Team 11 - Hot Java

Findings and Reporting Information Console (FRIC)

Feasibility Report

Version 1.2.7

February, 28 2020

Document Control

Approval

The Guidance Team and the customers will approve this document.

Document Change Control

Initial Release	3.0
Current Release	3.0
Indicator of Last Page in Document	$
Date of Last Review	02/28/2020
Date of Next Review	02/28/2020
Target Date for Next Update	03/06/2020

Distribution List

This following list of people will receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members: Ms. Elsa Tai Ramierz and Ben Robertson

Customer: Cyber Experimentation & Analysis Division (CEAD)

Software Team Members: Jesus Gutierrez, Cynthia Sustaita, Joaquin Hidalgo, Fernando Marquez, Lauren Eagan

Change Summary

The following table details changes made between versions of this document

Version	Date	Modifier	Description
1.0.0	02/19/2020	Lauren Eagan	-Modified document's header
1.0.1	02/19/2020	Fernando Marquez	-Added use case diagrams, use case

			descriptions, and actors descriptions
1.0.2	02/19/2020	Cynthia Sustaita	-Added use case scenarios for level 1 and level 2
1.0.3	02/19/2020	Jesus Gutierrez Joaquin Hidalgo	-Added distribution list
1.0.4	02/19/2020	Jesus Gutierrez Joaquin Hidalgo	-Filled out document control section
1.0.5	02/19/2020	Hot Java	-Divided sections as a team: >Joaquin and Fernie will work on existing solution >Cynthia and Lauren will work on Consideration >Jesus will work on section 1
1.0.6	02/19/2020	Jesus Gutierrez	-Completed section 1.1 and 1.2
1.0.7	02/21/2020	Joaquin Hidalgo	-Added 2 existing solutions. 2.2 & 2.3 -Added intro paragraph to section 4 (solutions)
1.0.8	02/21/2020	Fernando Marquez	-Added existing solution 2.1
1.0.10	02/21/2020	Jesus Gutierrez	-Added section 1.0 and section 1.3
1.0.11	02/21/2020	Lauren Eagan	-Added content to sections 1.0-1.2 -Created outline for requirements definitions

1.1.0	02/21/2020	Lauren Eagan Cynthia Sustaita	Edited level 2 use case diagram
1.1.1	02/22/2020	Lauren Eagan	Added and separated needs based on need type into section 1.3
1.1.2	02/22/2020	Cynthia Sustaita	Added consideration 3.1 and 3.3
1.1.3	02/24/2020	Hot Java	Revised current progress of report and broke down considerations to individual assignment.
1.1.4	02/25/2020	Joaquin Hidalgo	-Added 2 considerations for the section 3.4 (communication system)
1.1.5	02/25/2020	Fernando Marquez	-Added section 3.2 Data syncing and subsections 3.2.1 and 3.2.2
1.1.6	02/25/2020	Jesus Gutierrez	-Created section 3.6 Report Generation System and subsections 3.6.1 and 3.6.2
1.1.7	02/25/2020	Cynthia Sustaita Joaquin Hidalgo Jesus Gutierrez Lauren Eagan	-Polished considerations section and constructed solution #1 as a team.
1.1.8	02/26/2020	Cynthia Sustaita	-Added individual options onto the solution 1. Re-worded requirements.

1.1.9	02/27/2020	Hot Java	Added missing details into solution 1 and started and completed solution 2
1.2.0	02/27/2020	Joaquin Hidalgo Cynthia Sustaita	Added use case elements definitions
1.2.1	02/27/2020	Cynthia Sustaita Joaquin Hidalgo	Completed solutions comparison table
1.2.2	02/27/2020	Cynthia Sustaita	Edited use case diagram level 2 and labeled diagrams. Added IEEE references to the References section
1.2.3	02/28/2020	Fernando Marquez	Update table of contents Fix formatting of report(Spacing, style, etc...) Revise draft report for any typos and errors
1.2.4	02/28/2020	Joaquin Hidalgo Cynthia Sustaita Fernando Marquez Lauren Eagan	Revised report based off of Ben's comments Restructure Use Case model level 2 Add Use Case Scenarios and descriptions Added event related needs and mapped to corresponding consideration
1.2.5	02/29/2020	Lauren Eagan	Added conclusion
1.2.6	02/29/2020	Cynthia Sustaita	Based on Ms. Elsa's feedback, fixed grammatical errors, added naming to

			comparison table, Added introduction sentences to sections in 1.4 Updated level 1 use case diagram Stated which needs are met for each consideration
1.2.7	02/29/2020	Cynthia Sustaita Jesus Gutierrez Lauren Eagan	Completed and updated document based on feedback.

Table of Contents

1. Introduction

In this document, we will discuss the system we are developing, as well as its constraints. This section will cover in depth the purpose of our feasibility report and what we are trying to achieve with it. Following that, we describe the justifications behind the considerations for the system that we will be researching. Additionally, the next section will define a thorough comprehension of all the requirements needed for the system. Finally, our use cases will be discussed to conclude this section, which covers our: use case model 1, use case model 2, the descriptions of the models, the description of actors in the model, and the scenarios for each model.

1.1 Purpose of the Feasibility Report

The purpose of this document is to document the result of a feasibility study, which is the analysis of all of the systems relevant components. These components consist of the technical feasibility, resource feasibility, financial feasibility, operational feasibility, and schedule feasibility. This assessment will take into account the needs of the client and determine potential solutions, as well as evaluate them to decide which to move forward with for further analysis. To begin, existing solutions that represent a solution to a majority of the requirements were researched in order to verify the feasibility of those implementations. In addition, the considerations of each need of the system were clarified to ensure they were accounted for. To add to this, possible solutions to these considerations were researched in depth. These solutions were then weighed against each other, with the advantages and disadvantages of each detailed out. Finally, the previous section will be used to pick the optimal solution.

1.2. Justification for the Proposed System

The purpose of the system is to provide a useful and efficient system for users to document and efficiently store findings in an organized manner in regards to cyber vulnerabilities experiments as well as provide mitigation recommendations. The system will aid a CEAD lead analyst to manage 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 in an efficient manner. Visual representations of analysts progress will be available to further improve the awareness of the lead analyst throughout an event. Motivation towards creating this project is to provide a useful and efficient system for users to document and efficiently store findings in an organized manner in regards to cyber vulnerabilities experiments as well as provide mitigations recommendations.

1.3 Requirements Definition

The following list contains the requirements [1] for the system that was gathered from the RDD [2], Team 11 interview report, Team 1 CS5385 interview report, memo, consolidated needs and considerations, and client's presentation provided by the clients.

1.3.1 Task Management

The following includes all the requirements regarding the management of tasks.

1. The lead analyst should have the ability to create a task.
2. The lead analyst should have the ability to assign a task
3. The lead analyst should have the ability to edit a task
4. The lead analyst should have the ability to leave a task unassigned.
5. The analysts should have the ability to prioritize tasks and subtasks.
6. The analysts should have the ability to pick their tasks.
7. The analysts will have the ability to sort their tasks.
8. The system will show a visual representation of tasks' progress.
9. The system should allow an analyst to promote a subtask to a task.
10. The system should allow an analyst to demote a task to a subtask
11. The system must allow tasks to be categorized by completeness (i.e. assigned to, in progress, completed, not do-able).
12. The analysts have the ability to search by criteria for a specific task

1.3.2 Data Storage

The following includes the requirements regarding data storage.

