**Finding Potential Brownfields through the Nearest Neighbor Algorithm**

**1. Inception**

**1.1 Team Identification**

Team Name: Fern Force Five

Name: Email: GitHub Account:

Matt Judy judym1@tcnj.edu mjudy94

Kylie Gorman gormank2@tcnj.edu gormank2

Erin Wojciechowski wojciee1@tcnj.edu wojciee1

Thomas Borgia borgiat1@tcnj.edu tborgia10

Jake Orben orbenj1@tcnj.edu orbenjc

Nate Milkosky milkosn1@tcnj.edu nmilkosky

**1.2 Objectives and Overview**

*1.2.1 Problem Statement*

Habitat for Humanity requires information about the pollution history of plots of land in the Trenton area. If the organization receives a land donation, they would like an assessment of the quality of land. This pre analysis is to serve as a means to save on costs for soil contamination testing and time for planning, by directing Habitat for Humanity away from properties that will more than likely be too contaminated to pass the soil contamination test. If the soil is found to be contaminated, they are required by law to remediate the site before any further construction can take place. Testing and remediation has cost Habitat for Humanity a great deal of time and money in the past. In order to aid the foundation in future construction endeavors, this system will provide an estimation of the possibility of a specific site being polluted.

*1.2.2 Objective*

We intend to implement the existing code from last semester that dealt with the nearest neighbor algorithm using the C++ language and the interface of the web application will be programmed using HTML. To achieve our goal, we will incorporate information from the database class. We will use data that pertains to local contaminated areas in order to determine the likelihood of whether or not the property in question is contaminated. We also will be working with Habitat for Humanity and the journalism class to complete the project. Habitat for Humanity will be involved to refine the product and ensure that it works as requested and the journalism class will be assisting with the gathering of data for the system.

*Changes*

The user interface required HTML and PHP.

*1.2.3 Description of End Product*

The end product should be able to determine, using the nearest neighbor algorithm, the potential pollution of the site in question based on negative influences in the area. The final product will be integrated within the SOAP website and also potentially within a mobile application from another group. Our specific component will work with other modules created by other groups within the class to create a system for Habitat for Humanity to assist with all future construction projects in the area.

*1.2.4 Importance of the Module*

This module is especially significant because it will be designed to determine the likelihood of whether or not a specific plot of land is a brownfield. As a result, this module has the potential to save Habitat for Humanity countless hours and a great deal of funding. Remediation of soil is very time consuming, labor intensive, and expensive. If Habitat for Humanity knows ahead of time whether or not soil is contaminated, they will be able to determine if it is advantageous to begin construction on that piece of land. The potential to discover brownfields before construction begins is incredibly beneficial to the foundation.

*1.2.5 Similar Systems / Approaches*

An alternate system which uses a similar nearest neighbor algorithm is estimated land value. Real estate companies will select a property and determine its value based on factors in the vicinity. These factors can be positive, neutral or negative. The algorithm will be given data about these factors and will calculate accordingly and give an estimated value of the influences in the area which is then linked to the value of the land. Other approaches in which the nearest neighbor algorithm has been used are in land satellite imagery to determine characteristics of forests and determine their change over the years. This algorithm has even been used is to determine emotions using nodes on the face.

