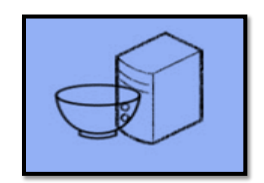
**Group Database Implementation Project Report**

New Modern Sign Project Management and Customer Relations Database Service

Courtesy of: New Modern Sign



CIS3365 - 2022SP - 21909

Database Management

META SOLUTIONS GROUP TEAM MEMBERS

Hibbard, Schelling M

Mason, Kameron A

Momin, Shafin S

Nguyen, Andy P

Nguyen, Peter Q

Patel, Krishnali

Rahman, Ahmed N

Syed, Taqi

Ghafour, Mohamad

April 14, 2022

Table of Contents

[Executive Summary: 3](#_Toc100843330)

[Client Information: 4](#_Toc100843331)

[Benefits and Costs: 5](#_Toc100843332)

[Project Approach: 6](#_Toc100843333)

[Solution: 7](#_Toc100843334)

[Testing Process: 14](#_Toc100843335)

[Project Improvements: 15](#_Toc100843336)

[Project Database Maintenance Issues: 16](#_Toc100843337)

[Lessons Learned: 17](#_Toc100843338)

[Project Summary: 18](#_Toc100843339)

[Presentation Slides 19](#_Toc100843340)

[Resources: 20](#_Toc100843341)

[Appendices: 21](#_Toc100843342)

# **Executive Summary:**

New Modern Sign, a growing small business based in the greater Houston area started in 2013. The owner Azmir Islam started off with a small clientele. These design projects once finished were outsourced to different printing shops. In the next 8 years the company grew to be a full-fledged sigh business, employing over sixteen full time employees.

Growth in business does not always lead to implication of business processes and practices and systems. To manage this expansion of incoming projects and the prediction of further growth there needed to be a more centralized process. This process would be to help, collect data, track, and keep up with each individual project. New Modern sign required a solution that would allow them to expand even more by managing projects with a centralized database.

This database would allow each of the employees to view or edit things such as the due date of a project, aspects of a project (permit information, customer name, address of installation, height, width etc.), and control human resources (who works on what part of the project). Bulk loading the data with .csv files is another important aspect we would like to have for this system because it allows us to migrate the current data quickly and efficiently to our new database.

The Meta Solutions group has taken on a project as part of the University of Houston College of Technology to analyze the current system New Modern Sign utilizes and deliver a new alternative based on the needs of the client.

# **Client Information:**

As a student in high school, Azmir Islam, the now owner of New Modern Sign, was approached by a friend who told him about a business opportunity in which the previous owner of the company passed away and the family of the deceased were wanting to sell the business. After many months of research and contemplation on whether to purchase the business or not, Azmir and his family purchased the business and started to learn about how to run the business. Azmir fell in love with the business and loved every aspect of it, so he went to college to learn how to further scale his business and learned design in his free-time and eventually he had the opportunity to start running the business by himself. When running the business throughout college, the shop would get small contracts since the shop only had one printing machine and two workers including Azmir. Fast forward to now, the business has scaled and grown exponentially, and the organization gets dozens of large-scale contracts a month and it currently holds over 15 employees. However, since there are a lot of contracts, employees, and task to do during the process of a project, the owner, Azmir Islam, needs a centralized database to gather up all their data in one place to keep it organized. This is due to multiple aspects of the business having data that relates to the business process. Along with having unorganized data, Azmir did not have an application that is convenient when it came to updating or deleting the data. With all the data being stored in excel files without being organized, Azmir Islam has requested Meta Solutions group to create a database with a functional GUI that allows the company to update data, delete data and even combine tables within the GUI.

# **Benefits and Costs:**

There are many benefits when implementing the database created by Meta Solutions Group. The first benefit is that all the data being stored will be easily accessible along with the data being organized. The GUI is also user friendly and is simple to use. When creating the database for New Modern Sign, the project It was concluded that there would be no initial monetary cost for the client as Meta Solutions Group will not be requesting funds for this project. Concerns are raised due to the possible liabilities for using the new system. After taking into consideration the possible opportunity cost (for which the client may give up when deciding to work with MSG over other groups) and time, it was concluded that the potential upside will be more than the risk which is relatively low. As this application is developed by students at the University of Houston for a class project, Meta Solutions Group has agreed to create a functional GUI database system for $0. Therefore, New Modern Sign has incurred at no charge for the database. When it comes to maintenance cost, the client would incur server cost to have the application run on the computers for the business. There will be no troubleshooting charges or even support charges for the client.

# **Project Approach:**

By the end of CIS 3347, our team had an acceptable design idea that we would want to follow for CIS 3365. However, a lot more concepts were covered in the latter class. Consequently, the team had to change multiple aspects of the project design that were previously thought to be exactly what would suit the client’s business case.

The ERD

The ERD was revamped from the group up due to many inconvenient elements and entities. For example, there were multiple many-to-many relationships which were taken care of by creating associative tables that changed the relationships to two or more one to many relationships instead; with the associative table acting as a buffer between the two.

Multiple tables were also either discarded or completely revamped for a much smoother flowing business case. For example, the Project Details table was completely discarded from the final ERD of the previous class. The Order table was added as a center entity to keep track of multiple other entities such as Customer, Production, Payment, Installation, Permit, Employee (Designer), and HOAARC tables. Finally, most of the other attributes of every table were completely changed (other than the Employee table and any table that acts as a child table to it such as Designer, Fabricator etc.).

The Backend

From the start of CIS 3365, it was made clear that the backend will be hosted on a remote Microsoft SQL server. To prepare for it, each team member made locally hosted SQL servers, where they would test all of their individual scripts. Finally, for the implementation of the scripts, individual scripts were brought into a single local SQL server and tested. After that the Scripts were simply replicated onto the remotely hosted SQL server with all data being transferred there for testing purposes as well.

The Front-end:

The team knew that they would have to code up a GUI from scratch and somehow be functional with the SQL server hosted database. Luckily, some of the team members were able to pick up GUI programming throughout the class.

For the GUI application the Python PyQT5 module was used. This is a package for any python-based IDE that allows the programmer to make specific UI aspects by using a language the students were already comfortable with.

The Pyodbc module was used to connect the front end to the back end, and once the GUI was at a prototype stage; it was tested with randomly generated data to check for any bugs and troubleshoot beyond that point.

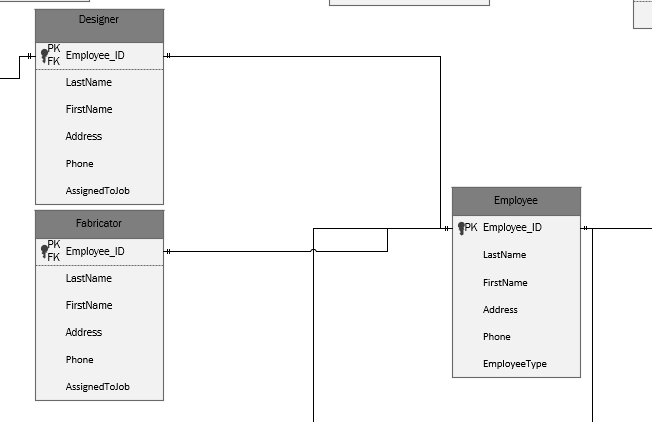
# **Solution:**

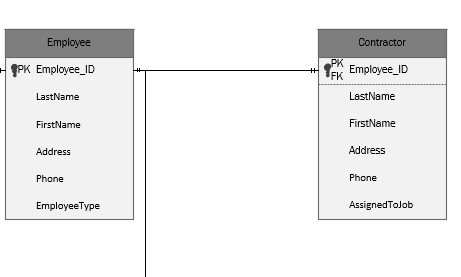
**Key Decisions:**

* Team MSG will need to meet client guidelines and follow their already set business process while also following ERD criteria and legalities
* Each proposed aspect of the project must solve one or more problem that the client is currently having with their business processes
* A GUI application must be created for client to access all database information from
* Team MSG is to assume that the client does not have experience with database management, therefore, the proposed application must be simple and effective

**ERD requirements**

To meet the set of business rules that New Modern sign follows on a daily basis, team Meta Solutions Group created an ERD to diagram how each entity in the work process relates to another. Take for example:



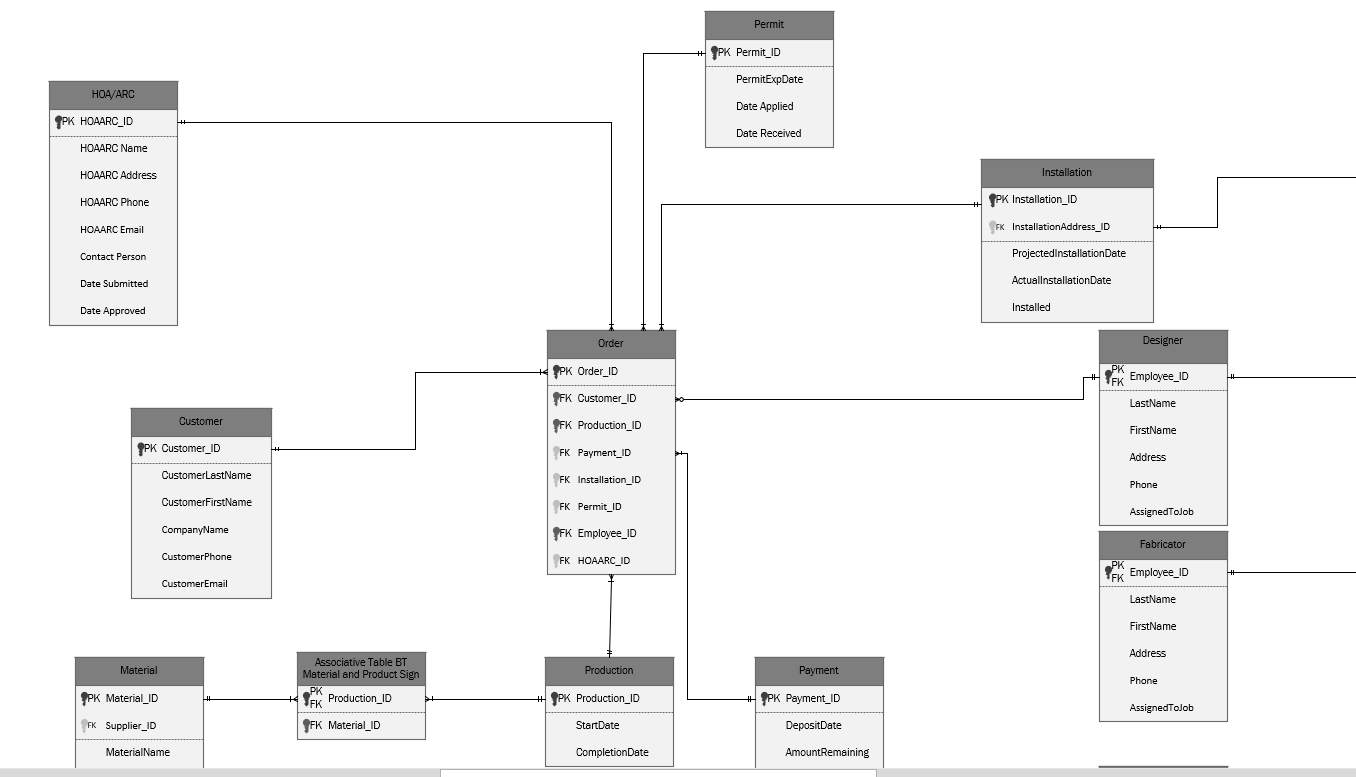


Illustrated above is an example of how different *types* of employees relate to the actual *Employee* entity. The Employee entity; at least in the case of New Modern Sign, holds a hierarchical structure where every Employee has a designated type. So, the Employee entity has several child entities such as: Designer, Fabricator, Welder, Contractor, and finally Driver.

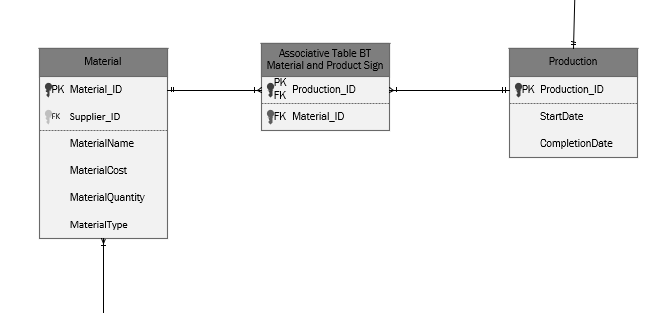
Perhaps the main business rule is the fact that almost everything connects to the Order entity. For example:

* One customer can place many Orders
* One HOA/ARC can be associated with many Orders
* One permit can cover multiple orders (if made by the same customer)
* One Installation job can have multiple orders
* One designer can work on multiple Orders
* One Production job can have multiple orders involved
* One payment can cover multiple orders

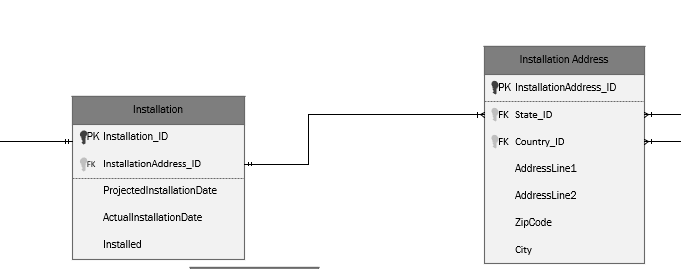
As you can see, the Order entity is in essence the center of almost all activity in New Modern Sign. Therefore, the ERD needs to reflect that:



When creating the ERD, MSG also realized that there was a slight issue in the form of what seemed like a many to many relationship. There was definitely a hidden entity that existed, which took care of that because there shouldn’t be a many to many relationship between entities. However, MSG was not told about such an entity. Therefore, to meet guidelines of the client while also having a legal relationship; an associative table entity was added between the Material and Production entities.



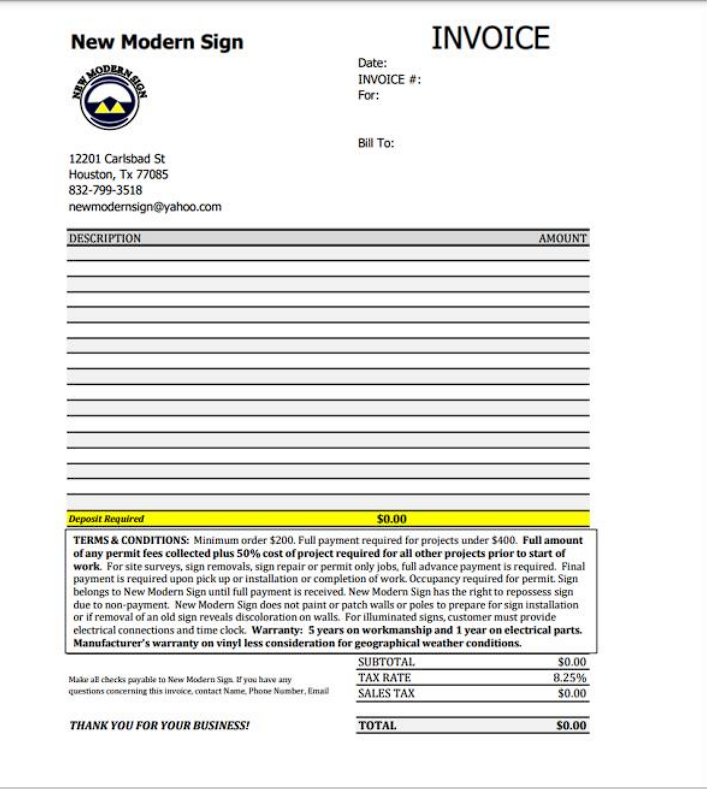
Another complex relationship that was requested by the client is the Installation table to the Installation address table. While it might seem confusing at first, simply put: One Installation job (which in this case is the Installation entity) can have multiple installation addresses (in this case the installation address entity). This is because one customer will most likely have one Installation job (for one set of Orders made by the customer). However, the set of orders from the customer might require installations in different locations. Therefore, both an Installation entity and an Installation Address entity was created



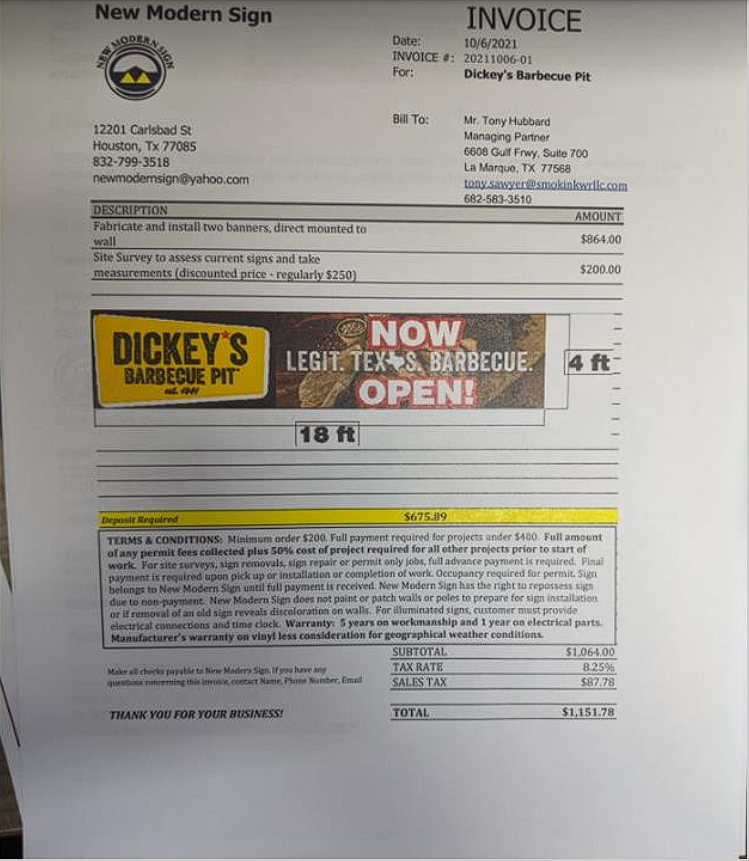
The Full ERD will be given in the appendices section

**Forms and Reports**

Since the beginning of the project, the client made it pretty clear that it did not rely on the use of many forms. The only form that they utilized was a physical form and they insisted that it stay physical only.

