

Mitigation Repository

Design & Architecture Document

Prepared by David Glennan

Team: River Otters

Date: 5/03/20

Version: 3.0

Table of Contents

1.	Purpose	4
2.	Scope	4
	2.1 Assumptions, and Limitations.....	4
3.	Human Interface Design.....	5
	3.1 Overview of Interface Design	5
	3.2 Screen Images	6
	3.2.1 Home Page.....	6
	3.2.2 Search Page	7
	3.2.3 Create Page	8
	3.2.4 Forking.....	9
	3.2.5 Edit Mitigation	10
	3.2.6 Delete Mitigation	11
	3.2.7 Login/ Create User Page	11
4.	Use Cases & Scenarios	12
	4.1 Use Case Search	12
	4.2 Use Case Add	13
	4.3 Use Case Manage.....	14
5.	Architecture Overview	15
	5.1 Logical View	15
	5.1.1 Package Diagram	15
	5.2 Development View	16
	5.2.1 Component Diagram Search	16
	5.2.2 Component Diagram Add	17
	5.3 Physical View	18
	5.3.1 Deployment Diagram.....	18
6.	Design Overview	19
	6.1 Search Functionality.....	19
	6.1.1 Analysis Class Diagram Search	19
	6.1.2 Activity Diagram Search	20
	6.1.3 Communication Diagram Search.....	21
	6.1.4 Sequence Diagram Select Search	22
	6.1.5 Sequence Diagram Display Detailed	23

6.2 Add Functionality	24
6.2.1 Analysis Class Diagram Add	24
6.2.2 Communication Diagram Add	25
6.2.3 Sequence Diagram Add General	26
6.2.4 Sequence Diagram Forking	28
6.3 Manage Functionality	29
6.3.1 Analysis Class Diagram Manage	29
6.3.2 Activity Diagram Manage	30
6.3.3 Communication Diagram Manage	31
6.3.4 Sequence Diagram Manage	32
6.4 Data Design	34
6.4.1 ER (Entity Relationship) Model	34
6.4.2 Physical Model	35
7. Tools Used	37
7.1 Tools, and purpose	37
8. Glossary of Terms	38
9. Revision History	39

1. Purpose

This document will describe the architecture and design decisions that are going to be implemented in the Mitigation Repository. The Mitigation Repository is based on the user requirements specified in the User Requirements Documents. This document will make use of several diagrams to depict the aspects that make up the Mitigation Repository.

2. Scope

This design will be accomplished in an Agile development and will be updated using results from future sprints to complete the design of the Mitigation Repository. Due to the Agile environment several of the design decisions made for the components of the Mitigation Repository are subject to be updated and or changed.

2.1 Assumptions, and Limitations

As stated above this document represents the design based off the Requirements Documents. As such only requirements stated in the Requirement Document are represented as features in this document. Due to this being a student project some software used was chosen due to the software being a freely available product. An example of these limitations include, the use of MySQL rather than Oracle or other paid products that offer additional features.

3. Human Interface Design

3.1 Overview of Interface Design

The Mitigation Repository will allow the user to perform searches, add, edit, and delete mitigations in the Mitigation Repository. This includes all information related to a mitigation including the author, system, and the security controls. The system allows the user several methods to perform searches to display mitigations to the risks the user is trying to solve. The user can search for mitigations by entering the security controls consisting of the category and type of the risk. If the user does not know the security controls the user can search by the title of the mitigation. Also, the user can search by most recent, this brings up the twenty-five most recent mitigations.


The user then is brought to a search results page that shows the mitigations corresponding to their search. The search results page is split in two sides to help the user read the information. The left side of the search results page shows the brief view of the mitigation's that correspond to the user's search. This includes the title and part of the description. When a user clicks on the mitigation it brings the mitigation up on the right hand of the screen in a detailed view. This view shows the title, description, system name, version, and the security controls of the mitigation the user clicked on.

The system also allows the user to create a new mitigation from the home page where a user can click on the create new mitigation button, which takes the user to the create new mitigation page. In this page the user can enter the required information to create the mitigation. The required information to form a mitigation is the title and description of the mitigation, the security controls that the mitigation corresponds too, and the system name and version and author of the mitigation.

3.2 Screen Images

3.2.1 Home Page

The screenshot below shows the Home page of the Mitigation Repository. This is the starting point for all users. From this point the user can search, create mitigations, and login to a user account.

Mitigation Repository  Login Guest Access

Search Mitigation

Search Mitigation By Title

Search By Title

Category:

Choose Category ▼

Type:

Choose Type ▼

Search By Category and Type

25 Most Recent Mitigations

Create new Mitigation

Create new Mitigation

6

3.2.2 Search Page

The screenshot below shows the search results page after searching for a mitigation. After selecting a mitigation on the left hand side by clicking on the mitigation it brings up the right hand of the screen. If the user is a general user they can fork the mitigation, if they are a admin they can edit or delete the mitigation.

Search Results

Refine Search

Choose Category ▼

Choose Type ▼

Choose Operating System ▼

Edge Test

Mitigation ID:156

Operating System: Windows Version: 10

Author: Kristen Stansfield

Administrative

Detective

A User Created Mitigation

Mitigation ID:155

Operating System: Windows 10 Version: 45664

Author: Theresa Morris

Administrative

Preventative

fork

Mitigation ID:152

Operating System: edit Version: edit

Author: fork fork

Physical

Preventative

editing

Mitigation ID:146

Operating System: edit Version: edit

Author: Edge Test Edge Test

Physical

Detective

Edge Test

Mitigation ID: 156

Link to this mitigation: [156](#)

Children: [156](#)

Fork Mitigation

Author: Kristen Stansfield

Created on:2020-05-03 18:25:21

Modified on: 2020-05-03 18:25:21

Operating System: Windows Version: 10

Testing to make sure everything works on Edge and just making this mitigation for fun

Administrative

Detective

3.2.3 Create Page

The diagram below shows the create new mitigation page. On this page a user can add an original mitigation not already represented in the database.

Mitigation Repository

Login

Logged in as Dave

Create a Mitigation

Author First Name

First Name

Author Last Name

Last Name

Mitigation Title

Mitigation Title

Operating System

Operating System

Operating System Version

Version

Enter Mitigation Description

Mitigation Description

3.2.4 Forking

The screenshot below shows the other method to add new mitigations, forking. As seen in the screenshot below the page pre-populates the forms with the old mitigation information.

ry

Login

Logged in as admin

Fork Mitigation

Author First Name

First Name

Author Last Name

Last Name

Mitigation Title

Internet connected computer

Operating System

Windows 10

Operating System Version

9090

Enter Mitigation Description

to properly mitigate this, we need to properly remove the system from the internet, as it is not


Category: Choose Category

Type: Choose Type

Fork Mitigation

3.2.5 Edit Mitigation

The screenshot below shows the edit mitigation page. From here a user can edit the any of the fields that need to be updated or corrected.

Mitigation Repository  Login Logged in as admin

Edit Mitigation

Author First Name

First Name

Author Last Name

Last Name

Mitigation Title

Admin permissons fixed

Operating System

Linux

Operating System Version

20.18

Enter Mitigation Description

Admin for new system...

Category: Choose Category ▼

Type: Choose Type ▼

Edit Mitigation

3.2.6 Delete Mitigation

The screenshot below show the delete mitigation page. From here the user can delete a mitigation by clicking on the “yes” button. Notably the page shows if the mitigation has any children mitigations the user knows before deleting.

Mitigation Repository Login Logged in as admin

Delete Mitigation

Are you sure you would like to delete this mitigation?

Trying to fix my hardrive

Mitigation ID: 118

Link to this mitigation: [118](#)

Children: [118](#)

Author: Andrew Wozny
Created on: 2020-04-12 03:13:56
Modified on: 2020-04-12 03:13:56
Operating System: Windows 10 Version: 19.1
Hardrive is corpted

Technical Detective

3.2.7 Login/ Create User Page

The screenshot below shows the login page and create new user page. The user can log in on the left hand to access features that correspond to their roles. On the right hand side a root/super user can create new users for the mitigation repository.

Mitigation Repository Login Guest Access

Login

Username:

Password:

Create new User

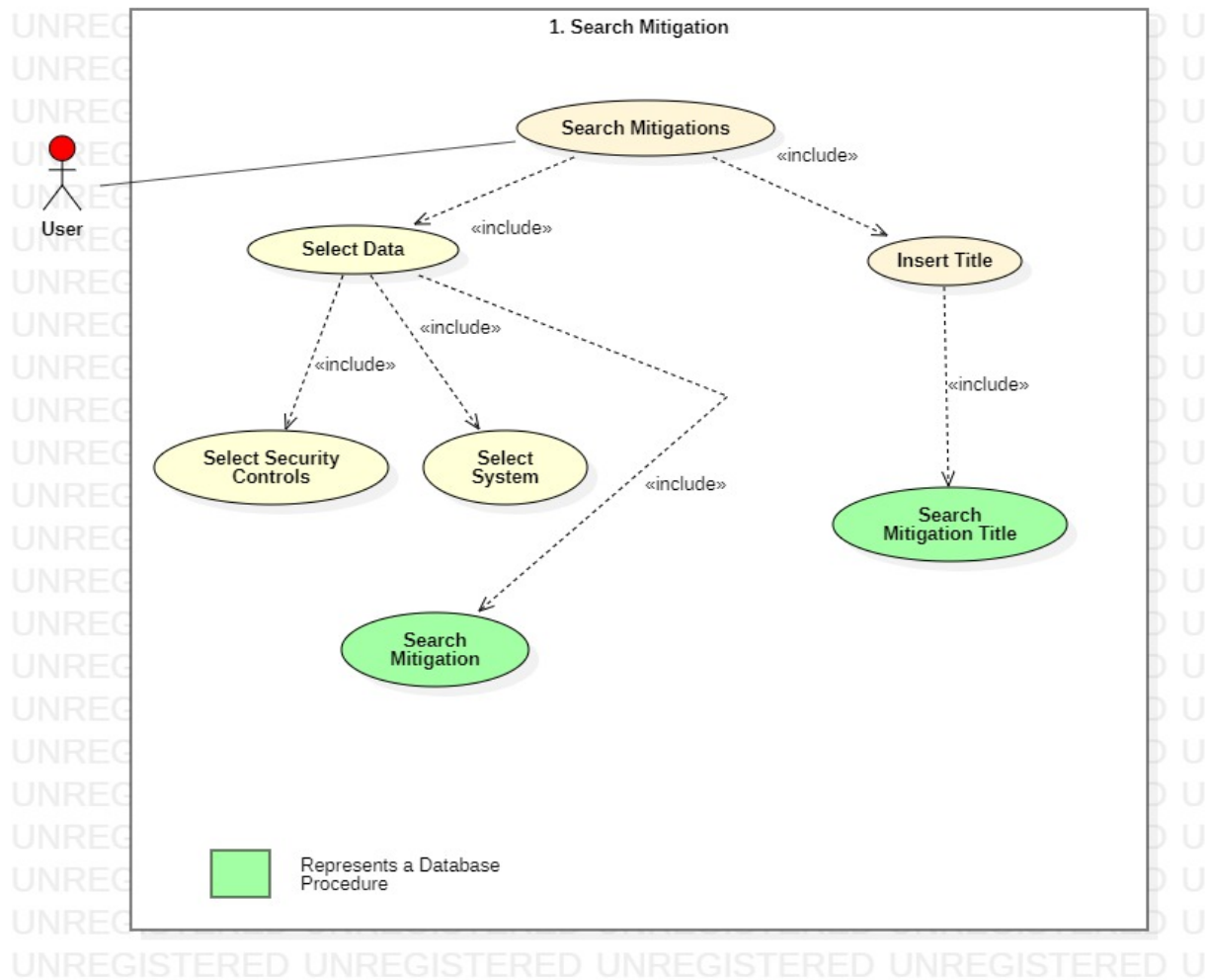
Username:

Password:

4. Use Cases & Scenarios

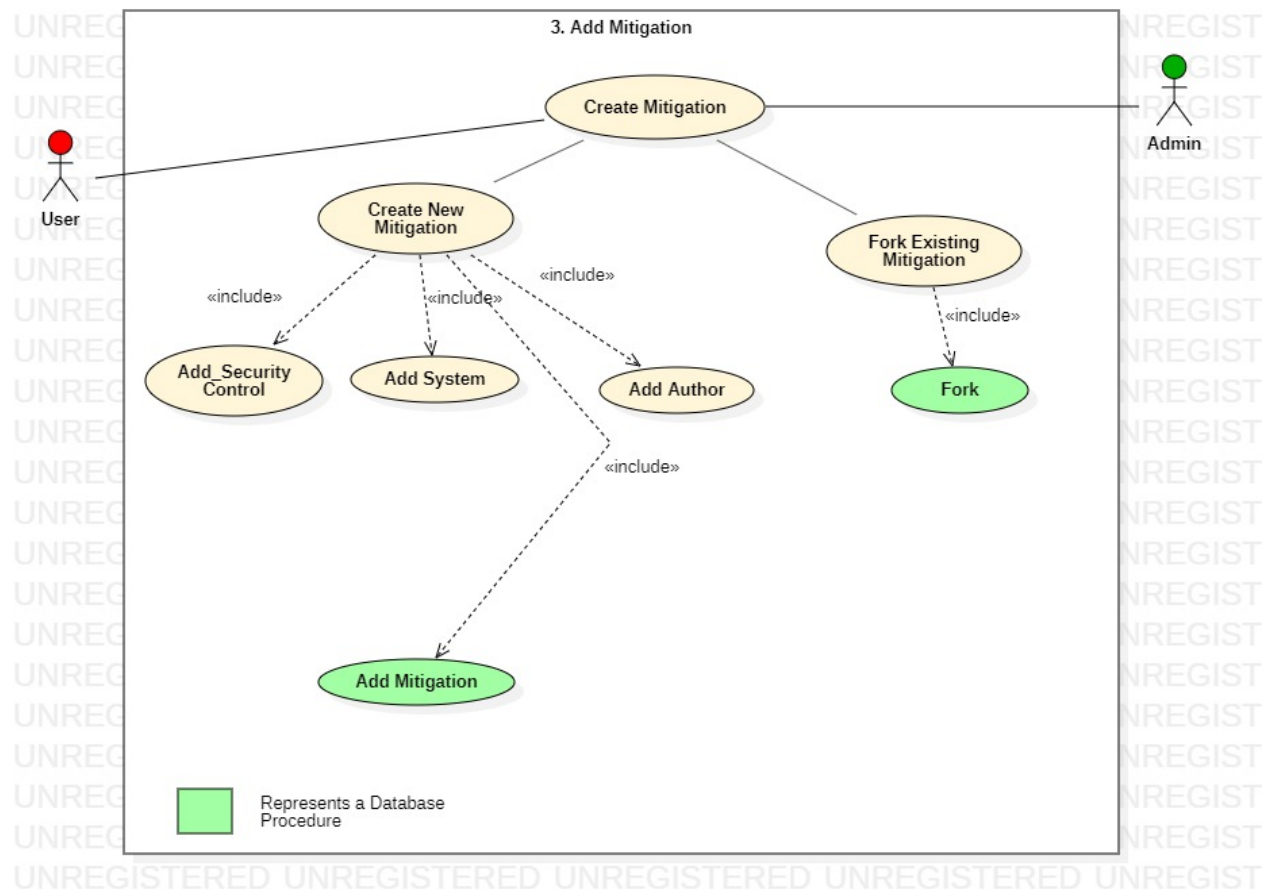
4.1 Use Case Search

The use case diagram below shows in the Mitigation Repository how a user would use the system to search for a mitigation. As can be seen below the user has two options to search for a mitigation from the database repository. Starting on the left the user can select data, which includes both the security controls and the system. Then this information can get passed to a database procedure that will return the results of the search. On the right we can see a user could also search by entering a title, which calls a different procedure that returns the mitigation results.



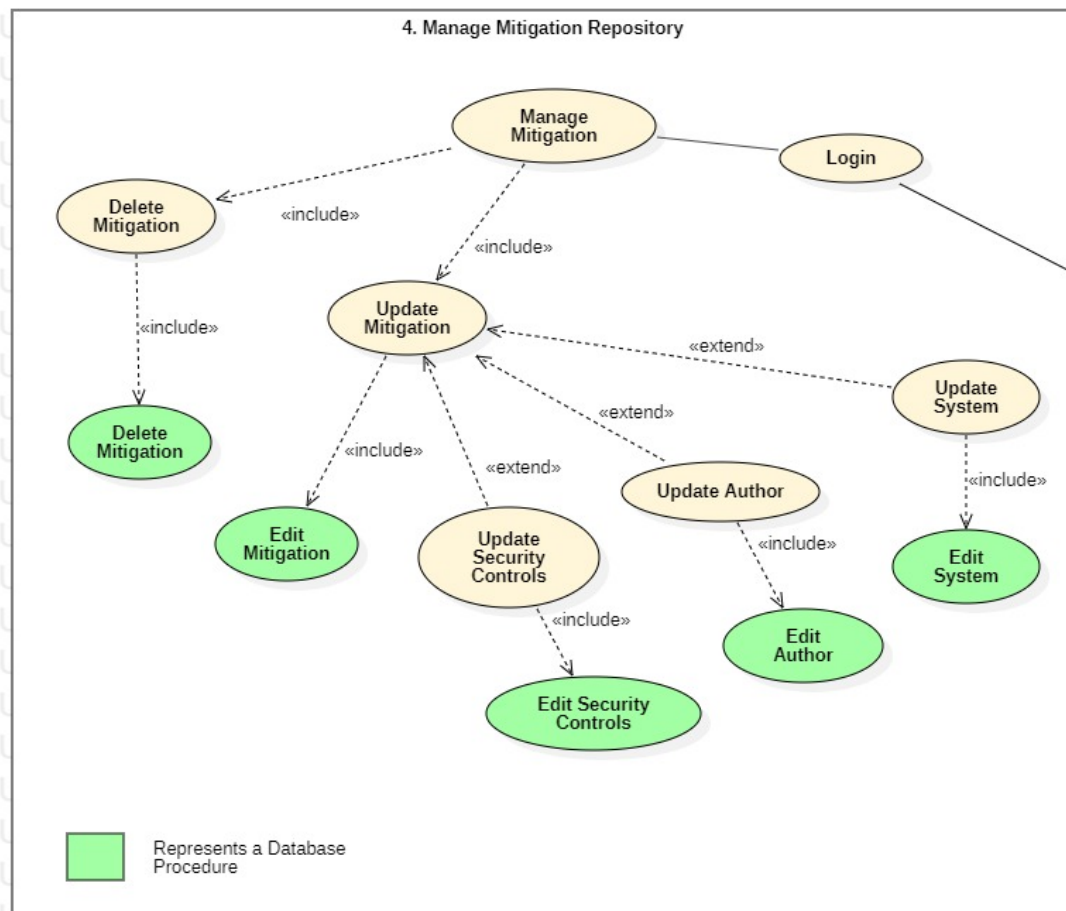
4.2 Use Case Add

This use case below shows in the Mitigation Repository the steps a user or admin would take to add a new mitigation to the repository. The user does not have to be an admin to add a new mitigation, however the steps a user must take will be explained in greater detail in the architecture views. As explained in the User Requirements Documents this can be done from the web interface and does not require a user to have any knowledge of the database level operations to enter data. The diagram shows there are two types of mitigations to be added to the repository. As seen below a user can enter a unique mitigation that includes security controls, system, and author to form a mitigation. Or shown on the right hand a user could fork off an existing mitigation. Forking the mitigation copies the attributes of parent mitigation while allowing the user to change the child mitigation description and author. This is done as this is not an exact clone of the parent mitigation.



4.3 Use Case Manage

This use case below shows in the Mitigation Repository the steps an admin would take to manage mitigations into the repository. As explained in the User Requirements Documents, these actions take place at the web interface layer and do not require the user to manage at the database level. The diagram shows two paths namely deleting or updating a mitigation. If an admin chooses to delete a mitigation the user would select the mitigation to manage and delete the mitigation. If an admin chooses to update a mitigation, there are several properties of a mitigation the admin might want to change. Editing a mitigation includes editing the description, security controls, operating system, and author. As shown by the extend line in the diagram these operations may not take place if the admin does not require to update these attributes.