*1.2.6 Why/ What Aspect of the Module is Innovative*

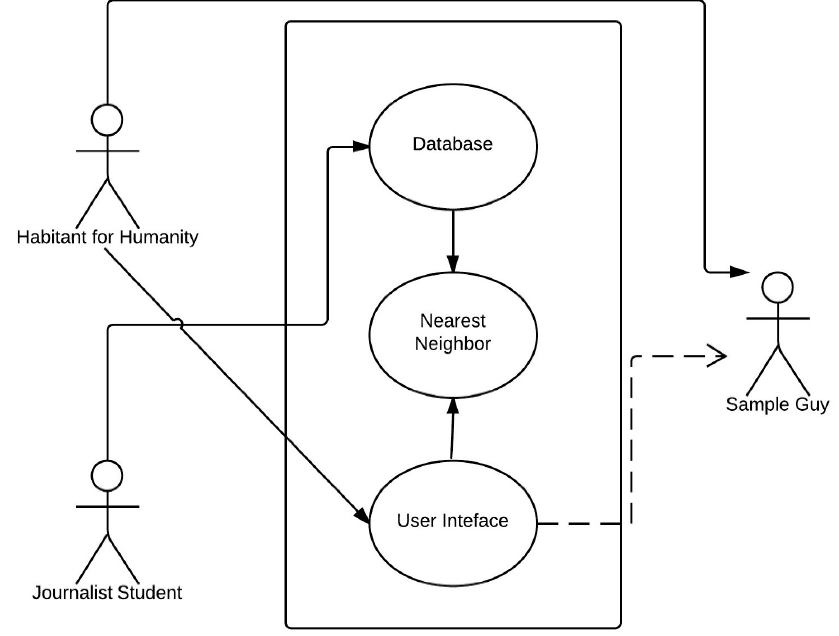
The module is innovative for Habitat for Humanity because they currently do not possess any systems to assist in their discovery of contaminated soil. The use of the nearest neighbor algorithm will allow us to take other contaminated sites and implement that information to help determine the likelihood of contamination in other nearby sites. This specific component of the overall system will provide insight into possible contaminated soil and has the ability to save the foundation a great deal of money.

*1.2.7 Technologies and Computer Science Concepts the Team will*

*Learn/ Implement:*

Our team will need to learn the basic concepts of the nearest neighbor algorithm to determine what data will need to be queried and how it is implemented in the algorithm. Our team will have to continue to learn and improve our existing C++ skills in order to understand the existing code written by the students last semester so we can make changes and integrate this specific component into the entire system. We will also have to learn the syntax for HTML to incorporate the module into the web based application

*1.2.8 A diagrammatic representation*



**2. Elaboration**

**2.1 Requirements Modeling and Analysis**

*2.1.1 Security*

There are a few security measures we can take in our project to make it more secure against outside attacks. First of all, we will attempt to manage our inputs properly. Since SQL injection is such a common type of attack on websites, we must protect the system from improper inputs. Input length, type, syntax and amounts can all be monitored in the system to help mitigate the risk of improper input attacks. If a user input does not conform the standard that are set for the aforementioned properties, then it will simply be rejected by the system. Another method of protection that we can utilize is output management. By separating control information from the actual data, we can help to ensure that attackers will not be able to manipulate internal commands such as queries and requests. We can also attempt to use libraries and frameworks that have been developed with security measures in mind. One final overarching method is to integrate security design into every step of the project. By taking security into account throughout the project life-­cycle, more weaknesses and vulnerabilities could be detected, allowing us to attempt to combat them sooner.

*2.1.2 Backup and Recovery*

Backup and recovery is required to protect against data loss and then reconstructs the system after any data loss. Therefore, it is crucial that data will be copied and saved in a protected file in case any data is lost. For this project we will be using GitHub as part of the Backup and Recovery implementation. This means that that the code will be stored in a repository. Putting the code in a repository allows each person on our team to pull the code, change and work on his or her assigned section, and push the code without fear of overwriting. GitHub also manages backups which allow us to retrieve a previous version of the code in case anything goes wrong while updating the system. Thus, several members can work on the code at the same time without the fear of damaging any previous code because an older, working version will be saved and available.

