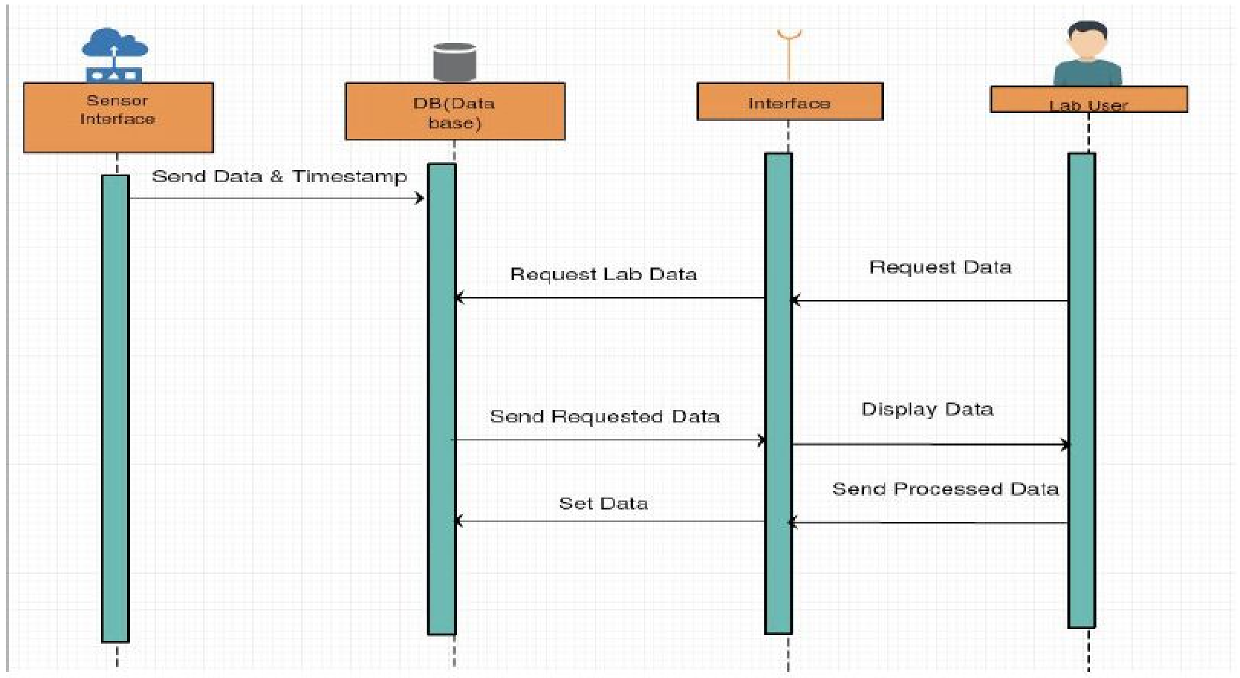
The following are the items to be inspected for the Water Quality & Flood Forecast Software Simulator

Meetings Held: Our groups meets every week to work on both Coding and Development Project. We have 10 Meeting Minutes plus couple in between the during the week to prepare for the Presentation and Coding Demo.