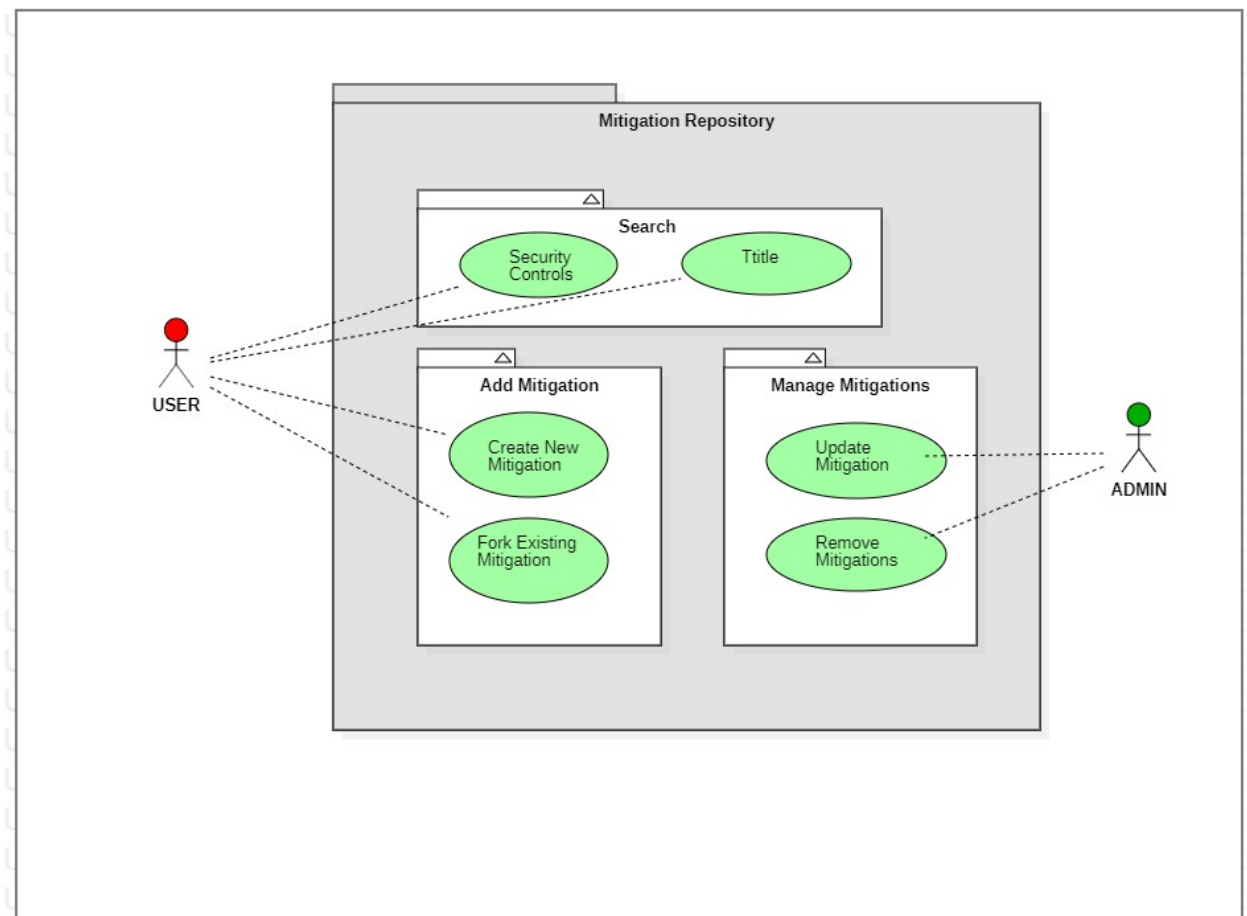


5. Architecture Overview

5.1 Logical View

5.1.1 Package Diagram

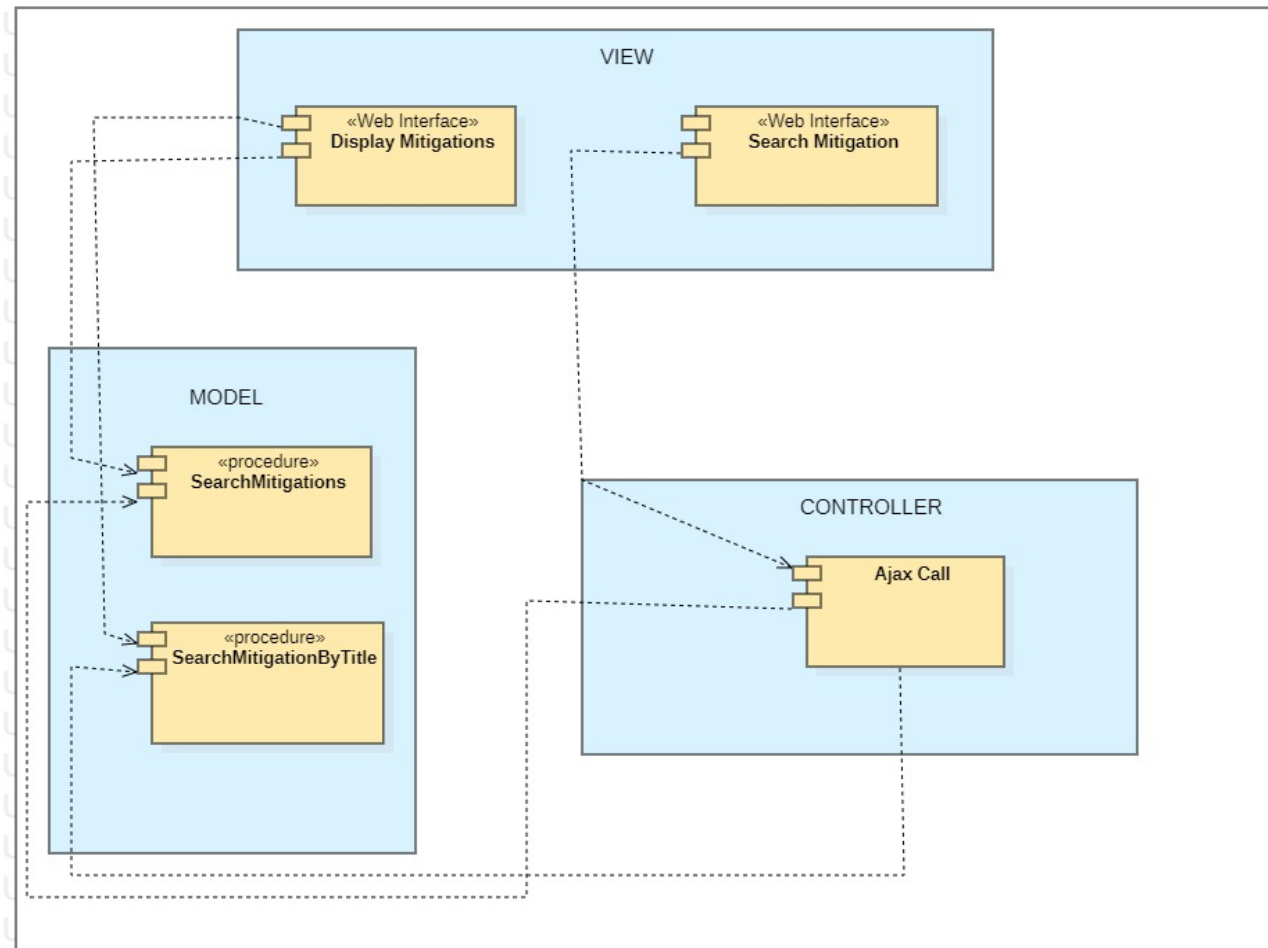
The diagram below shows how a general user can search and add new mitigations to the repository. However only an admin level user can manage mitigations. Managing the mitigations includes both updating mitigations (i.e. fixing spelling errors, incorrect info, etc.). Managing also includes deleting out of date mitigations as the need arises. These actions reflect meaningful changes in the repository that every user will see, and as such should be not done by every user. The admin can still search and add mitigations, but the general user cannot manage mitigation.



5.2 Development View

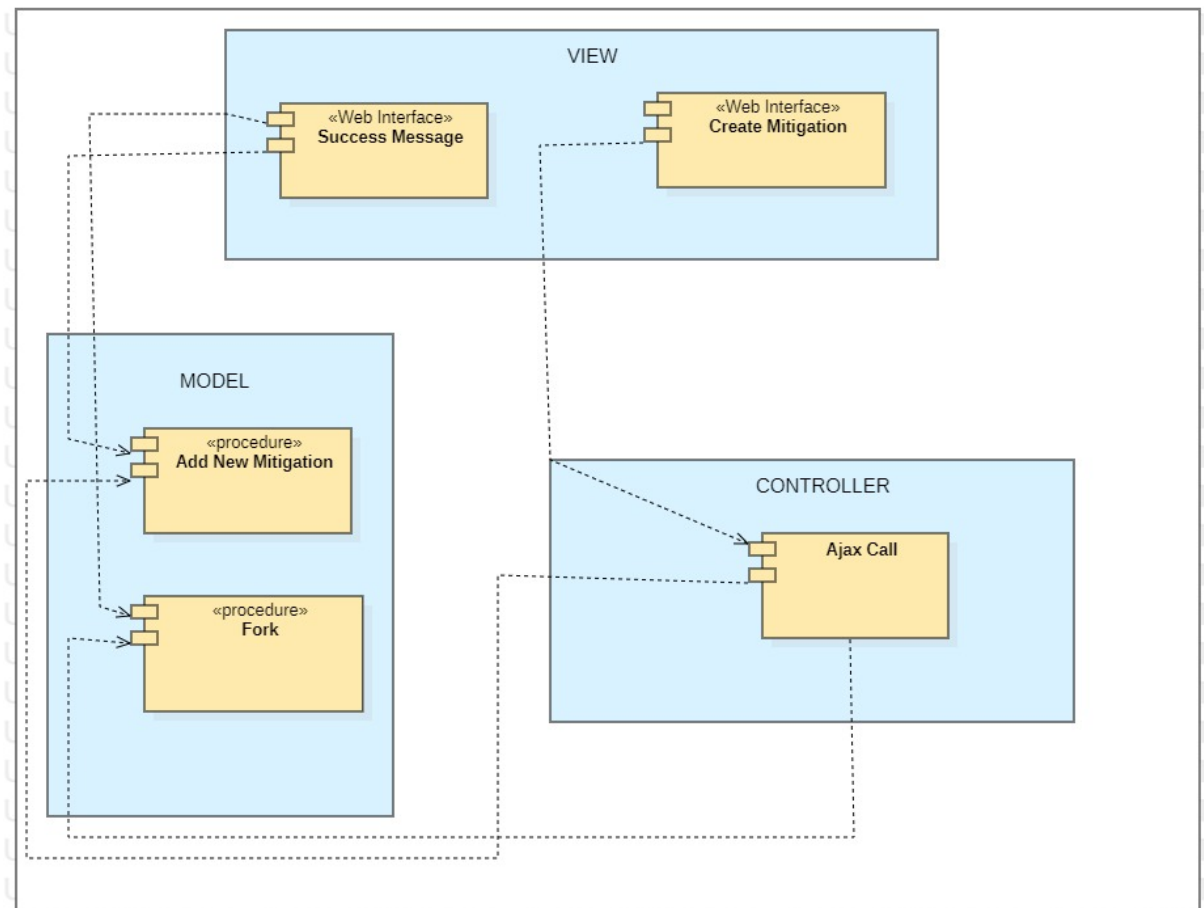
5.2.1 Component Diagram Search

This diagram shows components of the Mitigation Repository broken up into three categories. The view is representative of the Web Interface the user will interact with. The controller as labeled represents Ajax Calls which are used by in the system to make asynchronous calls so to the user the view looks like it never goes to another page. This Ajax call is a call to a database procedure. This system uses many calls to different database procedures. This is done to limit the PHP workload and to simplify the process. The model represents database procedures get information from the database to be displayed onto the view.



5.2.2 Component Diagram Add

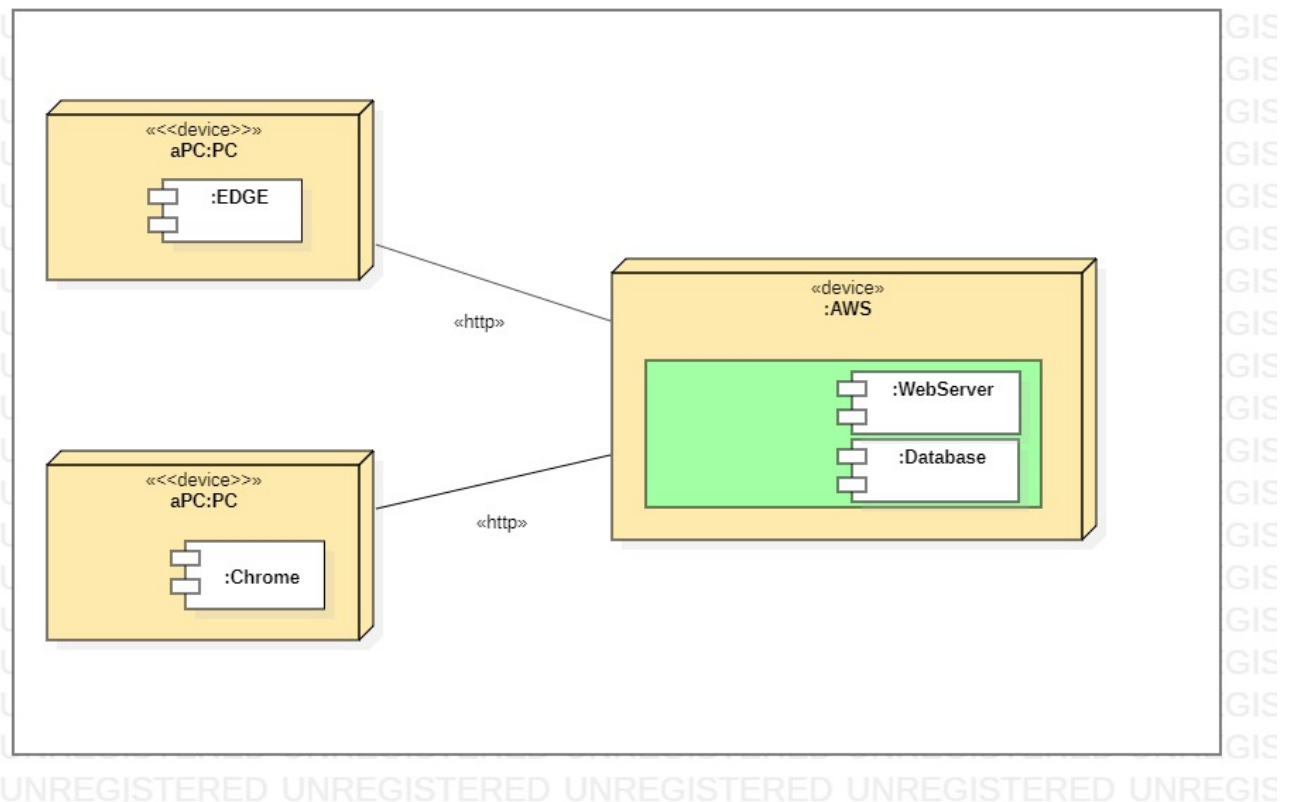
The component diagram below has identical subsystems (view, controller, model) to the diagram above. This diagram however shows that the same system is in place for adding mitigations to the Mitigation Repository as was for searching for mitigations. As seen in the diagram it follows the same pattern, first start with the user in the Interface, and making Ajax calls to database procedures. These procedures then add the data the user imputed that was passed by the Ajax calls. This system model is how all data the user sees or adds to the system occurs in the Mitigation Repository.



5.3 Physical View

5.3.1 Deployment Diagram

Due to the limitations and assumptions listed earlier, this document will be of less value as in the future the solution will not be available on AWS. But for documentation the diagram showcases the deployment of the solution as currently modeled. Notably both the web server and database occupy the same space as both are hosted on an Amazon Web Server. Also, of interest the Mitigation Repository should work regardless of browser as shown on the left side of the diagram.

