

## **After Action Report**

SSE Capstone - University of Regina

Safety Toolbox

Winter 2024

Mackenzie Kot

Mikayla Peterson

## Table of Contents

### **1.0 Introduction**

#### 1.1 Background and the Why

### **2.0 Design**

#### 2.1 Discuss Scope and Objectives

#### 2.2 Discuss Stakeholders

### **3.0 Prepare**

#### 3.1 Setup and Picking Our Tech Stack

#### 3.2 Architectural Diagrams

### **4.0 Implement - Discussion and analysis**

#### 4.1 What was expected to happen?

#### 4.2 What actually occurred?

#### 4.3 What went well and why?

#### 4.4 Potential Improvements

### **5.0 Disseminate**

#### 5.1 Future Work

### **6.0 Conclusion**

## List of Figures

Figure 1: Project Requirements Document.....	2
Figure 2: System Architecture Diagram.....	3
Figure 3: Entity Relationship Diagram .....	4
Figure 4: Use Case Diagram - IT .....	4
Figure 5: Lofi Diagram of Home Screen .....	5
Figure 6: Lofi diagram of Toolbox Talk page .....	6
Figure 7: Lofi diagrams of the Attendance dropdown .....	6
Figure 8: Lofi diagram of Certification page.....	7
Figure 9: Attendance page.....	8
Figure 10: Home page .....	8
Figure 11: Certifications page .....	9

## **1.0 Introduction**

### **1.1 Background and the Why**

To ensure that companies are compliant with their standards, Occupational Health and Safety enforcers and auditors can request a number of different types of documents from companies. Some of the types of documents they may request are attendance records for Toolbox Talks (an informal safety meeting that occurs at the beginning of every shift) and training records. Organizing and tracking down these documents can become a time-intensive task especially if a company has a mixture of both physical and digital records. It becomes even harder to keep track of these records if they are never recorded in the first place and someone is just remembering certain documents. One such company that needs help getting all of this documentation in order is Ralph McKay Industries Inc., a global tillage tool manufacturer. Ralph McKay has looked into already commercially available software such as Bamboo HR and SiteDocs. These already available software products do not work for Ralph McKay for a variety of reasons including them being more focused on HR or being primarily focused on incident and accident tracking and reporting. They also wanted to keep their data on their own servers since they are not comfortable with their data being located on other companies' servers. After discussing licensing with Ralph McKay, we decided to keep the application open source. We did this so that if we are unable to keep maintaining the application, Ralph McKay will be able to make any modifications they need, and since no proprietary information is stored in our application code they were okay with this agreement.

## **2.0 Design**

### **2.1 Discuss Scope and Objectives**

During an initial meeting with Ralph McKay, they discussed major pain points they had with their current tracking process. The biggest overall pain point was that they were having problems keeping track of exactly where all of their documents were stored. They established that tracking Toolbox Talk attendance, Toolbox Talk topics, and training certifications are what they need to improve the most. They also emphasized that due to the type of work and machinery present in the production plant, this application needs to be desktop-based since workers are not allowed to access their cell phones in the production plant. They also wanted to be able to generate reports with the information stored in the application.

To prevent scope creep, we decided that we should initially only focus on developing features that were related to tracking certifications and Toolbox Talk items. These are outlined in detail in Figure 1 below, which is our project requirements document.

