**Software Development Capstone Summary**

# Summary

## The Business Problem

The management of security consulting assignments, personnel and engagements (both remote and onsite) has been traditionally performed using a variety of generic business tools and applications. This presents an opportunity to create a single pane of glass solution that can efficiently maintain and manage the tracking of multiple security consultants, their credentials/qualifications, past, present, and upcoming assignments, as well as quick, easy, and secure access to reports/deliverables generated from their respective engagements.

## Plan of Implementation

I will begin by defining specific requirements and objectives for the final solution. This exercise will allow me to brainstorm advanced features and functionality that will be included and implemented into the solution. Once a solid understanding of the applications requirements and features are thoroughly defined, a design document will be created that details the architectural design, data flows, and class diagrams that will describe the applications functionality. In addition, a test plan for unit testing will be created to ensure the data flows and use cases of the implementation are sound.

With a defined set of requirements, objectives, features and functionality, implementation of the solution will begin, starting with the database design (Data Description), the UI design (View) and base classes (Model); and ending with controller classes (Control) where functionality is implemented. A locally stood up MySQL instance will be used for initial testing, eventually migrating the datastore platofmr to a cloud-based SQL instance to ensure data availability, accessibility, and security.

## Software Development Lifecycle Details

### Requirements Collection and Analysis

* 1. **ISSUE: Fix the date “off-by-one” problem. (DONE)**
  2. The Security Consultant Engagement/Assignment Tracking System (SCATS) should be a stand-alone application with a backend SQL database, protected by a secure login prompt, where user access management is handled by an administrative account. (DONE)
  3. **The administrative account credentials should be changed upon the first login. All subsequent user accounts created should be managed via the administrative account.** 
     1. **Funcationality needed to manage users and track user logins**
        1. **Manage Users (edit timestamps, change passwords, etc…)**
        2. **Options**
        3. **Quit - DONE**
  4. The SCATS application should provide the user with the ability to track and manage security consultants, their credentials, as well as past and upcoming assignments and engagements. (**DONE**)
     1. **Report Possibility: Generate report of all engagements**
     2. **Report Possibility: Generate report of all engagements by Consultant**
     3. **Report Possibility: Generate report of all Consultants and their Certifications**
  5. The SCATS application should provide the ability to add, update, and delete the Security Consultants on the team; their attributes, credentials, qualifications, and areas of expertise. (**DONE**)
  6. The SCATS application should provide the ability to add, update, and delete assignments and engagements of Security Consultants on the team. (**DONE**)
  7. The SCATS application should provide the ability to securely **store**, retrieve, and delete reports generated for each completed assignment/engagement.
     1. **Implement file encryption where the key is stored in the database and files are only accessible via the application.**
     2. **Open file(s) upon right-click <open> (DONE)**
  8. A visual representation of all Security Consultant assignments/engagements should be implemented within the solution. (DONE)
     1. **Highlight the current date (Gave up)**
     2. **Visually span engagements from start to finish visual (MEH)**
  9. The SCATS application should provide the user with the ability to generate multiple reports to detail works completed as well as Security Consultant productivity and efficacy.
     1. Implement a reports manager to generate various reports
  10. **Implement exception management for data entry (DONE)**