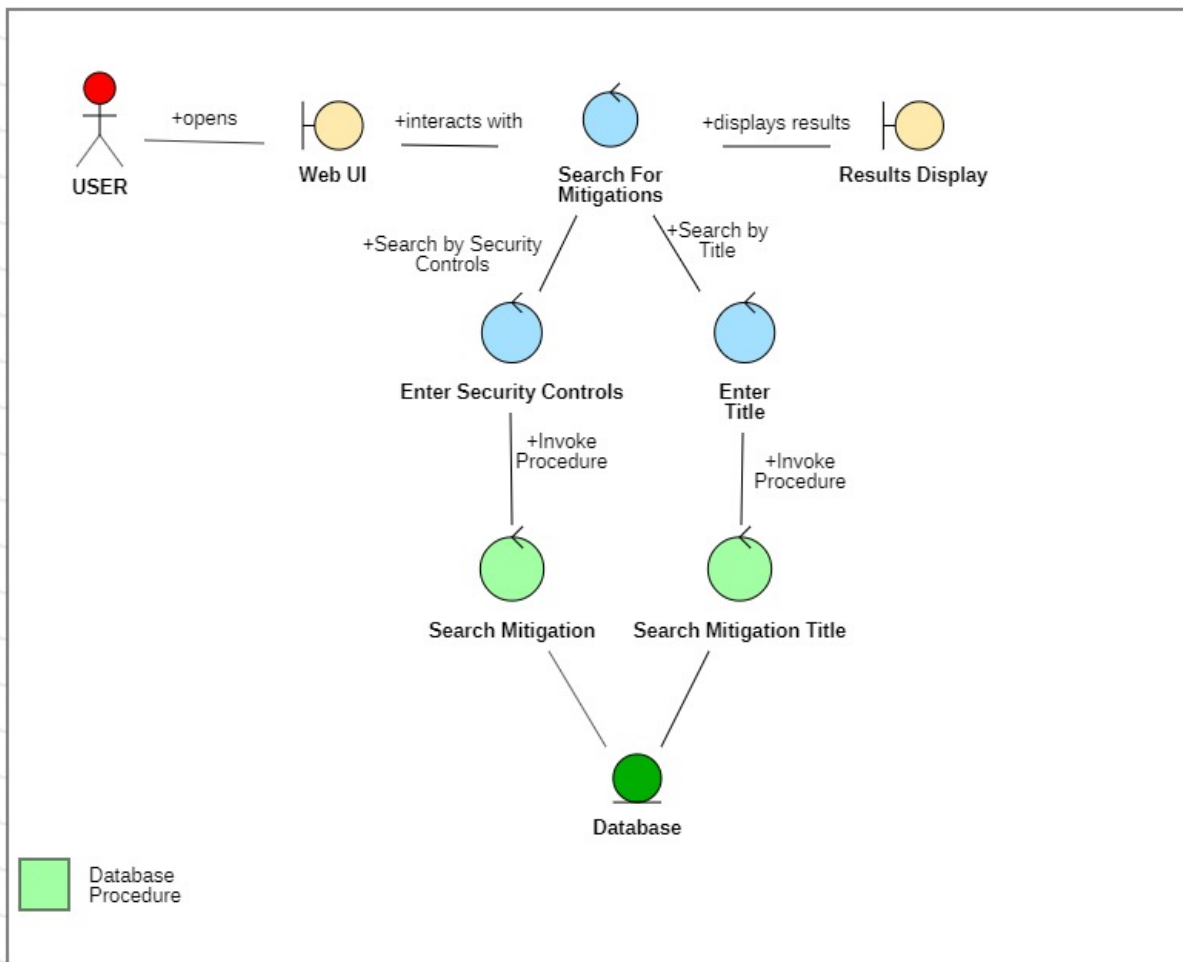


6. Design Overview

6.1 Search Functionality

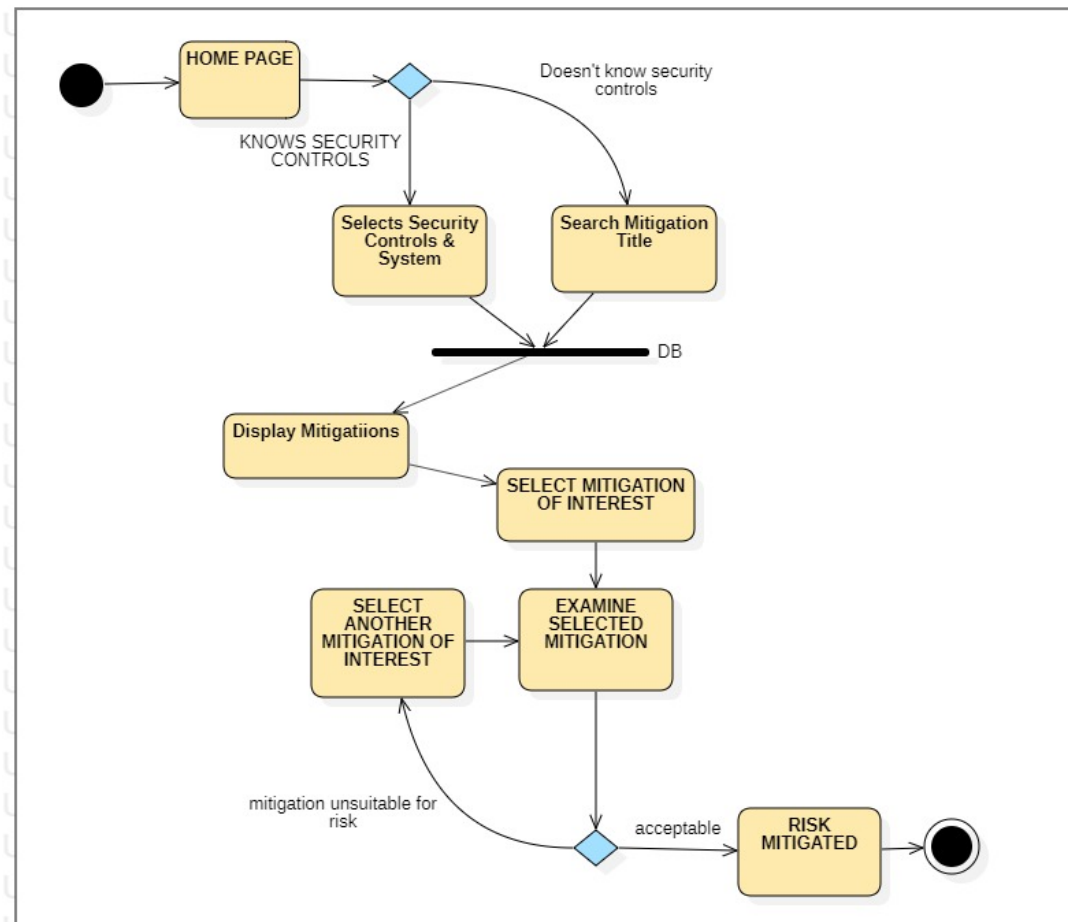
6.1.1 Analysis Class Diagram Search

The diagram below represents a user performing a general search in the Mitigation Repository. Going through the diagram we can see the user can search by either Security Controls or by entering a title. We can also see that both have separate procedure calls. This is done so the user can either know what exactly the mitigation is called (the title) or the user can search by selecting the Security Controls to find a mitigation of interest. This separation is achieved on the web interface with a search bar for entering the title and two drop-down buttons for selecting the Security Controls. The results returned by the procedures are then displayed on a results page.



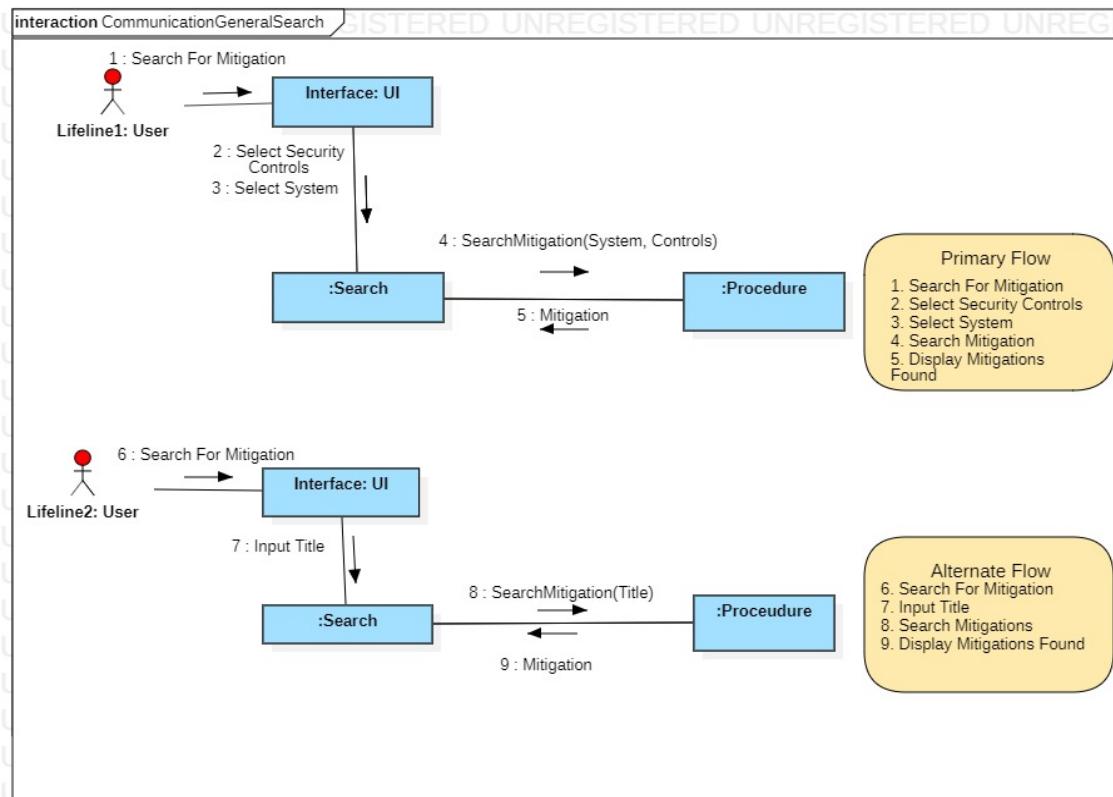
6.1.2 Activity Diagram Search

The diagram below shows a general search process happens in the system. It starts with the user at the home page and then choosing to search for a mitigation. They then have the choice to either search by entering security controls or by searching a title. Regardless of choice both then lead to procedure calls that will then search for the mitigations that correspond to the input. Then will display the mitigation results that correlate to what was selected or entered. Users can then select mitigation to display further information on the right hand of the screen. The user then can evaluate if the risk is mitigated by the mitigation listed or select another mitigation to find a suitable mitigation for the risk.