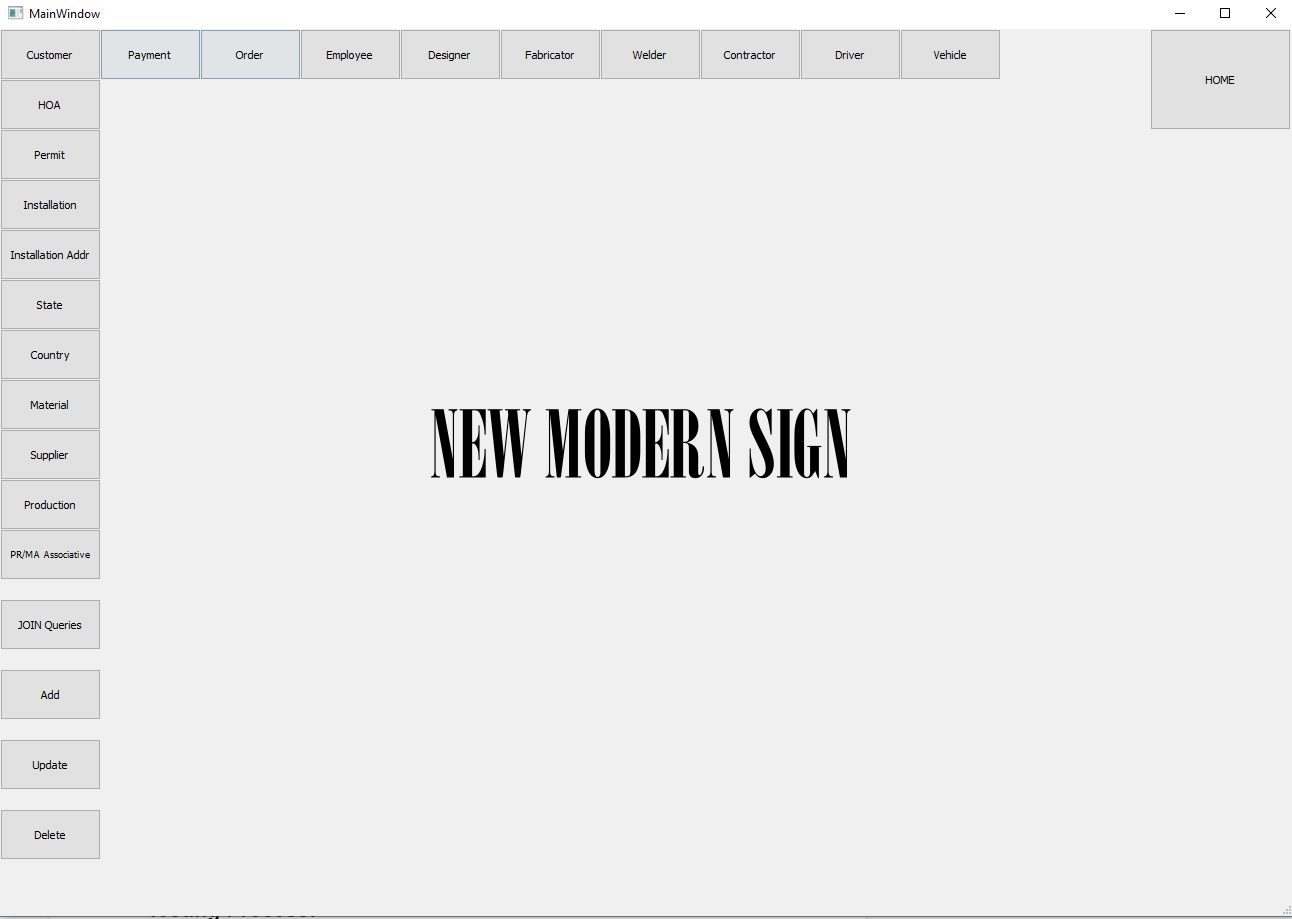


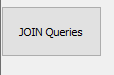
Blank

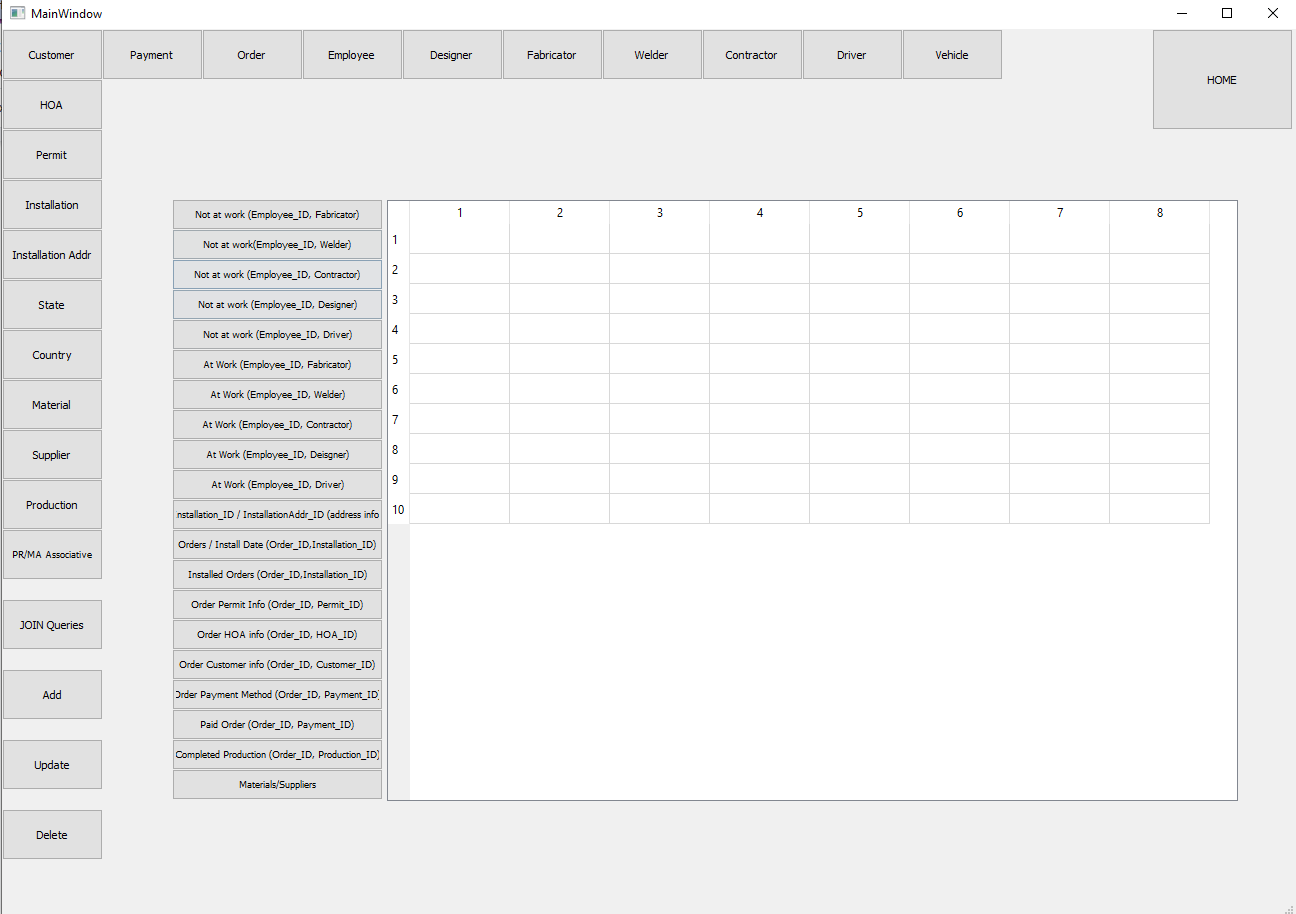


Filled Out

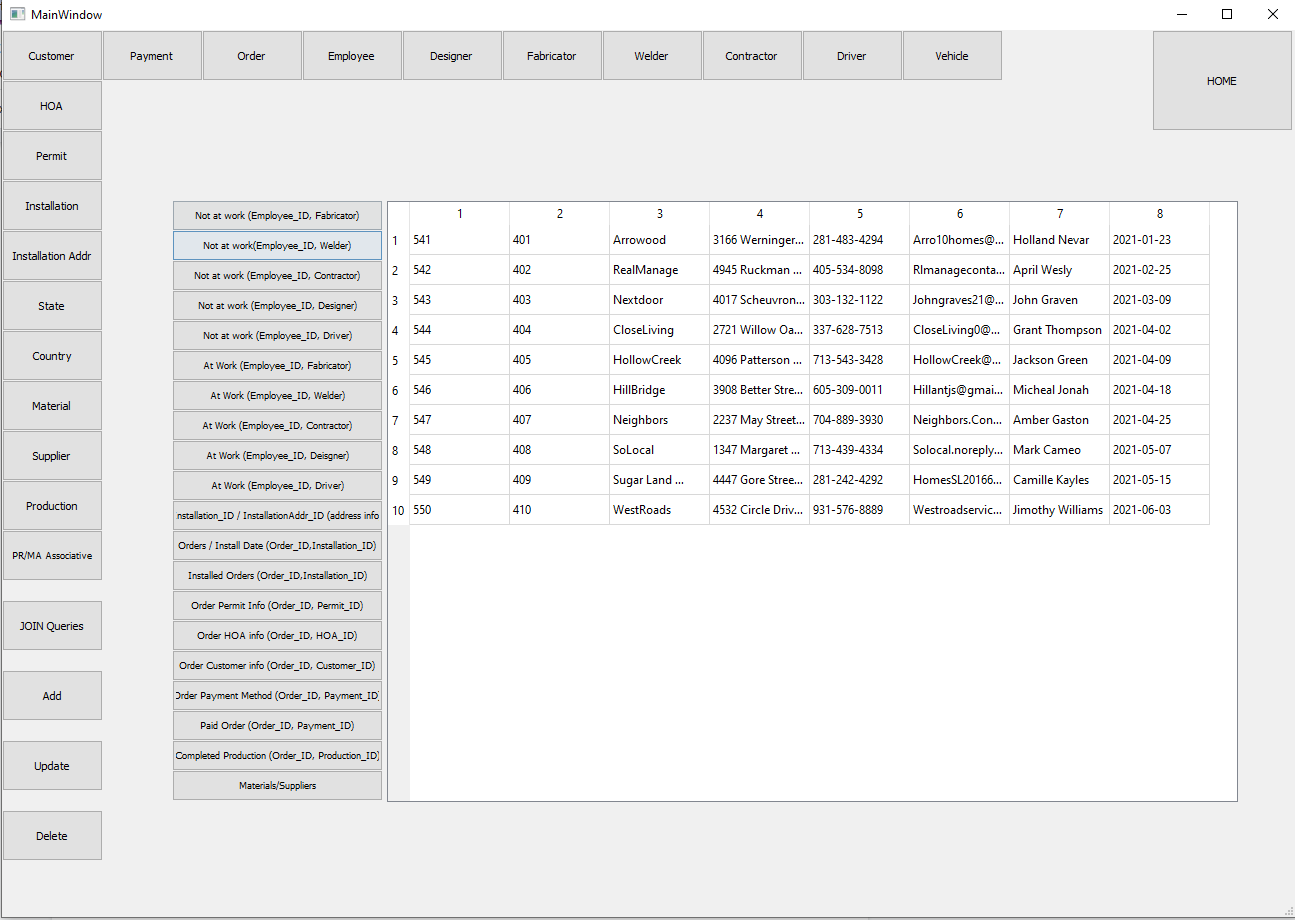
On the other hand, there were several sets of information that the client gave to us which could potentially be utilized in reports. Each of these reports were created using SQL in the form of JOIN queries and each of them provide unique information about entities and their relation to the daily business processes. These reports can be easily accessed in GUI application our team will provide to the client







Showing Order Permit Information (the GUI is a prototype; however, functionality will be the same). The GUI will be submitted with the rest of the project.