1. The system should automatically store completed events.
2. The analysts must have the ability to re-open old events.
3. The analysts should have the ability to duplicate old events.

1.3.3 Data Syncing

The following includes the requirements regarding data syncing.

1. Lead analysts must have the ability to schedule a syncing of data.
2. The analysts will have the ability to push data.
3. The analysts should have the ability to pull data.
4. System must allow analysts to transfer data wirelessly.

1.3.4 Report Generate System

The following includes the requirements regarding the generation of reports.

1. Lead analyst's have the ability to generate and export an ER-B report as a powerpoint.
2. Lead analyst's have the ability to generate and export a Risk Assessment report as an excel sheet.
3. Lead analysts should have the ability to generate a Final report as a word document.

1.3.5 Communication

The following includes the requirements regarding communication within the system.

1. Analysts' must have the ability to receive or send information between users through notifications (lead to analyst, analyst to lead, and analyst to analyst).
2. Analysts have the ability to work collaboratively between each other.
3. Analysts should have the ability to adjust schedule timing of notification.
4. Analysts' will have the ability to receive notifications in regards to scheduling.

1.3.6 User Management

The following includes the requirements regarding user authentication.

1. The system should prompt an analyst to log into their user profile upon entering the system.
2. The system must identify an analyst based on their IP addresses and initials.

1.3.7 Findings Management
The following includes the requirements regarding file management.

1. The analysts will have the ability to search by criteria for a specific finding.
2. The analysts should have the ability to create a finding.
3. The analysts should have the ability to edit a finding.
4. The analysts should have the ability to delete a finding.
5. The analysts should have the ability to attach a finding.
6. The analysts should have the ability to leave a finding unattached.
7. The analysts should have the ability to sort findings.
8. The analysts should have the ability to define a finding's category

1.3.8 Technical Components
The following includes the requirements regarding technical components of the system.

1. The system must be developed as a web application.
2. The system should have dual compatibility to be supported on both Linux and Windows.

1.4. Use Cases

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 facilitate our assumptions of the system to show the clients that way we can come to an agreement so when we finally deliver the system, it 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 [3]. The ideas being expressed in the Use Case diagram refers to behaviors of the system that is acted upon external actors.

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 [4] shown in Fig. 3. Association (also used in level 1) 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 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) [2].

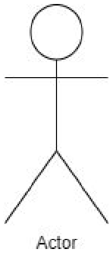


Fig 1: Actor notation

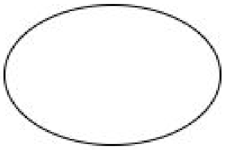


Fig 2: Use case notation

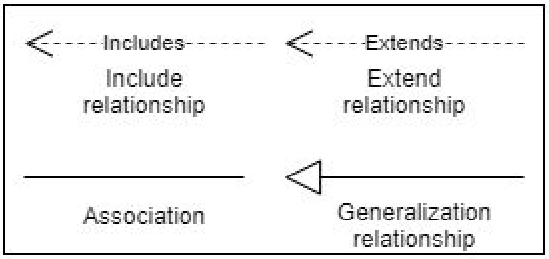


Fig 3: Relationship notation

1.4.1 Use Cases Model - Level 1:

The following figure (Fig. 4) represents a level 1 Use Case diagram to showcase the primarily use of the FRIC system:

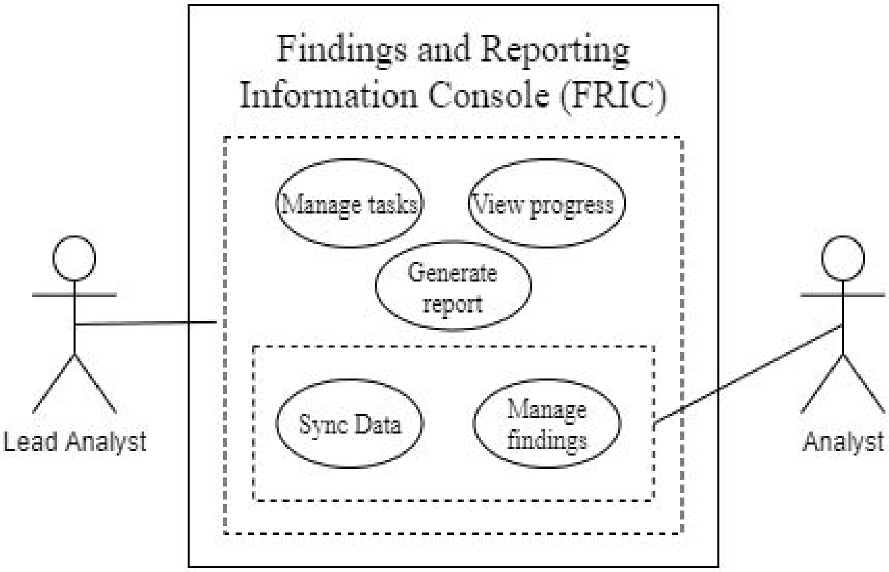


Fig 4: FRIC Level-1 Use Case diagram

1.4.2 Use Cases description - Level 1:

The following are detailed descriptions of our level 1 use cases: Manage tasks, manage findings, generate reports, sync data, and view progress.

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) 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.

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.

1.4.3 Actors Description - Level 1:

The following are detailed descriptions of our level 1 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.

1.4.4 Use Cases Scenarios - Level 1:

This section contains the detailed description of our level 1 Use Case diagram (Fig. 4) use case scenarios.

Use Case Scenario Name: Sync Data.

Description: The system allows all analysts to push data to the lead analyst or any other analyst and vice versa. If needed, an analyst or lead analyst can also pull data from another analyst to view their findings. .

Actors: Lead analyst, analyst.

Pre-condition: An analyst must have a task or subtask assigned and such assignment must have some kind of progress to push data for other analysts to see.

Trigger-condition: Analyst has findings they wish to share with their team.

Flow of events:

- Step 1: Actor: Selects option to push data.
- Step 2: System: Shows list of available IP's and initials to choose from.
- Step 3: Actor: Select an individual to share data with or sync to all.
- Step 4: System: Database pushes the information requested to be synced.
- Step 5: System: Displays confirmation the data has been synced.
- Step 5: End of use case

Alt: Actor: Selects to sync to all.

Use Case Scenario Name: Manage tasks

Description: The system allows a lead analyst to create/edit/delete 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.

Actors: Lead analyst

Pre-condition: A user has been assigned the position of lead analyst.