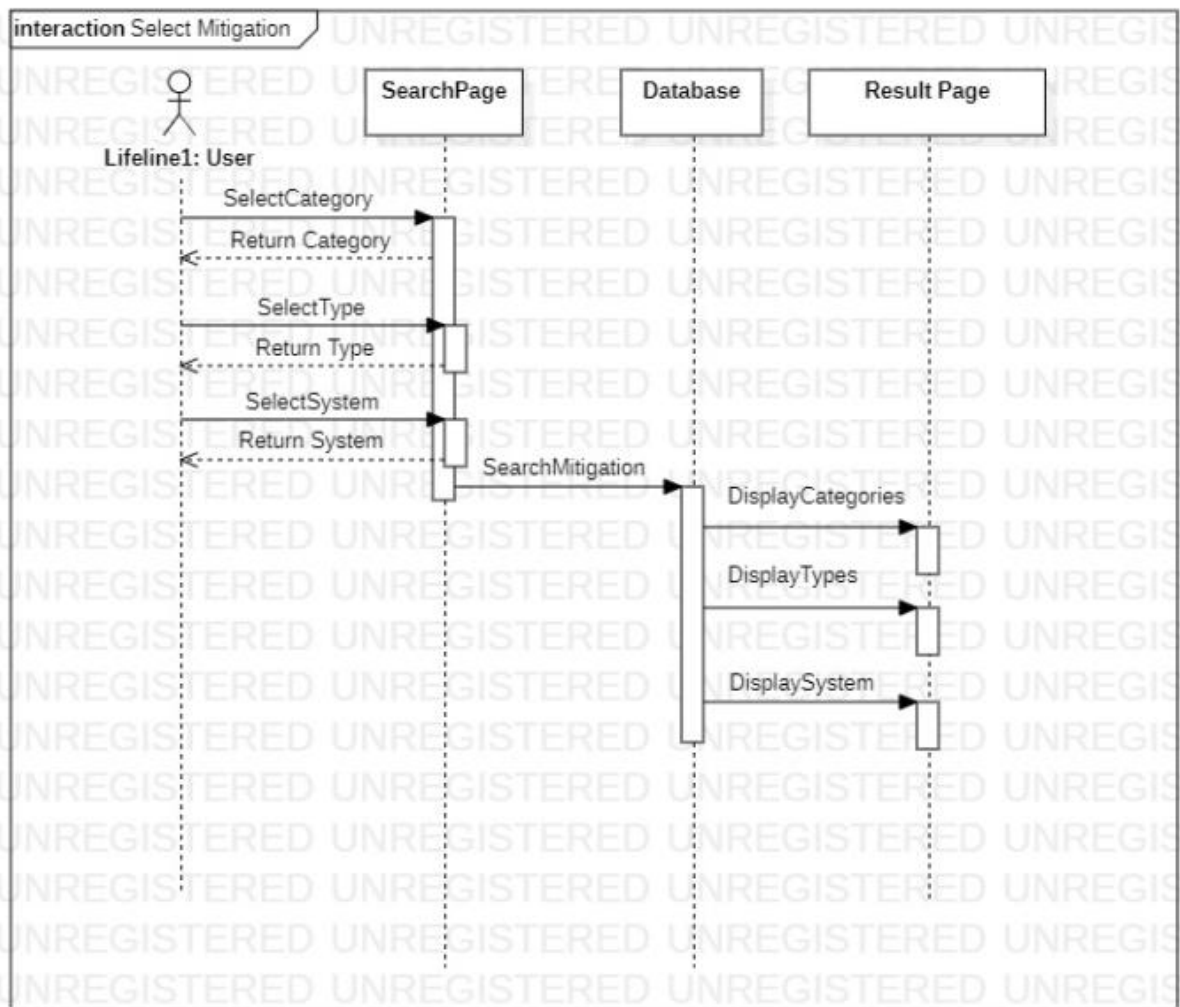
6.1.3 Communication Diagram Search

The diagram below shows when a user performs a search process in the system, separated into two flows. In both flows it starts with the user at the home page and then choosing to search for a mitigation. In the primary flow the user selects the security controls and system. This is then sent to a search function that makes an Ajax call to the database that then starts a procedure at the database level. This is then performed, and the result is passed back to the same function so that it can be displayed. The alternate flow shows a user enter a title which is sent to the search function which then sends the title as a parameter to a procedure. This procedure matches what the user inputted exactly with mitigation titles. This means if a user searched for the keyword “Mitigation” only mitigations containing “Mitigation” will be displayed to the user. In future iterations of the Mitigation Repository there will be an option to then filter these results by entering a more specific keyword in a filter box. This would then take the list and filter out mitigations not containing the keyword.



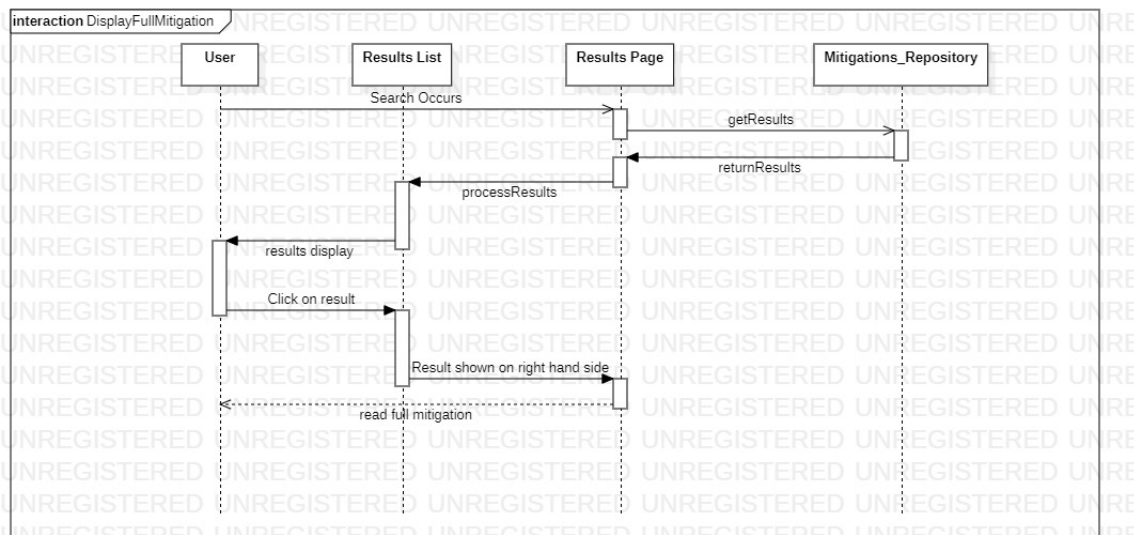
6.1.4 Sequence Diagram Select Search

When, a user of the Mitigation Repository wants to search a mitigation from the repository the following process occurs. The process starts with the user selecting the Security Controls from the drop downs displayed on the home page. Then the user can select the system that corresponds with the risk they are trying to mitigate. After this, a call is made by an Ajax method to call a database procedure. The Ajax method passes the users selected data as parameters to the procedure. The procedure then returns a data set to the Ajax method, which can then be formatted into a list to display on the results page.



6.1.5 Sequence Diagram Display Detailed

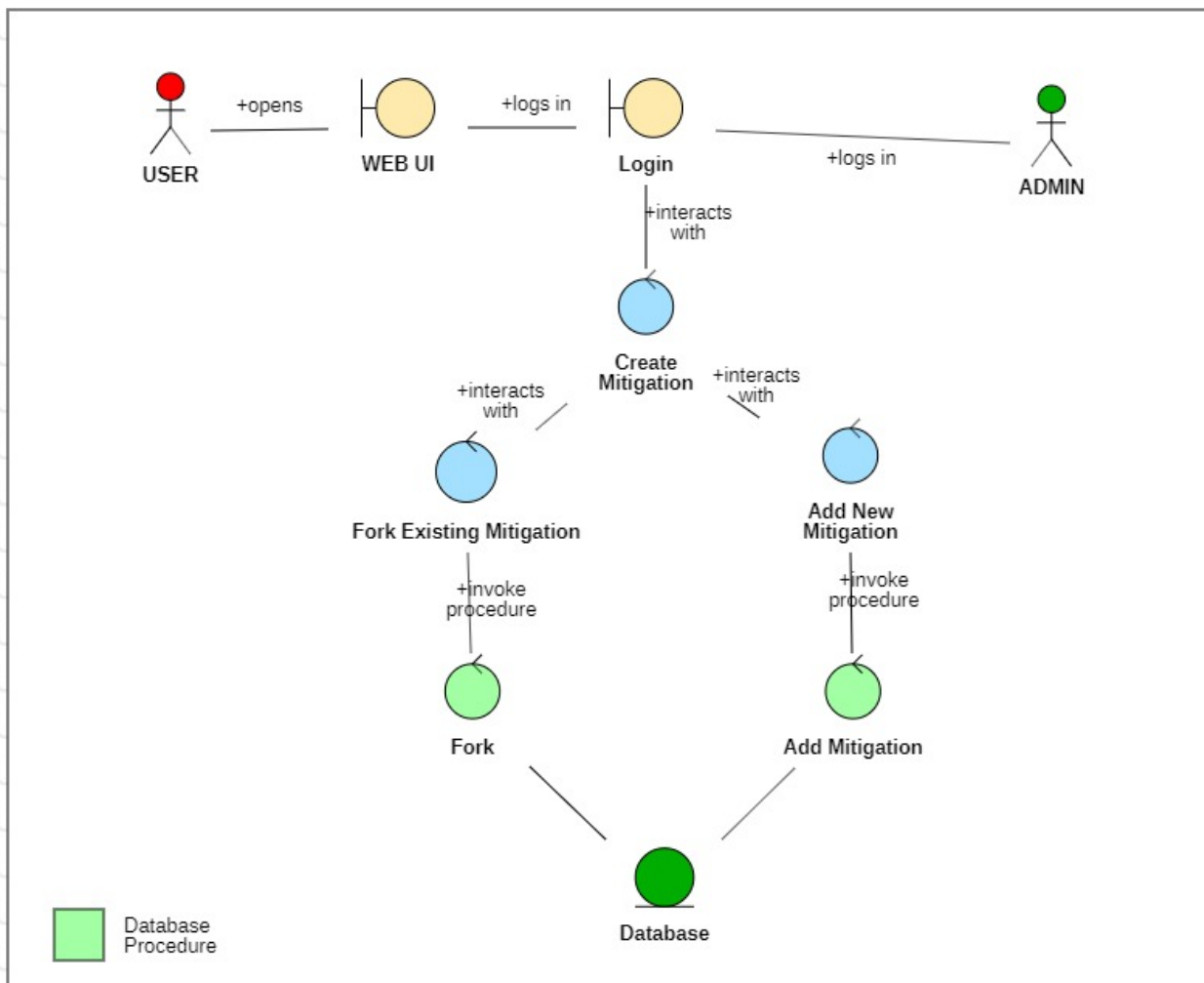
When a user of the Mitigation Repository wants to view the detailed information of the mitigation, the following process occurs. First the user will search for a mitigation as described in the previous diagram above. The user can use either search by title or by inputting the security controls. These results are then listed on the screen vertically on the left hand of the screen in a list, with a brief view of each mitigation. The brief view contains the title of the mitigation and part of the description. This enables the user to see a greater number of possible mitigations that might pertain to their search. To view all the information regarding each of the mitigations the user then clicks on the mitigation which makes a separate call using Ajax methods again calling a database procedure to display all the information on the right hand of the screen for the mitigation that was clicked on by the user. This is done by an asynchronous call using Ajax functions to call a procedure with the selected mitigation id number as a parameter. This procedure then returns the detailed mitigation information to the Ajax method. The method then displays the detailed mitigation selected by the user on the right hand of the page. This allows the page to update without requiring the user to change web pages to view more information. This can be repeated every time the user clicks on a new mitigation on the left-hand side an Ajax call is performed to display the information in greater detail.



6.2 Add Functionality

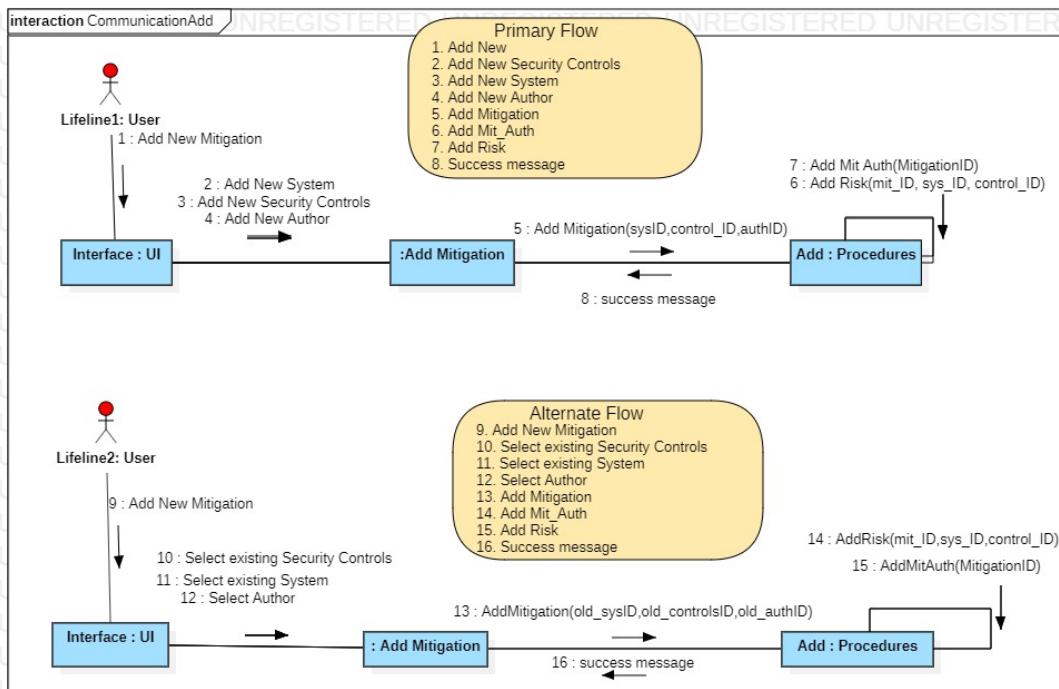
6.2.1 Analysis Class Diagram Add

The diagram below represents the user adding a mitigation in the Mitigation Repository. First to add a mitigation a user must login from a login interface. The login page is necessary for security of the system, so that only a user with the right qualifications can add a new mitigation to the repository. Once a user logs in, they interact with the create mitigation section of the user interface on the page. They can choose to either add a new mitigation or fork off an existing mitigation. These both get called by using jQuery Ajax calls to invoke different procedures in the database to carry out adding the mitigation to the Mitigation Repository Database.