# **Testing Process:**

* As for our team’s testing process for the project, we began by installing the necessary programs and components on the client’s computers. During the installation we discovered bugs within their software when wrong data was input while altering the add, delete, and update functions with the GUI.
* To fix these issues our group implemented a try and catch. Where the try executed the function program, and the catch where try executed the function program which catches the error. The error was then programmed so that it does not crash whenever an error is introduced.
* During our consultation meetings with the Professor, when our team was first developing the entity relational database, we had discovered numerous many-to-many relationships that functionally did not make sense to have. With the assistance of Professor Wu, we were able to locate the issues within our diagram. To address the many-to-many relationships within the ERD, we implemented an associative table between those relationships. After alleviating the issues within our ERD our group was able to begin working on the SQL scripts.

# **Project Improvements:**

Priority is given to additional pages. These may include a fully featured login page as typical of most applications. The login page can include a sign-in button and password reset options. There may also be a connection page afterwards which can be used to configure the device or remote server which the application connects to. The current database application is centered around only a main page which holds all widgets.

The second priority is for new widgets and buttons for better ease of access as well as a better-looking UI. The new widgets may include more drop-down lists and scrollbars, etc. The application can also use proper error messages for when user input is incorrect or not expected. As of currently, an error just does nothing. The widgets can be improved to be more responsive with extra functionality such as allowing users to search for specific reports or forms.

The final priority is on improving the look and feel of the GUI. The application may need complementary colors for appearance, widgets can also be recognized so that they provide a cleaner and more logical layout. We may also include additional functions/widgets that are not critical to the purpose of the database application but may provide a more professional feeling GUI. These can include the time, calendar, edit history functions.

# **Project Database Maintenance Issues:**

A huge issue with the database is that it’s not cascading. What this means is that in the database there is redundant data; the only way to delete overlapping data is by going in and doing it manually. Fixing this is quite an issue due to how difficult it is having to manually go in and delete the data that is redundant, we’ve been looking for a solution to solve this problem as a whole.

Another issue with the Maintenance of the Database is that there’s no recovery for the data. Once anything is deleted and removed, it is gone since it isn’t backed up. We’ve been working on finding a way to do a full data backup through the SQL Server so nothing is ever completely lost, but as of now there isn’t any form of recovery.

One of the most excruciating issues is how the GUI crashes when incorrect info is entered. If the input doesn’t match any relevant data in the database, the GUI just breaks down with errors and basically crashes. The problem lies with the execution, if the data isn’t a match the GUI just simply won’t execute.

# **Lessons Learned:**

**What worked:**

Updating the ERD allowed for less congestion in the relationships and created a better flow in the ERD. Changing the attributes of the tables made the ERD more comprehensive and easier to identify relationships. Catching an error in the program helped when inputting data as the software would crash when data was being inputted. Fixing that error helped continue with the GUI. Having issues with the remote connection at first, using local servers allowed us to test scripts and be able to correct them quicker before sharing the code.

**What didn’t work:**

The original ERD had some tables that were not necessary and was not fully comprehensive. The GUI would not execute when inputting wrong data and when changing the functions. Errors would occur and it would eventually crash. There were issues when attempting to delete data as it would not flow through but needing to do it manually. The GUI would not connect to the SQL script outside of the UH network/VPN.

**What to do differently:**

Identify proper paths in the ERD to reduce too many relationships that are not necessary, and make it flow better. Access the secure networks to allow scripts and the GUI to connect properly. Change some aspects of the application to provide a more appealing GUI.

# **Project Summary:**

The final product our team has produced according to the many previous reports and requirements ended up looking much like the prototype our team had showcased for our class and the company New Modern Sign at the end of 2021.

We were able to fulfill all the requirements for this system. Our weekly updates show the progress made as well as some of our decisions and hurdles over this past semester.

With a minor team change halfway through the project, duties were reassigned, and the original design did not change. Minor changes were made from the original ERD plans but thoroughly documented.

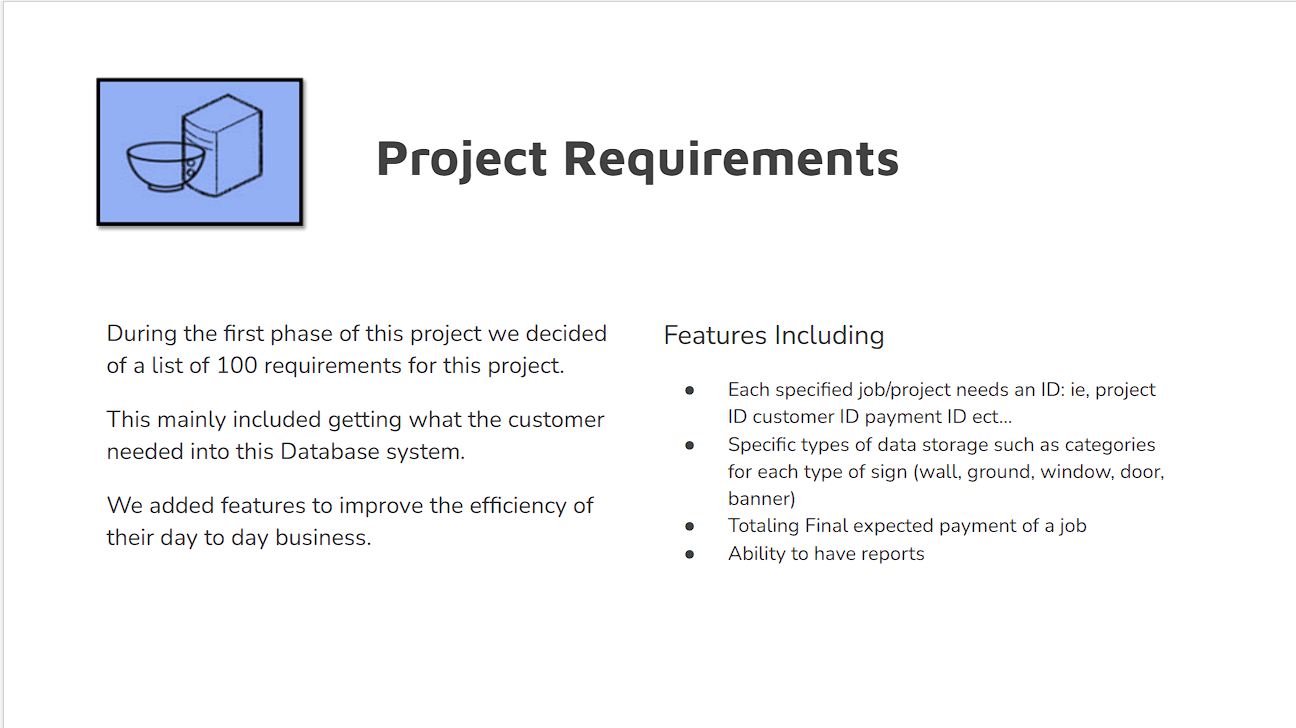
A sleek Modern design was chosen for the GUI and helps with the usability of the product. This simplified design allows us to show the main tables with little distraction. The main page displays all the main tables along with the joined tables. Upon selecting the table of your choice, you will be able to view, edit and add the data as desired.

Our finalized system is built according to the needs of our company New Modern Sign and their needs. We are able to input large amounts of raw data in the form of .csv files then once added to the system the data is easily manipulated. Reports can be designed and shown with ease within the new system. Not only are we able to load data in bulk we also implemented an easy way for individual data to be added within the system so users can replace their old forms with this new organizational system.

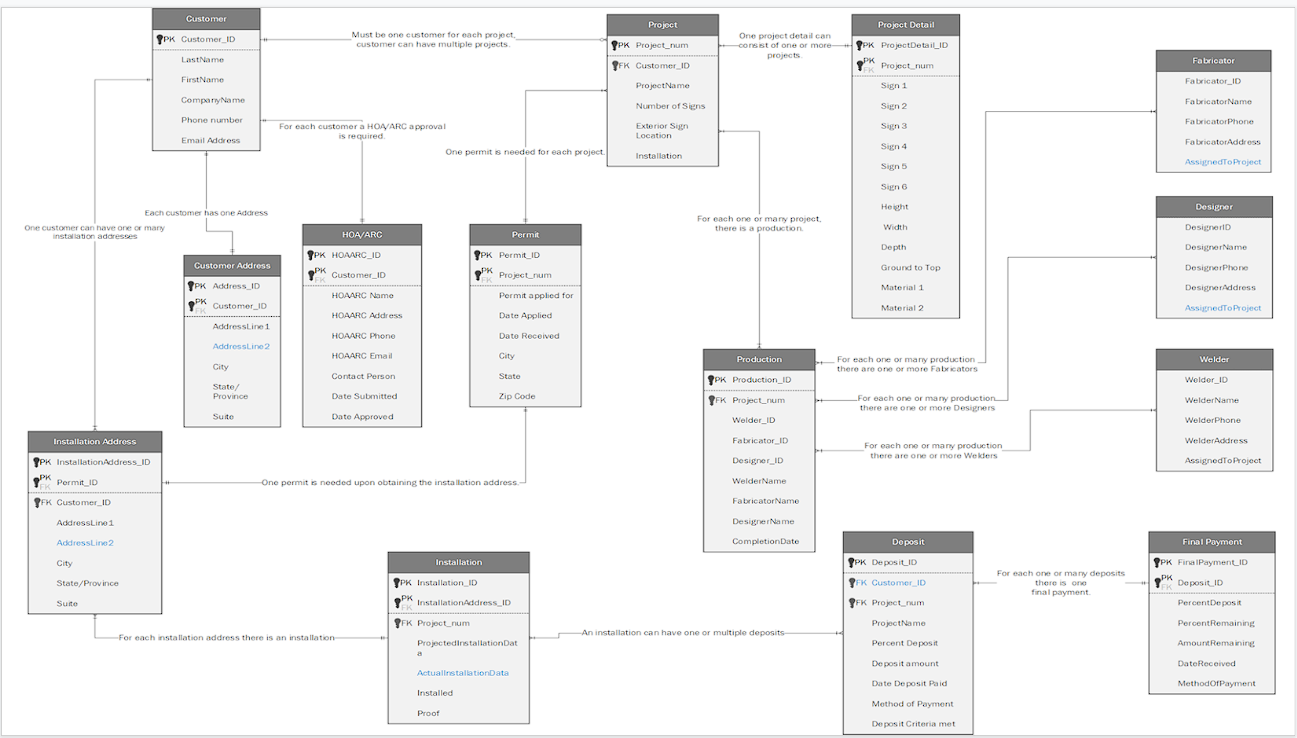
The Meta Solutions Group is proud to present our database solution to the New Modern Sign company. Not only do we hope it is used for years to come, but we also know it will help save time and energy with their data storage solutions as well as making reports on their data in the future.