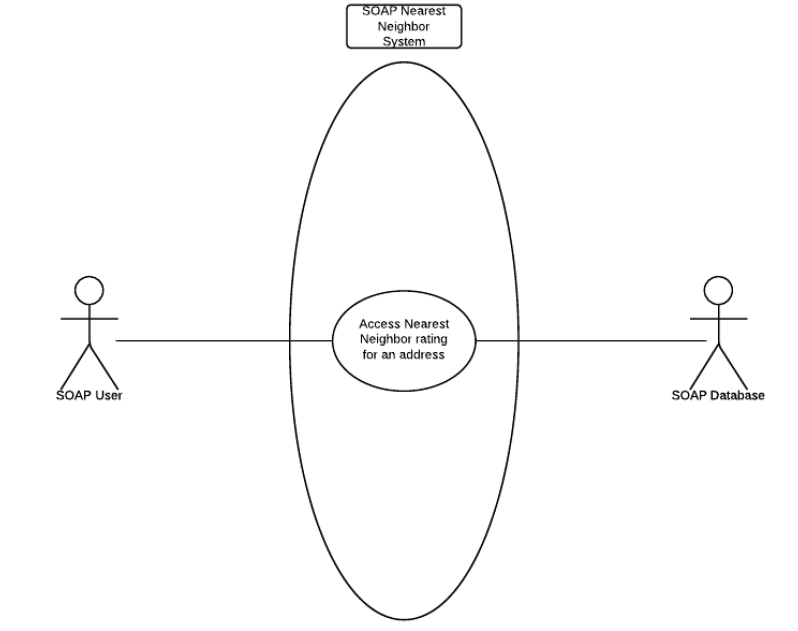
*2.1.3 Legal Issues*

There are several State and Federal Laws concerning the issue of development on contaminated land. Under New Jersey State Law, a brownfield is defined as “any former or current commercial or industrial site that is currently vacant or underutilized and on which there has been, or there is suspected to have been, a discharge of a contaminant." (N.J.S.A. 58:10B-­23.d). Under the Site Remediation Reform Act (SRRA) of 2009, responsible parties are obligated to remediate these contaminated sites in a timely manner. Once Habitat for Humanity purchases a plot of land, they become responsible for its state if they wish to build on it in the future. Therefore, our component of the system is very significant because it has the potential to discover these sites before they are considered for purchase. Other laws on the state level include Brownfield and Contaminated Site Remediation Act, Industrial Site Remediation Act, Underground Storage of Hazardous Substances Act, NJDEP Technical Requirements for Site Remediation, Brownfields Redevelopment, and Natural Resource Damages Act. On the federal level, the Small Business Liability Relief and Brownfields Revitalization Act must be considered. These laws restate that the responsible party is required to remediate these sites and state the required steps that must be taken. This process is very time consuming and labor intensive. It is crucial to determine whether or not a site is a brownfield before it is purchased. Further instructions are established by the New Jersey Department of Environmental Protection (DEP). These include the mandatory deadlines of remedial reports as well as the actual steps that must be taken to remediate the contaminated site.

*2.1.4 System Applications*

The primary application of this system will be to locate potential brownfields with low cost of cleanup to allow for organizations to clean and build on contaminated land in industrial areas. This will be accomplished using a "nearest-­neighbor" algorithm, which uses surrounding properties to determine the contamination in the desired property. The Nearest Neighbor Algorithm will work by first choosing a node, which will be used to indicate the desired location. We will then utilize the algorithm, data on that node and data on other nodes around it to help determine the likelihood of the desired location being a brownfield. Our system will allow Habitat for Humanity to determine the possibility of a land being contaminated before they purchase it, which will save the organization a great deal of time and money. We will integrate the code provided by the students last semester into the final web page.

*2.1.5 Use Case Diagram*



*2.1.6 Use Case Description*

Use Case: Access Nearest Neighbor rating for an address

Iteration: 1, last modification: October 15th

Primary Actor: SOAP User.

Goal in Context: To access the rating of an address based on the Nearest Neighbor algorithm.

Preconditions: SOAP is online and the Nearest Neighbor page is online; There is data to be manipulated.

Trigger: The SOAP user decides to access the rating of an address.

Scenario:

1. The SOAP user logs on to the SOAP website.

2. The SOAP user accesses the Nearest Neighbor page.

3. The SOAP user enters the address they would like to know the rating of into the address field.

4. The SOAP system returns the rating of that address using the Nearest Neighbor algorithm with data from the SOAP database.

5. The SOAP user views the rating.

Exceptions:

1. There is not enough data for the algorithm to calculate the rating of a location.

2. The SOAP user enters an invalid address. **Enter a valid address.**

Priority: High, the main function of the project.

When available: After Implementation phase.

Frequency of use: Moderate frequency.

Channel to actor: PC-­based browser and Internet connection.

Secondary actors: SOAP database.

