*{date}*

CIS 4911 Senior Project

Intelligence Inference Engine

Group Members

Jose Acosta

Lazaro Herrera

Mentor

Eric Kobrin

Instructor

Masoud Sadjadi

*{Copyright}*

Abstract

Table of Contents

1. Introduction

As the “Internet of Things” becomes larger and more connected, more and more of our devices become connected. Toasters, coffee makers, smartphones, items that were never connected are now as connected as your personal computer. As always, with greater connectivity comes greater risk at attack. Cyber-attacks on critical infrastructure can occur at any time and the new generation of tools to protect our domestic infrastructure cannot keep up with demands. What if a cyber-attack takes out power to the continental US, shuts down the traffic system of a city, disables the water processing plants of a state? Security researchers around the world are constantly monitoring the Internet for today and tomorrow’s threats but despite all of their knowledge, the information is not shared as easily as possible. What if you know that someone will be targeting critical infrastructure while a researcher in Cambridge finds the code that will make it happen? How do you make a cross-company, cross-country connection that could prevent a disaster?

As more devices come online and more of our infrastructure joins the Internet, some of the infrastructure that powers our lives becomes more and more reliant on the Internet and any vulnerabilities in them can cause issues. Power, water, traffic, financial, educational and many others are just some of the infrastructure that is constantly under attack by our enemies both foreign and domestic. A well-placed attack could cripple one or many of these necessary of society. Although the online communities of security researchers have many tools at their disposal, there is not one tool that allows connections to be made when you don’t know exactly what is being looked for. As an example, what if a researcher sees an attack that occurs every third Wednesday of every month that attacks only financial industries? They may not even be aware that a different researcher has seen the same attack originating from a single computer that directs others to attack, or that the attacks come from a specific sub-net. All of this data when properly connected in a single location can allow more flexibility when attempting a cyber defense. Knowing where the attack is coming from, and at what time can allow our more critical infrastructure to be more resilient to the future of cyber-attacks.

* 1. Problem Definition.

Currently, there exist many websites that catalog cyber-attacks, data breaches and vulnerabilities. Although all of these websites are critical to today’s security researchers, missing any of them can leave important data behind as well as not knowing the correct term to search for can cause some of the important information to be left behind. Consider the possibility that an attack may be filed as targeting “banks” yet the security researchers might be searching for an attack that is targeting “financial institutions”. Similar terms being used but a standard search engine does not take these into account.

* 1. Scope of system.
  2. Terminology - Definitions, acronyms, and abbreviations.

OSINT - Open-source intelligence

Cyber-attack - Any type of offensive maneuver employed by individuals or whole organizations that targets computer information systems, infrastructures, computer networks, and/or personal computer devices by various means of malicious acts usually originating from an anonymous source that either steals, alters, or destroys a specified target by hacking into a susceptible system {ref}

Triple store - {def} {ref}

* 1. Overview of document

This document will describe the current solution that is used to combat the problem and describes the limitations that the current system faces. This document also includes the project plan which details the breakdown of the work between the group members and what the responsibility of the group members are. Towards the end of the document, the use cases for the system will be described along with scenarios. The document will also contain all the various different UML diagrams that describe the system and how it functions and are used to make the use case, static, and dynamic models.

1. Current System

Currently, there exist many websites that catalog cyber-attacks, data breaches and vulnerabilities. Although all of these websites are critical to today’s security researchers, missing any of them can leave important data behind as well as not knowing the correct term to search for can cause some of the important information to be left behind. Consider the possibility that an attack may be filed as targeting “banks” yet the security researchers might be searching for an attack that is targeting “financial institutions”. Similar terms being used but a standard search engine does not take these into account.

1. Project Plan (**This deliverable only**)

Introduce the project plan section (one or two paragraphs)

* 1. Project organization – assignment of roles for this deliverable.
  2. Work breakdown – identification of milestones and deliverables (refer to project schedule in Appendix A and the diary in appendix B).

For our Trello board, we have decided to use color tagging to both identify the assignments of tasks and the completion status of tasks

Colors and Definitions for Priority

Blue - Card created by students

Purple - Card created by client

Colors and Definitions for Completions (only one active at a time)

Green - This task has a completion of 25% or more

Yellow - This task has a completion of 50% or more

Orange - This task has a completion of 75% or more

Red - This task has a completion of 100%

The following epics currently exist on our board.

