Questions:

- General questions:
- What are the possible threats to this company?
- Have you had an organizational cybersecurity system before? If yes, what were some things you liked about it and what were some things that needed improvement.
- Are there onsite security measures for the actual building?

System Requirements (in our system):

Design organizational system that addresses:

- Access control
    - How many people need to access their system on a day to day basis?
        - 2000-3000 people
    - How often are employees required to update their passwords?
        - No requirement, but from prior knowledge 90 days is the standard in general
    - Are employees required to use MFA?
        - No, but there will be an option to use it among account set up
    - Does everyone have access to everything or do certain people have access to certain information?
        - Yes right now everyone has access to everything, trying to fix that
    - Do the admins have the authority to deny and give access?
        - The admins should have to have the authority to deny and give access within the system
    - How is your information being stored currently?
        - Google drive, (needs addressing)
- Incident detection/response using data analysis
    - Is automation used in the system at all? Making appointments or sending emails?
        - Not currently, but being addressed in the system
    - Do you have an incident response plan?
        - Not currently, being addressed in the system
    - How would you rate your detection system?
        - Outdated, un-updated
- Security compliance
    - How would you rate your security compliance?
        - Uses windows defender
    - Do you have a firewall?
        - No, will be addressed

- Secure DevOps

- Is there training for employees in place?
  - No, not currently

System Requirements (in our system) continued:

- ☑ Firewall
- ☑ MFA
- ☑ Secure server to store information/data
- ☑ Decryption/Encryption
- ☑ Automation
- ☑ Access Control
- ☑ Intrusion detection system

UML Diagrams with Explanation:

- ☑ Use-case Diagram Link
- ☑ Activity Diagram Link

Use case diagram: Shows the interaction between users (actors) and the system

Employee signs in, their sign-in information is stored in the active directory. Once the employee signs in, they are prompted to then set up MFA through the MFA system. Each time the employee logs into their account they will go through MFA. After setting up MFA, the employee then has to be authenticated to pass the firewall as per the firewall “actor”. Once they pass the firewall, they can then be authorized to do appropriate actions by admin which will be stored in the active directory. The employee can now do their tasks, like set-up appointments, and this can be done through the secure encrypted database. If an incident were to arise in the database, the incident response plan would be put into place by admin.

Activity diagram: Represents the processes within the system

The user logs into the system, the active directory gives the yes or no if the user is supposed to be logging into the system. If it is a no, the user will not have access to the system, they will be back at the sign in page. If the active directory passes the user, they are prompted to do MFA to make sure it is really the user themselves signing into the system. After MFA, the user will be passed through the firewall. The user now has the privileges to make their appointments or do their assigned tasks. When the user inputs their said appointment times/does their tasks, the active directory checks if they can do their requested action. If yes, then the user can successfully input their data into the system. A logging request will be sent once this happens to validate the input. If the input is validated, then the user is completed with their tasks or has successfully made their appointment. If the input is invalid, incident response will then be implemented based on the issue at hand (appointment scheduling or task issue). Admin will have to decide how to communicate best to solve the issue. If the way they decide

does not work, then they will have to reapproach the issue. If the way they decide does work, then the recovery phase will be implemented and the incident will be resolved. Users will be made aware to not make the same mistake.

Group Timeline:

Timeline (Milestones, Deadlines, Responsibilities)

Milestones:

- **Weeks 6-8:** System design and modeling using UML.
- **Weeks 9-10:** Python scripting for automation and basic data analytics.
- **Weeks 11-12:** Advanced data analytics and machine learning implementation.
- **Week 13:** Risk management and disaster recovery planning.
- **Week 14:** Final integration and testing.
- **Week 15:** Project presentation and submission.

Deadlines: Check-ins week 6, 9, 12

- **System Design Document Due Week 8:** A detailed report including UML diagrams that illustrate the system architecture, components, data flow, and security processes.
- **Python Scripts and Automation Due Week 10:** Scripts for automating tasks such as log analysis, system monitoring, and basic threat detection.
- **Data Analytics Report Due Week 12:** Analysis using real or simulated data to demonstrate how the system can identify and respond to security threats.
- **Risk Management Plan Due Week 13:** A document outlining the system's risk management strategy, including incident response and disaster recovery plans.
- **Final Project Presentation and Submission Due Week 15:** A presentation summarizing the system design, implementation, and findings. Submit all documentation, code, and presentation slides.

Responsibilities:

- **Weeks 6-8:** Systems Modelers will work together to design the cybersecurity systems using UML diagrams. They will use UML to design three diagrams total: use-case, activity, and class to model system requirements and user interactions. They will also create a fourth diagram, sequence, to show the flow of data and control within the system. A system design document report will then be produced based on the diagrams, illustrating the system architecture, components, data flow, and security processes.
- **Weeks 9-10:** Python Developers will work together to implement python scripts to automate cybersecurity tasks. They will use python to create scripts (code) for log file analysis, system performance monitoring, and alert generation. Once this code is

created, they will develop automation for routine security checks like monitoring network traffic and will include error handling and logging to ensure reliability. The scripts (code) for the automating tasks of log analysis, system monitoring and threat detection will then be submitted in a brief report explaining their functionality.

- **Weeks 11-12:** The data analyst will work to integrate data analytics and machine learning into the system. They will collect and preprocess data relevant to the organization's security posture using data analytics techniques to identify patterns in the data. They will implement/use basic machine learning models to enhance threat detection capabilities. Lastly, they will visualize/present their findings using plots/dashboards in a data analytics report that will be submitted.
- **Week 13:** The project manager will work to develop a risk management plan document that will be submitted. The risk management plan will identify potential risks and vulnerabilities within the system, and a risk assessment matrix will be created to rate the likelihood of the risk happening. The risk management plan will outline incident response procedures, including detection, response, and recovery. The final part of the plan will include a disaster recovery plan addressing data backup, system restoration and continuity of operations.
- **Week 14:** All members of the group will come together to integrate all system components and test the system to ensure functionality. Unit testing, integration testing, and security testing will be done. We will then document any issues and how they could be resolved in a document to be put into the final presentation.
- **Week 15:** All members of the group will come together to create a presentation that includes an overview of the system design, implementation, and key findings. The presentation will also include a demo of the functionality of the system. The presentation will be submitted along with all documentation and code.

Role Assignment:

- Project Manager: Gabrielle
    - Oversee progress, ensure deadlines are met, coordinate communication. Develop the risk management and disaster recovery plan.
- System Modelers: Shac, Sailendra
    - Create and maintain UML/SysML diagrams for system design
- Python Developers: Ella and Sridevi
  Develop the python script for automation and the data analysis part.
- Data analysis: Mohamad
    - Perform data analysis and machine learning tasks; contribute to reporting.