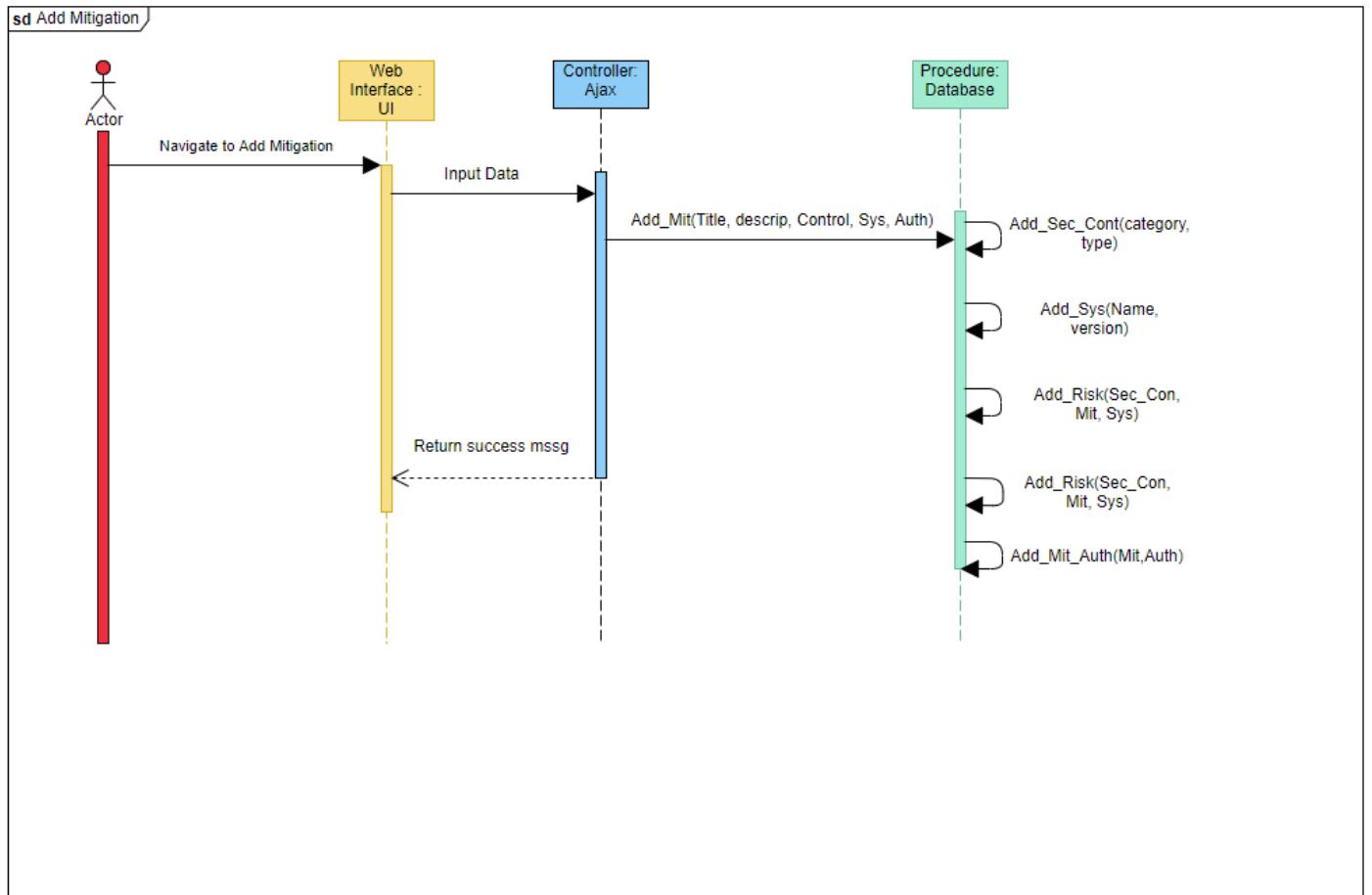
6.2.2 Communication Diagram Add

The diagram below shows a user adding a mitigation to the repository represented by two different flows. The primary flow shows the user adding a new mitigation that contains all new information not previously represented in the database. This new information includes all aspects of the mitigation including system, security controls, and author. The alternate flow shows a user adding a new mitigation with old information, meaning that the system or security controls already exist. Because the information already exists in the repository there is no reason to add them a second time, so this flow shows the user selecting previously existing ones and then follows the same steps. In both flows the Ajax function calls the same procedure to add the mitigation. This is because as described in Diagram 5.2.6 the procedure has several inner procedure calls. The inner procedure calls check to see if the inputted data exists before adding and will not overwrite existing rows in the tables. Then in the outer procedure the data is selected after the check to be used to form the necessary links in the database. This causes the procedure to perform two selects and an insert for each of the inputs. This however gives the benefit of simplifying the task into one procedure that can handle both cases as well as a mixture of the two at the price of querying time.



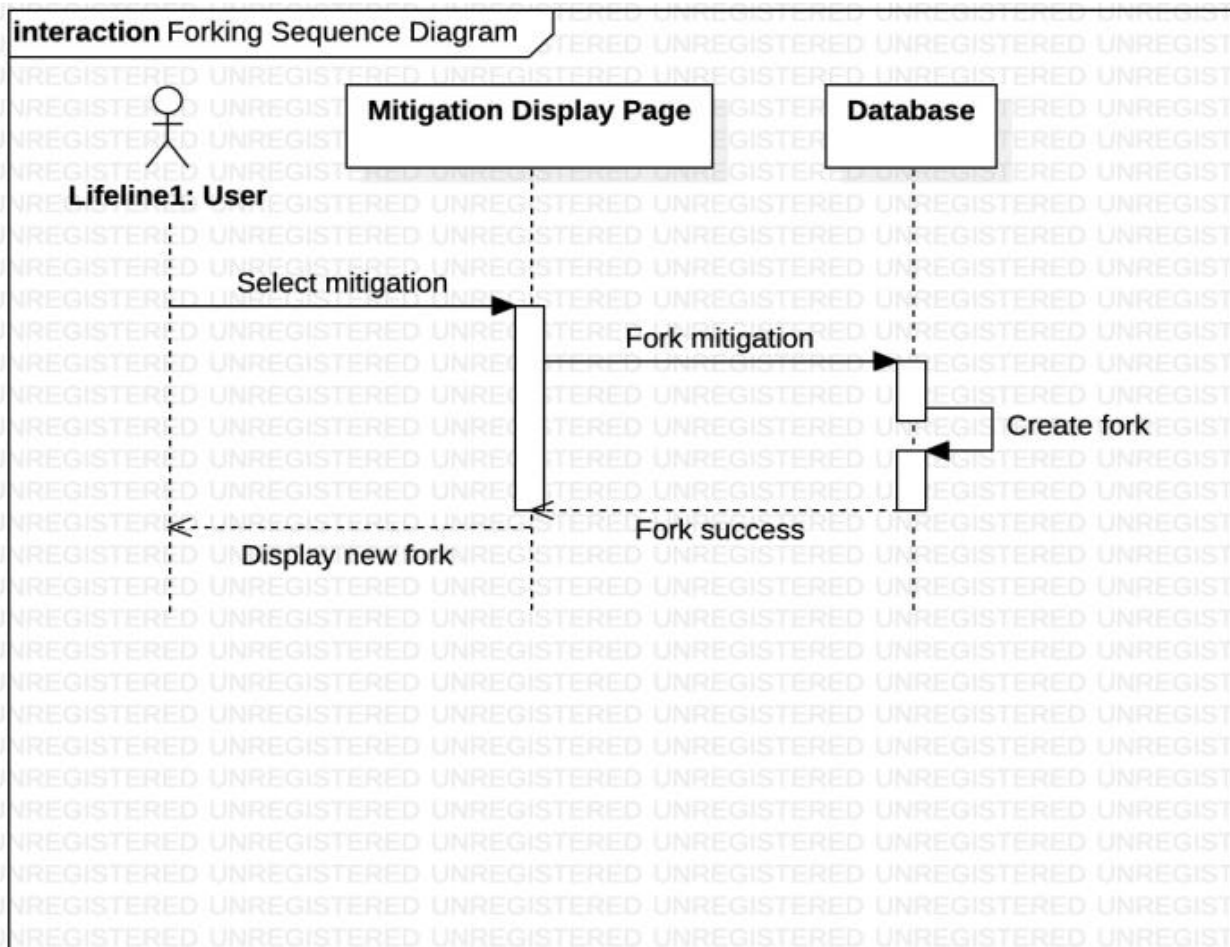
6.2.3 Sequence Diagram Add General

When a user of the Mitigation Repository wants to add a mitigation containing all new data to the repository the following processes occur. The first action for the user is to select the add mitigation button on the home page. Then the user will be taken to the add mitigation page where there are fields to enter the data. The fields to enter data include the Mitigation title, description. The remaining fields include the system, security controls, and author. Once the user enters the data and hits enter this will send an Ajax call containing the user inputted data to a procedure. This procedure then will make several procedure calls to add the new system, security controls. As a developer you could have the user enter one field at a time and only call one procedure at once. This however would require the web interface to be refreshed several times or taken to several pages to input all the data. Rather than refresh or change page the system was designed to input all the data, then refresh once if successful. If the mitigation and associated data is valid no null input, then the user will get a success message confirming the mitigation was added to the repository.



6.2.4 Sequence Diagram Forking

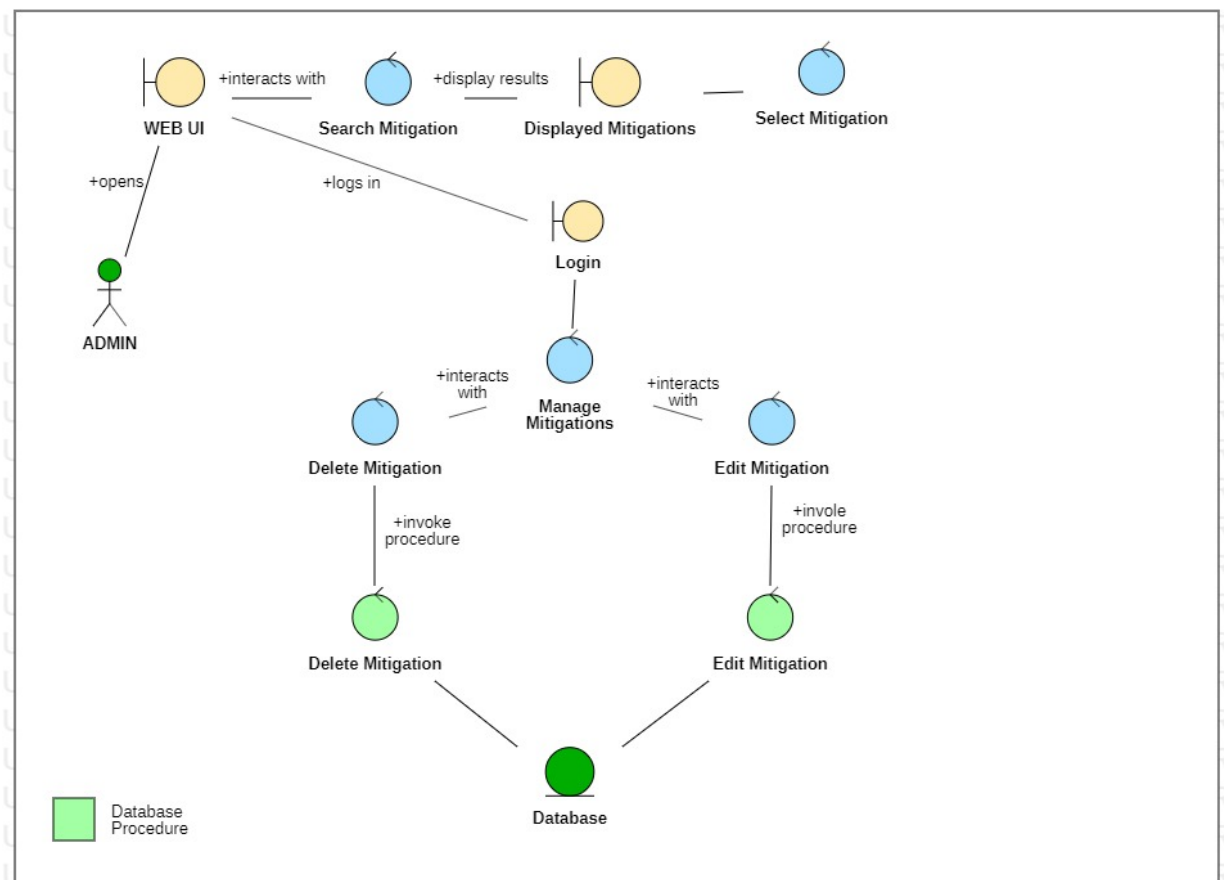
When a user of the Mitigation Repository wants to add a mitigation based off an existing mitigation in the repository the following processes occur. First the user searches and then selects an existing mitigation they want to fork, to create a new mitigation. Then the user will click to fork the mitigation they selected. They will then enter the new description and author into their respective text boxes. This inputted data will be sent from an Ajax call to a database procedure along with the id number of the mitigation selected. The procedure checks if the mitigation is valid by ensuring the parent mitigation exists before creating the new child mitigation. If successful, the user will get a success message confirming the mitigation was added.