Trigger-condition: At the beginning of the week, an event begins and is sent to the lead analyst.
Flow of events:

- Step 1: Actor: Lead analyst creates a task based on the event's testing plan.
- Step 2: System: Shows list of available analysts to assign this task to.
- Step 3: Actor: Selects an analyst and assign task.
- Step 4: System: Displays confirmation the task was assigned
- Step 5: System: Notifies analyst they were assigned a task
- Step 5: System: Database maintains a record of analyst's work (E.g. task progress, findings, etc).
- 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) 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 successful or unsuccessful, findings, its results and current progress.
Actors: Lead analyst
Pre-condition: 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: Actor: Lead analyst select the option to generate report
- Step 2: System: Shows list of properties to include: All findings/select findings successful/unsuccessful, results, and current progress.
- Step 3: Actor: Selects all findings, current progress, and submits report.
- Step 4: System: Shows list of available IP's and initials to choose from.
- Step 5: Actor: Selects who they would like to send it to.
- Step 6: System: Generates report and displays confirmation to actor.
- Step : End of use case

Use Case Scenario Name: View progress
Description: 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.
Actors: Lead analyst
Pre-condition: A task has been assigned to an analyst.
Trigger-condition: Lead analyst wishes to view the current progress of the analyst who is assigned the task.
Flow of events:

- Step 1: Actor: Lead analyst selects option to view progress.
- Step 2: System: Displays list of current tasks and the 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 : End of use case

Use Case Scenario Name: Manage findings

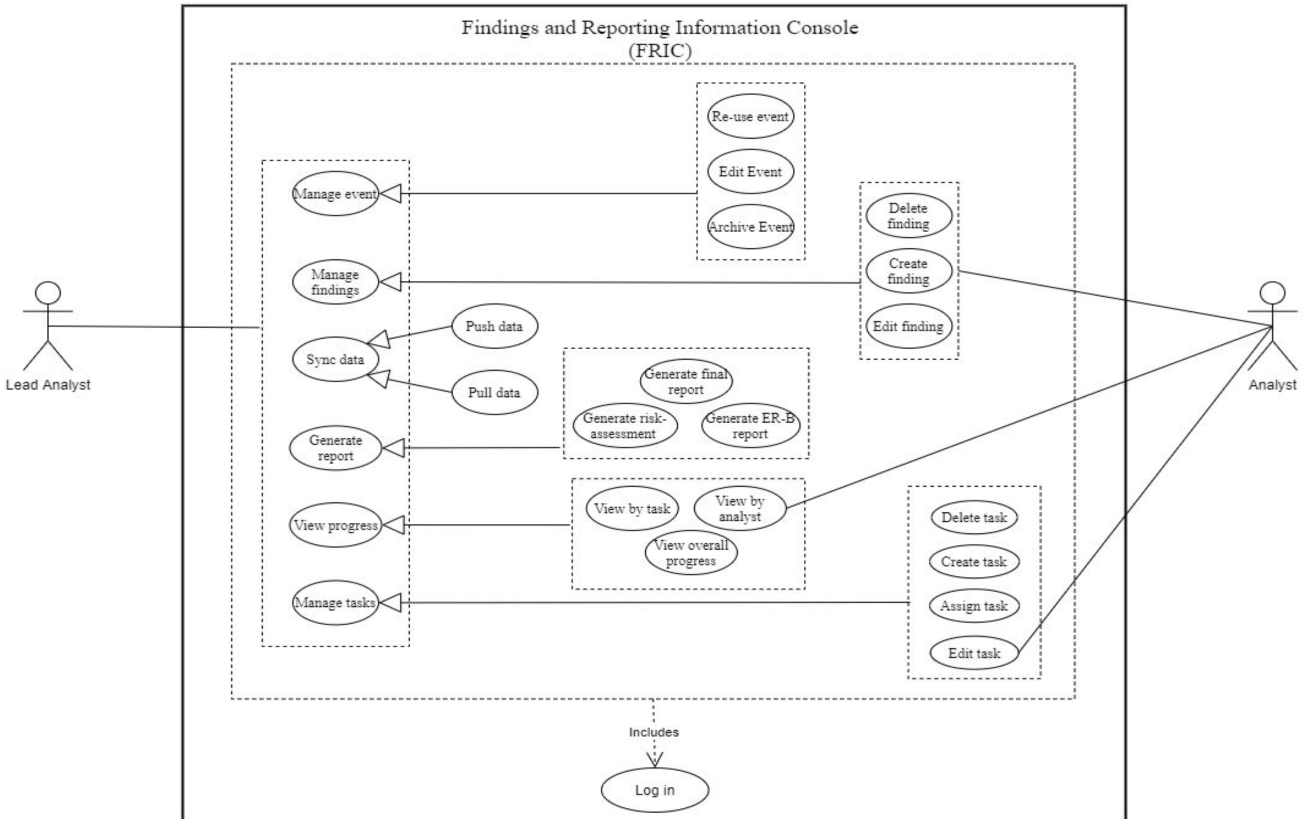
Description: The system allows any analyst to create/edit/delete a finding. Findings are to be tagged as vulnerable, informational or other. While managing findings, the analysts are allowed to edit a finding's attributes and search a finding by the type of tag it is or specific attribute. The system 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.
Actors: Lead analyst, analyst
Pre-condition: Lead analyst or analyst are assigned a task.
Trigger-condition: Lead analyst or analyst wishes to be able to create/optimize findings.
Flow of events:

- Step 1: Actor: Lead analyst or analyst selects edit/delete a finding.
- Step 2: System: Displays list of tagged findings, prompts to search a finding by a specific attribute.
- Step 3: Actor: Selects specific finding/s.
- Step 4: System: Displays information about that finding.
- Step 5: Actor: Lead analyst or analyst edits/deletes finding.
- Step 6: Actor: Selects completed.
- Step 7: System: Displays confirmation of finding edit/delete.
- Step : End of use case.

1.4.5   Use Cases Model - Level 2:

The following figure (Fig. 5) represents a level 2 Use Case diagram to showcase the primarily use of the FRIC system as well as the extension of the functionalities such as the include, extend, and generalization interactions between the actors and the system:

1.4.6 Use Cases description - Level 2:



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) 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.

1.4.7 Actors Description - Level 2:

The following are detailed descriptions of our level 1 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.

1.4.8 Use Cases Scenarios - Level 2:

This section contains the detailed description of our level 2 Use Case diagram (Fig. 5) use case scenarios:

Use Case Scenario Name: Create task
Description: The system allows the lead analyst to create a task under manage tasks. Creating the task is what usually the lead analysts know what to do before the event is started and ordered by the end-client.
Actors: Lead analyst
Pre-condition: A lead analyst has created an event.
Trigger-condition: Lead analyst initiates the task creation by selecting and pushes button, to create task
Flow of events:

- Step 1: Include <Analysts log-in>
- Step 2: Lead Analyst fills in information needed to describe the task (title, description, status, priority and due date).
- Step 3: System displays confirmation message.
- Step 4: Lead Analyst confirms task and clicks on the save button for this specific task.
- Step 5: System saves and updates the current task's under the system.
- Step 6: End of use case.

//Below assigning a task setup could go on the configuration page of the prototype
Use Case Scenario Name: Assign task
Description:. The system allows the lead analyst to assign a task after creating one. Assigning a task will give access to the user to update the task and see progress to it. Not always tasks get assigned, in alternative we all analyze to pick their own tasks.
Actors: Lead analyst, analyst
Pre-condition: A lead analyst has created a task.
Trigger-condition: A lead analyst creates a task and has to be done by people so you enter the info to create a task and now give assign tasks
Flow of events:

- Step 1: Include <Analysts log-in>
- Step 2: Lead A

Alt: Task doesn't need to be assigned to someone right away.

- Step .1 Lead Analyst can choose not to assign tasks.
- Step .2 System leaves the task assigned to section blank.
- Step .3 Lead Analyst can come back and assign this task to an analyst in the future.
- Step .4 System updates the current task's under the system.

Alt: Lead analyst allows analyst to choose their own tasks

- Step .1: Lead Analyst can choose not to assign tasks.
- Step .2: System leaves the task assigned to section blank.
- Step .3: Lead Analyst gives the ability to analyst(s) to pick their own tasks.

