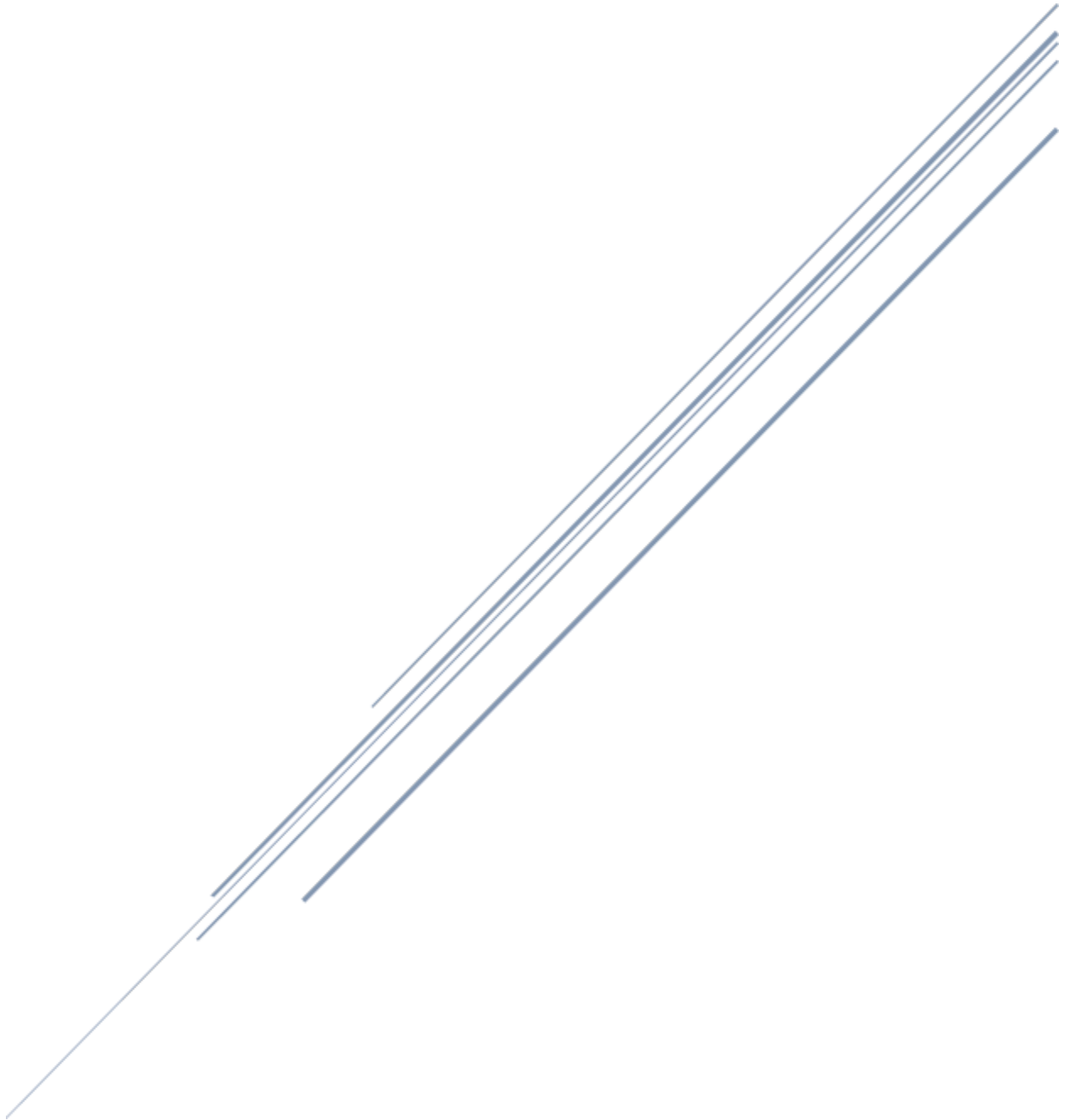


Visitor Management Tool - Thermal Exhaust Port (TEP)

A Capstone Report for CST499

by Carlos Orduna

Fall 2020



Executive Summary

Thermal Exhaust Port (TEP) is intended to become a visitor management tool used by two different types of users. An external group of users are granted access by an internal user to redacted visitor information for different dates. This project is expected to be expanded well upon past the capstone, the original deliverable has been identified as the baseline of a management tool. The main goal for this project is to create a way for these two users to share information. The two main objectives will achieve this goal by creating a database which will hold visitor information and a website that will display the data.

Minimizing contact between people in public spaces has become the norm due to SARS-COV-2 being introduced to our lives. This project will aim in reducing the amount of time it would take for visitors to check in to a lobby and interact with a receptionist or guard. This project has the potential to help reduce the amount it takes for people to be in a lobby by having their information be prefilled ahead of time and shared to the external staff.