PROJECT REQUIREMENTS DOCUMENT	
Project Name	Safety Toolbox
<b>Functional Requirements</b>	
Toolbox Talk <ul style="list-style-type: none"> <li>- Need to be able to track attendance for each talk date</li> <li>- Need to be able to generate attendance reports for each talk</li> <li>- Should be able to add a planned date for a Toolbox Talk along with a list of topics to be discussed</li> <li>- Possible library of Toolbox talk ideas/presentations(?)</li> </ul>	
Certification Tracking <ul style="list-style-type: none"> <li>- Should be able to filter by the certification type</li> <li>- Should be able to search by employee to see what certifications they have</li> <li>- Should be able to add/delete/edit a certification record in the system</li> <li>- Should be able to generate a list of who's training is expiring within a specific time frame (could have multiple grouping options i.e. by employee or training name)</li> </ul>	
<b>Technical/Performance Requirements</b>	
<b>Technical Requirements</b> <ul style="list-style-type: none"> <li>- Needs to work on Windows-based computers</li> <li>- All data processing and updating has to be done locally within the customer's own servers <ul style="list-style-type: none"> <li>- A data import job may be required to copy information from their main employee table to the application's employee table</li> </ul> </li> </ul>	
<b>Performance Requirements</b> <ul style="list-style-type: none"> <li>- Response times should be within an acceptable range (i.e. users shouldn't be waiting for 5 minutes for a screen to update after information has been changed in a table)</li> </ul>	

**Figure 1: Project Requirements Document**

During the second semester of the project, we expanded on the concept of having an in-application library. After a discussion with Ralph McKay, they mentioned that it would be helpful if the library would be able to keep track of training documents and work procedures. They also mentioned that it would be nice if they could generate training matrices. A training matrix gathers a list of all the training required for a specific job position and indicates whether or not the employee has completed that training.

## 2.2 Discuss Stakeholders

There are six key stakeholders we have identified. As the capstone class facilitator, Dr. Tim Maciag is a stakeholder. As the team developing this project, Mackenzie Kot and Mikayla Peterson are stakeholders. Our project mentor, Dr. Craig Gelowitz is another stakeholder. George Galloway and Martin Peterson, an Operations Superintendent and Controller at Ralph McKay respectively, are also stakeholders. Galloway and Peterson were continually contacted throughout the duration of the project to gather user feedback on the application's progress.

We had identified safety coordinators/managers in companies as our north star customers since they would be the most involved in the process of creating and managing safety documentation. Carryover customers include employees, like machine operators, who would need to be able to view these safety documents. Initially, this application was developed with industrial manufacturing companies in mind, but the application is general enough for other types of companies to use it. We chose to work with Ralph McKay specifically on this application since they genuinely want this type of application and they were willing and available to provide input on it.

## 3.0 Prepare

### 3.1 Setup and Picking Our Tech Stack

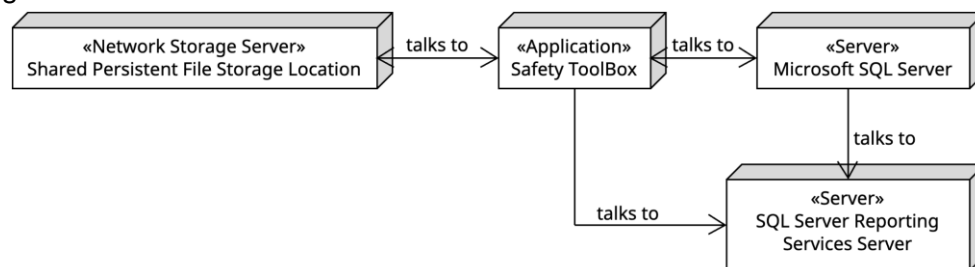
When we first met with Ralph McKay, we learned that they would want a desktop application over a web-based application. Later when we met with IT, we learned that their

setup already contained Microsoft's SQL Server Management Studio (SSMS). With this information, we decided to use .NET MAUI for the app and use SSMS as our backend. We chose .NET MAUI as our main technology for a few reasons. The first reason we chose it is because neither of us had worked with .NET MAUI specifically before, but both of us had worked with .NET/C# before, and we wanted to use the opportunity of capstone to learn something new. The second reason we chose .NET MAUI was due to a suggestion that a mobile app version might be useful to some companies that may use the app. Since we knew Ralph McKay wanted a desktop application for sure, we needed a technology that would lend itself well to porting the application to different operating systems, which is something MAUI aims to do. While it was never officially planned for our capstone deliverables, we wanted the ability to port the application to platforms other than Windows with little effort if we so decided.