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 as a lead analyst, there are tasks in the system..
Trigger-condition: Lead analyst has noted any tasks that they wish to remove from the system to be deleted.
Flow of events:

- Step 1: Include <Analysts log-in>
- Step 2: System Displays tasks as a list under the manage tasks section.
- Step 3: Lead Analyst Selects the option to delete task(s).
- Step 4: System Gives delete option: delete task and all associated subtasks or findings, delete only subtask or finding, delete only task.
- Step 5: Actor Selects delete task and all associated subtasks or findings.
- Step 6: System displays confirmation message
- Step 7: Lead Analyst Archive unwanted task(s).
- Step 8: System updates the current task(s) under the system.
- Step 9: End of use case.

Use Case Scenario Name: Edit task
Description: The system allows the lead analyst and 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, analyst
Pre-condition: A user has logged in as a Lead Analyst or Analyst.
Trigger-condition: Lead analyst or Analyst wants to update the current task.
Flow of events:

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

Use Case Scenario Name: Delete finding
Description: The system allows any analyst to delete any findings that are under a task whether it was chosen or assigned to.
Pre-condition: An analyst must be assigned to the task or system to be given this permission
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: Include <Analysts log-in>

- Step 2: Any Analyst selects the option to manage findings.
- Step 3: Any Analyst selects the option to delete a finding.
- Step 4: Any Analyst selects the certain finding that is going to be removed.
- Step 5: System displays confirmation message
- Step 6: System updates the new system with the certain finding deleted.
- 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 or findings. 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 as a lead analyst or analyst
Trigger-condition: Lead analyst or analyst wants to create a new finding
Flow of events:

- Step 1: Include <Analysts log-in>
- Step 2: Lead Analyst/Analyst selects the option to create a new finding.
- Step 3: System gives reading and writing permission to lead analyst/ analyst.
- Step 4: Lead Analyst/Analyst: selects to add appropriate information towards the document.
- Step 5: Lead Analyst/Analyst submits new finding into the system.
- Step 6: System displays confirmation message.
- Step 7: End of use case.

//NOT NEEDED ANYMORE
Use Case Scenario Name: Relate to task
Description: Extension towards “Create Finding”. Once the new finding has been created, the system will allow for the lead analyst or analyst to relate that finding to a task.
Pre-condition: Lead analyst or analyst has created a finding
Trigger-condition: Lead analyst or analyst wants to relate that new finding to a task
Flow of events:

- Step 1: Include <Analysts log-in>
- Step 2: System displays list of findings created, or findings that are not assigned to a specific task(s).
- Step 3: Lead Analyst/Analyst: Select finding(s) to relate to a task(s).
- Step 4: Lead Analyst/Analyst: Select task(s) for specific finding(s).
- Step 5: System displays confirmation message.
- 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.
Pre-condition: An analyst must be assigned to the task or system to be given this permission
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 1: Include <Analysts log-in>
- Step 2: Any Analyst selects the option to manage findings.
- Step 3: Any Analyst selects the option to edit a finding.
- Step 4: Any Analyst selects the certain finding that is going to be edited.
- Step 5: System gives reading and writing permission to the user.
- Step 6: Any Analyst selects to re-write the previous text section.
- Step 7: Any Analyst confirms the changes and saves.
- Step 8: System displays confirmation message
- Step 9: System updates the new system with the new finding time stamp.
- Step 10: End of use case.

Use Case Scenario Name: Pull data

Description: The system allows any analyst to receive (pull) data that has been pushed by other analysts on the system that they are working on.

Pre-condition: Any analyst must have saved their progress and have data available to pull from other analysts.

Trigger-condition: Analyst when clicks on button to pull data. A change must have been made to the overall main project since last time clicking on, pull data button.

Flow of events:

- Step 1: Include <Analysts log-in>
- Step 2: System prompts confirmation message to pull data from all analysts.
- Step 3: System prepares available data to be received by the who is pulling data.
- Step 4: System shows the differences between the current data on local computer and overall data from the main project.
- Step 5: Analyst looks at differences made between their data and data they are about to pull and selects manually selects what differences they want to keep.
- Step 6: System updates the local system with the new updates and time stamps changes.
- Step 7: End of use case.

Alt:

- Step 2.1: Analyst prompts confirmation message to pull data from selected specific group (just analyst).
- Step 2.2: System prepares available data based on the group of people analyst wants to receive data from (just analyst).
- Step 2.3: System shows the differences between the current data on local computer and specified group's project data.

Use Case Scenario Name: Push data

Description: The system allows any analyst to upload (push) their own data that has been made on their local computer to the main projects for other people to recieve.

Pre-condition: Any analyst must have saved their progress and have data updated since the last time they pushed.

Trigger-condition: Analyst when clicks on button to push data. A change must have been made to the overall main project since last time clicking on, push data button.

Flow of events:

- Step 1: Include <Analysts log-in>
- Step 2: Any Analyst selects to share information with all analysts.
- Step 3: System prompts confirmation message to push data.
- Step 4: System prepares data to be received by the main project.
- Step 5: System main project receives updated data and merges with previous main project.
- Step 6: System updates the main project with the new changes and time stamps changes.
- Step 7: System makes new updated main project available to pull for selected group of analysts.
- Step 8: System gives out notification to selected group, that this analyst has pushed new data.
- Step: End of use case.

Alt:

- Step 2.1: Analyst selects specific group (just analyst) to push data.
- Step 2.2: System recognizes option chosen to make data available only to certain group selected.
- Step 2.3: System based on chosen group selected, it gives the updated data from local computer available to be pulled.

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 would 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: 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: Include <Analysts log-in>
- Step 2: Lead analyst select the option to generate report.
- Step 3: System shows list of properties to include: All findings/select findings successful/unsuccessful, results, and current progress.
- Step 4: Lead analyst selects all findings, current progress, and submits report.
- Step 5: System shows list of available IP's and initials to choose from.
- Step 6: Lead analyst elects who they would like to send it to.
- Step 7: System generates report and displays confirmation message.
- Step 8: End of use case.

Use Case Scenario Name: View progress
Description: 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.
Actors: Lead analyst
Pre-condition: A task has been assigned to an analyst.
Trigger-condition: Lead analyst wishes to view the current progress of the analyst who is assigned the task.
Flow of events:

- Step 1: Include <Analysts log-in>

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

Use Case Scenario Name: Re-use event
Description: The system allows a lead analyst to reuse previous events that falls under the manage event. Here you can pull any event and reuse archived data.
Actors: Lead analyst
Pre-condition: A lead analyst must be logged into a lead analyst
Trigger-condition: Lead analyst must click on manage event and pick a specific event to reuse
Flow of events:

- Step 1: Include <Analysts log-in>
- Step 2: System gives reading and writing permission to the user
- Step 3: Lead Analyst selects
- Step 4: Lead Analyst reuses the finding or task
- Step 5: Lead Analyst can edit or archive event
- Step 6: Lead Analyst saves changes and confirms any changes done to previous versions of findings or task
- Step 7: System displays confirmation message
- Step 8: System will update previous version with new timestamp and initial of who changed previous version
- Step 9: End of use case

Alt:

- Step 5.1: Lead analyst archives event
- Step 5.2: End of use case

Use Case Scenario Name: Edit event
Description: The system allows a lead analyst to edit any previous event that falls under manage event. Here you can view and edit anything under the event but will time-stamp the most updated version.
Pre-condition: A lead analyst must be logged into a lead analyst
Trigger-condition: Lead analyst must click on manage event and pick a specific event to edit/view.
Flow of events:

- Step 1: Include <Analysts log-in>
- Step 2: System retrieves the event from the database
- Step 3: System gives reading and writing permission to the user.
- Step 4: Lead Analyst selects certain pieces of event or data within event like task or finding
- Step 5: Lead Analyst edits the finding
- Step 6: Lead Analyst clicks the save button and confirms the changes and saves
- Step 7: System displays confirmation message
- Step 8: System updates the piece of data with a new timestamp and initial of who is changing the data artifact
- Step 9: End of use case.