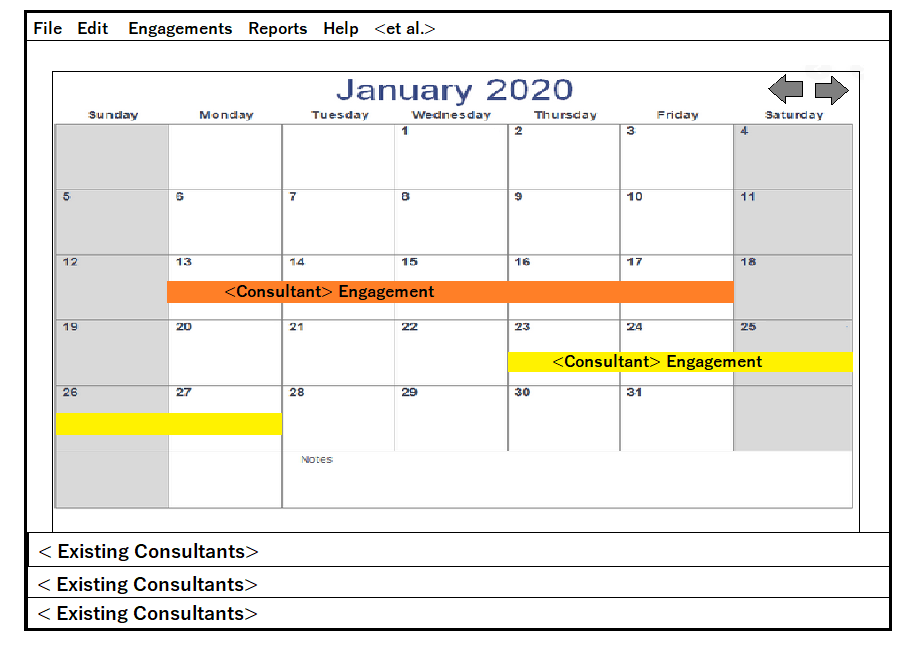
## Design

### General Use Case

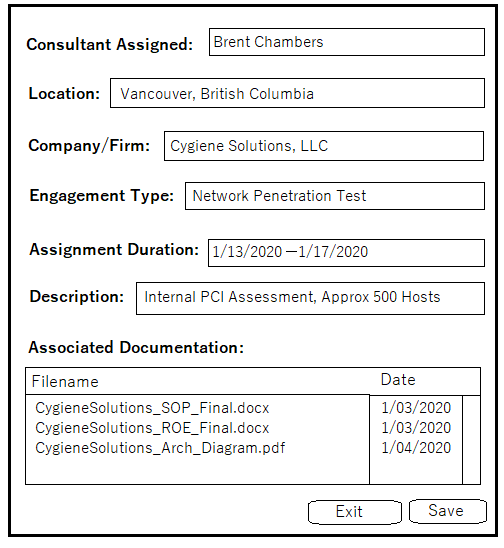
The primary purpose of the SCATS application will be to track security engagements, their assigned consultants, as well as their status and documentation generated through each phase of the consulting process. Because of this, engagements will be the primary focus, to ensure all work previously performed, in need of assignment, and to be scheduled in the future is presented first hand.

### Visual Mockup

The main screen of the SCATS application will provide the user with a graphical representation of the current month’s engagement/assignment schedule. Active assignments/engagements will span the scheduled length of time on the calendar and provide additional detail of the engagement upon a mouse click event.



