Change Summary Table

Version	Name	Date	Description
1.0.0	Hot Java	04/15/2020	Created first draft from sections 2.1 thru 2.4
1.0.1	Lauren Eagan	04/16/2020	Modified Use case diagram level 2 structure based on Elsa’s feedback.
1.0.2	Joaquin Hidalgo	04/16/2020	Edited ‘Create task’ Use case scenario based on Elsa’s feedback
1.0.3	Cynthia Sustaita	04/16/2020	Added items to section 2.4 and 2.5. Rephrased section 2.3 based on Elsa’s feedback.
1.0.4	Fernando Marquez	04/16/2020	Rephrased section 2.1 based on Elsa’s feedback.
1.0.5	Cynthia Sustaita	04/17/2020	Updated Use case diagram level 2 based on Ben’s feedback. Such changes include association between analyst and manage findings. Rephrased transaction include use case.
1.0.6	Lauren Eagan	04/17/2020	Updated and revised Use case scenarios based on Elsa’s and Ben feedback.
1.0.7	Cynthia Sustaita	04/17/2020	Rephrased section 2.1 based on Elsa’s feedback. Avoided

			using words that states that FRIC provides mitigations.
1.0.8	Lauren Eagan Joaquin Hidalgo Cynthia Sustaita Fernando Marquez	04/17/2020	Created missing Use Case Scenarios for Log-in, Transaction record log, push, pull, and edit task from an analyst perspective.

2. General Description

2.1. Produce Perspective

Findings and Reporting Information Console (FRIC) is a system for Cyber Experimentation and Analysis Division (CEAD) analysts to document and efficiently store findings in an organized manner with regards to cyber vulnerability experiments as well as provide mitigation recommendations. The system will aid a CEAD lead analyst to come up with tasks, and subtasks as well to assign them to CEAD analysts. The system will also be allowed to have multiple analysts that are able to sync data together in order to complete the given assessment as a team. The team will be able to keep track of the work of the users that will allow the status of progress to the system. The FRIC system is an independent and self-contained product that doesn't depend on any other system.

2.2. Product Features

A Use Case diagram is a representation of a user's interaction with a specific system, in this case, FRIC. These Use Case diagrams are made to validate our understanding of the system along its primarily functionalities. This is a great way to model our assumptions of the system for the clients in order to ensure the system is in agreement by all parties when it is delivered, and is exactly what the client wants and needs. A use case falls under the umbrella of a Unified Modeling Language (UML) which is a standardized way of using specific language to express ideas.

The ideas being expressed in the Use Case diagram (Fig. 4) refers to behaviors of the system that are acted upon by external actors in addition to the main functionalities provided by the system (FRIC). Actors (Fig. 1) represent external entities (E.g. humans and machines) that interact with the system by exchanging and/or providing information to the system with the goal of completing an event. A use case (Fig 2.) describes what happens in the system when an actor interacts with it to execute the use case as well as the abstract behavior of the system which is what the system is primarily used for. eneralization interactions between the actors and the system shown in Fig. 3. Association represents the interaction of an actor with specific use cases. An include relationship projects what common features that exist in 2 or more of the use cases. An extended relationship projects optional behavior to a base use case and lastly, a generalization relationship demonstrates inheritance from one element to another.

One last Use Case element that we're making use of is the Use Case scenario. Such an element represents a list of actions needed in order for an actor and the system to accomplish a certain goal. Each scenario contains the following: use case scenario name, description (brief description of use case), actors (list of actors involved in the system), pre-conditions (description of what must be true before entering the scenario), trigger condition (description of what initiates the scenario), flow of events (steps that occur as the actors and the system react while attempting to reach a goal) and optionally, an alternative (subflow of alternate steps, if applicable)

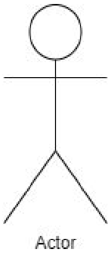

Fig 1: Actor notation

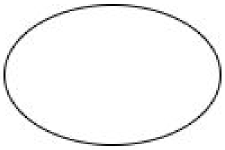

Fig 2: Use case notation

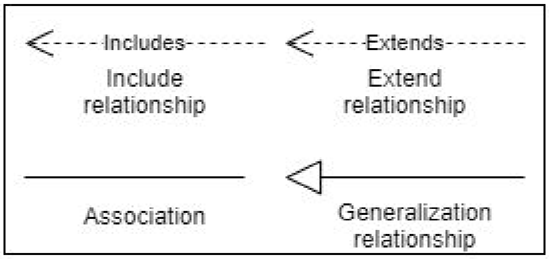

Fig 3: Relationship notation

2.2.1. Level 2 Use Case Diagram

The following figure (Fig. 4) represents a level 2 Use Case diagram to showcase the main functionalities of the FRIC system:

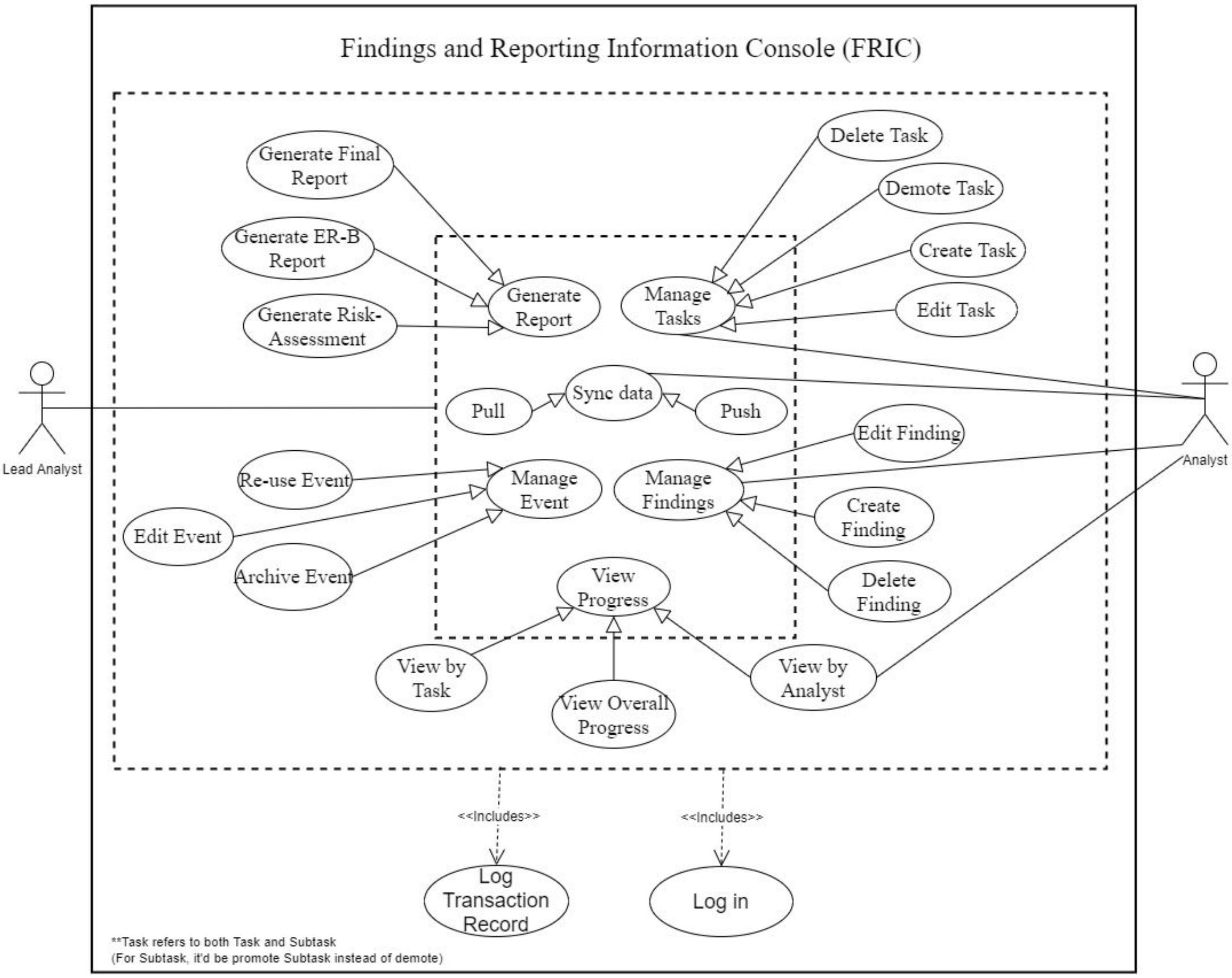

Fig 4: FRIC Level-2 Use Case diagram

2.2.2. Description of Actors

The following are detailed descriptions of our level 2 actors: Lead analyst and analyst.