6.3 Manage Functionality

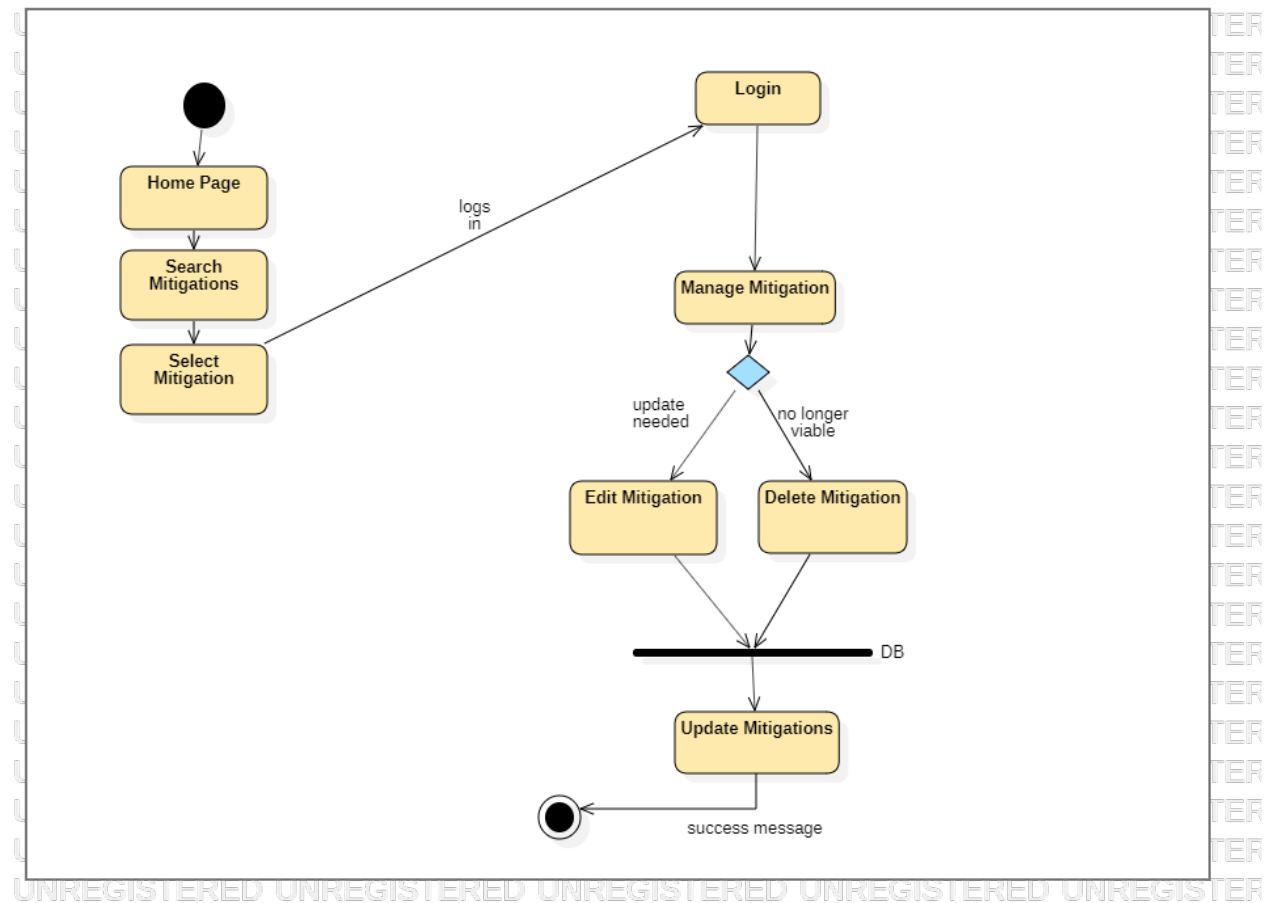
6.3.1 Analysis Class Diagram Manage

The diagram below represents the steps an admin would take to manage a mitigation in the Mitigation Repository. First the admin can search for mitigations and select mitigations to further inspect them. When doing this they may spot an issue with the mitigation that needs updating, so the admin can log in and edit the mitigation (fix spelling, correct mistakes, delete). The Mitigation Repository was designed so the admin does not have to be at the database level to fix small issues. This helps maintain security while allowing a way for the admin to fix the issues of the system. As shown below the admin (user) does not have to log in to perform searches however, to edit a mitigation the admin must log in. This enables the system to provide the necessary functionality without compromising security.



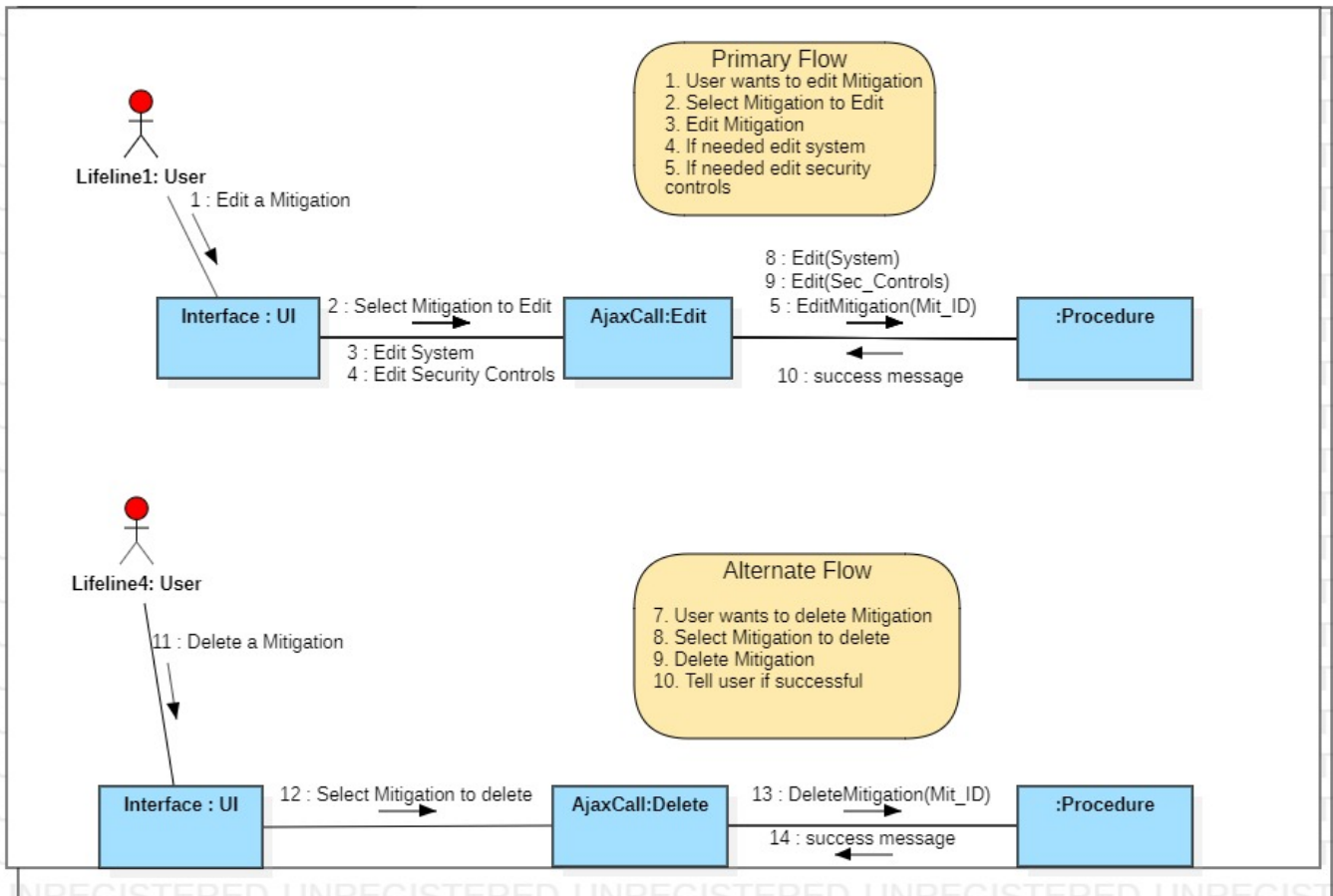
6.3.2 Activity Diagram Manage

The diagram below shows the steps a user of the mitigation repository would take to manage the database. Similarly, to add a mitigation to manage a mitigation a user must login to make changes. This is done after searching and selecting a mitigation that requires updating or deleting by the user. Then after the user chooses how to manage the mitigation calls are made to procedures in the database to update the mitigation.



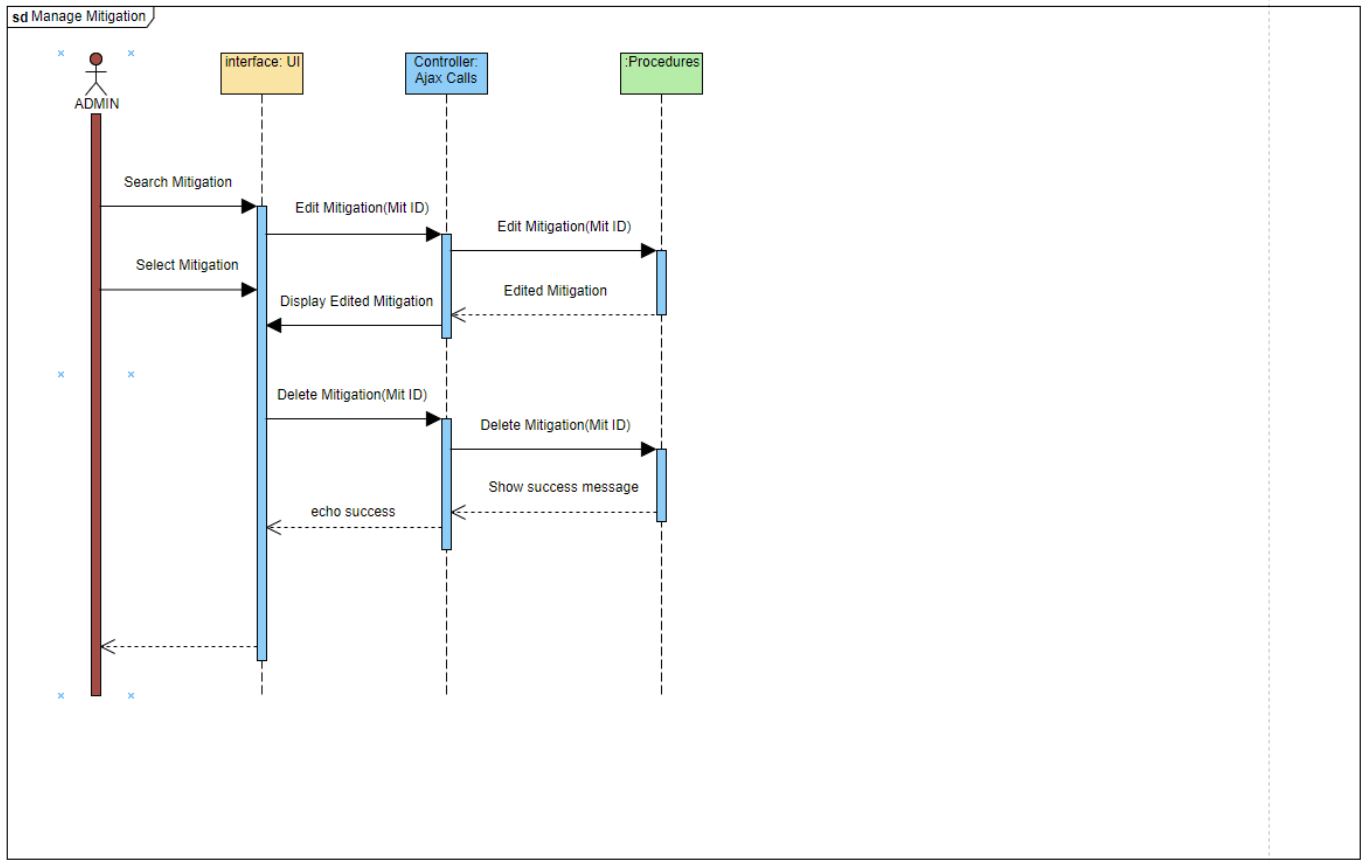
6.3.3 Communication Diagram Manage

The diagram below shows a user managing a mitigation which is broken up into two flows. The primary flow is a user that wants to edit a mitigation by first selecting the mitigation. Then an Ajax call is made to the database where an edit mitigation procedure can be used to update the mitigation. Also, a user might also edit the system or security controls that correspond to the mitigation they selected. These changes would be entered and then call database procedures to update. The alternate flow is when a user determines a mitigation needs to be deleted. First once again the user selects a mitigation, then an Ajax call runs a procedure to delete the mitigation and then return a success message back to the users.