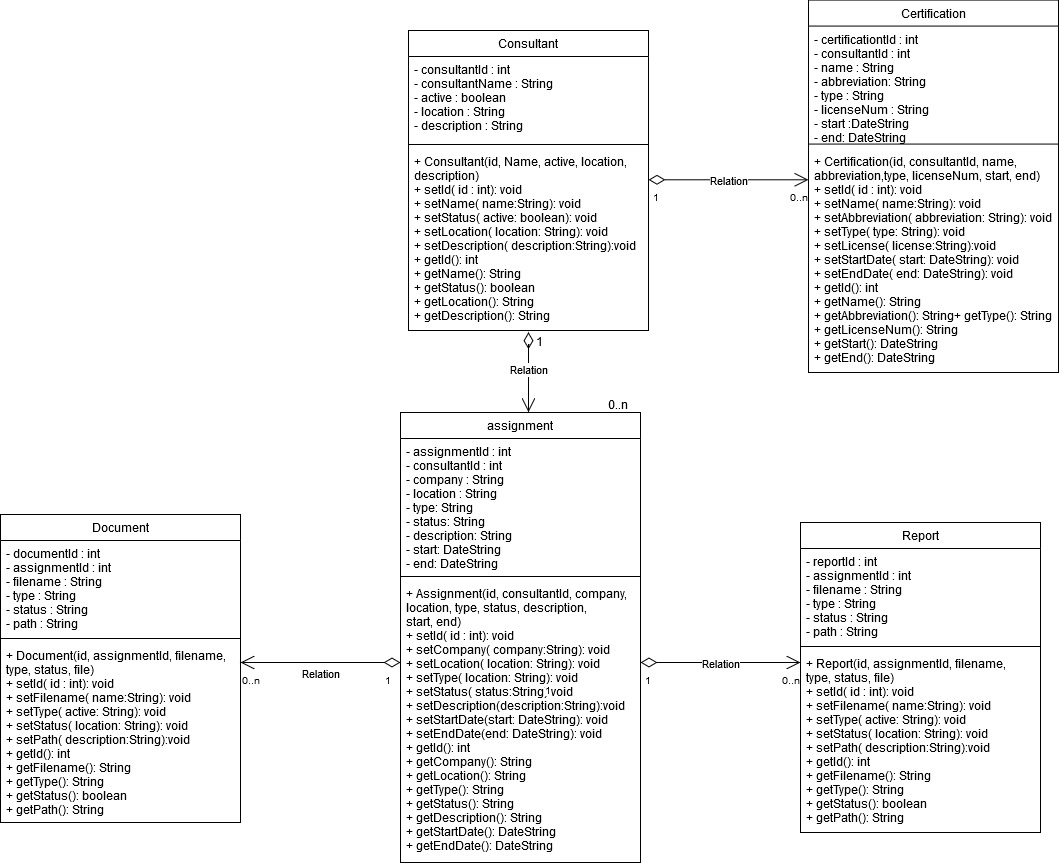
When clicked upon, a popup window will appear that will show the consultant assigned to the engagement, the type of engagement (Network, Web, Redteam, Social Engineering, Development, etc.), the location, the duration, associated documentation (Rules of Engagement (RoE), Statement of Work (SoW), etc.) and the hours of approved active operation.



### Data Design

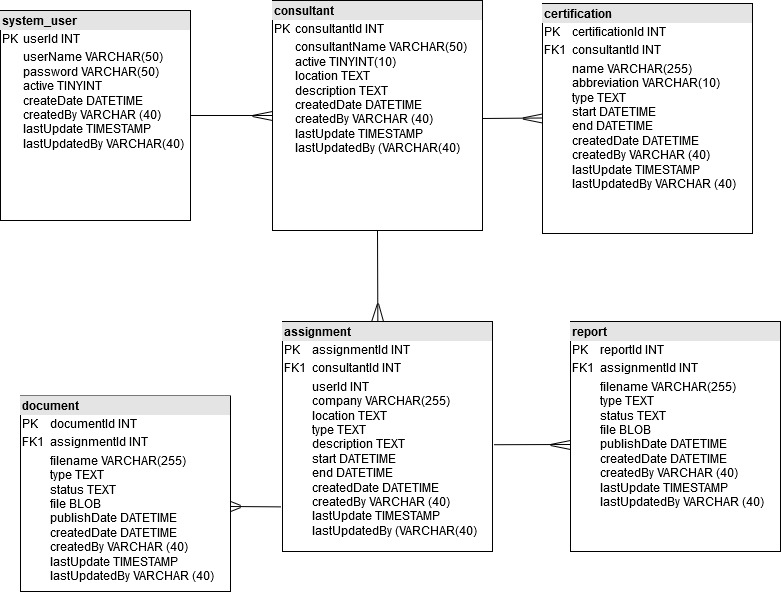
The SCATS solution is based upon the management and interaction between five primary data classes; Consultant, Certification, Document, Assignment, and Report. The Consultant data model defines each individual consultant added to the SCATS system for management and tracking. Because consultants can only be assigned engagements if they’re qualified and certified, the Certification data model will be used to outline each certification obtained by the consultant. Certifications data fields track the certification name, its type, license number, acquisition date, status, and expiration date.