Some of these are receiving checklists to be converted into stories.

- Setup Development Environment (Mulgara)

- Setup Development Environment (Web Server + Bootstrap)

- Develop Primary Feature (Data Entry Web Form)

- Test Primary Feature (Data Entry Web Form)

- Develop Primary Feature (Data Retrieval Web Form)

- Test Primary Feature (Data Retrieval Web Form)

- Develop Secondary Feature (RDF Web Crawler)

- Test Secondary Feature (RDF Web Crawler)

- Develop Secondary Feature (Custom Reusable Queries)

- Test Secondary Feature (Custom Reusable Queries)

- Develop Secondary Feature (Confidence Ranking)

* 1. Cost Estimate – cost to develop the software system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cost Matrix | Weeks Required | Jose - $30 hour / 12hr week | Lazaro - $30 hour / 12hr week | Total Cost |
| Data Entry Form Implementation | 2 | $720.00 | $720.00 | $12,672.00 |
| Data Entry Form Testing | 1 | $360.00 | $360.00 |  |
| Data Retrieval Form Implementation | 2 | $720.00 | $720.00 |  |
| Data Retrieval Form Testing | 1 | $360.00 | $360.00 |  |
| RDF Crawler Implementation | 3 | $1,080.00 | $1,080.00 |  |
| RDF Crawler Testing | 2 | $720.00 | $720.00 |  |
| Custom Queries Implementation | 1 | $360.00 | $360.00 |  |
| Custom Queries Testing | 0.6 | $216.00 | $216.00 |  |
| Confidence Ranking Implementation | 3 | $1,080.00 | $1,080.00 |  |
| Confidence Ranking Testing | 2 | $720.00 | $720.00 |  |

1. Proposed System Requirements

Introduce the chapter (one or two paragraphs)

* 1. Functional Requirements – describes high-level functionality

- The system shall allow users to submit data to be stored through some type of web form

- The system shall store the data in one of the existing semantic web triple- or quad-stores such as Mulgara or Jena

- The system shall allow the data to be query-able directly using Sparql or Datalog

- The system shall allow a user to set up predefined queries which are accessible by other users

- The system shall be able to gather data by itself through the use of web-crawlers

- The system shall be able to set a confidence interval for the data that is being collected from user inputted data

For each functional requirement state the associated non-functional requirements, if any, for *Usability, Reliability, Performance,* and *Supportability*.

* 1. Analysis of System Requirements

Analysis models – contains the complete functional specification and is mainly for the designers and programmers. This section describes the diagrams in the Appendices B - D and validates the models against the use cases.

Scenarios

Scenario {#}: Data Entry Success

Bob, a security analyst at company X, has gathered some data on attacks that have been happening to Company Y. Bob wants to share this data to see if further connections can be made. Bob goes to the web site that is being hosted at URL Z, and clicks on the button to submit data. When Bob gets to the web form, he will fill out the various field that are required. Once Bob has filled out the form, he will click on the submit button. The data that Bob entered will then be sent to the Mulgara triple store and Bob will receive a confirmation telling him that the data has been entered.

Scenario {#}: Data Entry Fail : Back End Down

Bob, a security analyst at company X, has gathered some data on attacks that have been happening to Company Y. Bob wants to share this data to see if further connections can be made. Bob goes to the web site that is being hosted at URL Z, and clicks on the button to submit data. When Bob gets to the web form, he will fill out the various field that are required. Once Bob has filled out the form, he will click on the submit button. The web form will attempt to connect to the back end to enter the information Bob has provided, but it is down. Bob will remain on the current web form and receive an alert informing him that the back end service it currently down and to please try to enter the information again later.

Scenario {#}: Data Entry Fail : Invalid Information

Bob, a security analyst at company X, has gathered some data on attacks that have been happening to Company Y. Bob wants to share this data to see if further connections can be made. Bob goes to the web site that is being hosted at URL Z, and clicks on the button to submit data. When Bob gets to the web form, he will fill out the various field that are required. Once Bob has filled out the form, he will click on the submit button. Once the submit button has been pressed the information on the web form is checked to verify it is valid. It is found that one or more of the fields that Bob has filled out contain invalid information. Bob will remain at the current form and an x will appear next to any field that is incorrectly filled out. Bob can now fix the invalid fields and retry or he can exit the page and abandon entering data.