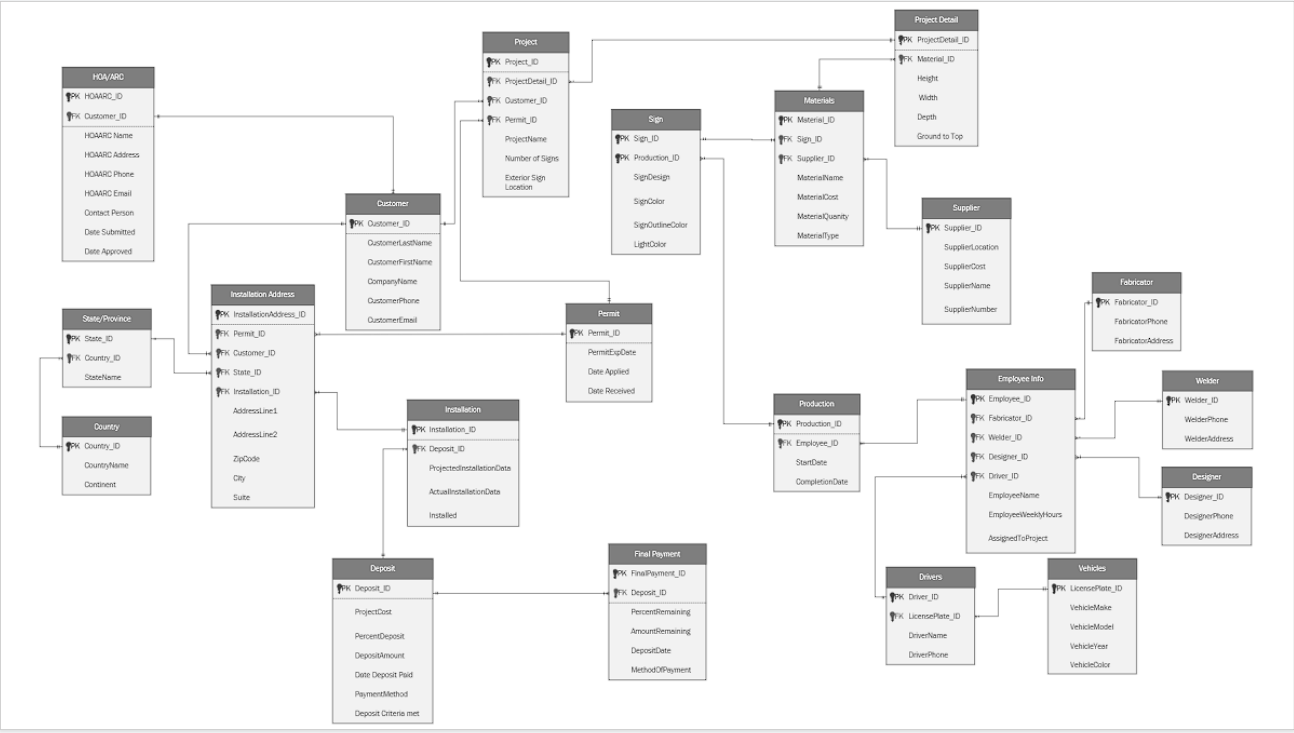
# Presentation Slides

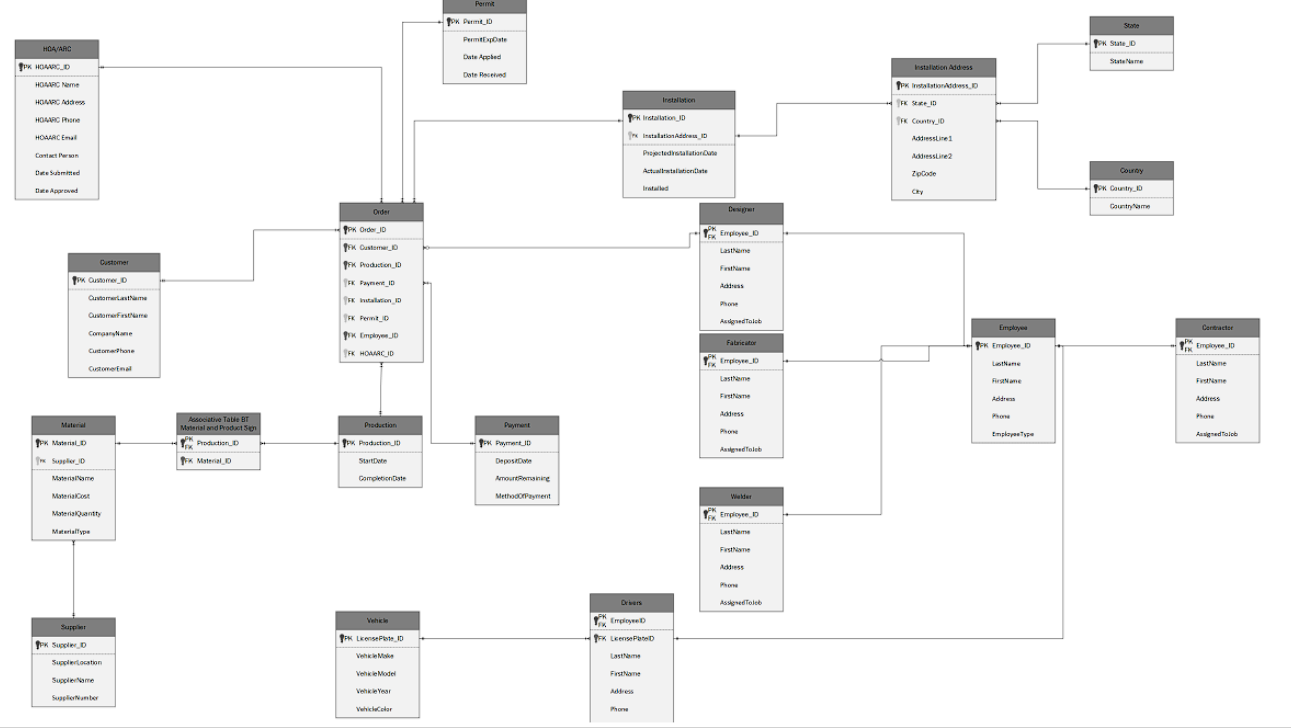


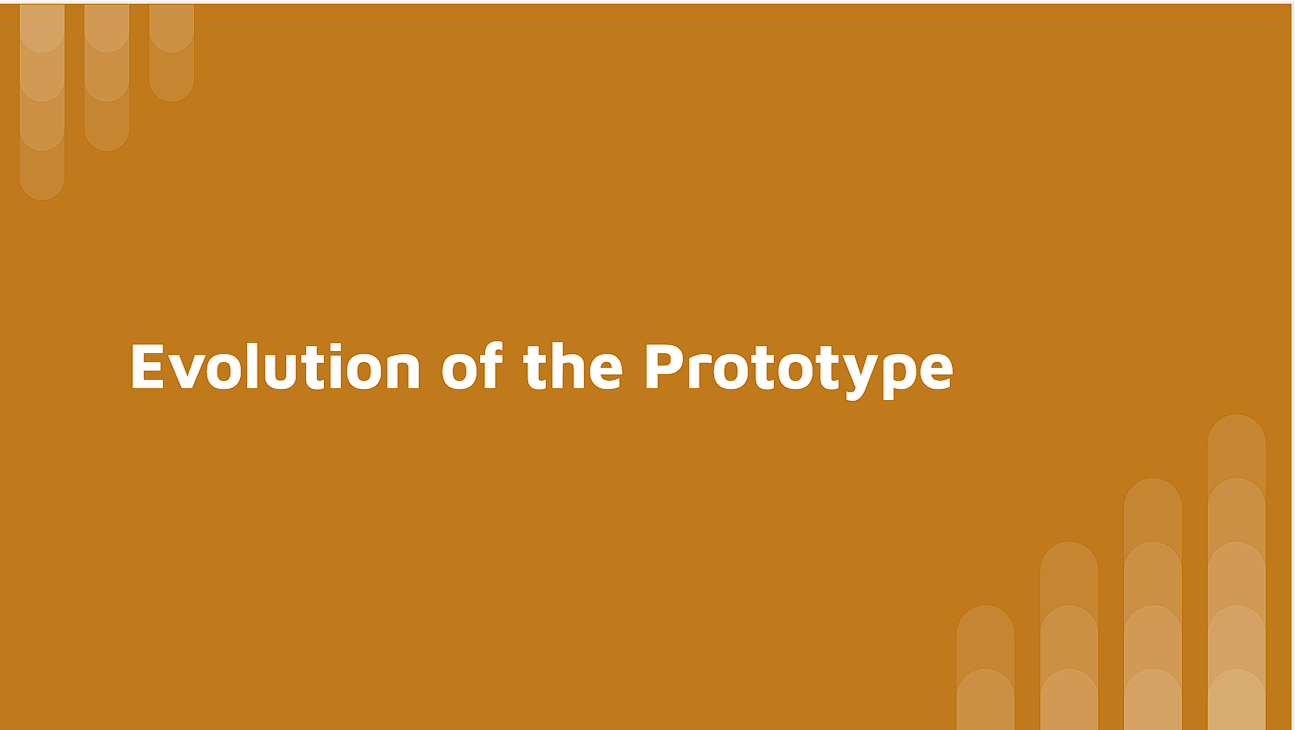


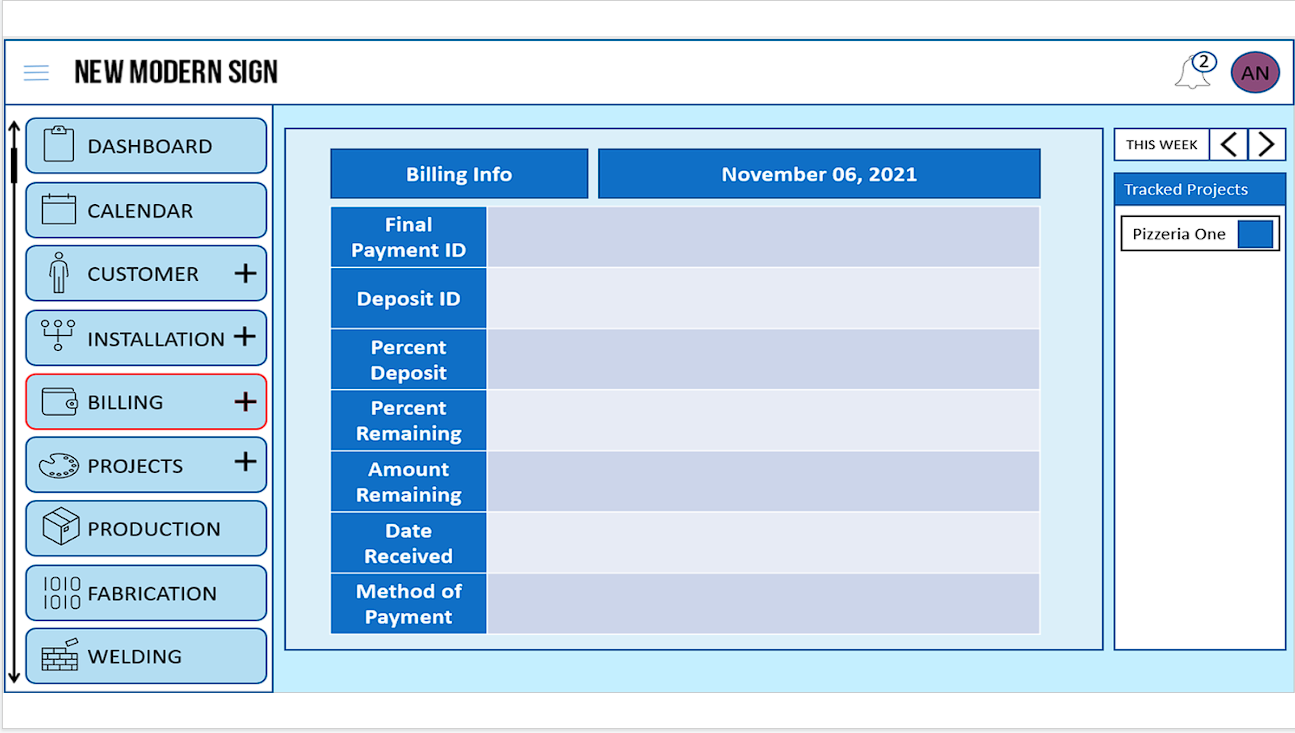


