Fern Force 5

Dr. Pulimood

CSC 415 - Software Engineering

15 October 2013

Requirements and Analysis: Nearest Neighbor Algorithm

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 mitage the risk of improper input attacks. If a user input does not conform the 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 seperating control information from the actual data, we can help to ensure that attackers will not be able to manipulate internal commands such as queies 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 vulnerabilites could be detected, allowing us to attempt to combat them sooner.

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 allows 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.

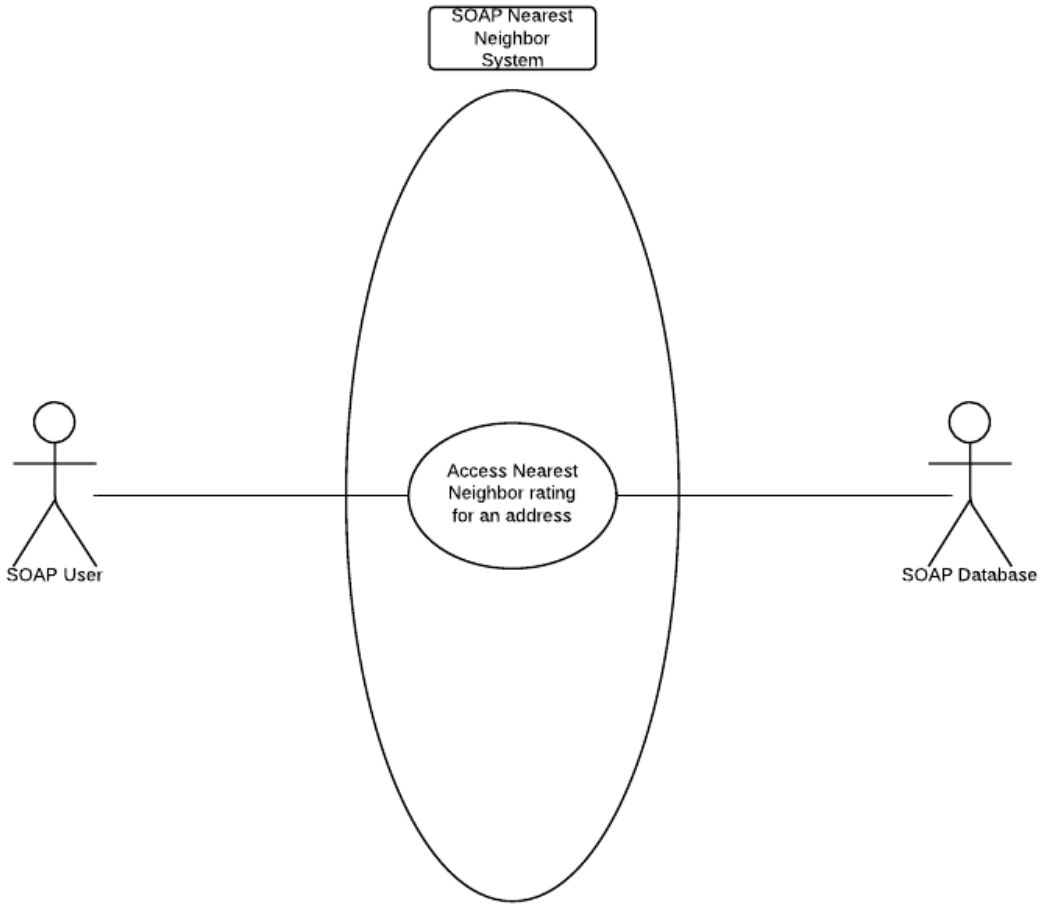
Legal Issues:

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 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 dsicover 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.

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.

Use Case Diagram:



Use Case Description:

Use Case: Access Nearest Neighbor rating for an address

Iteration: 1, last modification: October 15th by Nate Milkosky

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?

Analysis Class Diagram:

Fern Force Five
Analysis Class Diagram

Tom Borgia, Kylie Gorman, Matt Judy, Nate Milkosky, Jake Orben, Erin Wojciechowski