An Assignment data model is used to define past, active, and future assignments that will be undertaken by qualified consultants. The Document data model will be used to define the necessary documentation records for an assignment, ensuring all legal and regulatory compliance needs are set prior to active engagement. Likewise, a Report data model will be used to define the reports generated as an assignments status transitions from “Active” to “Reporting”.



### Database Design

The backend database will consist of five data tables detaling the data descriptions of a system users, managed consultants and their certifications; assignments, pre-assignment documents, and reports generated upon completion. The “system\_user” table will serve to manage users authorized to access the application and database. The “consultant” table will have a many-to-one relationship with the “system\_user” table, and serve to manage consultants to be tracked and managed. Certifications of consultants will be recorded and tracked within the “certification” table, ensuring all consultants have valid, active, and industry approved security certifications. When consultants are put on assignment, the “assignment” table will be used to maintain assignment/engagement records along with the “document” table, used to store documents associated with the assignment. Finally, the report table, with a many-to-one relationship with the assignment table, will be used to store reports generated upon the completion of an assignment.



1. Implementation
2. Testing
3. Installation/Deployment
4. Documentation/Maintenance

## Valdiation and Verification

The evaluation of this solution should be performed through the perspective of a team lead/project manager, responsible for maintaining and managing a group of security consultants, their credentials and qualifications, as well as their respective scheduled assignments.

## Programming Requirements and Deliverables

This project aims to produce a single stand-alone application that meets all defined requirements, goals, and objectives. The project will be developed using Java, JavaFX, and a SQL database backend for data storage.

The SQL database will be provisioned via AWS RDS web services to ensure data availability. Data protection will be implemented via AWE security controls.

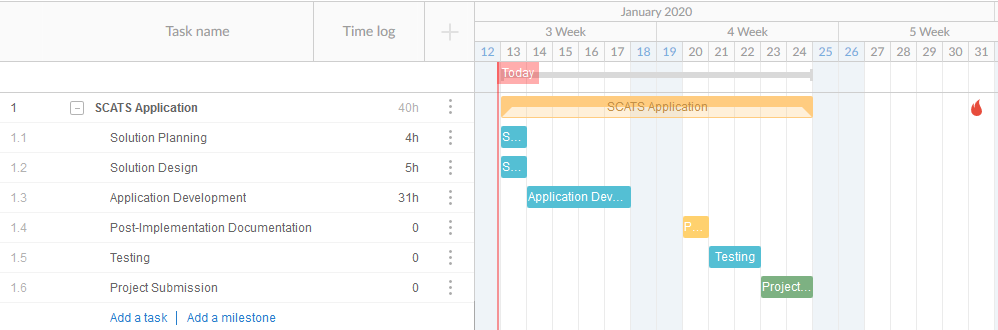
## Scheduling and Timeline

Application Type: **Stand Alone Java Application**  
Programming/development language(s) that you will use: **Java and SQL**  
Operating System(s)/Platform(s): **Windows 10 and MacOS**  
Database Management System you will use: **AWS Hosted MySQL Database**  
Estimated number of hours for:

* + 1. Planning and Design: **8 hours**
    2. Development: **40 hours**
    3. Documentation: **24 hours**
    4. Total: **73 total hours**

Projected completion date**: 1/30/2020**

**Gantt Chart Diagram**



# Post-Implementation Documentation

## Design Diagrams (Class and Design Diagram)

## Unit Testing

Use Case Scenarios:

1. Login to the SCATS Application and create a SCATS user
2. Onboard a new consultant (Name: Jesse Bradford), add a picture, and schedule him for Training in Vancouver, BC.
3. Generate a report of all Engagements completed within the last 6 months
4. Create 5 new engagements, and delete them all.
5. Attempt to access a SCATS stored report within the SCATS file system. Ensure the file’s contents cannot be read.

## Unit Test Results

## Source Code and Executable

## Hosted Resources

## User Guide Setup and Maintenance

## User Guide General Use

**Security Consultant Engagement/Assignment Tracking and Scheduling System (SCATS)**