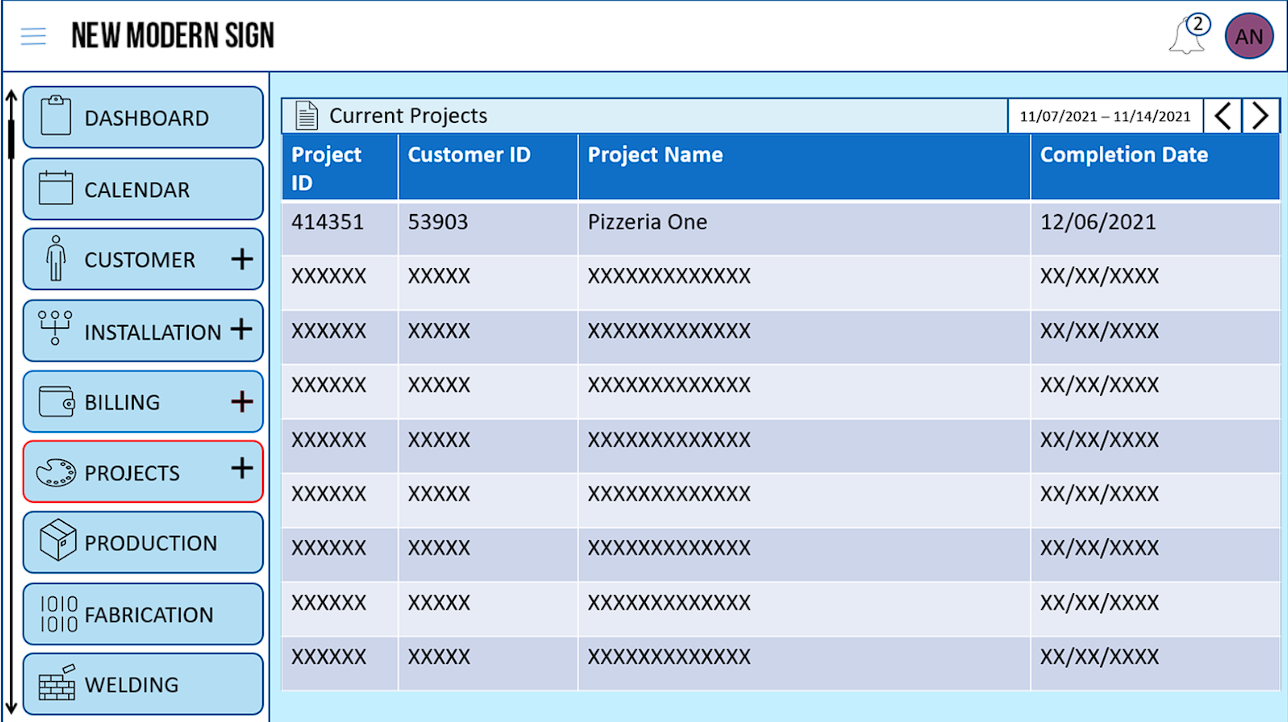


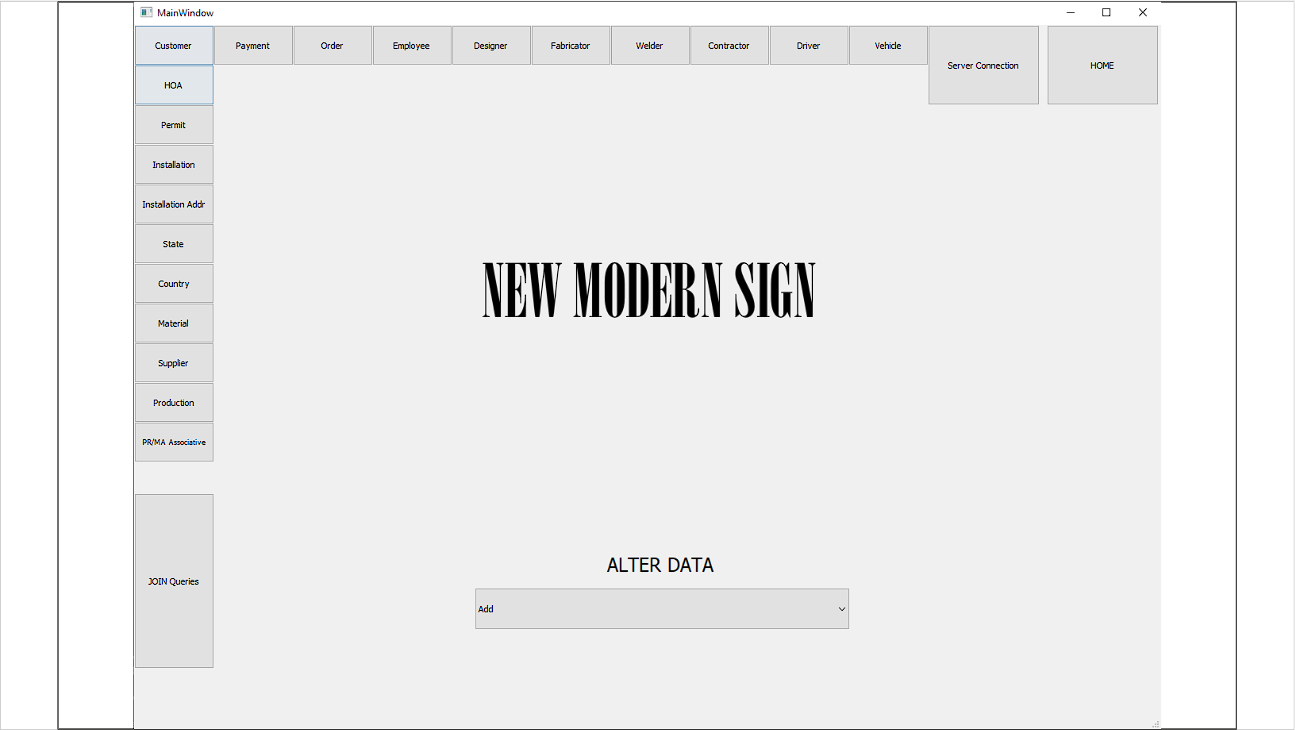


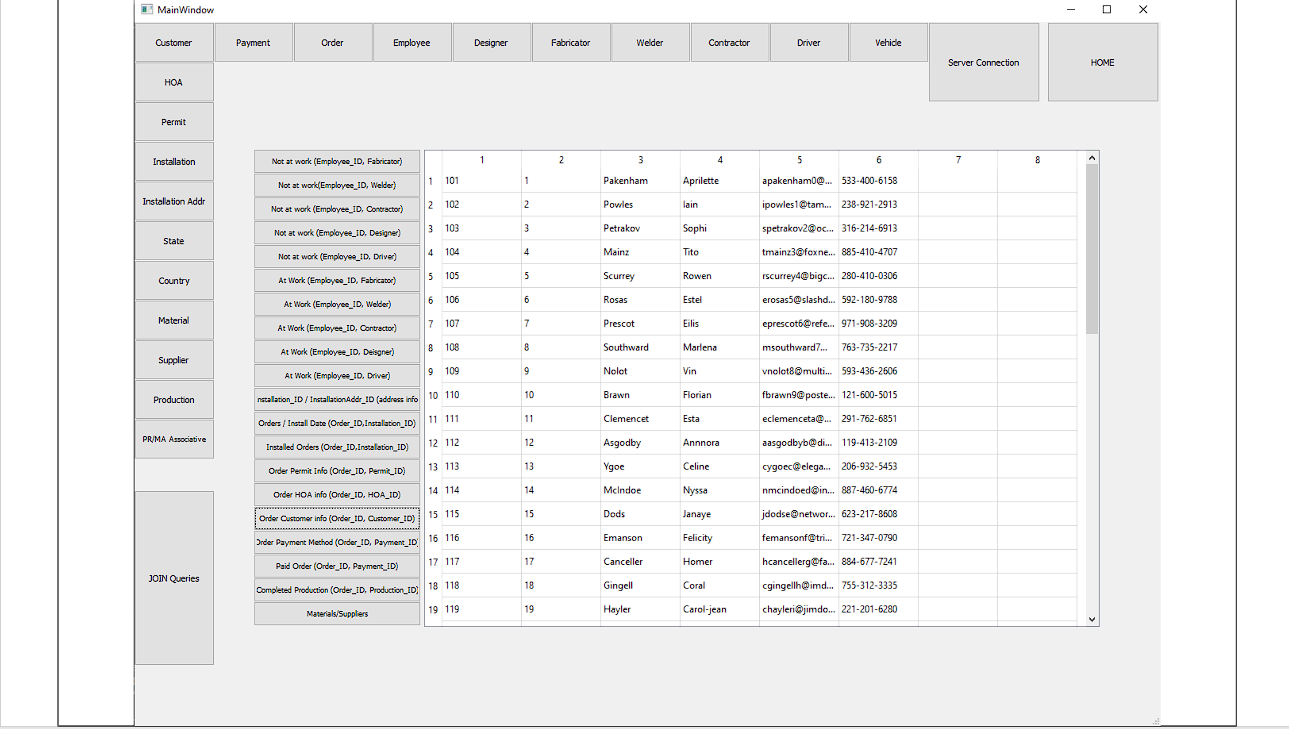


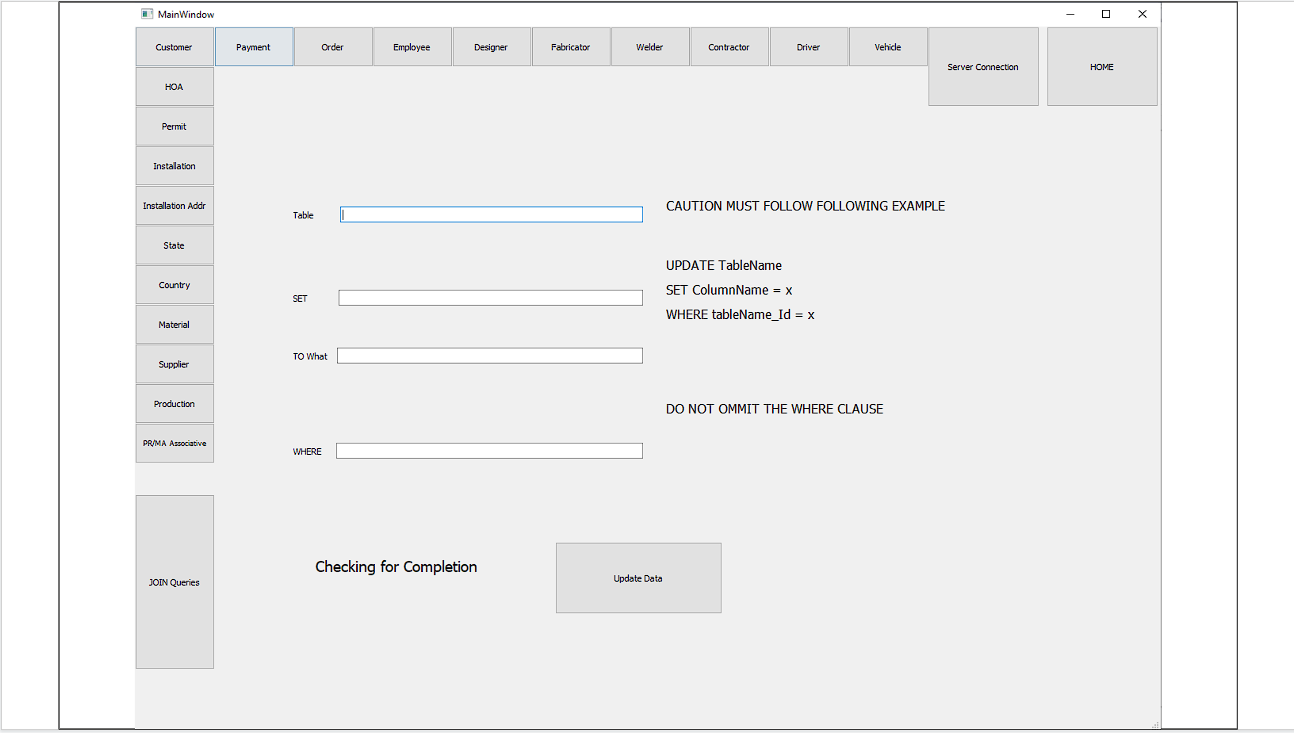


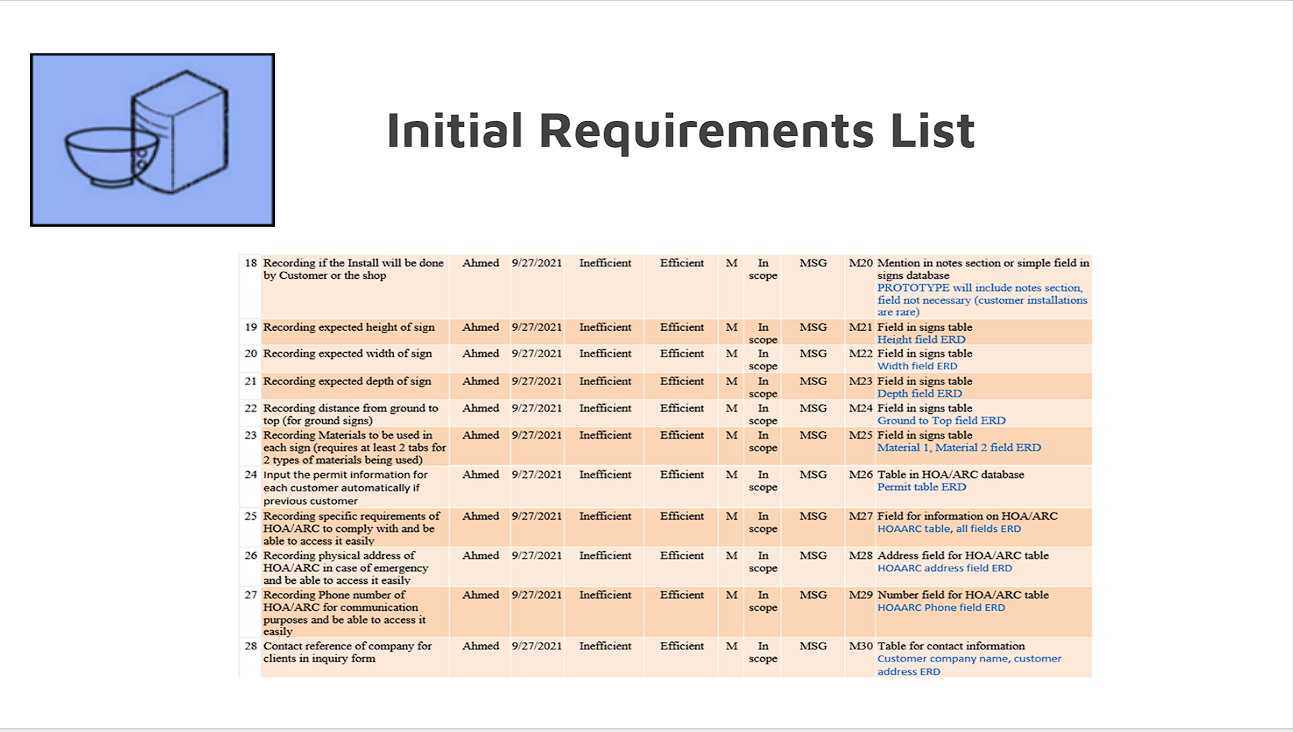