### 3.2 Architectural Diagrams

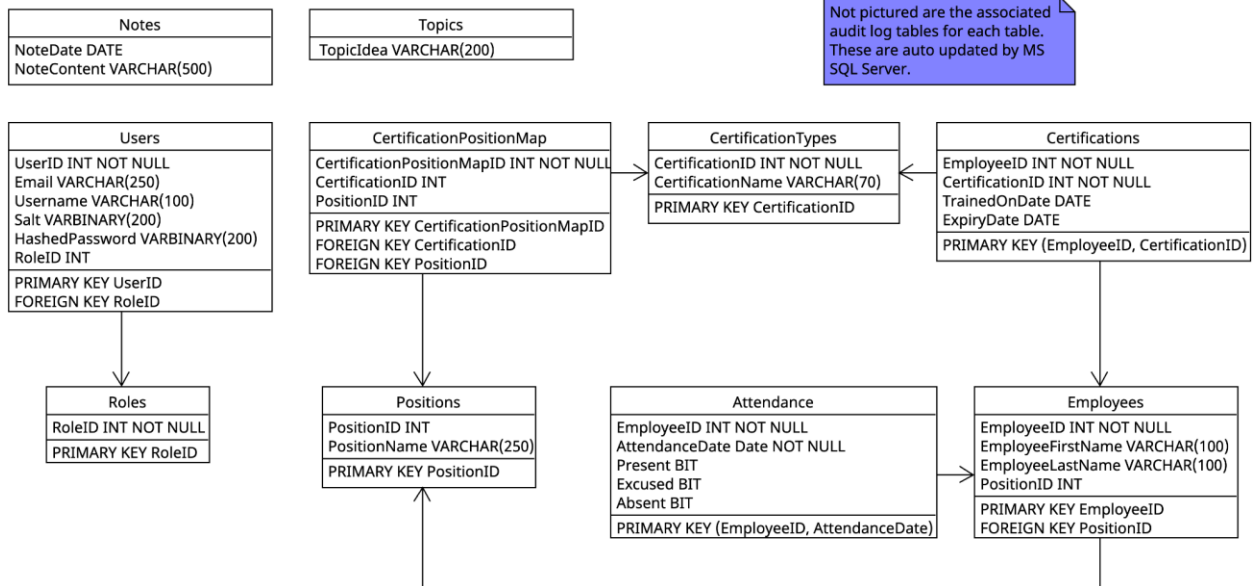
We created four types of architectural diagrams that were slightly tweaked throughout development to ensure that they remained accurate: entity relationship, simplified code architecture, overall system architecture, and use case diagrams. These diagrams were selected since they give the best overall look at how our application is actually going to be structured. We could have spent more time creating more diagrams for more specific scenarios (i.e. sequence diagrams for some specific interactions), but we felt at that point it would have been more useful to begin programming. We felt it was best to have the overall picture defined and worry about the hyper-specific details when it came time to implement them in the code.

Figure 2 below is the system architecture diagram. It illustrates how Safety Toolbox is intended to connect to Ralph McKay's SQL server, their reporting server, as well as their network storage location.