- **Lead Analyst:** Lead analysts are a part of the Cyber Experimentation and Analysis Division (CEAD) that execute cyber experiments and create tasks based on such results and assign them to one analyst.
- **Analyst:** An analyst is a person from the CEAD that utilizes the FRIC system to work on tasks and to exploit and assess vulnerabilities that will be in sync with a lead analyst.

2.2.3. Description of Use Cases

The following are detailed descriptions of our level 2 use cases: Manage tasks, manage findings, generate report, sync data, view progress, and manage event:

- **Manage tasks:** The system allows a lead analyst to create/edit/delete tasks and sub-tasks, as well as relate a sub-task to a task, a task to a sub-task, or a task to a system. Additionally, this system allows a lead analyst to assign such tasks or sub-tasks to other analysts. While managing tasks, the lead analyst should be able to edit a task's or sub-tasks' attributes to help explain the progress of their work.
- **Manage findings:** The system allows any analyst to create/edit/delete a finding. Findings are to be tagged as vulnerable, informational or other. While managing findings, the analysts are allowed to edit a finding's attributes and search a finding by the type of tag it is or specific attribute. The system will also allow an analyst to attach a finding to a subtask. If no subtask exists under a task, the finding then can be attached to a task then be able to change that finding to a subtask later on, once created.
- **Generate report:** System allows analysts to generate an Emerging Result Brief (ER-B), a Risk assessment, or Final report and export a formatted technical report for PM use straight from the program. Such reports would be made up of observations regarding all the activities that were acquired during the event as either successfully or unsuccessfully, findings, its results and current progress.
- **Sync data:** The system allows all analysts to push data to the lead analyst or any other analyst and vice versa. This gives the opportunity for analysts to share what data they have and keep up to date with other people's data.
- **View progress:** The system allows analysts to view progress of the system, task or subtasks. The progress is the current state of the system, task or subtask. In-order to give analysts the ability to view the progress of the system, task or subtask as either not started, in progress, not do-able, completed or past due.
- **Manage event:** Manage events help the users setup and organize the natural assessment and penetration tests performed over a time period. Since this involves touching the system, by keeping history of past events the lead is able to go back and duplicate certain events to reuse in the future in order to save time on future projects. A lead is also able to go back and edit past events for their own benefit to view or even rewrite data in past events. Lastly, events are able to be archived to remove unwanted events and projects to keep the system more maintabale over time.

2.2.4. Use Case Scenarios

This section contains the detailed description of our Use Case diagram (Fig. 4) use case scenarios:

Use Case Scenario Name: Create task (Lead analyst assigns tasks)

Description: Create task with denying permission analyst to sign up for tasks. The system allows the lead analyst to set the configuration page preferences by either making all tasks under this specific event with only allowing the analyst to be assigned a task.
Actors: Lead Analyst
Pre-condition: A user has logged in and authenticated as a Lead Analyst, Lead Analyst chooses to assign tasks
Trigger-condition: Lead analyst initiates create task.
Flow of events:

- Step 1: System updates the task template for this event to provide Lead analysts with the ability to assign tasks.
- Step 2: Lead Analyst selects create task under event.
- Step 3: Lead Analyst fills in information needed to describe the task (analyst assigned, title, description, status, priority and due date).
- Step 4: System creates the task on all systems, and set's the tasks progress category to assigned.
- Step 5: System updates the task on all systems.
- Step 6: Include <Record transaction log>
- Step 7: End of use case.

Use Case Scenario Name: Create task (only allowing analyst to pick tasks)
Description: Create task with permission for analyst to sign up for tasks. The system allows the lead analyst to set the configuration page preferences by either making all tasks under this specific event with only allowing all analysts to pick a task.
Actors: Analyst and Lead Analyst
Pre-condition: A user has logged in and authenticated as a Lead Analyst, lead analyst allows analysts to pick from a list of tasks.
Trigger-condition: Lead analyst initiates create task.
Flow of events:

- Step 1: System updates the task template for this event to only allow analysts to pick their own tasks.
- Step 2: Lead Analyst selects create task under event.
- Step 3: Lead Analyst fills in information needed to describe the task (title, description, status, priority and due date).
- Step 4: System creates the task on all systems for the task to have progress category of, not assigned
- Step 5: System updates the task on all systems.
- Step 6: System updates user, task data store ,and timestamp
- Step 7: Include <Record transaction log>
- Step 8: End of use case.