6.3.4 Sequence Diagram Manage

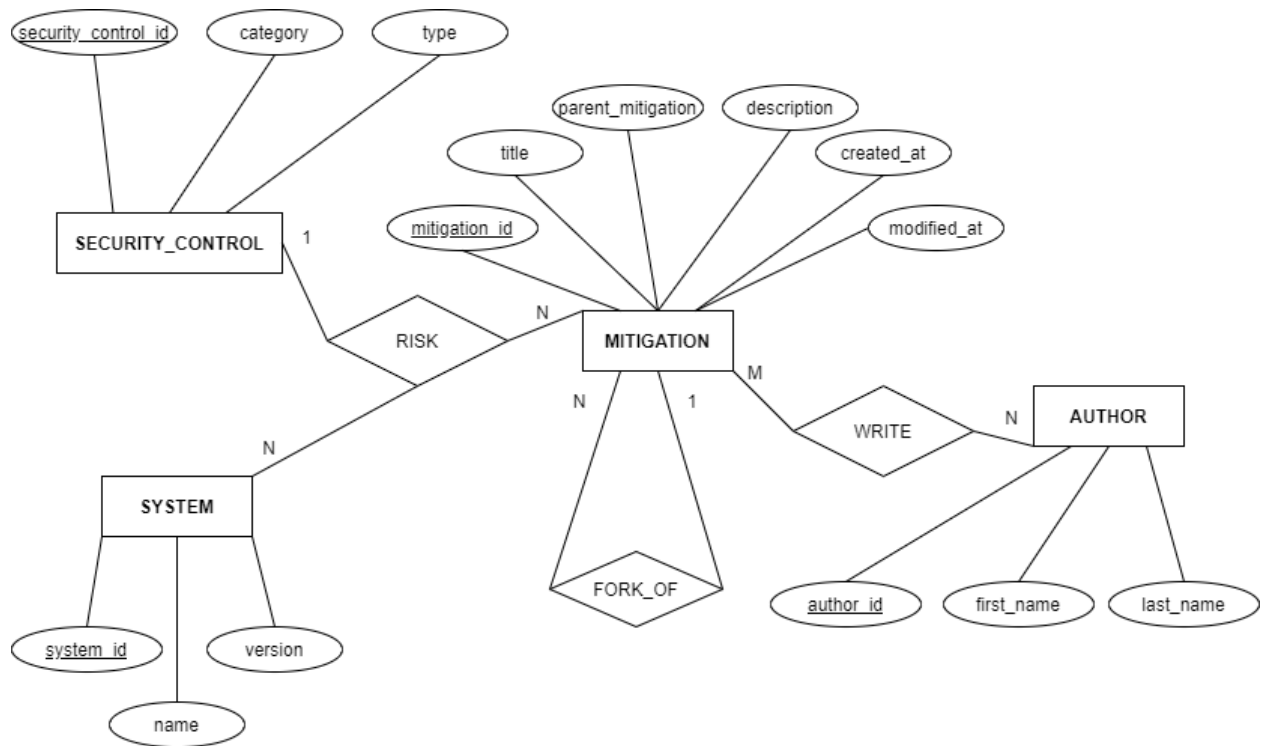
When a user of the Mitigation Repository wants to manage a mitigation in the repository the following processes occur. The user will first select the mitigation to manage. This can be done first by searching for a mitigation and choosing to manage the mitigation. Then depending on if the user wants to delete or update the mitigation, the user will either select to update or select to delete the mitigation. To update the user will click on an update button to be taken to the edit mitigation page where there are fields to update the data. The user can choose which fields need updating and then input the data, which is then handled by an ajax call to database procedures corresponding to the input. For example, if the user updates the description the system will call the update mitigation procedure. If the user decides to delete the mitigation, an Ajax call will be made to a procedure containing the id of the mitigation the user wants to delete. This procedure then deletes the mitigation from the table. When this happens the foreign keys in the linking tables Risk and Mitigation_Auth(see pg. 37-39) are deleted from their associated tables. If deleted the user will get a success message displayed to the screen.



6.4 Data Design

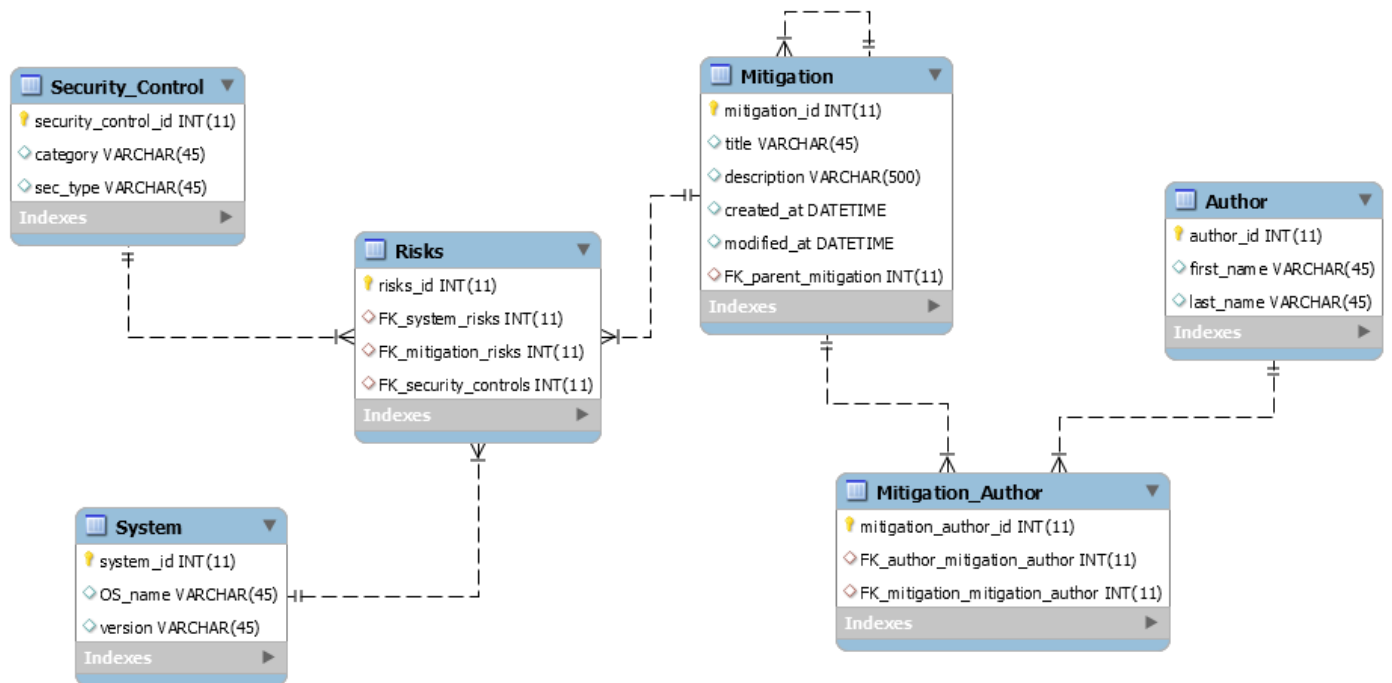
6.4.1 ER (Entity Relationship) Model

The diagram below is an Entity Relationship (ER) Model of the Mitigation Repository database.



6.4.2 Physical Model

The diagram below is a Physical Model of the database comprising the Mitigation Repository. The Security_Control table models the security controls that reduce a risk, consisting of a category and type. The System table represents the applicable system (Operating System) to the mitigation consisting of the name and version. The Mitigation Table represents the mitigation that will mitigate the risk the user faces. The Mitigation Table allows for a title, description, and created and modified dates to allow the user to search by most recent for example. The author table represents the author who add the mitigation to the repository. The Security controls and system are both used as inputs for the users when searching in the Mitigation Repository and are required to be connected to the mitigation table. This link is referred to as the Risks table and contains three foreign keys that link the tables together. The foreign keys mitigation_risks, and mitigation_mitiagtion_author is set to delete cascading. This is so when an admin deletes a mitigation its links in the Risks and Mitigation Auth table are deleted however the system, security controls and author persist. This is modeled this way as several mitigations can use the same system so deleting a system or security control can break other mitigations.



7. Tools Used

7.1 Tools, and purpose

1. **Ajax**: Ajax was used to help create asynchronous interfaces when displaying mitigations to the user. This allows the user to stay on one web page but display different content.
2. **DataGrip** v2019.3: This tool was used in writing Database procedures and general database overview. This tool was chosen due to familiarity with JetBrains IDE's and ease of use on setting up connection to Database.
3. **GitHub**: This tool was used for keeping development team members in sync and the ability to test/rollback changes made to files and code.
4. **jQuery** v3.4.1: This library was used to help implement Ajax in the web development. This tool was chosen to help simplify the process of implement Ajax.
5. **MySQL Workbench** v8.0.19: This tool was used to initialize the tables in the database and create the foreign key primary key relationships to the tables. This tool was chosen due to familiarity among development team members, as well as desired functionality.
6. **Notepad ++** v7.8.5: This tool was used for working with HTML and CSS files for the web UI used in the project. This tool was chosen due to familiarity and ease of use in working with HTML, and CSS filetypes.
7. **PHPSTORM** v2019.3: This tool was used for working with jQuery and Ajax functions and methods used to make the web interface. This tool was used for due to support for jQuery and Ajax.
8. **StarUML** v 3.2.2: This tool was used to create the diagrams composing this document. This tool was chosen due to it being a free UML diagram editor that could handle all the needs to make the diagram comprising the document.

8. Glossary of Terms

Entity Relationship Diagram: a diagram to visualize data relationship and dependencies between entities inside of a database.

Forking: the ability to clone a mitigation so users can work on mitigations to add different solutions to the same mitigation.

Repository: a central table for all storage locations. It is used to implement version control and can store multiple versions of a data record. There are countless types of mitigations in cyber security so using centralized data, it ensures that all the data is being pulled and stored in one location.

Risk Mitigation: a strategy to prepare or decrease the presence of threats faced by data center. Usually illustrated in steps for a user to follow to reduce possible negative effects on a system's data center.

Security Control: a way to reduce or mitigate a risk to assets (ex: physical property, information data, hardware systems). These are classified by multiple criteria which is given a category (ex: directive, preventative, detective, etc.) and a type (ex: physical, administrative, or technical)

Use Case Diagram: a diagram to visualize the different roles throughout the system and how those roles interact with the system.

9. Revision History

Version	Date	Updated By	Comments
1.0	03/23/20	David Glennan	Initial document creation
2.0	04/03/20	David Glennan	Revised diagram descriptions, two diagrams updated, and a third removed.
3.0	05/03/20	David Glennan	Reordered document by functionality rather than order by diagram.