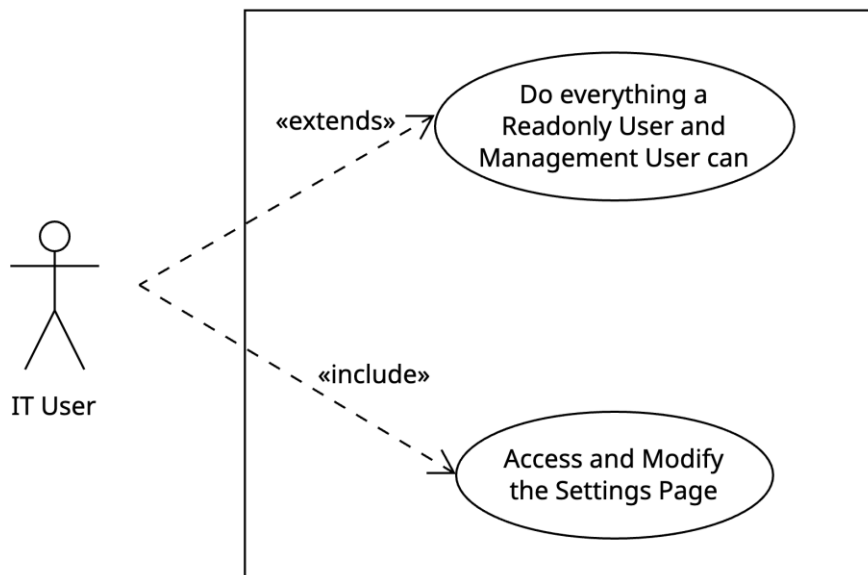
**Figure 2: System Architecture Diagram**

Figure 3 below is the entity relationship diagram. With the exception of the Notes and Topics tables, our database structure is normalized to reduce the duplication of specific types of data in the database.



**Figure 3: Entity Relationship Diagram**

We ended up having a total of four use case diagrams. We have three different access levels within our application (readonly, management, and IT) as well as an admin setup account. To make access easier to manage, and the use cases easier to conceptualize, the higher access tiers have all the same access as the levels below them with some additional use cases added. The only exception to this is the admin setup account, which is only able to access and modify the in-application settings. Figure 4 below shows how the IT role has the same access the management and readonly users do as well as the ability to access and modify the in-application settings page.



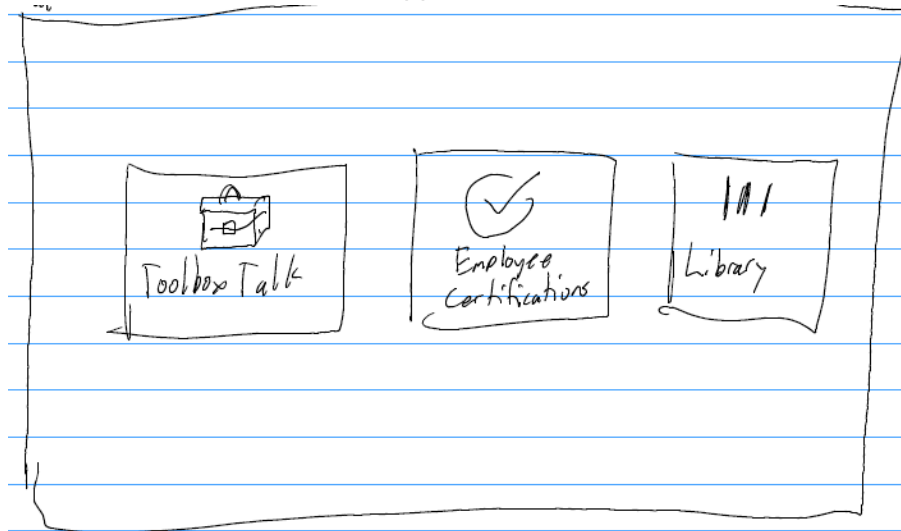
**Figure 4: Use Case Diagram - IT**

## 4.0 Implement - Discussion and analysis

### 4.1 What was expected to happen?

Prior to one of our initial meetings with Ralph McKay, we drew up some lofi diagrams of what each page may look like. At the time we didn't have any experience with MAUI so we weren't sure what was already built into it, but the lofi diagrams would at least serve as a general guideline for us, and provide a good picture of what we were planning to Ralph McKay.

Figure 5 below is a lofi diagram we made for the home page. The main idea was to use tiles to access the three main areas of the application.