Alt:

- Step 5.1: Lead analyst just views the certain event or data artifact with no time-stamp update
- Step 5.2: Lead analyst exits the certain event
- Step 5.3: End of use case.

Use Case Scenario Name: Archive event
Description: The system keeps a history of completed events to allow them to be viewed and/or reused.
Pre-condition: A lead analyst must be logged in as a lead analyst
Trigger-condition: Lead analyst must click on manage event and pick a specific event to edit/view.
Flow of events:

- Step 1: Include <Analysts log-in>
- Step 2: System gives reading and writing permission to the user
- Step 3: Lead Analyst chooses to conclude the event
- Step 4: System displays option to archive an event
- Step 5: Lead Analyst clicks archive completed event
- Step 6: System displays confirmation message of archived event
- Step 7: System displays a confirmation message
- Step 8: End of use case

2. Existing Solutions

This section contains the needs of an occuring system that exists as a possible solution for what our system is trying to implement. We will use all sources of information that were given to determine an, a-like software program or system that closely resembles our system that currently exists as an existing solution.

2.1. Existing Solution 1

SureSync (Interview report - Appendix B - 27:01)

Syncing data and collaborations with other users is a key factor to our system. A lead analyst and analyst need to be able to sync report(s), finding(s) and data using a system such as a push/pull. Software Pursuits inc. is an existing solution for our system because it replicates a push and pull system in either a one way or multi-way synchronization. An example from our system is that analysts should be able to pull data at any point from other analysts to view progress of task(s), finding(s) and/or report(s), hence the need to constantly synchronize data. An analyst should be able to push data at any point to other analysts to sync their data, finding(s) and/or report(s) and vice versa. Additionally Software Pursuits gives the ability to allow users to collaborate between other users using SureSync as well as the ability to assign certain tasks to users as well as the monitoring of the data being handled, meaning that progress of tasks could be easily visualized. SureSync is a product from Software Pursuits that offers compatibility to either OS and/or web applications plus a one way push and pull system, job progress/status and other additional features for user collaborations. This solution handles backups making use of automated SQL queries. This existing solution meets the needs under the following sections: 1.3.1, 1.3.2, 1.3.6, 1.3.7, 1.3.8

2.2. Existing Solution 2

GitLab:

GitLab is a platform that supports both project planning and source code management [5]. Developers have the ability to create repositories that can contain both documentation and code. GitLab allows users to collaborate on the same repo, meaning that certain tasks can be assigned to specific users with certain restrictions. These repositories can be edited and deleted after creation as well as searched by specific criteria. Additionally, GitLab counts with a storage system that counts with the ability to look at old repositories as well as clone them. Since it counts with code management features, developers have the ability to both push and pull code. GitLab also counts with a subgroup features that provides a hierarchy of behaviors, meaning that certain projects have either a bigger priority or significance than others. Developers that make use of this platform can work in different environments such as Windows or Linux. The following needs are met through 1.3.1, 1.3.2, 1.3.3, 1.3.6, 1.3.7, and 1.3.8.

2.3. Existing Solution 3

Intel Bucket (Interview report - Appendix C Recording #4 - 9:25)

Intel Bucket is one existing solution that we obtained from the interview report. All the data in the projects like the tasks, attachments and findings can be stored in to the hard-drive which as stated in the interview report, hard drives do not have a long lifetime. Intel Bucket is the way to maintain past projects and keep reimaging hard-drives. The way the CEAD currently works is that all analysts sync with each other first. Then once synced, all analysts continuously push and sync their data to leads before terminating the event. After, the lead has the whole project saved into their own hard-drive and then the lead's copy of the hard-drive gets uploaded into a secure system. This existing solution meets the needs detailed in: 1.3.1, 1.3.2, 1.3.3, 1.3.5, 1.3.6, 1.3.7, 1.3.8..

3. Consideration

This section contains the different aspects to be considered to develop the proposed system. Each of these considerations is explained detailing why it is necessary as a major component of the system as well as external application or services represented as options. Additionally, each option has a detailed description in addition to a list of advantages and disadvantages of making use of such options in regards to our requirements.

3.1. Database

A database is an easily accessible collection of data, stored for use at a later time. This consideration is mapped to the needs under Data Storage: 1.3.2. The options for this considerations are as follows: relational or non-relational database. Option 3.1.1 meets needs 1.3.2.1 and 1.3.2.2 while option 3.1.2 meets needs 1.3.2.2 and 1.3.2.3

3.1.1. Relational database - MySQL

A relational database stores and provides access to data related to another set of data. As its names implies, a relational database is based on a relational model in which data attributes are stored in columns and data records are stored in rows [7]. This structure facilitates the relation of one set of data to another.

Pros:

- Automated queries can be performed by controlled systems.
- Multiple systems can interface with the same database.
- MySQL is an open source relational database

Cons:

- Complex relational database systems can lead to isolated databases where the information cannot be shared from one system to another.
- Relational databases require all data to be structured due to their schema.
- Its queries could easily lead to injection attacks.

3.1.2. Non-relational database - MongoDB

A non-relational database does not follow a relational model in contrast to a relational database [6]. This type of database does not specifically make use of relationships within sets of data.

Pros:

- There’s no need to follow a specific schema design.
- Non-relational databases handle unstructured data
- Non-relational databases are open source most of the time, therefore there won’t be a need to worry about a license.
- Less risk of injection attacks due to their schema-free structure.

- MongoDB is an open source non-relational database.

Cons:

- If there's a need to relate one set of data to another, data would have to be "joined" manually.
- Non-relational databases are fairly new in comparison to relational-databases, therefore resources and integration tools are more scarce.

3.2. Data syncing

For the system provided, the system uses data syncing to communicate between analysts. This allows an analyst to collaborate with other analysts to view certain aspects of the system. This consideration is addressed in section 1.3.3 Data Syncing. The consideration is available on more platforms and can remotely access data. Option 3.2.1 meets all the needs under section 1.3.3 Option 3.2.2 meets needs 1.3.3.2 and 1.3.3.3.

3.2.1. GoodSync

GoodSync is a file server system that offers different services on multiple platforms [8]. It offers a secure file transfer and syncing between users. It also offers features file backup and runs as a background service with no user interaction.

Pros:

- This tool allows the transfer of real time data, meaning that changes are able to be seen immediately.
- Offers the ability to backup files.
- Supports end-to-end encryption, ensuring security of data.
- Team members have a supporting version of history control, that way they can see all past updates and contributions.
- This tool provides remote file access, meaning that you can access the files anywhere at anytime.
- One way or two way synchronization across multiple locations and platforms

Cons:

- When doing File Transfer Protocol (FTP), chances of security breach is highly likely.

3.2.2. PieSync

PieSync is a real-time intelligent customer data synchronization platform. [9] This platform supports control of data flow as it provides real time delivery, meaning that collection of data is immediately reflected.

Pros:

- Gives an unlimited amount of user team members that can be added.
- It has a high amount of customization allowed for dashboard and synchronization.
- PieSync is an open source platform.
- Has a feature which automatically keeps track of API changes.