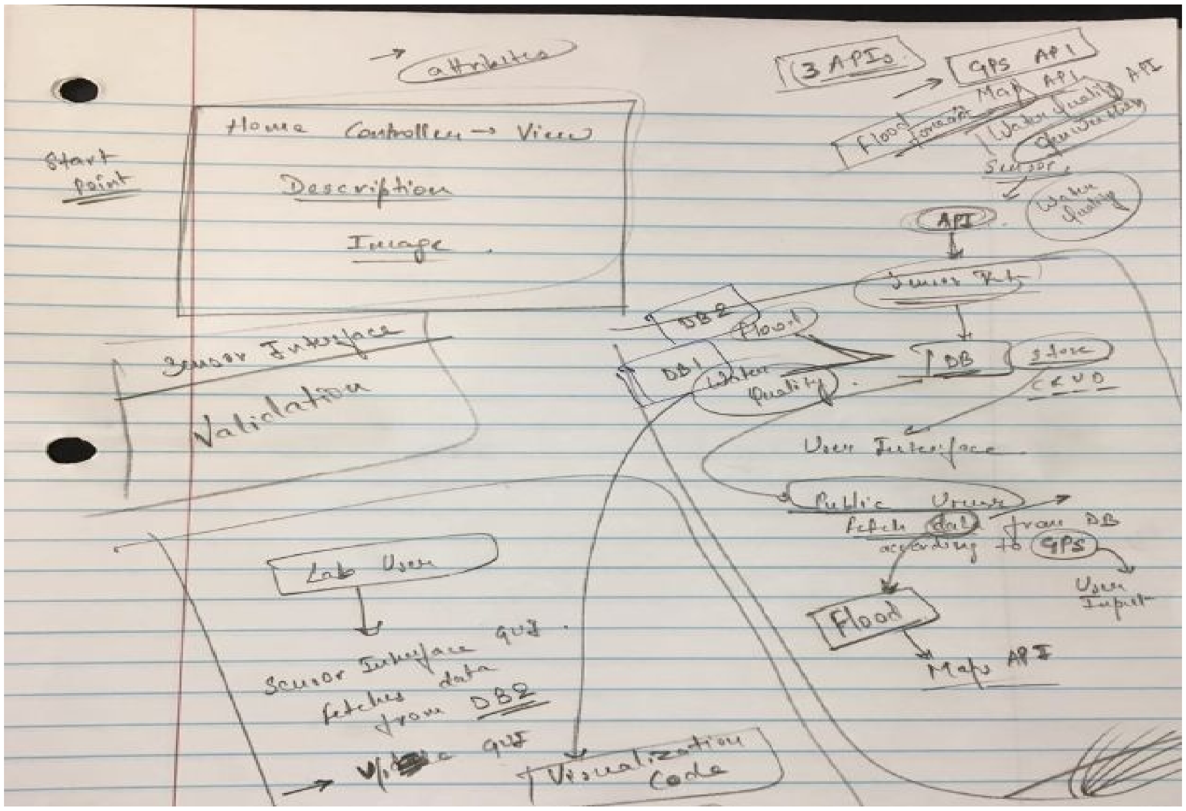
Water Quality and Flood Forecast Simulator Documents: Also, We made this DB diagram by hand and nor reference anywhere online.

Activity Diagram - interaction between sensors, database and users based on roles





Brainstorming Design Documents of Water Quality & Flood Forecast Simulator:



3 Inspection Results

1. .Net MVC Frameworks setup.

   Created By: Charvi Virani

   Inspected By: Everyone in group (Dhaval Patel, Kruneet Patel, and Michael Gallagher).

   Inspection Performed: October 5th, 2018.

   Result of Inspection: We inspected that the .NET MVC connected properly with Visual Studio also running in all the platforms Windows and Mac. Also, we made sure that local host server was connected properly and web page was rendering properly before working on the Database or the Front-end part of the project.

2. Creating a Front-End User Input GUI for Sensor Interface.

   Created By: Dhaval Patel

   Inspected By: Kruneet Patel and Charvi Virani

   Inspection Performed: October 6th, 2018.

   Result of Inspection: We inspected that all the name are needed for the Sensor Interface are in GUI such as Resistance, Voltage, Temperature, pH level, Dissolved Oxygen, and Radio Isotope Name with corrects units. Also, the code that was written is clean and commented properly so that other members in the group can edit it later. Also, interface and textbox aligned properly. The submit button for the user input is working properly so that it is saved in the database properly for data validation of the interface.

3. Creating a Front-End User Input GUI for Laboratory Interface.

   Created By: Dhaval Patel & Kruneet Patel

   Inspected By: Michael Gallagher, and Charvi Virani

   Inspection Performed: October 6th, 2018.

   Result of Inspection: We inspected that all the name are needed for the Laboratory Interface are in GUI such as Resistance, Voltage, Temperature, pH level, Dissolved Oxygen, Radio Isotope Name, Stable Isotope Name, Current Speed, and Water Level above the warning level with corrects units. Also, the code that was written is clean and commented properly so that other members in the group can edit it later. Also, interface and textbox aligned properly. The submit button for the user input is working properly so that it is saved in the database properly for data validation of the interface. The Generator Graphical Report and Back to List button inspected properly so that other group member could use it to generate the graphical report after the data validation.