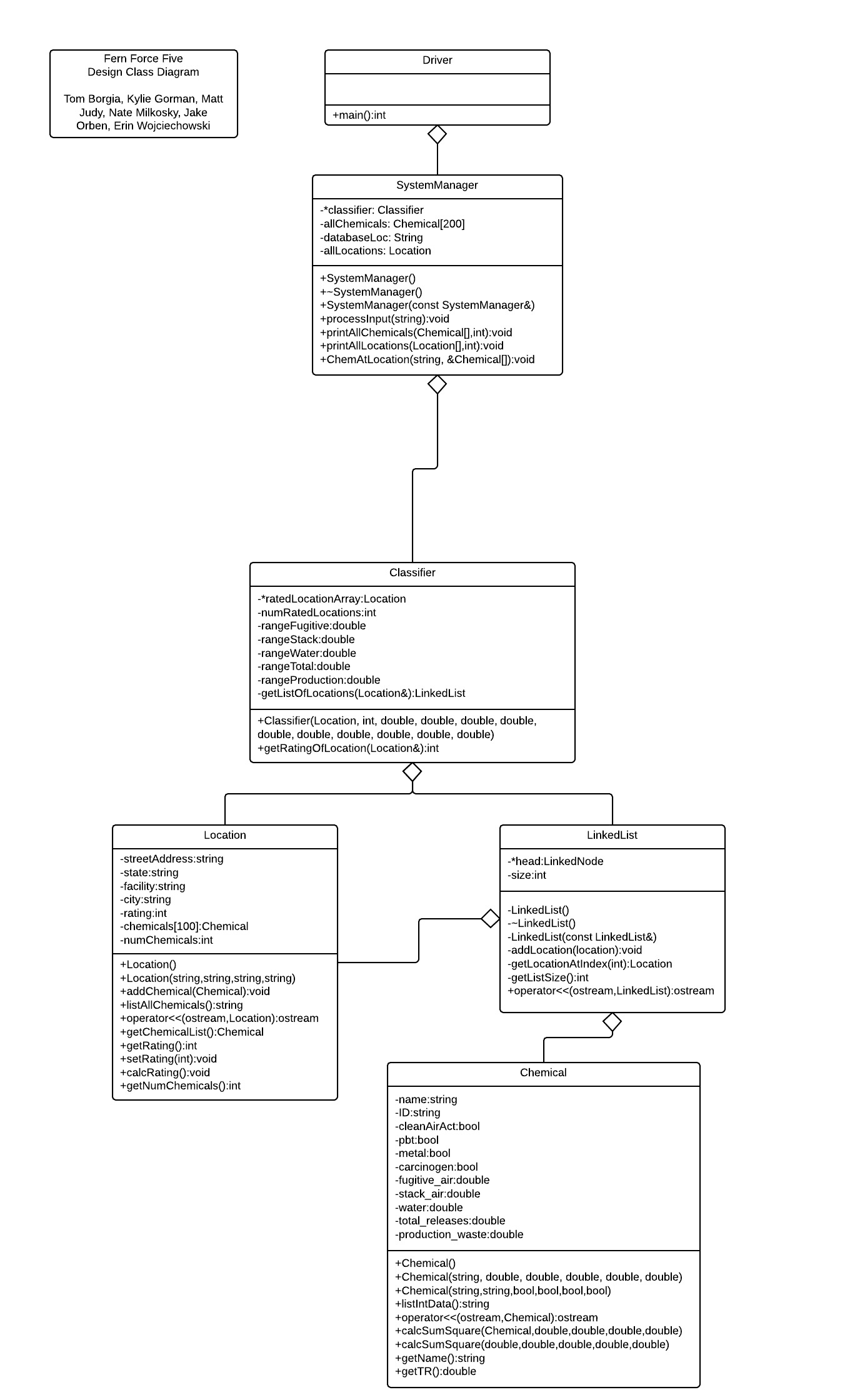
Channels to secondary actors: Data file is created by the SOAP database and read by the SOAP nearest neighbor system.