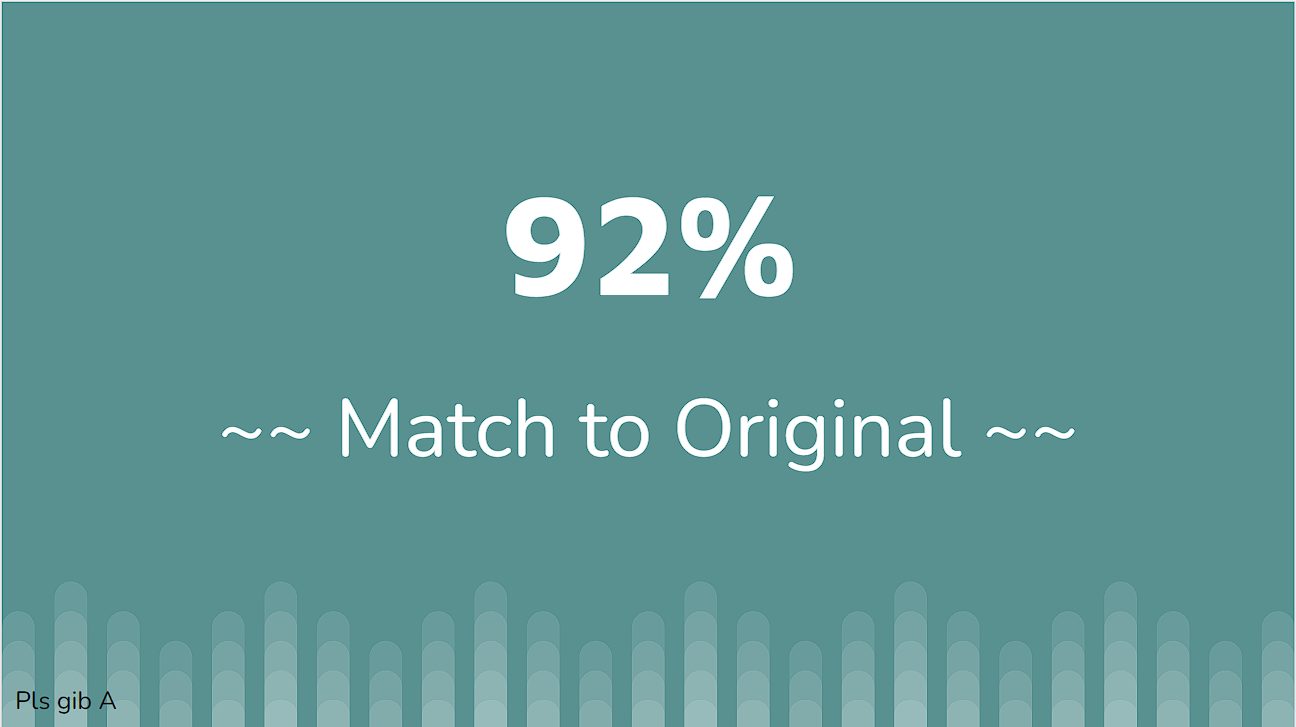
















# **Resources:**

Coronel, C., & Morris, S. (2019). *Database systems: Design, implementation, and management*. Cengage.