Use Case Scenario Name: Delete task
Description: The system allows the lead analyst to delete a task under manage tasks. Deleting a task is given to the lead analyst to archive any unwanted tasks.

Actors: Lead analyst

Pre-condition: A user has logged in and authenticated as a Lead Analyst, there are tasks in the system..

Trigger-condition: Lead analyst logs in and has selected the tasks they wish to have deleted.

Flow of events:

- Step 1: System displays a list of all tasks.
- Step 2: Lead Analyst selects the option to delete task(s).
- Step 3: System displays delete options: delete task, delete task and all associated subtasks, delete subtasks.
- Step 4: Lead analyst selects delete task and all associated subtasks or findings.
- Step 5: System displays confirmation message
- Step 6: Include <Record transaction log>
- Step 7: End of use case.

Use Case Scenario Name: Edit task (All access)

Description: The system allows the lead analyst to edit a task under manage tasks. Editing the task gives a lead analyst the ability to change the title, progress, description, status, priority and due date of a given task. Editing a task gives an analyst the ability to edit the progress of a given task.

Actors: Lead analyst

Pre-condition: A user has logged in and authenticated as a Lead Analyst, a task to edit must be available.

Trigger-condition: Lead analyst needs to modify a task.

Flow of events:

- Step 1: Lead Analyst selects the option to manage the tasks.
- Step 2: Lead Analyst selects to edit a certain task.
- Step 3: Lead Analyst overrides the previous edit and saves.
- Step 4: System displays confirmation message.
- Step 5: Include <Record transaction log>
- Step 6: End of use case.

Use Case Scenario Name: Edit task (Progress only)

Description: The system allows the Analyst to edit a task under manage tasks. Editing a task gives an analyst the ability to edit the progress of a given task.

Actors: Analyst

Pre-condition: A user has logged in and authenticated as an Analyst, a task to edit must be available.

Trigger-condition: Analyst needs to update the progress of a task.

Flow of events:

- Step 1: Analyst selects the option to manage the tasks.
- Step 2: Analyst selects to edit a certain task.
- Step 3: Analyst overrides the previous edit and saves.
- Step 4: System displays confirmation message.
- Step 5: Include <Record transaction log>
- Step 6: End of use case.

Use Case Scenario Name: Delete finding

Description: The system allows an analyst to delete a finding if and only if they are the author of such finding. .
Actors: Lead Analyst, Analyst
Pre-condition: A user has logged in and authenticated as a Lead Analyst or Analyst
Trigger-condition: Analyst has reviewed the current findings and knows that this certain finding is no longer usable, thus deletes finding.
Flow of events:

- Step 1: Analyst selects the option to manage findings.
- Step 2: Analyst selects the option to delete a finding.
- Step 3: Analyst selects the finding that is going to be removed.
- Step 4: System displays confirmation message.
- Step 5: System updates the new system with the certain finding deleted.
- Step 6: Include <Record transaction log>
- Step 7: End of use case.

Use Case Scenario Name: Create finding
Description: The system should allow both a lead analyst and an analyst to create a finding. The new finding will consist of a title, description, status, priority, and due date of the given/new task.
Actors: Lead analyst, Analyst
Pre-condition: A user has logged in and authenticated as a Lead Analyst or Analyst
Trigger-condition: Lead analyst or analyst wants to create a new finding
Flow of events:

- Step 1: Lead Analyst or Analyst selects the option to create a new finding.
- Step 2: System gives reading and writing permission to lead analyst/ analyst.
- Step 3: Lead Analyst or Analyst: selects to add appropriate information towards the document.
- Step 4: Lead Analyst or Analyst submits the new finding into the system.
- Step 5: System displays confirmation message.
- Step 6: Include <Record transaction log>
- Step 6: End of use case.

Use Case Scenario Name: Edit finding
Description: The system allows any analyst to edit any specific findings that are under a task whether it was chosen or assigned to.
Actors: Lead analyst, analyst
Pre-condition: A user has logged in and authenticated as a Lead Analyst or Analyst.
Trigger-condition: Analyst has reviewed the current findings and knows that this certain finding is no longer up to date and wishes to edit any attributes about this finding.
Flow of events:

- Step 2: Lead Analyst or Analyst selects the option to manage findings.
- Step 3: Lead Analyst or Analyst selects the option to edit a finding.
- Step 4: Lead Analyst or Analyst selects the certain finding that is going to be edited.
- Step 5: System gives reading and writing permission to the user.
- Step 6: Lead Analyst or Analyst selects to re-write the previous text section.