Cons:

- Only a two-way integration of data, meaning it is not possible to integrate with a third service.
- This tool is not flexible in customization.

3.3. Project management system

A data management system would facilitate the managing of data for the system's users. This consideration is mapped to the needs under 1.3.1 Task Management and 1.3.7 Findings Management. This consideration could be Tuleap or Taiga. Both of these options make use of scrum, which is a set or practices of communication and flexibility in order to carry out a project. Option 3.3.1 meets needs all of the needs under 1.3.1 and 1.3.7. Option 3.3.2 meets most of the needs under sections 1.3.1 and 1.3.7, except for needs 1.3.1.3 and 1.3.7.5

3.3.1. Tuleap

Tuleap is a General Public License scrum platform used to host software projects. This platform is widely used by teams to track the lifecycle of their software project [10]. This platform supports project management, source control, code review, continuous integration, and documentation.

Pros:

- Tuleap offers an end-to-end traceability, meaning that deliverables can be associated with each other.
- Tuleap is an open source tool.
- Offers user profile management that allows users to see everyone's tasks.
- Host multiple workspaces.
- Allows customization of the features this tool provides, meaning that users are allowed to modify what features to use or not .
- Scalability and robustness.
- Allows the creation, deletion, and edition of projects.

Cons:

- External integration is difficult to obtain (E.g. github).
- Tools such as documentation and communication are not as developed as others (E.g. code review).

3.3.2. Taiga

Taiga is a project management platform that makes use of scrum and kanban to make their project agile. Kanban is a board used to keep track of project's progress, notes, and specific details [11]. This platform is also used by teams to track the lifecycle of their software project.

Pros:

- Taiga is an open source platform.
- User profile management that allows users to see everyone's tasks.

- Allows the creation, deletion, and edition of projects.
- Allows customization of the environment.
- Allows the creation, deletion, and edition of projects.

Cons:

- Big learning curve as it requires users to be familiar with scrum methodology
- Restrictive free tier service
- Non-scalable as it only allows certain amount of users per project

3.4. Communication system

This section will be about the communication system needed for the analyst to communicate within the system allowing them to ultimately work more collaboratively. This consideration section is for the communication system. It addresses needs under, Communication 1.3.5 on this report. The considerations will be different API on how they better fit what the system requires. Option 3.4.1 meets all the needs under section 1.3.5. Option 3.4.2 meets need 1.3.5.1.

3.4.1. ServiceNow

ServiceNow is a software that is considered to help certain people to accomplish tasks and manage groups of people [12]. It specifically targets employee's, customer and IT workflows. In general, this software program is designed to help facilitate task's, activities and processes.

Pros:

- Able to adjust notifications based on due-date.
- Able to send/receive notifications based on your placement of the scope of the project (lead to analyst, analyst to lead, and analyst to analyst).
- Ability to adjust the due date/ notification timing of existing due dates
- Ability to receive notifications in regards to scheduling. Also will allow the other person to select a time/data scheduling that doesn't already conflict with your existing schedule.

Cons:

- Could be difficult to implement and not python related language
- Could be a security threat depending on the level of securence needed for the system

3.4.2. Twilio

Twilio is a great online communication system to send chats, notifications and videos to other users using the system. [13] This API is able to be used with the web and mobile platforms. Twilio is very flexible due to the fact that it gives the power to the developers to build their custom communication system for their own needs.

Pros:

- Send push notifications through one single API on web

- Opt into specific channels or groups that you want to notify (lead to analyst, analyst to lead, and analyst to analyst)
- Flexibility to receive notifications different ways (web, email, sms)

Cons:

- Must be used with either web, Android or IOS apps
- Cannot adjust the timing of notifications
- Offers trial but is not free to use

3.5. User Management System

The user management system's purpose is to differentiate between the lead analyst and the analyst profile. Each user profile has a set of privileges that need to be verified are being applied to the correct user. This consideration would address the considerations part of section 1.3.6 User Management. Option 3.5.1 meets all needs under section 1.3.6. Option 3.5.2 meets need 1.3.6.1.

3.5.1. Auth0

Auth0 is a flexible solution that adds authentication and profile management to applications [14]. It gives convenience and security to the process of adding user management to a system.

Pros:

- It generates logs to track user interaction.
- Provides pre-built authentication methods but also allows the customization of authentication methods.
- Integration process is thorough but easy.

Cons:

- Customization features can be unclear and not intuitive to set up.
- It can be limited when it comes to how much you want to customize your authentication process.

3.5.2. Okta

Okta is an Identity management API for software teams building systems. It is a cloud-based user data store with a private deployment option [15]. It would be a good choice for using as a user management system, because it is easy to manage the data as well includes pre-built roles. Although, it does detract from usability as it can have many unneeded features.

Pros:

- Easily model and manage data, including pre-built roles.
- Faster speed to market for application development and launch.
- Lower cost of application development and maintenance.
- Easily integrates with tools.
- Multiple levels of administrative rights allow organizations to distribute responsibilities to different teams and allowing them to assist in the administration.

Cons:

- Some reports don’t reflect current user assignments.
- Not intuitive for set up.
- Can have too many unneeded features.

3.6. Report generation system

The report generation system’s purpose is to take data from a source such as a database or an excel sheet, and with this data use it to produce a document in such a format that will satisfy particular human readership, either in word, excel, or powerpoint format. This consideration would address the needs under Report Generate System: 1.3.4. Both Option 3.6.1 and option 3.6.2 meet all needs under section 1.3.4

3.6.1. ZohoAnalytics

ZohoAnalytics is used to embed, scale, and also monetize self-service reports [16]. This tool has a private pulling and pushing option for the lead analyst to easily generate and blend given data into one single report. ZohoAnalytics is an easy-to-use tool that provides the option to analyze performance by given metrics of current lead analyst.

Pros:

- Get and blend data from multiple sources to create cross functional reports.
- Allows customized visualization of data.
- It features an AI-powered assistant that allows users to ask questions and get intelligent answers in the form of meaningful reports.
- Embedded analytics.
- Tool provides smart data alerts and forecasting.
- Analyze performance by the given metrics.

Cons:

- Customization features can be unclear and not intuitive to set up.
- Could limit customization of excel reports.

3.6.2. Izenda

Izenda is a flexible solution in analyzing all types of data by using a dashboard. [17] Izenda also has the ability to generate and export risk assessment reports, ER-B reports, and even final reports which can be accessed through the user’s account.

Pros:

- Embed, scale, and monetize self-service reporting.
- Data is accessible from any device by remote access.
- Flexible, meaning it allows izenda to be scaled alongside our application.
- Izenda gives the ability for users to self-service on every report.
- Inherits our platform’s security model to maintain compliance with any privacy or security requirements.
- Flexibly analyze data using charts, graphs, and filters.

Cons:

- Offers trial but not free to use.
- Could limit customization of excel reports.

4. Solutions

This section will contain proposed solutions based on different combinations of considerations' options provided in section 3. Each will further elaborate which requirements are met as well as requirements not met based on the features each of these options provide.