Table of Contents

Executive Summary	1
Table of Contents	2
PART 1	4
Introduction & Discussion	4
Introduction	4
Problem in Technology	5
Project Description	6
Stakeholders and Community	6
Justification for Project's Needed	7
Client	8
Project Goals and Objectives	8
Goals	8
Objectives	8
Feasibility	9
Environmental Scan	9
PART 2	11
Design Requirements	11
Final Deliverable	11
Function Decomposition	11
Approach/Methodology	12
Selected Criterion	13
Ethical Considerations	14
Legal Considerations	14
PART 3	16

Timeline/Budget	16
Usability Testing/Evaluation	18
Final Implementation	19
Conclusion	24
References	27
Appendix A	28
Appendix B	29

PART 1

Introduction & Discussion

Introduction

Checking in at a waiting room for a meeting or appointment is the first interaction a visitor has when they step into your company, so it makes sense to want to make it as seamless as possible for your potential clients, customers and employees. In most places it cannot be as simple as having an open lobby where guests can walk in straight to an office, classroom, or boardroom. In most corporations we expect a level of security in place to preserve a continuous flow of visitors and prevent access to any unscheduled or unwanted guests that may disrupt the workflow. This project aims on sharing this information ahead of time before the visitor arrives to minimize the amount it takes for them to sign in and access the facility.

Due to the current events in the world and the SARS-COV-2 virus being introduced to our lives, moving things electronically and minimizing the amount of exposure between people has become paramount for companies nowadays. It is in the best interest of companies to reduce the amount of people in waiting rooms and office space to reduce the spread of the virus. Reducing the amount it takes for people to sign in, while also decreasing the contact between a receptionist and visitor due to the information already being available and prefilled, may protect workers in their daily life.

My Primary approach will involve meeting with internal users (security and trade compliance personnel that receive visitor information) and external users (personnel that staff a lobby) to understand the information that should be saved in the database and that is necessary for a visit to take place. This will help in identifying the information that needs to be redacted from the primary database. Learning the structure of how the databases are maintained and controlled is paramount to being able to discern a flow of information needed. Once the applications used and primary fields to be shared are identified, I will create mockups for sharing queries through a local website.

Researching on what the market has to offer for the visitor management solution can also provide insight into what sort of features could be implemented in the future. Features implementing sign up forms or checking in features to extend to the visitors are viable and can be introduced after the capstone. As goals are met and new ones discovered, I can see some of these objectives being requested to be introduced to ease the signing in process.

Problem in Technology

Some companies may use an off-the-shelf visitor management software or external applications to verify guests via a personal identification number, biometrics or badge access. Others, such as hospitals, may handle this situation by having a person make an appointment for a visit and an external user will receive a list of the appointments expected for the day. Some businesses may not have the luxury of being able to share the whole visit and visitor information to this external user or post the information out in their waiting rooms with an off-the-shelf software due to general security concerns for where the data is being fed into. This can cause

unintended side-effects such as slowing down the visit sign in process or having an external user generally not be aware of who to expect for the day.

Project Description

For this project, I plan on creating a database where visitor information will be kept, along with tables that will support this information to create many different queries and a website in which only some of the information will be posted. The final goal is to make it possible to share redacted or truncated visit information while keeping the integrity of the data on the database unmodified. While later expanding upon this project to include many other management tools. The solution will involve having both, a linked database and a website in which data can be stored and shown to a user.

Stakeholders and Community

The primary stakeholders involved for the project would be internal users, the external users, and the people involved in visiting a building using this solution. Both, the internal users and external users would gain a symbiotic relationship through the use of the application, as the information will begin to be shared between the two groups. External users will have access to vet visitors coming in, and verify that they have been cleared beforehand, allowing them to either speed up the process or deny the access of those that have been previously rejected. This would prevent access to any individual that has been rejected or denied entry by the internal users, but not communicated to the external users ahead of time. This could prevent legal action if

paperwork has yet to be signed, such as non-disclosure agreements, and maintaining proper security measures.

Justification for Project's Needed

As new issues arise with the SARS-COV-2 pandemic, one of the main paths companies may be looking to condense their essential operations, while reducing and monitoring their visitors traffic. Having a Visitor Management System can assist in pre-registering visitors, verifying identifications and streamlining the process (Metcalf, 2020), allowing all of these operations to be performed with minimum interaction between either the visitor and the personnel verifying the visits. Introducing a system that can assist in reducing the spread of coronavirus. One of the approaches that Metcalfe delves into in his journal regarding Workplace Safety for Essential Services during the Coronavirus Outbreak is an “invite-first” process to mitigate visitors who may be dropping by unannounced. This process involves having a team pre-approve visits and, via pre-registration and approval flow (Metcalf, 2020), minimize the face-to-face interaction between people in a high traffic area such as a waiting room or lobby. TEP will try and recreate this approach by having users pre-input the information for visitors ahead of time and display them in a table. This will help organize and plan ahead on who may be coming into the facility. According to the CDC guidelines “the more closely you interact with others and the longer that interaction, the higher the risk of COVID-19 spread” (CDC, 2020). That’s why reducing the amount people spend

inside closed rooms, such as waiting rooms, may help in reducing the spread of the current pandemic..