**Figure 5: Lofi Diagram of Home Screen**

The lofi design for the Toolbox talk page in Figure 6 features dropdown components for the three things that belong under it. This was an early idea to keep everything sorted under the three main pages found on the home screen, and the Library page's lofi diagram used the same idea. Figure 7 shows how we planned the Attendance dropdown on the Toolbox Talk page to look. The Attendance dropdown would display attendance for that day only, with a button to view past attendance at the bottom. The topic ideas and notes dropdowns were also quite simple.



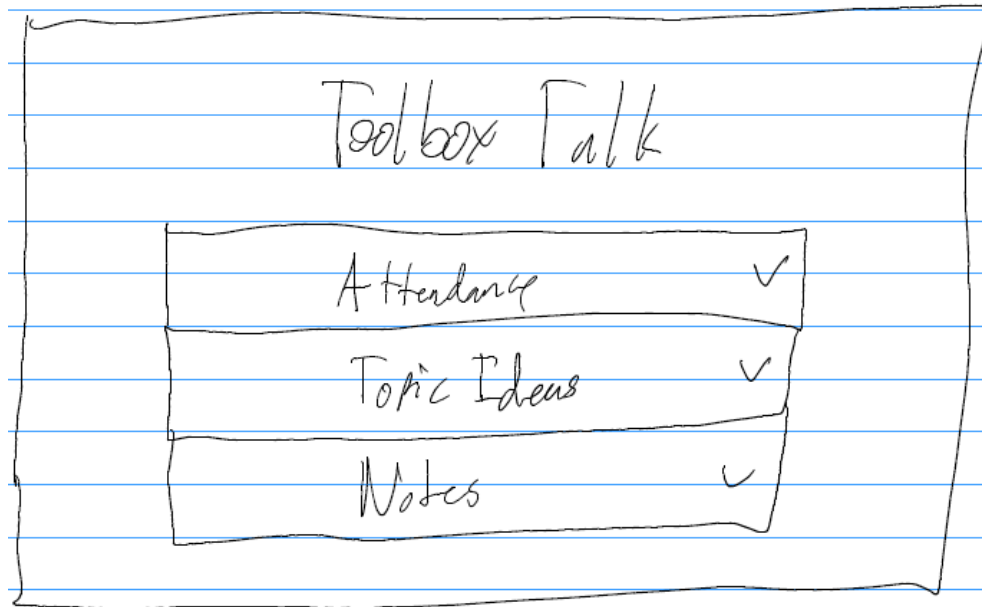


Figure 6: Lofi diagram of Toolbox Talk page

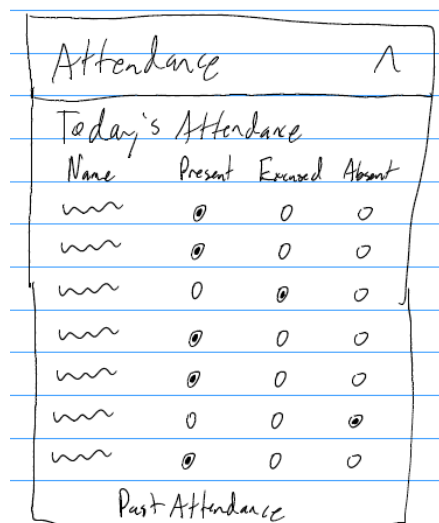
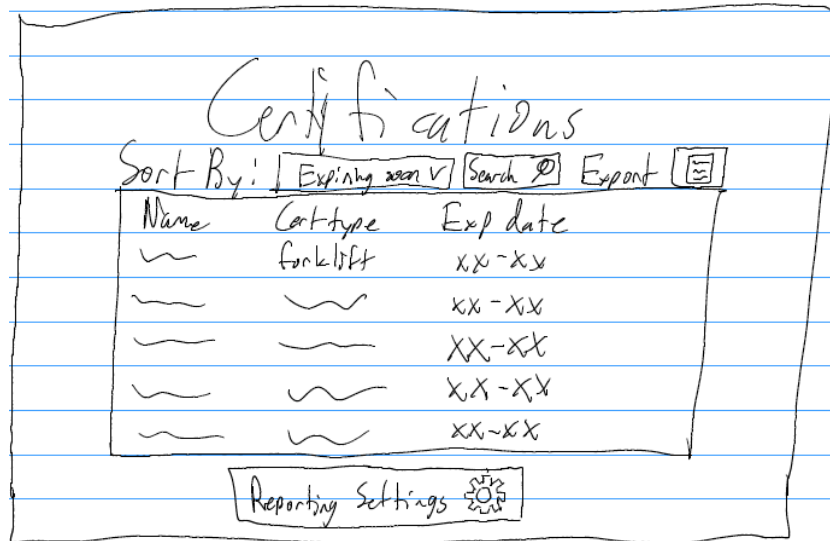