Scenario {#}: Data Retrieval Success: Custom Query

Bob, a security analyst at company X, has noticed that recently there have been a lot of attacks in the field his client’s company is in. Bob wants too see if his client’s company is at risk of an attack. Bob goes to the web site that is being hosted at URL Z, and clicks on the button to query the data that has been collected. In the next form, Bob clicks on the button to enter his own custom sparql query. Bob will enter his query into the text box provided and then click on the submit button. The query that Bob has entered will then be sent to the Mulgara server and the results will sent back to be displayed on the web client Bob is on. Bob can now view the results of his query.

Scenario {#}: Data Retrieval Success: Generated Query

Bob, a security analyst at company X, has noticed that recently there have been a lot of attacks in the field his client’s company is in. Bob wants too see if his client’s company is at risk of an attack. Bob goes to the web site that is being hosted at URL Z, and clicks on the button to query the data that has been collected. In the next form, Bob will be presented with several fields to fill out in order for the page to automatically generate a query. Once Bob has filled out the required fields, He will then click on the submit button. The web form will generate a query and send it to the back end Mulgara service. Once the results are retrieved, Bob will be moved to a new page showing him the results of his query.

Scenario {#}: Data Retrieval Failure : Back End Is Down

Bob, a security analyst at company X, has noticed that recently there have been a lot of attacks in the field his client’s company is in. Bob wants too see if his client’s company is at risk of an attack. Bob goes to the web site that is being hosted at URL Z, and clicks on the button to query the data that has been collected. Bob will fill out the necessary information for the type of query he wants to do. Once Bob has filled out the information, he will press submit. Since the back end is currently down, Bob will remain at the current page and he will receive a message informing him that the back end is currently down and he should retry at a later time. Bob can now wait and retry with his query or abandon his query.

Scenario {#}: Data Retrieval Failure : No Results For Desired Query

Bob, a security analyst at company X, has noticed that recently there have been a lot of attacks in the field his client’s company is in. Bob wants too see if his client’s company is at risk of an attack. Bob goes to the web site that is being hosted at URL Z, and clicks on the button to query the data that has been collected. Bob will fill out the necessary information for the type of query he wants to do. Once Bob has filled out the information, he will press submit. The query is sent to the Mulgara back end but no results are generated from the query. Bob will remain at the current page and receive a message informing him that his query generated no results. Bob can now change his query and retry or he can abandon his query.

Scenario {#}: Custom Query Saving Success

Bob, a security analyst at company X, has created a query that he thinks will retrieved information other people will want. Bob wants to share this query. Bob goes to the website that is being hosted at URL Z, and click on the button to query the data that has been collected. In the next form, Bob will click in the button that says “Custom Query”. Once Bob is at the custom query form he will fill out the text box provided. After Bob has filled out the text box provided with the query that he wants to save, he will click on the check box that says “Save Query”. Enabling the “Save Query” button will add two additional text boxes to the current form. Bob will fill out the new text boxes with a title for the query and for tags (separated by commas) to describe the query. Once the two new text boxes are filled, Bob will press the submit button. If the query was successful, Bob will be taken to a new page containing the results of the query and a message saying the query was saved along with the id for said query for quick access.

Scenario {#}: Custom Query Saving Failure : Back End Services Down

Bob, a security analyst at company X, has created a query that he thinks will retrieved information other people will want. Bob wants to share this query. Bob goes to the website that is being hosted at URL Z, and click on the button to query the data that has been collected. In the next form, Bob will click in the button that says “Custom Query”. Once Bob is at the custom query form he will fill out the text box provided. After Bob has filled out the text box provided with the query that he wants to save, he will click on the check box that says “Save Query”. Enabling the “Save Query” button will add two additional text boxes to the current form. Bob will fill out the new text boxes with a title for the query and for tags (separated by commas) to describe the query. Once the two new text boxes are filled, Bob will press the submit button. Because the back end services are down the query cannot complete or save. Bob will receive a message informing him that the back end is currently down and the query could not be saved and to wait and retry later. Bob can now wait and retry at a later time or he can abandon saving his query.