Client

I will be tailoring the project to the specifications and design for a company whose current specifications are set for an internal team that reviews visitor information from their employees ahead of time, and have external users such as receptionists or guards that man the waiting room. The final project may later be molded for the introduction to a company who may be interested in using it.

Project Goals and Objectives

Goals

- Introduce external users to an internal database's information
- Control what information is shared between the two parties involved
- Help external users be able to tailor their day depending on how busy their day may be
- Help ease the sign in process for a waiting room setting to avoid prolonged periods of waiting in a waiting room

Objectives

- Parse through a visit form (in a database) to collect information and store it
- Create queries for day visits to display in a table

- Establish a new internal access to review visits
- Share data via internal network to an external user
- Future Objective - Notifications to process owners or sponsors of the visit when a visitor arrives
- Future Objective - Expand upon the tools available to external and internal users

Feasibility

Environmental Scan

As pointed out under the Problems in Technology section, some of the companies that look for a visitor management system solution would resort to an online software for their waiting rooms. The way they work is by having a device placed in the waiting room for guests to sign a form and the information is retained for as long as the customer would like. These devices are shared between individuals and placed on high traffic areas. Looking at an article written by Ernest Hamilton titled “Top 5 Visitor Management Systems in 2019” his recommendations come to be Sine, Proxyclick, Envoy, SwipedOn and iLobby, from least to most recommended (Hamilton, 2019).

The software that stood out the most to me was Envoy, as it comes with many features tailored around security, compliance, and efficiency. Security features include signing proper documentation, such as non-disclosure agreements, and visitor screening (Envoy Visitor, Deliveries, and Rooms Management, n.d.). The software also allows the owners to review the

data and export queries to an external system; however, most of the information is initiated by the visitors and not the stakeholders.

There are many enterprise solutions available. Looking at Envoy's list we see a robust library of software available for companies for purchase. These softwares often may save data on the cloud or connect to external servers for the data management. This can be risky for some companies who may want to control the data locally. A lot of these softwares require the visitor to enter the information when they arrive, which involves handling the same device (computer or tablet) between multiple people. The focus for TEP is to minimize this contact, the focus will be on having a user populate the tables of time. There are many good updates that could be incorporated into TEP but due to time constraints will have to be implemented past the capstone proposal.

Tailoring the software or application up to the internal users standard is what will set this project aside from an off-the-shelf or subscription-based solution. This will allow users to control where the data is stored and limit the signing process to only those internal users.

PART 2

Design Requirements

The detailed specifications for the basis of the project is to create a system in which users can create visitors and portray the visitors on a list formatted to other users. The requirements involve working with a database for a back end, and a form or list for a front end. Due to this being a solo project, the workload will be light and shrunken down to meet specifications. User friendliness is a key focus for the tool development, as users with varied levels of experience should be able to navigate through the final product.

Final Deliverable

The Final deliverable will include a schema and a website form and list. which will create a main visitor table along with other supplementary tables. These supplementary tables will be prefilled during creation. The final website will contain two major sections for users to add visitors and to review visits. The final deliverable will be more a proof of concept that meets the goals and objectives of the capstone project, intended to be used to build upon it well after the capstone project is completed.

Function Decomposition

The major functions needed for the final product are a list of visits and a form for the internal users to work on and ease of visitor input. The database will work as a back end that will

contain all of the visitor information while an internal website will display the information back for anyone to see.

The database will have fields for visitor information, names, company and date of visit, with a sponsor and facility as foreign keys as seen in the sample below.