Figure 7: Lofi diagrams of the Attendance dropdown

The Certifications page in Figure 8 below has a fair amount of detail to it, being one of the main ideas for the app in the first place. It would display a list of certifications, with ways to search and sort them. There is an export button and a reporting settings button, which were planned to generate SQL reports of certifications using SQL Server Reporting Services (SSRS).



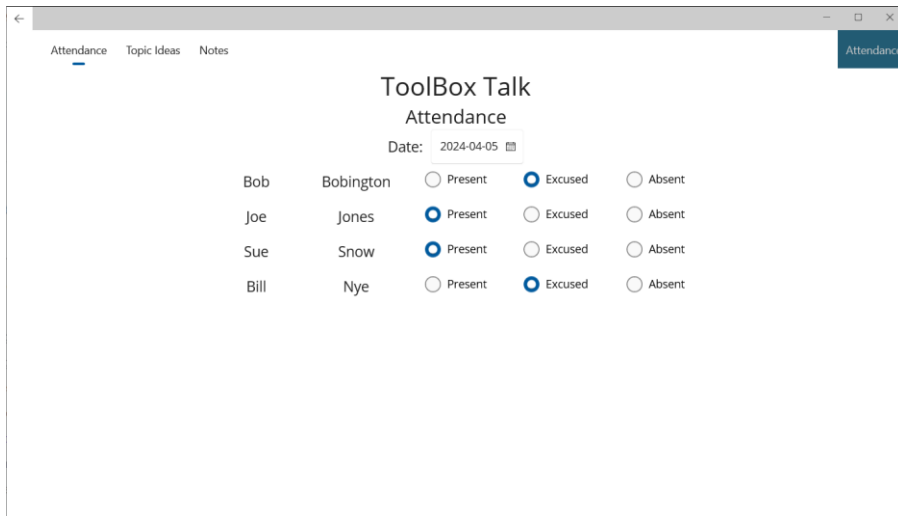
**Figure 8: Lofi diagram of Certification page**

#### **4.2 What actually occurred?**

There were quite a few changes between the lofi diagrams we created and the final version of the app.

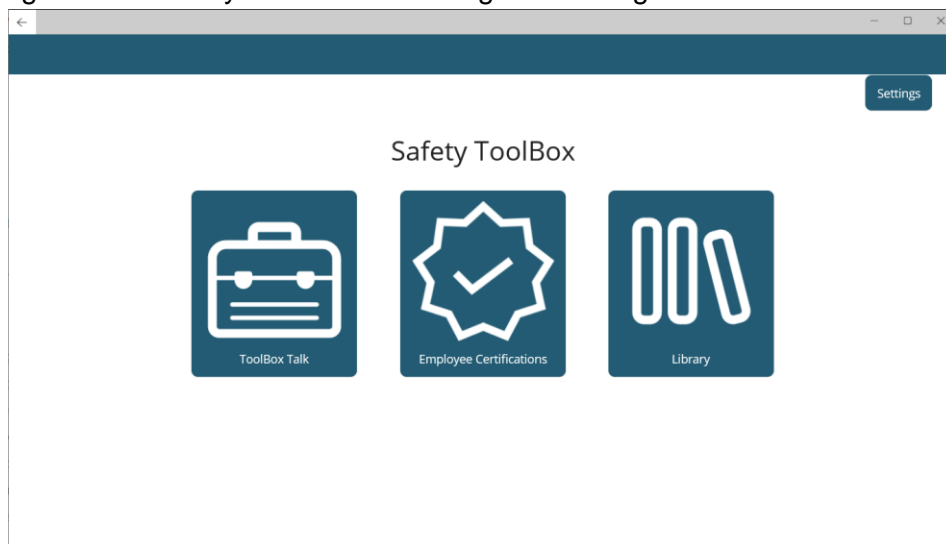
While working on our app, we changed our scope a bit to exclude the reports we originally planned to include from SSRS. This was for a few reasons. First, we had some issues connecting to Ralph McKay's server, so we moved it lower on our priority list. Then in March Ralph McKay had their own issues with their report server, and it was breaking every few days. This is when we decided to remove it from our current scope and move it into our list for future work. This allowed us to focus more on the app itself as well.

