|  |
| --- |
| September 29th 2018 – Stephen Nielson - CS 313 - Brother Burton |

|  |
| --- |
| Executive Summary |

# Project 1 - Therapist Admin History Tracker

The therapist admin history tracker will be a prototype application to evaluate the use of Amazon Cognito and SQL database triggers to build a HIPAA compliant audit trail of the activity of therapist administrators.

## Target Audience

This product is intended primarily for the HIPAA compliance officer and HIPAA IT administrator who are charged with monitoring and ensuring HIPAA compliance of the therapists for a HIPAA covered entity.

A secondary audience is the administrators of therapists who are in charge with managing the day to day billing, scheduling, and intake documents of a client.

## Proposed Prototype

This project will create a Angular 2 driven portal backed by a Heroku PHP system using the Postgress database. The system will allow administrators to add and update clients as well as therapists. Administrators can create basic appointments between clients and therapists. As data is viewed and or modified the system will generate a read only audit activity log of all interactions with system data by the logged in system user.

## Milestone 1

The first milestone will include an implementation of the database structure. A preliminary ERD for the database structure is included as Appendix 1. As seen in Appendix 1 the database entities will be ‘Client’, ‘Therapist’, ‘SystemUser’, ‘Appointment’, and ‘ActivityLog’.   
Appointment will represent a many to many relationship between Client and Therapist and contain the appointment data. Client, Therapist, and Appointment will have foreign key relationships to SystemUser to track which SystemUser created as well as last modified the data.

Triggers will be designed to insert data into ActivityLog whenever Client,Therapist, SystemUser and Appointment have any Create, Read, Update, or Delete (CRUD) activity. The application system user will be granted SELECT privileges to the ActivityLog table to prevent audit history modification.

## Milestone 2

This milestone will include the integration of the Amazon Cognito system for user authentication. SystemUser’s will be manually inserted into the SystemUser table and authenticated via Amazon Cognito. The angular 2 frontend will retrieve a list of therapists and display it on a therapists page. The frontend will also retrieve a list of clients and display it on a clients page. Finally, there will be an activity log page that will display a list of all of the records that have been selected/viewed in the system and the SystemUser who did the activity.

## Milestone 3

This milestone will have a page to see a list of appointments and the ability to create and update an appointment between a therapist and a client. The Activity Log page will display the list of records that have been updated / changed. A stretch exercise will be to include pages that will also update/create therapists and client records. To ensure data integrity all Insert/Update statements will be wrapped in transactions and I will use prepared statements to mitigate any SQL injection exploits. Data will be sanitized by removing all HTML elements from text input fields.

## Risk Mitigation Strategy

I have never used the Amazon Cognito system before and have extremely limited experience with MySQL database triggers. This is also my first attempt at the ActivityLog design pattern. There is some inherent risk in using these new technologies due to current unforeseen unknowns and the time constraints of this project. In the event that we can’t integrate Amazon Cognito I will utilize my own login authentication libraries I have built for Angular2.

My strategy for risk reduction with the SQL triggers is in the event I just cannot make them work I will insert the records as additional SQL statements at the business API layer in php.

I do expect a small measure of risk in the time requirements for this project due to having to build the PHP project by hand instead of using an API library such as Sylius or Laravel to do much of the heavy lifting for me. I have attempted to limit the scope of the number of database objects I will have to implement CRUD REST endpoints for. During the implementation of the project if I need to reduce the scope of the project we will skip over the Therapist frontend, Therapist endpoints, and remove the Therapist connection to the Appointment table.

## Conclusions

I believe this project is an achievable goal and that it is worth pursuing. It allows me to experiment with several new technologies I have limited exposure to. It helps me to determine if we should, upon conclusion of the prototype, roll these technologies into some of the products we develop at my company Discover and Change. I have a risk mitigation strategy to ensure the project is still completed on time and meets the class objectives / expectations.

## Appendix 1 – Database ERD