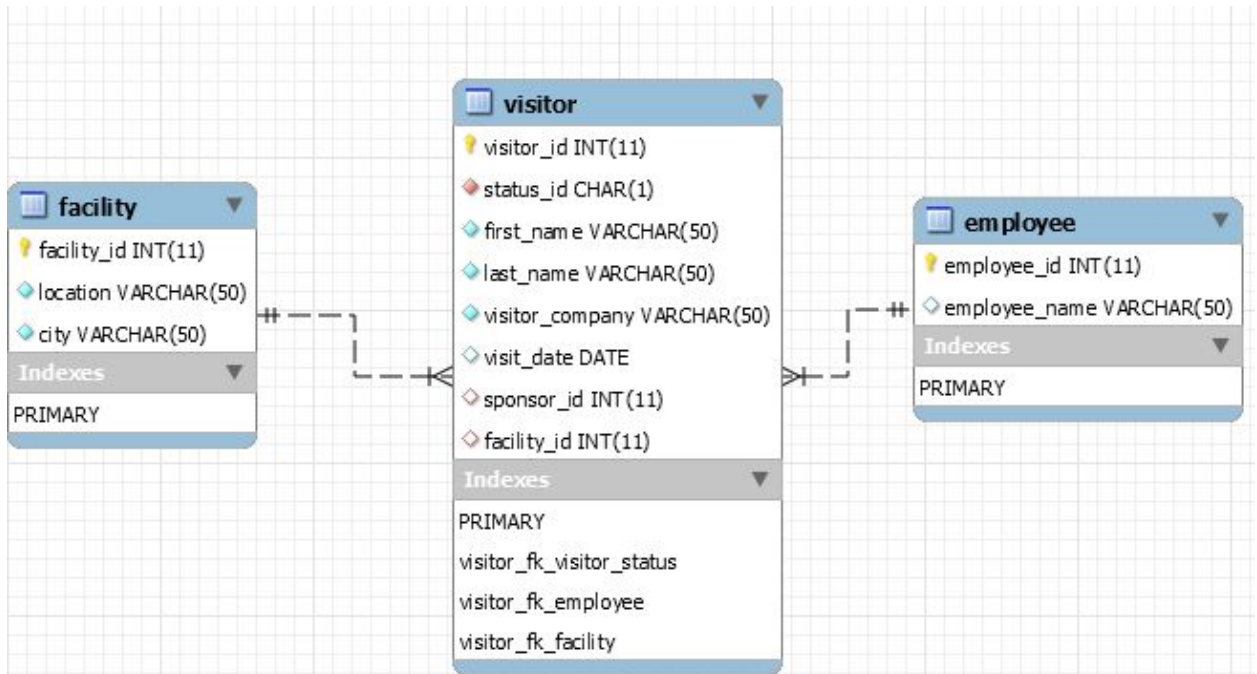


Figure 1. EER sample diagram for a visitor table with foreign tables

Final forms will contain fields to input information for the visitor table without modifying the other two, as the employee and facility tables are not the primary focus on this project.

Approach/Methodology

A working database is paramount and I will be using previous labs from our database course to create a schema for a visitors table with smaller support tables related to it. I have

created two different versions for this, one on MS Access upon request, with an internal form and one version on MySQL which is linked to an internal website using Java & Spring Boot and hosted locally. The Access version had to be put on hold due to constraints on available resources and time frame on the project, which shifted gears to a quick and internal solution. The final database design on MySQL will follow teachings from CST363 to create a SQL schema and build tables through a script. A visitor table will be used with two support tables, visitor sponsor and facility. Once the database is created and tested, the online form will connect to these tables via entities and contain two versions, an input form and a display list for users to go through. I will be using teachings from CST438 to link these two using spring boot and Java. These forms and online lists will have a similar look and feel using bootstrap and CSS code to modify the fields theme. I will be using teachings from CST336 to style the site and introduce a general theme throughout the internal websites. I will be testing the data entry and verifying that the database is updated when new fields are introduced, as well as making sure things are printed in an organized fashion.

Selected Criterion

Based on the selected specifications, the goal is for a simple to use form with an organized list. Performance is not a big dictator on the selected choices, as memory allocation and control will be handled by storing data inside the database and only accessing it when needed. I will be using MySQL for the final product due to working locally with a personal computer and not having access to enterprise licenses to other software. This will help create a proof of concept in which a visitor management tool could be expanded upon. The major constraint in this project is the fact of it being a solo project. Without the ability to split the work,

time management will need to be monitored to make sure deadlines are met. To meet these deadlines, the project has been split into two sections, database and website work. The final deliverable has also been truncated in order to have a proof of concept rather than a complete off the shelf solution.

Ethical Considerations

The major ethical concerns that may arise with the capstone project is that now that visitors may be kept in a database and checked, those that have been denied access to the building will be known to a third party external user. The personnel who are denied entry may be done so due to failed background checks or internal decisions within the client's team, which is not shared with external users. Sharing such information may result in oversharing of personal data and can cause assumptions on the reputation within those on the denied list. Spilling of this data can also be considered to be mishandling of information and could affect the internal users ability to manage further sensitive information. This is why sanitizing the information before sharing is important to keep the privacy of those involved, by only sharing all the information with those who have been properly trained, while funneling the least amount of information necessary to any external users.