Knowing nothing about MAUI when we created our lofi diagrams, they contained elements that do not exist in MAUI. An example of this is the dropdowns, which only exist as a picker tool and not as a component that can hold children in MAUI. While this is something we could have coded ourselves, we chose to go with TabbedPages for the sub-pages instead, which was a better idea in the long run since it allowed for more room on each page. This meant that the Attendance and Library pages got a more improved layout than the one we planned in our lofi diagrams. To take the Attendance page in Figure 9 as an example, the final version contains a date picker at the top of the screen to reflect what date the attendance being shown was taken rather than the separation between 'Today's Attendance' and previous dates discussed in section 4.1.



**Figure 9: Attendance page**

There are also parts of the app that stayed true to our original vision. For example, the home page in Figure 10 is nearly identical to our original lofi diagram.



**Figure 10: Home page**

The Certifications page, seen in Figure 11 is also largely the same as our lofi diagram, with the only significant difference being the number of columns of data shown, and the removal of the reporting buttons meant to utilize the SSRS reports, reflecting our slight change in scope.

First Name	Last Name	Certification	Trained On Date	Expiry Date	View PDF	Edit Certification
Joe	Jones	Cranes/Slings	2021-08-17	2024-01-20		
Joe	Jones	Lock Out Tag Out	2021-04-30	2024-02-15		
Bob	Bobington	Tower Motor Safety	2019-12-20	2024-04-30		
Joe	Jones	Lifting	2024-04-05	2024-06-13		
Sue	Snow	Bio Hazard	2020-07-15	2024-07-20		
Bob	Bobington	WHIMIS Training	2020-12-20	2024-08-30		
Joe	Jones	WHIMIS Training	2021-12-20	2024-08-30		
Sue	Snow	Fire/Emergency		2024-08-30		
Sue	Snow	WHIMIS Training	2022-12-20	2024-08-30		

**Figure 11: Certifications page**

These are just a few examples, but if you examine our app, you can clearly tell that we used the lofi diagrams as a starting point for every page.

### 4.3 What went well and why?

Besides moving our reporting functionality to the backlog due to issues with Ralph McKay's servers, we were able to accomplish what was outlined in our project requirements document. Despite having issues with MAUI, which will be discussed later, we were able to work around those issues to create a fully functioning product that we are proud of. We are happy that we were able to get some parts of the app implemented that stayed true to what we initially envisioned; we are also happy with the changes we made in order to make the application work.

Although this is not code-related, something that went well that is worth mentioning is that we worked well together as a team and were able to behave with professionalism. Since we have known each other for a long time, we were comfortable and honest with each other when one of us would have to confront the other about project progress and other things. If we were not comfortable with each other and were unable to handle this type of interaction, the project would have taken a turn for the worse and likely would not have finished.