- Step 7: Lead Analyst or Analyst confirms the changes and saves.
- Step 8: System displays confirmation message
- Step 9: Include <Record transaction log>
- Step 10: End of use case.

Use Case Scenario Name: Pull data
Description: Lead Analyst or Analyst are able to receive (pull) data that has been submitted by other analysts that differ from the current user's data on the local computer system they are working on.
Actors: Lead Analyst, Analyst
Pre-condition: A user has logged in and authenticated as a lead analyst or analyst, there must be a Lead Analyst or Analyst with data different from another Lead Analyst or Analyst.
Trigger-condition: A Lead Analyst or Analyst needs changes made to data by another analyst.
Flow of events:

- Step 1: Lead Analyst or Analyst selects to pull data from the system.
- Step 2: System verifies to the previously existing data.
- Step 3: System displays confirmation.
- Step 4: System updates the data.
- Step 5: Include <Record transaction log>
- Step 6: End of use case.

Use Case Scenario Name: Push data
Description: Lead Analyst or Analyst are able to share (push) data to other lead analysts or analysts.
Actors: Lead analyst, Analyst.
Pre-condition: A user has logged in and authenticated as a Lead Analyst or Analyst, there must be a Lead Analyst or Analyst with data different from another Lead Analyst or Analyst.
Trigger-condition: An Analyst makes changes to data and wishes to update other analysts.
Flow of events:

- Step 1: Lead Analyst or Analyst selects to push data.
- Step 2: System verifies changes to the previously existing data.
- Step 3: System displays confirmation.
- Step 4: System updates the data.
- Step 5: Include <Record transaction log>
- Step 6: End of use case.

Use Case Scenario Name: Generate report
Description: System allows Lead Analysts to generate an Emerging Result Brief (ER-B), Risk Assessment report, or a Final Technical Report, and export a formatted technical report for PM use straight from the program. Such reports will be made up of observations regarding all the activities that were acquired during the event as either successful or unsuccessful, findings, its results and current progress.
Actors: Lead analyst
Pre-condition: A user has logged in and authenticated as a Lead analyst, event is in progress and has some progress or findings.

Trigger-condition: Lead analyst needs to send a report of the event and it's current progress.
Flow of events:

- Step 1: Lead analysts select the option to generate reports.
- Step 2: System displays properties to include: All findings/select findings successful/unsuccessful, results, and current progress.
- Step 3: Lead analyst selects all findings, and any current progress they would like to include.
- Step 4: Lead analyst submits the report.
- Step 5: System generates report and displays confirmation message.
- Step 6: Include <Record transaction log>
- Step 7: End of use case**.**

Use Case Scenario Name: View progress
Description: The system allows analysts to view progress of tasks, subtasks, or overall. The progress is the current state of the system: assigned, not do-able, not started, in progress, completed, past due or transferred .
Actors: Lead analyst
Pre-condition: A user has logged in and authenticated as a Lead Analyst, tasks have been created under an event.
Trigger-condition: Lead analyst wishes to view the current progress of a specific task.
Flow of events:

- Step 1: Lead analyst selects option to view progress.
- Step 2: System displays list of current tasks and any analysts they have been assigned to.
- Step 3: Actor selects the analyst they wish to view.
- Step 4: System displays current progress of the task to lead analyst.
- Step 5: Include <Record transaction log>
- Step 6: End of use case.

Use Case Scenario Name: Edit event
Description: The system allows a Lead Analyst to edit any event that falls under manage event.
Actors: Lead analyst
Pre-condition: A user has logged in and authenticated as a Lead Analyst, there are current events in progress.
Trigger-condition: Lead analyst must select manage events and pick a specific event to edit/view.
Flow of events:

- Step 1: System retrieves the event from the database.
- Step 2: System gives reading and writing permission to the user.
- Step 3: System displays event details.
- Step 4: Lead Analyst edits an event detail.
- Step 5: Lead Analyst saves and confirms changes.
- Step 6: System displays confirmation message
- Step 7: System updates the data.
- Step 8: Include <Record transaction log>
- Step 9: End of use case

Use Case Scenario Name: Re-use event

Description: The system allows a lead analyst to reuse previous events that fall under manage event. A lead analyst will be able to pull any event and reuse archived data.