Legal Considerations

A denied personnel list may be followed with a negative connotation to the people on it, which is also why making it such information private and controlled is important and necessary to the client. Having such information go out into the public could have legal repercussions from

the people whose identity may be compromised and damaged. Those revealed to be barred from accessing the facility could look into legal action and sue the company on the grounds of their reputation being tarnished and for defamation, because their name was a part of a denied entry list. All data entered will use test and fictional information to avoid any personal information.

Besides these legal issues, the software used will need to use active licenses and any non original code will need to have the proper authorization from the originators before being able to be used on a release that will be given to a client. Software in use will be acquired legally and will be used with proper active licenses.

PART 3

Timeline/Budget

Originally, the project involved working alongside a client under my own proposed timeline. The project began in early September with the database being the first of the items on the timeline (*figure 2*). The project continued sailing through until the sharing hub portion of the proposal timeline.

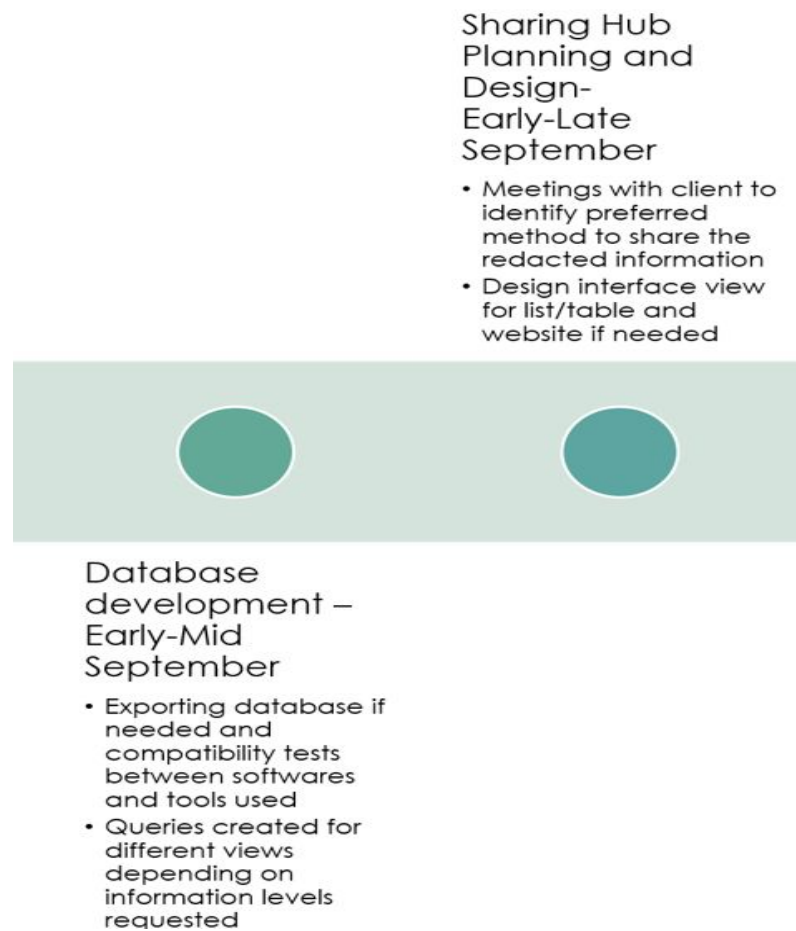


Figure 2. Part 1 of original proposed timeline



Figure 3. Part 2 of original proposed timeline

During the sharing hub planning portion, the client’s information technology specialist indicated that we will not be able to use an existing server in order to host any internal websites by the time we were ready to do testings and suggested to create a local proof of concept locally on a personal computer. This caused my scope to change and proposal to shift to a less ambitious project due to new time constraints. With new freedom on software to be used, due to not having limitations by the client while using my personal computer, I switched softwares for the database and started over. A lot of features I wish could have been part of the original proposal were put in the icebox to meet deadlines and overhead course work. I was not able to meet every goal I had set in mind by the time that testing was to take place, so the scope had to be lowered to provide a proof of concept.

Usability Testing/Evaluation

My target audience for the project usability test involved people with different levels of computer skills. During the testing process, I set forth three different tasks for the users to complete. This was pretty straightforward, add a new visit, edit an existing visit and delete a visit. I figured that the main focus for my project was to be accessibility, anybody should be able to look at the project and understand what they can do. The tests were done in person on my local laptop while I ran it as a spring boot application. During testing, I had two different versions of the project, one with MS Access and another with MySQL as the base database. Both versions were tested by two client representatives, and I had one more person test the MySQL version to meet the requirements of having three people. The Access version was not complete as it did not include an internal website, instead it used internal forms to create visits, so all they were asked to do for this version was to create a visit and edit a visit. For purposes of this project the focus was on the Capstone version and the access was reviewed second for my own input into the second version.