4.1. Solution 1

Non-relational DB, GoodSync, Tuleap, ServiceNow, Auth0, Izenda:

Non-relational DB: The reasoning behind making use of a non-relational database falls under data structure. Requirements 1.3.2.2 and 1.3.2.3 are met since a non-relational database allows users to see documents that have been uploaded to the database: ability to re-open old events. These documents would contain tasks and finding's details as JSON objects making it, making readability not an issue. Additionally, a non relational database allows duplicate documents, therefore analysts would be able to re-create and upload old events. On the other hand, this type of database does not support automatic savings, therefore analysts would have to manually query to the database. Thus requirement 1.3.2.1 is not met.

Goodsync: The main reason GoodSync is considered is that it has a one way or two way push and pull system which is a key feature for this system. The requirement 1.3.3.2 and 1.3.3.3 is met since the system would allow to sync data between users in either direction. Requirement 1.3.3.1 is met since the system allows automated, scheduled and real time backup and synchronization without any user interaction . Lastly requirement 1.3.3.4 is also met since the service is accessible on a wide variety of platforms. The user can access the data store on any device wirelessly or transmit data to the data store securely with end to end encryption.

Tuleap: Requirements under 1.3.1 and 1.3.7 are met with the use of this tool in addition to Auth0 in order for the system to identify which users can do certain actions. By differentiating them, Tuleap would allow a lead analyst to manage tasks and both lead and analyst to manage findings by providing an option of creation, edition, and deletion of a task and/or finding. Since Tuleap has the ability to associate sets of data, this deliverable also allows a lead analyst to assign a task to a specific user, in this case, an analyst as well as allowing either a finding to be attached or unattached to a task. Additionally, this task can also be left unassigned for analysts to be able pick their own tasks. Tuleap also provides a hierarchical structure that can represent the priority of a task as well as the promotion or demotion of a task to subtask or vice versa. This hierarchical structure also provides the ability to sort tasks and findings by a specific attribute as well as searching since it allows users to customize the environment, therefore both lead and analyst can customize these settings based on a preferred criteria. Lastly, Tuleap provides visual representation of progress through bars allowing the visualization of each task's progress.

ServiceNow is the solution we considered to include due to the fact that it satisfies more requirements than my other option. It has the ability to have communication and help to manage customer needs for IT, customers and employees to manage tasks and activities satisfying requirement 1.3.5.2. Specifically it also satisfied the need stated in the document in reference to 1.3.5.1. ServiceNow has the capability to send and receive notifications inside a certain scope for example, it can send it just for the lead, or just the analyst or both to benefit our system. Lastly, it satisfies the requirements 1.3.5.3 and 1.3.5.4 to schedule a set time to push a notification to other users and adjust when that notification is scheduled to be set in case the user wants to make a change.

Auth0: The reasoning behind using this user management option is it is the ideal option to satisfy the requirements listed in 1.3.6.1 and 1.3.6.2. The system needs to prompt an analyst to log into their user profile to differentiate between the roles. Once logged in to the system, the profile chosen will be used to take the analysts' IP addresses and initials to keep track of each analyst's contributions. Auth0 will allow for the user login to be customized so that we are just verify what role the person is using, as well as their IP address and initials. Additionally, it comes with the feature to keep logs of each users' interactions.

Izenda: This option is ideal because the tool allows Izenda to be scaled alongside the application and generates a final report regarding the merged data under 1.3.4.3 by giving the lead analyst real time data and self service analytics. Another need that this tool meets is the requirement under 1.3.4.2, where the lead analyst is allowed to blend all new gathered data then generate and export a risk assessment report by using a secure login system where they can then adjust specific functions, create groupings, format reports, set up filters, create pivot tables, and add data visualization. Lastly, the requirement under 1.3.4.1, the lead analyst could also generate and export a ER-B report from captured/merged data by following the easy-to-use tool Izenda.

The combination of these tools meet the needs under section 1.3.8 as they all are compatible with both Windows and Linux OS and are capable of being integrated in a web application.

4.2. Solution 2

Relational DB:MySQL, PieSync, Taiga, Twilio, Okta, ZohoAnalytics:

Relational DB: MySQL: The use of a relational database would allow the system to automatically store events as soon as an analyst marks them as completed as this approach allows automatic MySQL queries to make an update, therefore need 1.3.2.1 is met. Additionally, this database would also provide the ability to open old events through specific queries, fulfilling need 1.3.2.2, however, the attributes must be set through a specific schema, On the other hand, need 1.3.2.3 is not met as a relational database does not allow duplicate primary keys, therefore a replica of an event cannot be done.

PieSync: The use of PieSync is considered for this system because it offers the flexibility to synchronize either one way or two way between companies which would satisfy requirements 1.3.3.2 and 1.3.3.3. Additionally, it also offers the ability to personalize who to synchronize with and update users if any updates have been made to the database. Requirements 1.3.3.1 and 1.3.3.4 unfortunately are not met since one requirement can't schedule when to sync data and the other can only sync data through supported apps.

Taiga: The majority of requirements under 1.3.1 and 1.3.7 are met with the use of this tool. Taiga counts with an admin system that would allow the system to differentiate who's a lead and who's a non-lead. Thanks to this, the restriction of certain actions are taken care of. Taiga would allow a lead analyst to manage tasks and both lead and analyst to manage findings by providing an option of creation, edition, and deletion of a task and/or finding. Taiga unfortunately does not count with association of projects, meaning that a lead cannot directly assign a task to an analyst nor a finding to be attached or unattached to a task, therefore requirements 1.3.1.3 and 1.3.7.5 are unfulfilled. This tool also counts with hierarchical structure that can represent the priority of a task as well as the promotion or demotion of a task to subtask or vice versa. This hierarchical structure also provides the ability to sort tasks and findings by a specific attribute as well as searching since it allows users with expertise to customize the environment, therefore both lead and analyst can customize these settings based on a preferred criteria. Lastly, Taiga provides visual representation of progress that not only would provide a task progress, but an overall progress of each analyst. Although this tool fulfills a big amount of requirements, it only allows certain number of users per project, putting a constraint on collaborators.

Twilio is the next option we chose due to the fact that it has less amount of needs than our other option. We still consider Twilio as a solution due to the fact that it satisfies requirement 1.3.5.1 to send and push notifications to others users in the system. This notification system is allowed to send to different people and groups of varieties of the scope so not everyone gets bombarded with notifications. On the other hand this solution given is not able to satisfy requirement 1.3.5.2 to work together. Even though you are able to communicate with each other, you are not able to sync information and progress of tasks through this system. Also requirements 1.3.5.3 and 1.3.5.4 do not satisfy the needs of scheduling a certain notification push to other users. Due to this statement, you are not allowed to adjust the timing of the schedule notification push to notify the lead analyst in our system.

Okta: This software is a good fit for the system, as it meets one of the requirements needed. Requirement 1.3.6.1 is met by the function in Okta which provides a login feature to the system. It gives a secure way for the analysts to verify which role they will be using. Unfortunately, it does not meet the other requirement 1.3.6.2 needed in user management. Which needs the functionality of keeping track of an analysts' contributions to the system via their IP address and initials.

ZohoAnalytics can obtain and blend data in order to combine and generate the given data. The lead analyst has the ability to not only generate but also export risk assessment reports and final reports by using the login feature that the system provides and entering the adequate data into the tool where it then prompts the lead analyst if they would like to generate either a risk assessment report or a final report. Furthermore, this tool meets the requirements discussed in section 1.3.4. Fortunately, this tool also covers the requirement in section 1.3.4.1, which includes generating and exporting a ER-B report following a similar process such as requirements 1.3.4.2 and 1.3.4.3.

The combination of these tools meet the needs under section 1.3.8 as they all are compatible with both Windows and Linux OS and are capable of being integrated in a web application.

