**Open a GitHub account for each of the team with a public repository (will be**

**used for your individual project).**

**Open a team account on GitHub (do not use your personal account!), create a public repository and submit that link. Your team will work out of this repository.**

https://github.com/Security-System-Company/COSC-310-Project

**Choose a software development cycle that is appropriate for this project and for your team. Give a brief rationale as to why you choose this SDLC rather than others.**

We chose the waterfall develop cycle. This is because the software we are developing for the security system is one that requires all parts of a step to be complete to move on to the next. To maintain a high level of security the requirements should be well understood at the beginning of the project and should only have minimal changes made to it. We also want the requirements to remain as tight as possible. Because of this lack of change it is easy to complete this system in a very step my step process, also to ensure everything necessary is complete before moving on.

**Create a project plan and submit this document. This is meant to help you do some upfront design, role assignment, task breakdown, and hours monitoring. Specifically, the report will include:**

**A brief description of your project (i.e., one paragraph), what you are doing, what features and functionality the system will have. The URL to your team repository should be included here.**

Our project is a security system for an office building. It should be able to scan preauthorized NFC cards to allow specific employees in specific doors within the working hours.  <https://github.com/Security-System-Company/COSC-310-Project>

**Your chosen SDLC and rationale for its suitability (max. one paragraph).**

We chose the waterfall develop cycle. This is because the software we are developing for the security system is one that requires all parts of a step to be complete to move on to the next. To maintain a high level of security the requirements should be well understood at the beginning of the project and should only have minimal changes made to it. We also want the requirements to remain as tight as possible. Because of this lack of change it is easy to complete this system in a very step my step process, also to ensure everything necessary is complete before moving on.

**In point form:  A listing of all the phases of the SDLC, where each phase should have at least 3 tasks. Here, make sure at least two of the tasks have subtasks. Give your phases, tasks, and subtasks meaningful labels -- run your labels by the TA if you're not sure they are easy to understand by a third party.**

REQUIREMENTS

* Researching existing items
  + Real Life
    - University
      * Campus housing doors and how they work
      * Classroom and lecture hall doors
  + Media
    - Office buildings
      * Emergency situations
      * Visitors
      * High level access
  + Research
    - Case study - MentCare
      * Security memory/logs
* Deciding which items from each
  + Using NFC cards like the key cards used on campus
  + Having specific visitor cards
  + Creating a similar system to MentCare’s activity logging system
  + Emergency measures to ensure doors do not remain locked when quick exist is necessary
  + Quick response time
* Narrowing down requirements to what is practical and within our capabilities

SYSTEM AND SOFTWARE DESIGN

* History
  + Assess the languages and coding software we know
    - Java (processing, android studio), python, SQL
* NFC
  + Options that would allow us to stimulate NFC cards tapping on a door
    - Android studio would allow for emulating and assigning cards
* Visuals
  + Which software is best suited for this situation
    - Processing
* Assign roles based on ability
  + Split responsibility based on how the tasks can divide
    - NFC cards
    - Visual Representation
    - Project Management

IMPLEMENTATION AND UNIT TESTING

* NFC/Phone App
  + Android studio (Java)
  + Research for implementation
  + Implemented basic features of the android studio app for user functionality
  + Implemented NFC scanning capabilities of android phone
* Visual/Website
  + Processing (Java)
  + Made an office layout using online software
  + Resized the software to necessary specs
  + Added visuals to processing
  + Coded doors different colours to represent when they open and close
  + The doors stay open (green) for three seconds and then close (red)
* Testing
  + Basic functions for both NFC and Visual aspects to later expand to connect the two

INTEGRATION AND SYSTEM TESTING

* Register cards with app
  + Find cards
  + Connect to app
* Add a networking interface to allow for phone app to communicate with software to emulate the situation
  + Find interface
  + Connect it to the phone app
  + Connect to secondary software
* Connect the two separate programs
* Test various situations
  + Safety measures
  + Doors opening
  + Doors failing to open
  + Logging interactions

OPERATION AND MAINTENANCE

* Find holes in testing
* Make sure no new issues arise
* Make sure code is up to date as outside software (Processing and Android Studio) are updated

LIST OF LIMITATIONS

* We cannot emulate someone holding open a door or any other suspicious activity with our program.
* Our software can currently only emulate one person use.
* Unable to test the functionality of the NFC Scanner app due to not currently having NFC cards.
* Currently do not have networking solution implemented to allow processing java application and android studio application to communicate with each other.
* **A Work Breakdown Structure (WBS) showing task assignment to each team member, estimated duration of the task in hours (round to nearest half hour), and actual duration of task in hours (round to nearest half hour). Provide a brief explanation of your WBS.**

So many hours but I will only be using the last two to do everything at once xoxo

* **A Gantt chart of your tasks, showing start and end dates for each task, and showing dependencies across tasks. Provide a brief explanation of your chart. Don’t forget that this assignment is not just about coding.  So, when you develop**

Gantt charts are actually super cool this looks like fun