4. Leaflet Map API with front-end user input for latitude and longitude.

   Created By: Charvi Virani

   Inspected By: Kruneet Patel and Michael Gallagher, Dhaval Patel

   Inspection Performed: October 30th, 2018.

   Result of Inspection: For some reason the Google API was making bills charges, so we decided to use the Leaflet Map API. We inspected are does the web page with the Leaflet

API have all the necessary fields needed to accomplish this task, and also connected to Open Weather API and Laboratory User Interface. We had Latitude, Longitude, and Area Name as the fields we made sure when user input the AreaName = "Chicago" with correct Latitude and Longitude that it is pointing to the correct location we was then used by the Open Weather API to show the weather forecast.

5. Manage Sensor and Laboratory Database from lab user input. on MySQL Workbench

   Created By: Michael Gallagher & Dhaval Patel

   Inspected By: Charvi Virani and Kruneet Patel

   Inspection Performed: October 7th, 2018.

   Result of Inspection: We made sure that once that database was created for Sensor and Laboratory Interface on MySQL Workbench, we made it is connected properly with the Visual Studio. All the necessary tables in the both databases are needed for this project. Making sure it save the data on the database once the user enter in the input clicks the save/submit button on the web page.

6. Data Validation is done for Sensor and Laboratory Database to give correct result for the Public User Interface.

   Created By: Kruneet Patel and Charvi Virani

   Inspected By: Dhaval Patel and Michael Gallagher

   Inspection Performed: October 10th, 2018.

   Result of Inspection: We made sure that once the database is connected using MySQL Workbench. Once we input the data in the sensor interface it is then validated by interface. We inspected that validation algorithm is working perfectly and handling all the correct data based on the criteria of our project. Once the data is validated it is then passed on the databases to generate report. Similar inspection procedure was for the data validation process of the laboratory interface.

7. Graphical Report from Laboratory Interface.

   Created By: Michael Gallagher

   Inspected By: Everyone(Dhaval Patel, Kruneet Patel, Charvi Virani)

   Inspection Performed: November 10th, 2018.

Result of Inspection: Once the data validation for the Laboratory interface was generated successfully. The graphical report is generated using Graphical API and querying the data which was saved under the Lab interface. After the data validation. We did inspection on the API is it generating correct graphical report and querying in the correct data from the laboratory database.

IV Recommendations and Conclusions

The items covered in the above have passed their testing and inspection process. The next step is successfully use the Open Weather API since we issue fetch data from the Open Weather API, also because of the time constraint we are unable to finish it on time. Once the Open Weather API is successfully implemented. The next step is using the Flood Forecast API in our project.

V Project Issues

1 Open Issues

Our software system will allow access from a large number of users. Designing database centers that are capable of providing such intensive user access has always been a challenge of software engineering. We want to make sure the data access is highly expandable and elastic, while at the same time it has to be tolerant to partial failures and provide consistent data.

The client company's manufacturing process depends on this system to provide continuous water quality monitoring. At the same time, we need to provide our public users access to the flood warnings. Such high availability will be hard to achieve.

Another issue is the cable communication between the client company database and the sensors. Some of the sensors will be physically far away from the data center, so we need to find durable cable to guarantee continuous communication.

We could use wireless communication methods to resolve the problem. Current technology can provide high fidelity wireless communication for over 10 miles. But doing this might introduce even more problems such as signal interference and possible eavesdropping.

2 Waiting Room

Currently graphic reports are only bar charts, a future release should have a whole range of data visualization to choose from.

Also, need to decide on and implement an alert system for public users that will be faster and more efficient for a greater number of users of the system

3 Ideas for Solutions

The graphical report functionality could be written in Javascript to allow for dynamic loading of visualizations.

Alerts could be sent via mobile phone numbers instead of the current system that uses emails.

4 Project Retrospective

Creating a workable interface was relatively easy and quick to get working. More time should be allotted for back-end development in the future: integrating APIs proved to be a challenge that caused problems for every part of the project.

VI Glossary

The same definition and terms that are used from the previous coding/ development report.

VII References / Bibliography

[1] Robertson and Robertson, Mastering the Requirements Process.

[2] A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.

[3] J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.

[4] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.

VIII Index