I loaded the local website for them and had each user do the three tasks requested. The website at this point had no UI design, as it was a white page with a list and hyperlinks. The testers did not run into any issues as they were quick to edit visits and delete them. They tested with fake data and tried to use data that could break the project. The forms found under edit and new visit creation were quickly tested and reviewed. The main remark all three had was that they would like to see a more professionally looking website instead of a blank slate. The other

comment that was brought up was that they would like to see more fields for each visit (see Appendix A for reference form submitted). My testers did not find any issues with the listed commands, but focused more on the look and feel for the website, a task that I took to fix after the testing session in order to give it a less generic feel.

Final Implementation

For the final implementation of the project. I went from having a MS Access database that was linked to many different support tables such as employees, facilities, access level, and visitor companies and will be what this project will evolve into in the future. Originally the visitor table was to have foreign keys for each of the support tables, and queries were created in order to control these other fields within the input (figure 4).

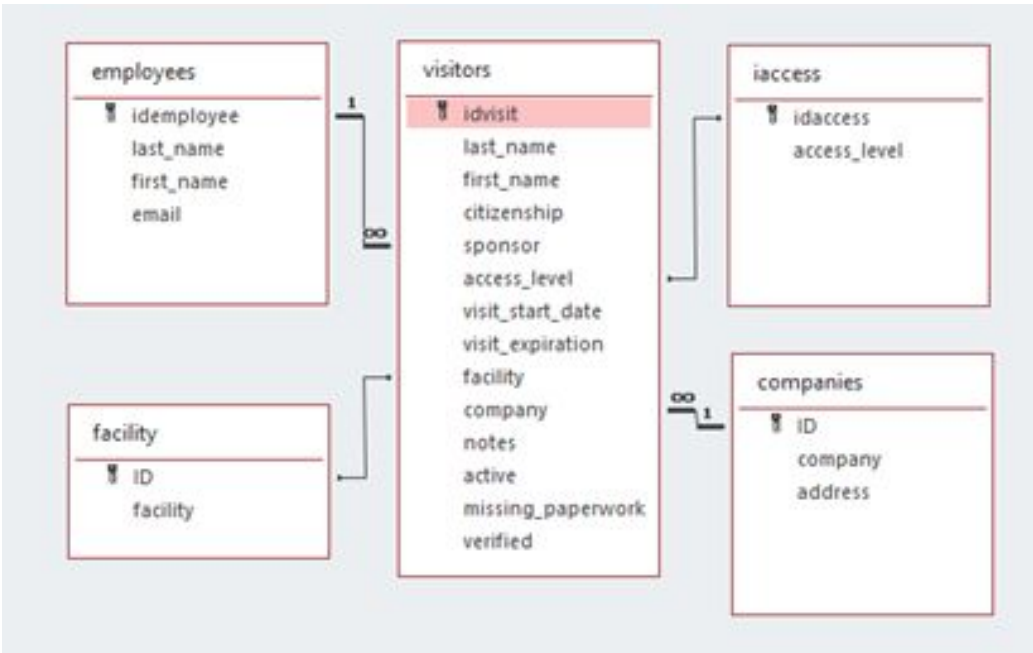


Figure 4. MS Access original database ERR design

Due to new scope and shorter timeline however, the tables in use and the concept behind them shrunk. I decided to go with a single table to create a general concept and expand upon it as time allowed (figure 5).

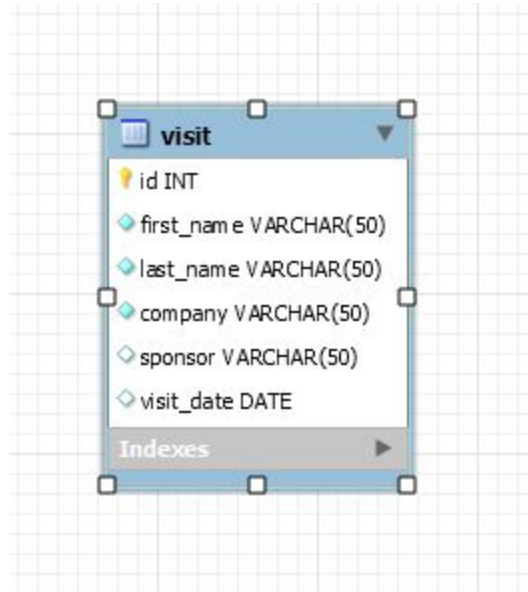


Figure 5 -Table used for proof of concept

I opted in to have the most common fields and the minimum information that could be needed for a visit to take place. The table includes the visitor's name, their company, date of visit, and their sponsor's email address, this is what was used in order to create a baseline for the website and begin working on where the data will be displayed. I drafted a script to create the schema that will accompany the website and matches the ERR design above (figure 6).

```

1 • DROP SCHEMA IF EXISTS tep;
2 • CREATE SCHEMA tep;
3 • USE tep;
4
5 • CREATE TABLE `visit` (
6     `id` INTEGER NOT NULL AUTO_INCREMENT,
7     `first_name` VARCHAR(50) NOT NULL,
8     `last_name` VARCHAR(50) NOT NULL,
9     `company` VARCHAR(50) NOT NULL,
10    `sponsor` VARCHAR(50) NULL,
11    `visit_date` DATE NULL,
12    PRIMARY KEY (`id`)
13 )ENGINE=InnoDB DEFAULT CHARSET=utf8;
..