Scenario {#}: Custom Query Saving Failure : Query already exists

Bob, a security analyst at company X, has created a query that he thinks will retrieved information other people will want. Bob wants to share this query. Bob goes to the website that is being hosted at URL Z, and click on the button to query the data that has been collected. In the next form, Bob will click in the button that says “Custom Query”. Once Bob is at the custom query form he will fill out the text box provided. After Bob has filled out the text box provided with the query that he wants to save, he will click on the check box that says “Save Query”. Enabling the “Save Query” button will add two additional text boxes to the current form. Bob will fill out the new text boxes with a title for the query and for tags (separated by commas) to describe the query. Once the two new text boxes are filled, Bob will press the submit button. The query is attempted to be saved but it is found the query already exists. Bob will be taken to a new page containing the results of his query but will also receive a message informing him that the query already exists and give him the query’s id.

Scenario {#}: Looking Up Custom Queries Success

Bob, a security analyst at company X, wants to look up some data, but he is not entirely sure on what he specifically wants to look for. Bob would go to the website that is at URL Z. Once Bob is at the website, he will click on the button to query the data that has been collected. At the query form, Bob will now click on the “Search” button which will take him to a new form. At this new form, Bob can fill out a text box in which he add tags to define what he wants to search for. Bob will click on the submit button when he has put in all the tags he wants to search for. The current saved queries will be compared to the tags Bob has input and relevant queries will be sent back to Bob’s client and be displayed for Bob to look at.

Scenario {#}: Looking Up Custom Queries Failure: Back End Services Down

Bob, a security analyst at company X, wants to look up some data, but he is not entirely sure on what he specifically wants to look for. Bob would go to the website that is at URL Z. Once Bob is at the website, he will click on the button to query the data that has been collected. At the query form, Bob will now click on the “Search” button which will take him to a new form. At this new form, Bob can fill out a text box in which he add tags to define what he wants to search for. Bob will click on the submit button when he has put in all the tags he wants to search for. Because the back end services are down the search cannot be completed. Bob will remain at his current page and receive a message informing him that the back end services are current down and to try at a later time. Bob can now wait and retry his search or he can abandon it.

Scenario {#}: Looking Up Custom Queries Failure: No Results

Bob, a security analyst at company X, wants to look up some data, but he is not entirely sure on what he specifically wants to look for. Bob would go to the website that is at URL Z. Once Bob is at the website, he will click on the button to query the data that has been collected. At the query form, Bob will now click on the “Search” button which will take him to a new form. At this new form, Bob can fill out a text box in which he add tags to define what he wants to search for. Bob will click on the submit button when he has put in all the tags he wants to search for. The back end services will attempt to find queries that match the parameters given by Bob but none are found. Bob will remain at his current page and receive a message informing him that his search parameters yielded no results. Bob can now modify his search and retry or he can abandon his search.

Scenario {#}: Confidence Interval

Bob, a security analyst at company X, wants to look up some data, Bob would go to the website that is at URL Z. Once Bob is at the website, he will click on the button to query the data that has been collected. At the query form, Bob will now click on the “Search” button which will take him to a new form. At this new form, Bob can fill out a text box in which he add tags to define what he wants to search for. Bob will click on the submit button when he has put in all the tags he wants to search for. The current saved queries will be compared to the tags Bob has input and relevant queries will be sent back to Bob’s client and be displayed for Bob to look at. Confidence Interval will give Bob the results sorted by the evidence that was collected as reference when the data was initially input.

Scenario {#}: RDF Data Explorer

Bob, a security analyst at company X, is curious about all the data that has been entered into the system but does not want to know any specific thing about the data. Bob would navigate to the RDF Data Explorer page. At the …

Scenario {#} RDF Data Explorer Failure: No Data

Scenario {#} RDF Data Explorer Failure : Back End Services Down

Name:

Store Data

Participants:

User

Events:

1. From the homepage of the website, the user must click on the “Contribute Now” button.
2. The user will be taken to a form which must be filled out.
3. On the form the user may enter his/her name or leave it blank.
4. The user must fill out the prefixes textbox with the prefixes they will be using for their properties.
5. The user must enter the subject of the data he is about to enter in the “Subject of Lead” textbox
6. The user must been enter at least one property of the subject by selecting a property from the properties drop down selecting the appropriate prefix in the prefix drop down and entering the value of the property in the textbox next to it.
7. If the user wants to submit more than one property they may click on the “+” button to add more property segments . If they want to remove any extra property input segment they can click on the “-” button.
8. The user clicks on the “Submit Lead” button.
9. The user will be taken to the Data Entry form and a message will appear on the top of the page notifying the user if their data was accepted or not.