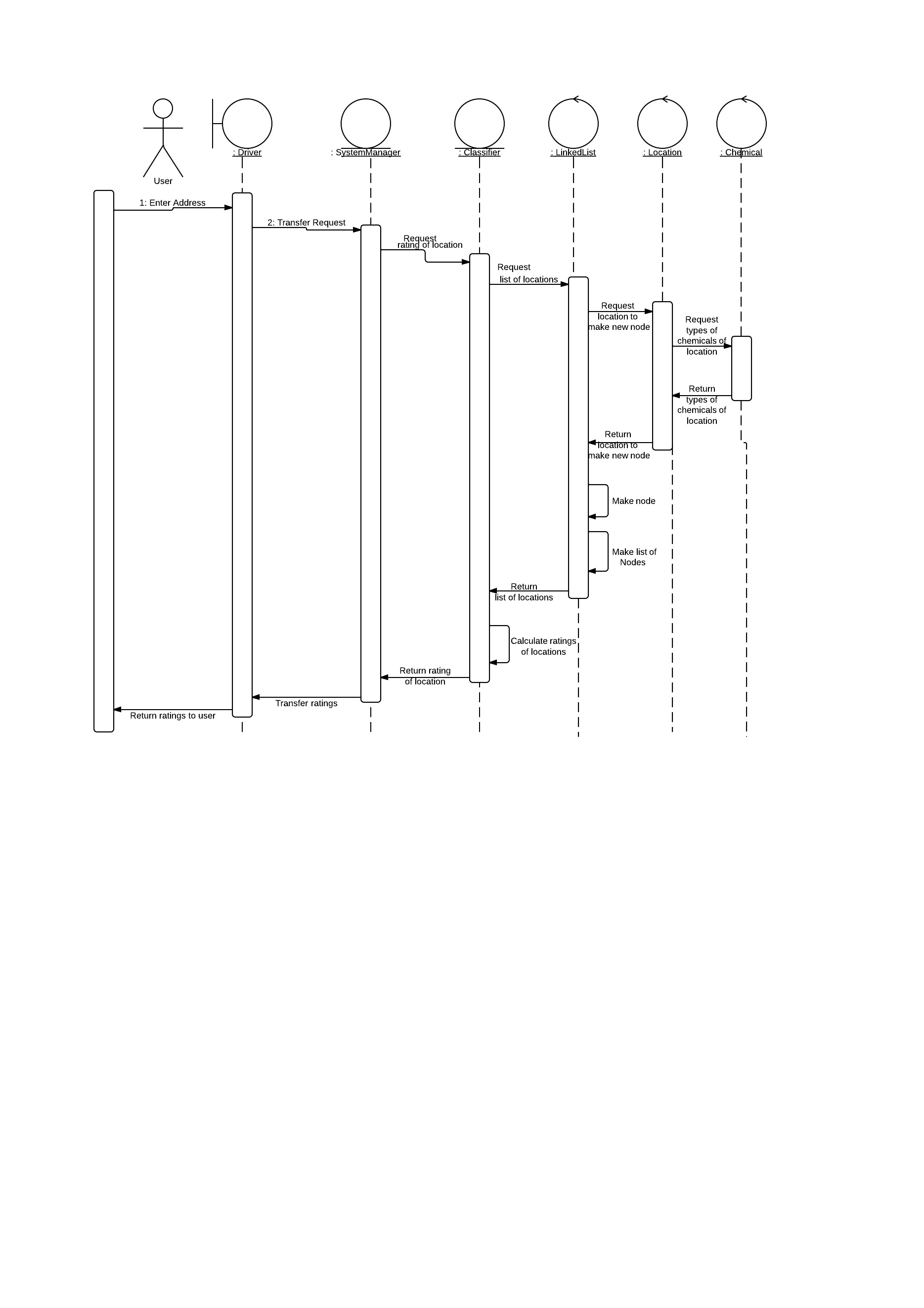
Open Issues: Will access be available to everyone?