```

Figure 6 script created for the visit's table

I played around with having a larger database. I created a bigger table with a supplementary facility and employee tables. I drafted another script that would create this and add fields for both employees and facilities at the moment of creation (see Appendix B). I did not implement this on the final deliverable during this iteration, but this is the new desired path towards the next future update on the project.

For the sharing of information hub, I decided to do something similar we did during our Software Engineering course. I created a new Maven project combined with a Spring Boot Application. I decided to use the JPA repository as an interface between the software and the database. I created a visit Entity class (figure 7) that matched the criteria that the database was holding. Implementing the date was tricky as displaying the information was straightforward. I had to set the date as a “Temporal” in order for the interface to be able to write the date field into

the database on the edit and create new visit forms. Getters and setters were generated for the visit class as well.

```
@Entity
public class Visit {
    private Long id;
    private String first_name;
    private String last_name;
    private String company;
    private String sponsor;
    @DateTimeFormat(pattern = "yyyy-MM-dd")
    @Temporal(TemporalType.DATE)
    private Date visit_date;
}
```

Figure 7. Visit Class

I used a service and controller class to access the data on the project and to travel between the pages. To keep things simple, I used commands to list all the visits, save edits to the visits and to delete entries on the database depending on the ID selected, all this can be found under the service class of the visit. Under the controller, I created the website's handlers that will send the id to be edited or deleted, as well as having the redirecte pages for the user when they click on specific buttons.

When the application launches, the home page lists all the visits that are on the database (figure 8). The homepage links to the find all visits function and query in order to display them all in order of creation (with a focus on adding filters in the future).

Visit ID	First Name	Last Name	Company	Sponsor	Visit Date	Edit/Delete
1	Ian	Williams	Apple	white@company.com	2020-10-15 00:00:00.0	Edit Delete
2	Dennis	Simmers	Dell	white@company.com	2020-10-20 00:00:00.0	Edit Delete
5	James	Solis	Toshiba	dorian2@company.com	2020-10-08 00:00:00.0	Edit Delete

Figure 8 Home page

I edited the page feel to it by using bootstrap. The edit and delete button capture the ID of the row selected in order to do the calculations and add it to the browser’s URL. This gives quick access to the user to modify the correct row or delete the visit without any hassle. I added a “New Visit” button for the users to click on at the bottom of the list, and on the header. I used similar edit and new visit forms when displaying them to the users (Figure 9).

Edit Visit

Visit ID: 1

Visitor's First Name:

Visitor's Last Name:

Company:

Sponsor's Email:

Date of Visit:

[Save](#)

Create New Visit

Visitor's First Name:

Visitor's Last Name:

Company:

Sponsor's Email:

Date of Visit:

[Save](#)

Figure 9 - Forms for editing and creating visits

Users can easily edit visits and create new visits to display in the main page, once the visit is over, it can then be deleted. The main focus was to keep things as simple as possible so that any user with any computer skill can traverse the website without any issues. Other things

added to the website involve a navigation bar for the header, which as new modules get expanded upon, more buttons shall be added.

With the main functions completed, it is possible now to expand on the project to make it more complex and introduce new features. This implementation carries over what the baseline necessities are for a visit to take place and for the information to be shared between users while under the same network.

Conclusion

In conclusion, the main issue I wanted to solve was to lower the amount people spend in waiting rooms or lobbies and share information with a user that can pre fill information and share the information necessary for the visit to take place. I decided on a visitor management tool in which users can input information ahead of time so all that is needed would be to verify the person's identity. I originally worked alongside a client, but due to some software restrictions and time constraints I created two versions, one for my client and one for the capstone.

I created a visitor management tool that uses a MySQL database backend which holds all the data and a website on the frontend where the information can be shown and edited. The website was drafted to be simple to use and follow, with minimalist clutter in mind. The information on the website is already pre populated and easy to follow for those who may need to get access to it.

I learned that working on a project alone, depending on the scope proposed, may be a huge task. I had to lower my scope and expectations throughout the project to meet deadlines. I

will keep expanding on working on this project well past the end date of the program in order to incorporate and make the project more robust. We need to keep fighting despite the setbacks that may be put in front of us, getting past them is what helps us grow and helps us hone our skills.