Alternative Events:

* At step 8, if the user clicks the button and any of the data is invalid then an exception will occur that notifies the user there is invalid data.
* At step 1, or step 8, if the user clicks on the button and the system is down an exception will occur which will tell the user the system is currently down.

Entry Conditions:

User has data that they want to submit

User must be at homepage of the system

Exit Condition:

Data has been stored in the triple store and the user receives acknowledgment of it.

Exceptions:

Data that was input was invalid

The system is down.

Name:

Query Data

Participants:

User

Events:

1. From the home page, the user must click on the “Search Now!” button which will take them to the submit query form.
2. In the submit query form the user must enter the prefixes they will be using in the prefix textbox.
3. The user must then fill out one input segment by selecting at least one property from the drop down box along with an appropriate prefix from the prefix drop down and must enter a value in the text box.
4. If the user wants to submit more than one property they may click on the “+” button to add more property segments . If they want to remove any extra property input segment they can click on the “-” button.
5. The user must then click on the “Search Database” button.
6. The user will be taken to a page that has results of the query they submitted, if any exist.

Alternative Events:

* Instead of having the system generate a query for them, a user may submit their own query by selecting the “Custom Search” button, which will take them to the custom search form.
* At step 5, if the user clicks the button and any of the data is invalid then an exception will occur that notifies the user there is invalid data.
* At step 6, if no results are generated from the query that the user has used then an exception will occur which tells the user there are no results.
* At step 1, or step 5, if the system is down and the user goes through one of these steps then an exception will occur which will notify the user that the system is currently down

Entry Conditions:

User must be at homepage of the system

Exit Condition:

User has successfully entered a query and has gotten a list of results.

Exceptions:

User enters invalid data.

There are no results for the query specified by the user.

The system is down.

Name:

Use a custom query

Participants:

User

Events:

1. From the data retrieval page, the user clicks on the “Custom Search” button. This will take the user to the custom search page.
2. At the custom query page, the user will enter a query that they have generated themselves into the textbox provided.
3. The user will click on the “Execute Query” button
4. If the query is valid and produces results then the user will be taken to a results page.

Alternative Events:

* At step 2, If the user wants to use a query that another user has saved instead of one they generated themselves, then the user will enter the appropriate tags for that query in the tags textbox then click on “Find Queries”.
* At step 3, if the user has entered an invalid query then an exception will occur notifying the user that their query is malformed.
* At step 1, or 3, if the user follows these steps and the system is down an exception will occur notifying the user that the system is currently down.
* At step 4, if the user’s query provides no results then an exception will occur and the user will be notified that their query produced no results.

Entry Conditions:

The user must be at the data retrieval page

The user must have a query or the user must know the tags of a query that has been saved

Exit Condition:

User has entered a valid query and has gotten a list of results the query generated

Exceptions:

User entered an invalid query.

The query has returned no results.

The system is down.

Name:

Saving a custom query

Participants:

User

Events:

1. From the data retrieval page, the user clicks on the “Custom Search” button. This will take the user to the custom search page.
2. At the custom search page the user will enter the query that they want to store and enter the tags that will be used to identify that query in the tags textbox.
3. The user will click on the “Add Query” button.
4. The user will receive a message telling them whether the query they submitted was saved.

Alternative Events:

* At step 1, or step 3, if the system is down then an exception will occur and the user will be notified that the system is currently down.

Entry Conditions:

The user must be at the data retrieval page

The user must have a query and the tags he wants to save that query under

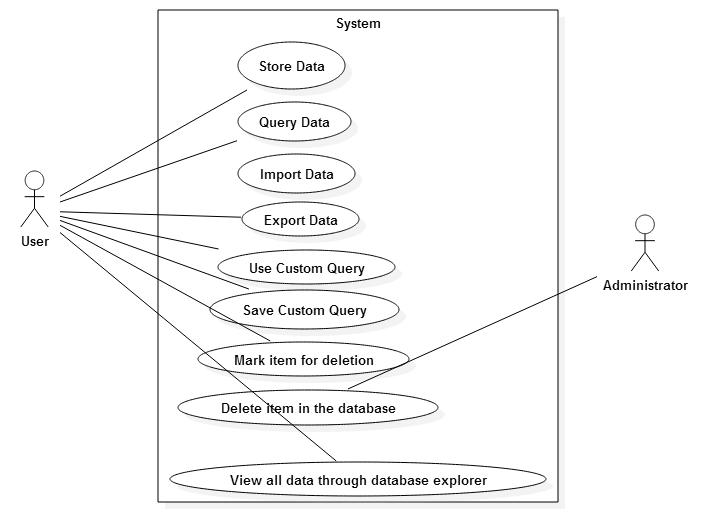
Exit Condition:

The query has been saved into the system.

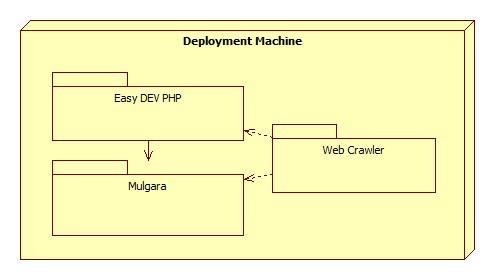
Exceptions:

The system is down.

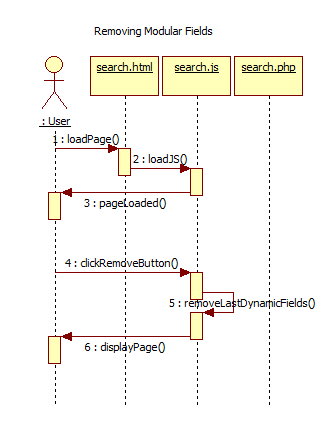
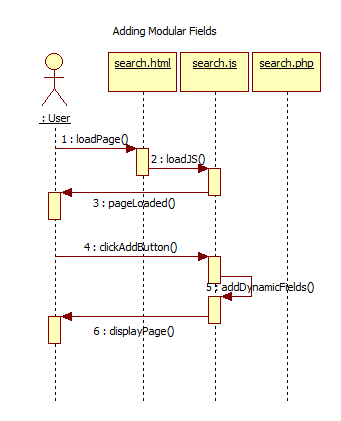
Use case model

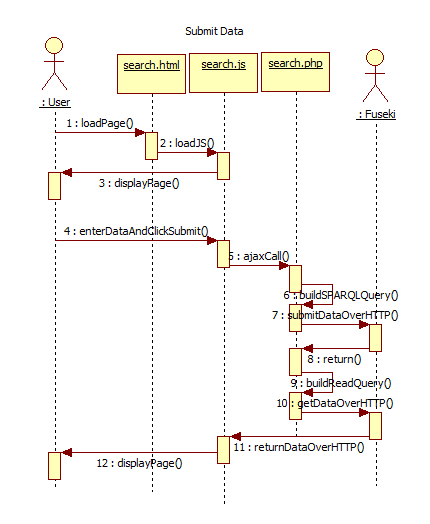
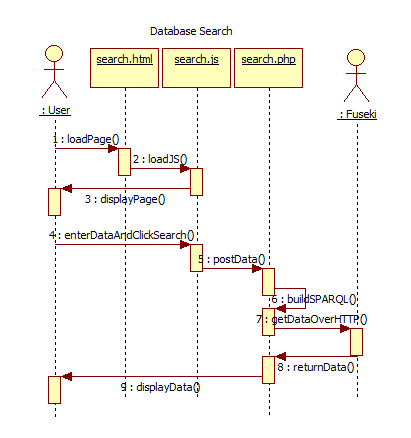


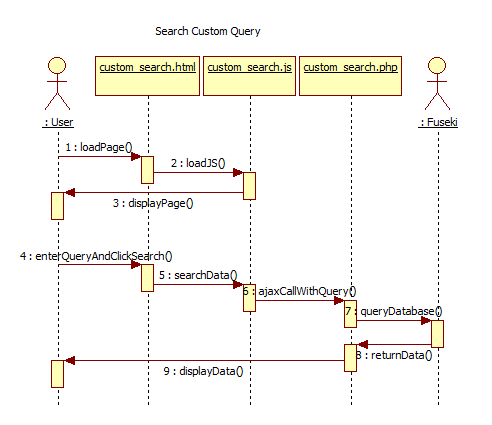
Static model e.g., object diagrams, class diagram