### 4.4 Potential Improvements

There are a few things that we would change if we were to do this project again. The biggest change we would make is to not use .NET MAUI and use something else instead. At the beginning of this project, neither of us had used MAUI before. A few months in we had discovered a few issues with MAUI (issues that, upon researching, had existed for years without being fixed), and had to use some inconvenient code workarounds to get the functionality we desired rather than the straightforward way we thought we could do it. This caused our code in some files to be less clean than others.

A simple example to illustrate the problems we had with MAUI was with the Attendance page's radio buttons. Radio buttons generally function in a group and are used in cases where

the user needs to pick one thing from a list. MAUI's radio buttons have a property for this functionality, called `RadioButtonGroup`, which can be set on each radio button. The problem, however, is that when you are developing a Windows application, setting the `RadioButtonGroup` value on a radio button has no effect. This is only a problem on Windows, this works fine on an Android app for example. Thankfully, there is another way to assign a group to a radio button, and that is by telling the parent of a radio button the group the button should be in. Due to the way MAUI Grids work and the fact that our radio buttons needed to be in different columns on this page, each radio button needed to have its own parent (we used `StackLayouts`) for no other reason than to assign a group to its child. This was one of the first issues we noticed, and certainly one of the simpler ones. When a basic function such as grouping a radio button does not work, it's a sure sign that other things won't operate as expected either, and we found out the hard way every time.

Outside of changing the use of .NET MAUI as our technology, there are a few changes we would make to our application. The application makes a lot of SQL queries to the database, and this really came to light during our code review. This means that there was a lot of repeated code that was almost the same, save for the query itself, any parameters, and reading the results after querying. It would be much cleaner and a lot less work if we had made a class specifically for handling the SQL queries, with different functions for insert, update, and select statements with optional parameters for the maximum number of query parameters we needed. The query itself would also be passed in as a parameter. The results would have to be returned to the caller of the SQL function to be read, since the caller would know the format of the results and be able to read them.

## **5.0 Disseminate**

### **5.1 Future Work**

We have already agreed to continue developing this application for Ralph McKay as volunteers; the project will be retaining its open-source status. As a side note, we will be marking our final commit before project day with a tag for marking purposes. We had issues deploying the application to their servers prior to Project Day due to some permissions problems with Microsoft SQL Server, so the first thing that we are planning to do is to get this sorted out. Throughout March, Ralph McKay was having issues with their reporting server not working correctly, thus we halted working on the reports as well; this work will have to be picked up again post-Project Day. In all likelihood, future modifications made to the application will likely be guided by requests made by Ralph McKay due to our agreement to continue working with them.

There are a couple of things that could be good to add to the application. Currently, there is no way to update the certification names and other pieces of information that only appear in several dropdowns in the application; they have to be updated in the actual SQL tables (Ralph McKay had no issue with this). To make this and other pieces of information (like the Certification and Position mapping) easier to update, it would be a nice quality-of-life feature to be able to adjust these things in the in-application settings. Incident and accident tracking is a fairly important feature that would also be a good thing to add to this application to further increase its usefulness to other companies outside of Ralph McKay. Adding a mobile application port would likely be particularly useful for non-industrial companies who need to access the

information in the application when they are away from their computer. The ability to add equipment inspection and maintenance information (like a preventative maintenance schedule) would also be useful to keep in this application to ensure equipment is kept in safe working order. Hazardous material management and a specific place to store material safety data sheets would also make sense as a feature. The ability to keep track of personal protective equipment issuance, usage, and compliance would also make sense for this application. A lot of these features could be added as a new icon on the main dashboard or as an additional library tab.

## **6.0 Conclusion**

While there is always room for improvement, and we do still have future work planned for it, this application in its current form is something we can be proud of.

While we were able to finish the application we had planned, there were definitely some lessons learned along the way and some things we would do differently if we were to do a project like this again. We've realized that .NET MAUI, while it is already a few years old, needs some more time to sort out its issues to be a more useful technology. In the future, we would likely pick a different framework to develop an application like this.

In conclusion, we successfully created the application that we set out to create.