I had the opportunity to work alongside a client for the duration of this project and I learned that sometimes the requests and restrictions put in by the client ask for us to be flexible. I was able to follow their proposed guidelines at first, but the final product would not have been done in time for the capstone due date. This, and advice from others, caused me to make the decision to make the jump and restart the project locally. I learned that we must adapt to changes if needed to complete and meet deadlines, whether those be set by a client, ourselves or by our teachers.

I was also able to play around with previous labs procured during the teachings throughout the online computer science program. I used what I learned in my teachings of the online computer science program and integrated them into my final deliverables; learning to be resourceful is definitely a skill needed to succeed in a field that keeps changing and growing rapidly.

This is by no means the last time I will work on this project, as I still plan on implementing many tools on it to make it stand out from other similar software available in the market. It currently may be in its infancy but there are many plans to make it expand. The client version has also not slowed down, the setbacks put forth are mainly there due to time constraints and it will be a matter of letting time go by and having a proof of concept available for

integration. I'm hoping that as I keep working on this project more and more, it keeps shaping to become a tool that companies may be able to incorporate into their infrastructure.

Overall this has been one of the most rewarding experiences that the computer science online program has offered to me and I'm grateful that I had the opportunity to tackle an issue and present a solution for it.

References

CDC (2020). Daily activities and going out. Retrieved from *Centers for Disease Control*:

<https://www.cdc.gov/coronavirus/2019-ncov/daily-life-coping/going-out.html>

Envoy visitor, deliveries, and rooms management. (n.d.). Retrieved from *Envoy*:

<https://envoy.com/>

Hamilton, E. (2019). Top 5 visitor management systems In 2019. Retrieved from *Tech Times*:

<https://www.techtimes.com/brandspin/238224/20190204/top-5-visitor-management-systems-2019.htm>

Metcalfe, K. (2020). Ensuring workplace safety for essential services during the Coronavirus outbreak. Retrieved from *Security Magazine*:

<https://www.securitymagazine.com/articles/91968-ensuring-workplace-safety-for-essential-services-during-the-coronavirus-outbreak>

Appendix A

Testing Group for TEP

Below is the information provided by the testers during testing for the project. This was provided as an assignment and below is the extract with each user's main points for the capstone project.

Tester 1 - Kaitlyn

- Very simple and easy to understand
- Would like a more engaging look (currently black and white)
- Would like different users
- Would like more fields to be added
- Would like to organize by fields

Tester 2 - Kelly

- No issues navigating through actions (edit and create visit)
- Would like a homepage and color

Tester 3 - Nikki

- Would like it being in its own software
- Would like more color and a theme
- Easy to see visits, add, edit and remove

Appendix B

Unused script for visitors database. Reserved for future project work

```
DROP SCHEMA IF EXISTS tep2;
```

```
CREATE SCHEMA tep2;
```

```
USE tep2;
```

```
CREATE TABLE facility (
```

```
    facility_id INTEGER NOT NULL AUTO_INCREMENT,
```

```
    location VARCHAR(50) NOT NULL,
```

```
    city VARCHAR(50) NOT NULL,
```

```
    CONSTRAINT facility_pk PRIMARY KEY ( facility_id )
```

```
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE visitor_status (
```

```
    status_id CHAR NOT NULL,
```

```
    description VARCHAR(50) NOT NULL,
```

```
    CONSTRAINT visitor_status_pk PRIMARY KEY ( status_id )
```

```
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```

CREATE TABLE visitor (
    visitor_id INTEGER NOT NULL AUTO_INCREMENT,
    status_id CHAR NOT NULL,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    visitor_company VARCHAR(50) NOT NULL,
    visit_date DATE NULL,
    sponsor_id INTEGER NULL,
    facility_id INTEGER NULL,
    CONSTRAINT visitor_pk PRIMARY KEY ( visitor_id )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE employee (
    employee_id INTEGER NOT NULL AUTO_INCREMENT,
    employee_name VARCHAR(50) NULL,
    CONSTRAINT employee_pk PRIMARY KEY ( employee_id )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

ALTER TABLE visitor ADD CONSTRAINT visitor_fk_visitor_status FOREIGN KEY ( status_id ) REFERENCES visitor_status ( status_id );
ALTER TABLE visitor ADD CONSTRAINT visitor_fk_employee FOREIGN KEY ( sponsor_id ) REFERENCES employee ( employee_id );
ALTER TABLE visitor ADD CONSTRAINT visitor_fk_facility FOREIGN KEY ( facility_id ) REFERENCES facility ( facility_id );

```

ERR for the script above.