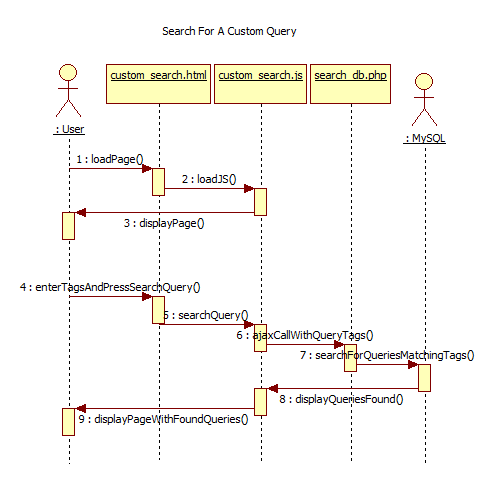


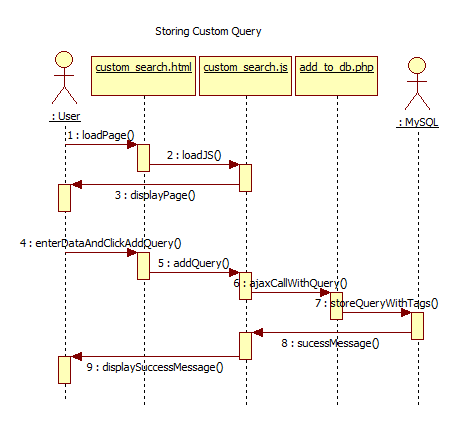
Dynamic model e.g., sequence diagrams or state machines