Actors: Lead analyst

Pre-condition: A user has logged in and authenticated as a Lead Analyst.

Trigger-condition: Lead analyst must click on manage event and pick a specific event to reuse.

Flow of events:

- Step 1: Lead Analyst selects copy event.
- Step 2: System displays confirmation message
- Step 8: Include <Record transaction log>
- Step 4: End of use case.

Use Case Scenario Name: Archive event

Description: The system allows a Lead Analyst to archive an event, which will keep a history of completed events to allow them to be viewed and/or reused.

Actors: Lead analyst

Pre-condition: A user has logged in and authenticated as a Lead Analyst, and there are events in progress.

Trigger-condition: Lead analyst must click on manage events and pick a specific event to edit/view.

Flow of events:

- Step 1: Lead Analyst chooses to conclude the event
- Step 2: System displays option to archive an event
- Step 3: Lead Analyst clicks archive completed event
- Step 4: System displays confirmation message of archived event
- Step 5: Include <Record transaction log>
- Step 6: End of use case

Use Case Scenario Name: Login

Description: The authentication for CEAD analysts who log onto FRIC.

Actors: Lead analyst, Analyst

Pre-condition: Must be an authorized user, a CEAD analyst.

Trigger-condition: Lead analyst or analyst must input credentials.

Flow of events:

- Step 1: System displays fields to enter credentials.
- Step 2: Lead Analyst or Analyst enters credentials.
- Step 2: System authenticates the credentials.
- Step 3: System gives reading and writing permission depending on the user.
- Step 4: Include <Record transaction log>
- Step 5: End of use case

Alt: Step 1:

- Step 1.1: Analyst inputs wrong credentials
- Step 1.2: End of use case

Use Case Scenario Name: Log Transaction Record
Description: FRIC will keep a transaction record of any activity registered by both the Lead Analyst and Analyst.
Actors: Lead Analyst, Analyst
Pre-condition: A user has logged in and authenticated as a Lead Analyst or Analyst.
Trigger-condition: Lead analyst or analyst must input credentials
Flow of events:

- Step 1: System generates and saves records for any performed activities by the Lead analyst or analyst.
- Step 2: End of use case

Use Case Scenario Name: Promote subtask
Description: Lead Analyst and Analyst are able to promote a subtask to the status of task.
Actors: Lead Analyst, Analyst
Pre-condition: A user has logged in and authenticated as a Lead Analyst or Analyst, a subtask must exist in the system.
Trigger-condition: Analyst selects promote subtask.
Flow of events:

- Step 1: System displays area to enter reason for promotion.
- Step 2: Analyst enters the reason for promotion.
- Step 3: Analyst confirms and saves.
- Step 4: Include <Record transaction log>
- Step 5: End of use case

Use Case Scenario Name: Demote Task
Description: Lead Analyst will reduce the higher action task to subtask.
Actors: Lead analyst
Pre-condition: A user has logged in and authenticated as a Lead Analyst, a task must exist in the system
Trigger-condition: Lead analyst selects demote a task.
Flow of events:

- Step 1: System displays area to input reason for demotion.
- Step 2: Lead analyst enters reason for demotion.
- Step 3: Lead analyst selects task parent to assign to the task being demoted.
- Step 4: Lead analyst confirms and saves.
- Step 5: Include <Record transaction log>
- Step 6: End of use case.

2.3. User Characteristics

The users that will be making use of the FRIC system will be CEAD analysts and lead analysts that will have knowledge in cybersecurity and a variety of technical skills such as computer usage and operating systems.

2.4. General Constraints

The following are known constraints of the system, these constraints will describe the factors which will determine the choices made in the development of the system.

1. The system shall not be accessible to unauthorized users.
2. The system shall be completed by the end of the Fall 2020 semester.
3. The system shall not make use of internet connection.

2.5. Assumptions and Dependencies

The teams' assumptions and dependencies for the FRIC system are the following:

1. The system shall be accessible as a web application.
2. The system shall be implemented in Python.
3. The CEAD will provide the Final Report template at a later time.
4. FRIC data will be saved into provided hard-drives