5. Comparison of Solutions

The purpose of this section is to weigh the differences between the previously proposed solutions in Section 4 and identify which of these solutions would fulfill as many requirements as possible. The following table (Fig. 6) demonstrates such comparison of weigh:

Fig. 6: Comparison matrix of proposed solution’s features	**Solution 1** *Non-relational DB, GoodSync, Tuleap, ServiceNow, Auth0, Izenda*	**Solution 2** *Relational DB - MySQL, PieSync, Taiga, Twilio, Okta, ZohoAnalytics*
Specific hardware and software requirements	This proposed solution meets the specific hardware and software requirements that have been decided until this point, which is OS compatibility.	This proposed solution supports the required hardware and software requirements that have been decided.
Time constraints	This project is expected to be delivered by the end of this year. Goodsync requires a license after a 30-day trial, therefore this tool would put a constraint. Additionally, ServiceNow’s services are limited after a demo trial.	Okta has a pricing after a 30-day trial as well as Twilio’s services.
Ease of use	The majority of the components that make up this proposed solution are user friendly except for Goodsync as it lacks customization,	ZohoAnalytics is not user friendly as it lacks customization and has too many features that can be unneeded just as Okta does.
Staffing levels and training required	Goodsync’s steps of installation take to set up, but the rest of this proposed solution’s component don’t require additional training to or expertise to make use of them.	Twilio’s environment can be customized, but it would require developers to build it from scratch, therefore experience is required. Additionally Taiga requires its users to be somehow familiar with scrum methodology. Okta’s and ZohoAnalytics has additional features that can make them no intuitive to use.
User preference	This proposed solution provides a better usability of this system	This proposed solution may lead to less usability of the system, as

	as these options provide features that can make it an intuitive one to build.	a few of the options are known for including too many features, making it less intuitive to build or use in the long term.
Security issues	This system will be dealing with sensitive data and this proposed solution does not introduce security risks.	A relational database is prone to injection attacks rather than a non-relational database, but the rest of this proposed solution's component are secure

6. Conclusions

The key elements comprising our system will be: database, data syncing, project management, communication, user management, and report generation. Throughout this report we have demonstrated an exhaustive list of the requirements needed for this project, how they relate to each other and how these components will fulfill the considerations of our system. Following that, we researched and found not only existing solutions to the system need, but individual options for each consideration that can be used to meet them. Further analysis was needed to determine which of these options would be feasible for the project. In order to conduct this analysis we took the options that worked together best and came up with two possible solutions, which were then compared with each other using certain standards. These standards are: specific hardware and software requirements, time constraints, ease of use, staffing levels and training required, user preference, and security issues (Fig. 6).

Using our comparison, we were able to determine that in order to achieve the best feasibility level for our system, solution one is the optimal choice. A few of our reasons include: it meets more of the needs than solution two, it supports multiple platforms, the comparison of standards tables shows it is more suitable and has more benefits, it has more detailed notification settings and allowances and lastly it gives more flexibility to customize what we prefer as we build our system. Overall, we believe we have achieved a demonstration of feasibility for our plan that will lead us into success with this project.

7. References

[1] C. V. Sustaita, L. Eagan, F. Marquez, J. Hidalgo, and J. Gutierrez, “Interview Report Team 11.”

[2] O. Perez, H. Vasquez, T. Provencio, J. Rivers, A. Cuevas, and V. Fonseca, “RDD - Findings and Reporting Information Console (FRIC).” .

[3] 'usecase_part1.pptx', The University of Texas at El Paso, 2020.

[4] 'usecase_part2.pptx', The University of Texas at El Paso, 2020.

[5] GitLab. (2020). About GitLab. [online] Available at: https://about.gitlab.com/ [Accessed 29 Feb. 2020].

[6] “The most popular database for modern apps,” *MongoDB*. [Online]. Available: https://www.mongodb.com/. [Accessed: 28-Feb-2020].

[7] J. Homan, “Relational vs non-relational databases,” *Pluralsight*, 30-Jan-2020. [Online]. Available: https://www.pluralsight.com/blog/software-development/relational-non-relational-databases. [Accessed: 28-Feb-2020].

[8] “Complete control and protection in one solution,” *GoodSync Enterprise | Data Backup, Synchronization, and Recovery Software Design for Business*. [Online]. Available: https://www.goodsync.com/business-file-server. [Accessed: 28-Feb-2020].

[9] “Sync contacts between your favorite cloud applications,” *PieSync*. [Online]. Available: https://www.piesync.com/. [Accessed: 28-Feb-2020].

[10] N. Tiwari, “Get to know Tuleap for project management,” *Opensource.com*. [Online]. Available: https://opensource.com/article/17/1/interview-Tuleap-project. [Accessed: 28-Feb-2020].

[11] P. Ruiz-Múzquiz, “Taiga.io,” *Taiga.io*, 22-Oct-2019. [Online]. Available: https://taiga.io/. [Accessed: 28-Feb-2020].

[12] *Push notifications*. [Online]. Available: https://docs.servicenow.com/bundle/orlando-servicenow-platform/page/administer/notification/concept/c_PushNotifications.html. [Accessed: 28-Feb-2020].

[13] “Twilio Notify: SMS, Push, and Chat Notifications,” *Twilio*. [Online]. Available: https://www.twilio.com/notify. [Accessed: 28-Feb-2020].

[14] “User Management,” *Auth0*. [Online]. Available: https://auth0.com/user-management. [Accessed: 28-Feb-2020].

[15] “Home,” *Okta*, 07-Feb-2020. [Online]. Available: https://developer.okta.com/. [Accessed: 28-Feb-2020].

[16] “BI & Analytics Software: Online Reporting Tool – Zoho Analytics,” *Zoho*. [Online]. Available: https://www.zoho.com/analytics/?utm_source=STH&utm_medium=Website&utm_campaign=ReportingTools. [Accessed: 28-Feb-2020].

[17] “Embedded BI & Analytics Platform,” *Izenda*. [Online]. Available: https://www.izenda.com/. [Accessed: 28-Feb-2020].

$

Feasibility Report Grading Criteria

Grade: 98.5

68.5 pts.

- **Complete 70**

Title Page (1/1)

Document Control (1/1)

Table of Contents (1/1)

Introduction (5/5)

- Purpose of the feasibility report (1/1)
- Justification for the proposed system (2/2)
- Summary of Requirements Definition (2/2)

Use Cases Diagram (13.5/15)

- Level 1 and Level 2 Use Case Diagram (5.5/6)
- Description of actors (2/2)
- Description of use cases (2/2)
- Use case scenarios (4/5)

Possible Solutions (40)

- Existing Solutions (5/5)
- Considerations (10/10)
    - Solutions (at least two present) (10/10)
    - Comparison of Solutions, including Comparison Matrix (10/10)
        - Specific hardware and software requirements
        - Time constraints
        - Ease of use
        - Staffing levels and training required
        - Client/User preference
        - Security issues
    - Sufficient detail is given to show understanding of and provide justification for a proposed solution (5/5)

Conclusions (5/5)

- Summary and recommendations

References (2/2)

20 pts.	• **Correct & Consistent - 20/20** • Information presented is referenced • Alternatives are truly feasible ways of meeting the requirements • Correct and consistent with the requirements definition and interview • Comparisons use the same criteria
10 pts.	• **Presentation 10/10** • Clear and concise • Correct grammar Professional presentation