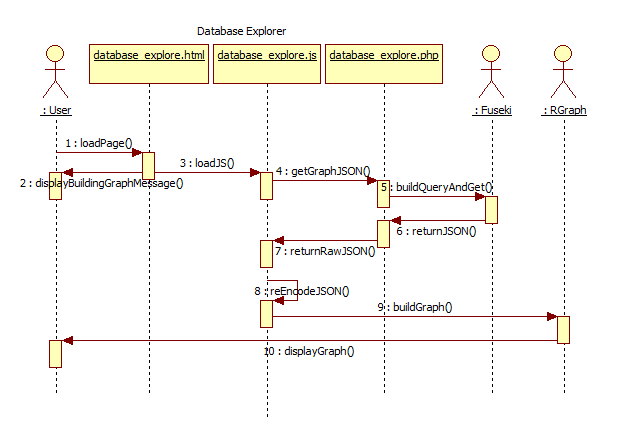




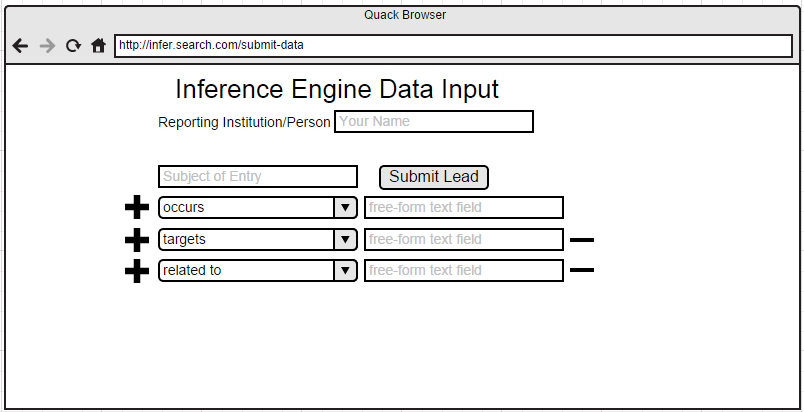


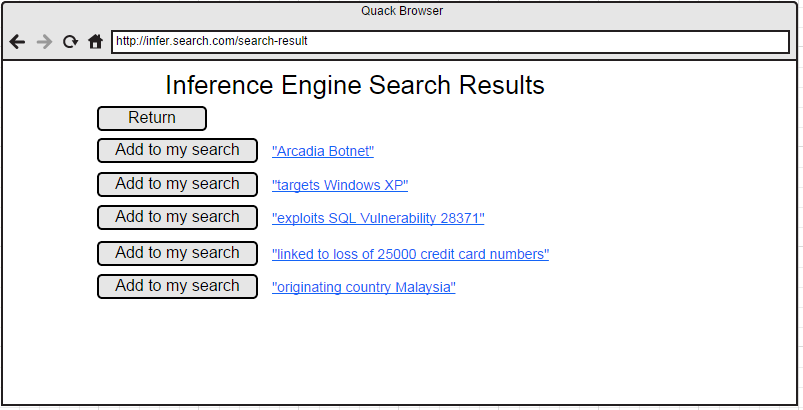
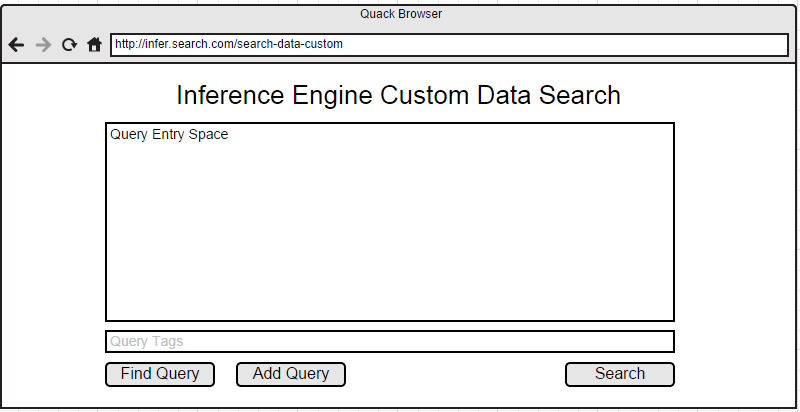
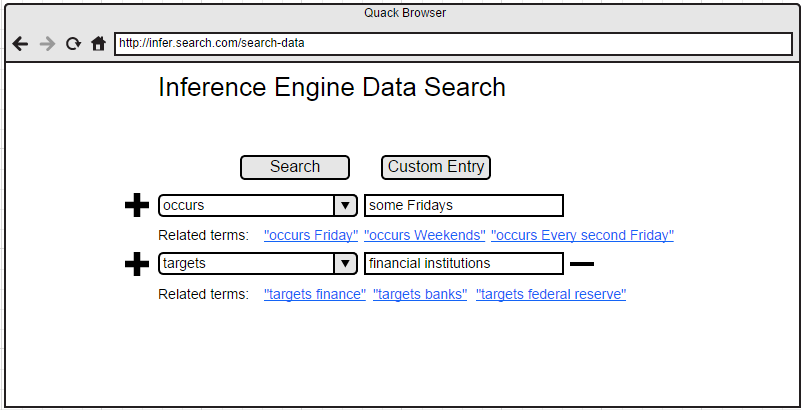


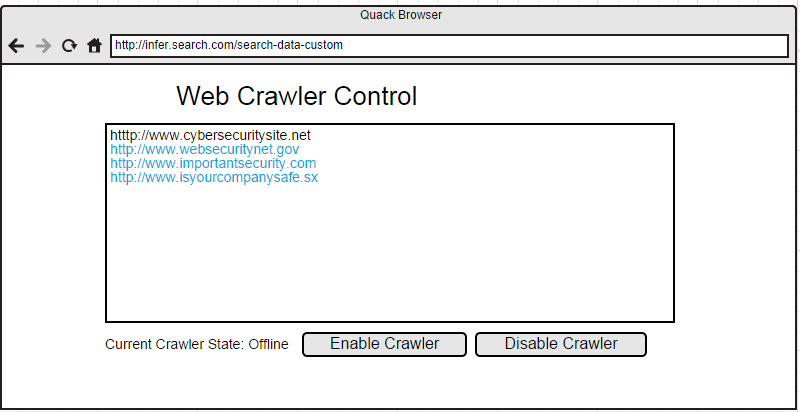




1. Glossary - define terms used in document, especially domain specific terms.
2. Appendix
   1. Appendix A - Complete use cases
   2. Appendix B - Use case diagram using UML
   3. Appendix C - Static UML diagram
   4. Appendix D - Dynamic UML diagrams
   5. Appendix E - User Interface designs.







* 1. Appendix F - Diary of meeting and tasks.

1. References

***Please email me the UML diagram in one file before the presentation.***