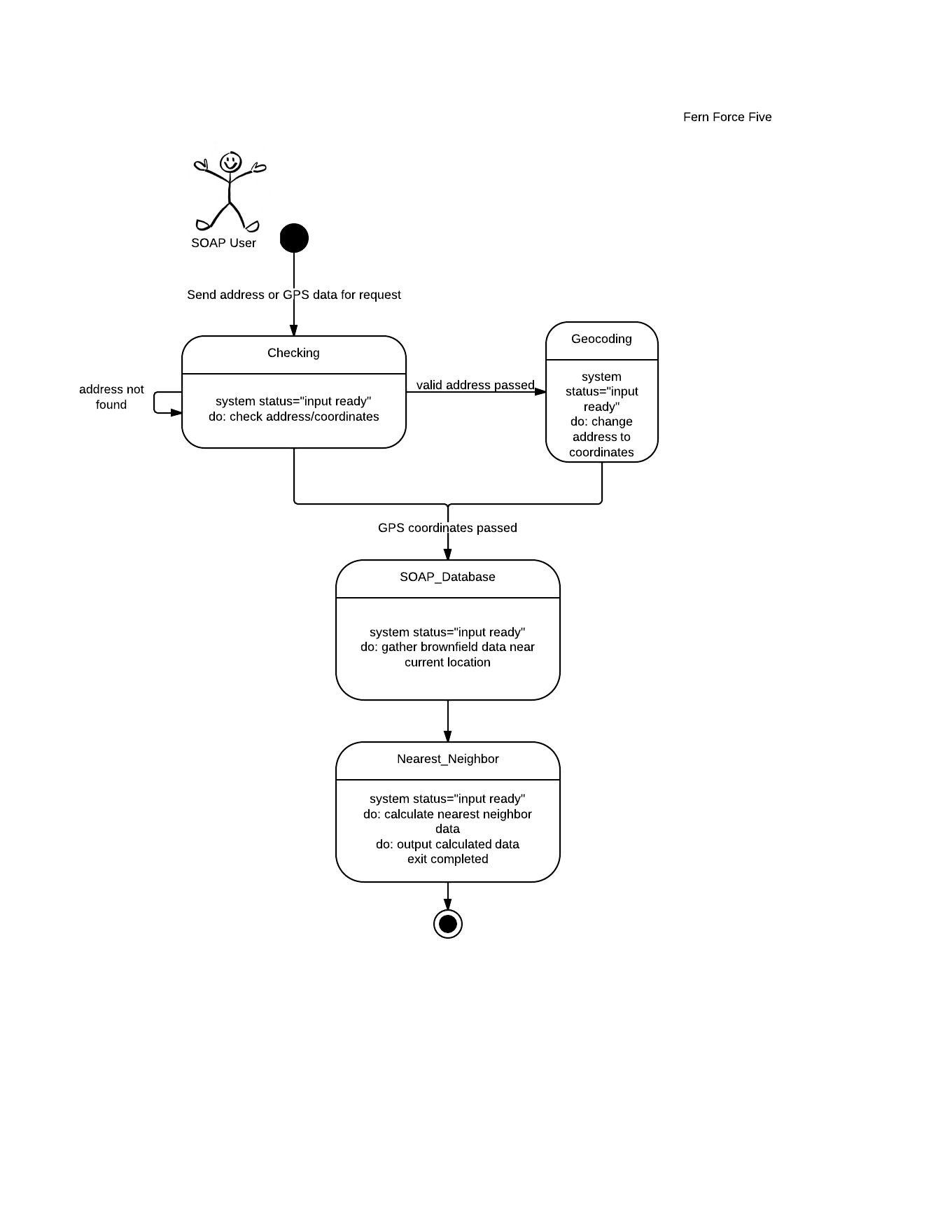
**Changes**

The system was modified to take in coordinates of the location in question. Therefore, the user should input coordinates and not an address.

**2.2 Design**

*2.2.1 Analysis Class Diagram*

*2.2.2 System Sequence Diagram*

*2.2.3 Basic State Diagram*

*2.2.4 User Interface Design*

The initial search page will contain a field labeled “Address”, and a button that will say search. This button will bring you to the results page, which will display the rating of the address entered, and any relevant information in a table.

Rule 1: Strive for Consistency

Keep same layout as soap system.

Rule 2: Frequent Shortcuts

Not applicable

Rule 3: Offer Informative Feedback

The results page will display the results of the search and calculations.

Rule 4: Design Dialogs to Yield Closure

The results page will tell the user that it has finished searching and display the results.

Rule 5: Simple Error Handling

Display an error if an improper address is input or if there is insufficient data. This will be described by the error message.

Rule 6: Easy Reversal of Actions

The results page will have a back button

Rule 7: Support Internal Locus of Control

The search will be responsive, and the information will be laid out to the user. All relevant links will be displayed.

Rule 8: Reduce Short-Term Memory Load

The results will be in a format that can easily copy and paste. The search result will have a unique URL, so they can be accessed later.