David-Engel. (n.d.). *Python SQL driver - PYODBC - python driver for SQL server*. Python driver for SQL Server | Microsoft Docs. Retrieved April 14, 2022, from https://docs.microsoft.com/en-us/sql/connect/python/pyodbc/python-sql-driver-pyodbc?view=sql-server-ver15

MikeRayMSFT. (n.d.). *Data types (transact-SQL) - SQL server*. SQL Server | Microsoft Docs. Retrieved April 14, 2022, from https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver15

PyQt tutorial. (n.d.). Retrieved April 14, 2022, from https://www.tutorialspoint.com/pyqt/index.htm

*PYQT5*. PyPI. (n.d.). Retrieved April 14, 2022, from https://pypi.org/project/PyQt5/

# **Appendices:**

HOAARC Table provides relevant HOAARC information for each order. Each order must have HOA/ARC approval.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| **HOAARC\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID for referring to HOAARC** |
| **HOAARC Name** | **y** | **n** | **50** | **VARCHAR** | **n** | **Name of the HOA** |
| **HOAARC Address** | **y** | **n** | **225** | **VARCHAR** | **n** | **Address of main office of HOA** |
| **HOA Phone** | **y** | **n** | **20** | **VARCHAR** | **n** | **Primary phone of HOA** |
| **HOA Email** | **y** | **n** | **50** | **VARCHAR** | **n** | **Primary email of HOA** |
| **Contact Person** | **y** | **n** | **50** | **VARCHAR** | **n** | **Person to contact for relevant HOA queries** |
| **Date Submitted** | **y** | **n** | **XX/XX/XXXX** | **DATE** | **y** | **Date entered for HOAARC** |
| **Date Approved** | **y** | **y** | **XX/XX/XXXX** | **DATE** | **y** | **Final approval date** |

Customer Table provides the most relevant information of each customer in the database.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| **Customer\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID for referring to each customer** |
| **CustomerLastName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Customer last name** |
| **CustomerFirstName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Customer first name** |
| **CompanyName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Name of company, the customer or order represents** |
| **CustomerPhone** | **y** | **n** | **20** | **VARCHAR** | **n** | **The phone number of customer** |
| **CustomerEmail** | **n** | **y** | **50** | **VARCHAR** | **n** | **Customer primary contact email** |

Supplier table provides information of the supplier for each material in the database.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Supplier\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID for referring to suppliers** |
| **SupplierAddress** | **y** | **n** | **50** | **VARCHAR** | **n** | **Office address of supplier** |
| **SupplierName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Name of supplier** |
| **SupplierPhone** | **y** | **n** | **20** | **VARCHAR** | **n** | **Primary phone number of supplier** |

Material table provides info on all the materials in stock. The business requires this for quick access to material information on each order.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Material\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY** | **y** | **ID for referring to materials** |
| **Supplier\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY** | **y** | **ID for referring to which supplier provides which material** |
| **MaterialName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Name of material** |
| **MaterialCost** | **y** | **y** | **(8, 2)** | **DECIMAL** | **n** | **Cost of material** |
| **MaterialQuantity** | **y** | **y** | **2^31-1** | **INT** | **n** | **How many is in stock** |
| **MaterialType** | **y** | **n** | **50** | **VARCHAR** | **n** | **The material type, what is it made of** |

Production table represents orders placed into production (the designing, welding, and fabricating part of the process).

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Production\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID for referring to order placed in production** |
| **StartDate** | **y** | **n** | **XX/XX/XXXX** | **DATE** | **n** | **Date put into production** |
| **CompletionDate** | **y** | **y** | **XX/XX/XXXX** | **DATE** | **n** | **Date of sign finished** |

Payment table provides deposit and amount information for each order. An order must receive a deposit in order to be placed into production.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Payment\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID for referring to payment** |
| **DepositDate** | **y** | **n** | **XX/XX/XXXX** | **DATE** | **n** | **Date of initial deposit received** |
| **AmountRemaining** | **y** | **y** | **(8, 2)** | **DECIMAL** | **n** | **The amount of money still remaining after deposit** |
| **MethodOfPayment** | **y** | **n** | **20** | **VARCHAR** | **n** | **Customer preferred method of transaction** |

Permit table: each order must have a requested and applied for permit for sign.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Permit\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID referring to each permit** |
| **PermitExpDate** | **n** | **y** | **XX/XX/XXXX** | **DATE** | **n** | **Expected date of permit expiration** |
| **Date Applied** | **y** | **n** | **XX/XX/XXXX** | **DATE** | **n** | **Initial date of permit application** |
| **Date Received** | **y** | **y** | **XX/XX/XXXX** | **DATE** | **n** | **Actual date received for permit** |

State table listing the names of all U.S.A states. We require a state table for repetitive entry in address fields of other tables.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **State\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID of each state** |
| **StateName** | **y** | **n** | **50** | **VARCHAR** | **n** | **The name of each state** |

Country table listing the names of all countries. We require country tables for repetitive entry in address fields of other tables.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Country\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID of each country** |
| **CountryName** | **y** | **n** | **50** | **VARCHAR** | **n** | **The name of each country** |

Installation Address table lists the relevant address information where the sign will be installed.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **InstallationAddress\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID for referring to each installation address** |
| **State\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY** | **n** | **The state for the installation address** |
| **Country\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY** | **n** | **The country for the installation address** |
| **AddressLine1** | **y** | **n** | **225** | **VARCHAR** | **n** | **Primary address** |
| **AddressLine2** | **n** | **y** | **225** | **VARCHAR** | **n** | **For additional address information such as suite or apt.** |
| **ZipCode** | **y** | **n** | **10** | **VARCHAR** | **n** | **Zip Code of installation address** |
| **City** | **y** | **y** | **50** | **VARCHAR** | **n** | **The city of the installation address** |

Installation table lists information on when the installation will be made. Connects with installation address table.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Installation\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID for referring to each installation process** |
| **InstallationAddress\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY** | **n** | **For connecting each installation process to the Installation Address table** |
| **ProjectedInstallationDate** | **n** | **y** | **XX/XX/XXXX** | **DATE** | **n** | **Estimated installation date given to customer** |
| **ActualInstallationDate** | **y** | **y** | **XX/XX/XXXX** | **DATE** | **n** | **Date of actual installation process** |
| **Installed** | **y** | **y** | **0 or 1** | **BIT** | **n** | **Has the order been installed?** |

Vehicle table presents all the vehicles used in the installation process.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **LicensePlate\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID referring to the license plate of each vehicle (Not the actual license plate number)** |
| **VehicleMake** | **y** | **n** | **50** | **VARCHAR** | **n** | **Make of vehicle** |
| **VehicleModel** | **y** | **n** | **50** | **VARCHAR** | **n** | **Model of vehicle** |
| **VehicleYear** | **n** | **y** | **10** | **VARCHAR** | **n** | **Year of vehicle** |
| **VehicleColor** | **n** | **y** | **10** | **VARCHAR** | **n** | **Color of vehicle** |

Employee table lists all the organization’s employees and roles.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee\_ID** | y | n | 2^31-1 | INT, PRIMARY KEY, UNIQUE | y | ID for referring to each employee |
| **LastName** | y | n | 50 | VARCHAR | n | Employee surname |
| **FirstName** | y | n | 50 | VARCHAR | n | Employee given name |
| **Address** | y | n | 225 | VARCHAR | n | Home address of employee |
| **Phone** | y | n | 20 | VARCHAR | n | Primary work phone number of employee |
| **AssignedToJob** | n | y | 0 or 1 | BIT | n | Is the employee currently assigned to any job or role? |

Designer table lists all the employees registered as designers.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **ID referring to designers** |
| **LastName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Employee surname** |
| **FirstName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Employee given name** |
| **Address** | **y** | **n** | **225** | **VARCHAR** | **n** | **Home address of employee** |
| **Phone** | **y** | **n** | **20** | **VARCHAR** | **n** | **Primary work phone number of employee** |
| **AssignedToJob** | **n** | **y** | **0 or 1** | **BIT** | **n** | **Has the employee been assigned to this specific job?** |

Welder table lists all the employees registered as welders.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **ID referring to welders** |
| **LastName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Employee surname** |
| **FirstName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Employee given name** |
| **Address** | **y** | **n** | **225** | **VARCHAR** | **n** | **Home address of employee** |
| **Phone** | **y** | **n** | **20** | **VARCHAR** | **n** | **Primary work phone number of employee** |
| **AssignedToJob** | **n** | **y** | **0 or 1** | **BIT** | **n** | **Has the employee been assigned to this specific job?** |

Fabricator table lists all the employees registered as fabricators.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| **Employee\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **ID referring to fabricators** |
| **LastName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Employee surname** |
| **FirstName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Employee given name** |
| **Address** | **y** | **n** | **225** | **VARCHAR** | **n** | **Home address of employee** |
| **Phone** | **y** | **n** | **20** | **VARCHAR** | **n** | **Primary work phone number of employee** |
| **AssignedToJob** | **n** | **y** | **0 or 1** | **BIT** | **n** | **Has the employee been assigned to this specific job?** |

Contractor table lists all employees registered as contractors.

| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **ID referring to contractors** |
| **LastName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Employee surname** |
| **FirstName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Employee given name** |
| **Address** | **y** | **n** | **225** | **VARCHAR** | **n** | **Home address of employee** |
| **Phone** | **y** | **n** | **20** | **VARCHAR** | **n** | **Primary work phone number of employee** |
| **AssignedToJob** | **n** | **y** | **0 or 1** | **BIT** | **n** | **Has the employee been assigned to this specific job?** |

Driver table lists all the employees assigned as drivers. Each driver uses a specific vehicle, and the table has a connection to the vehicle table.

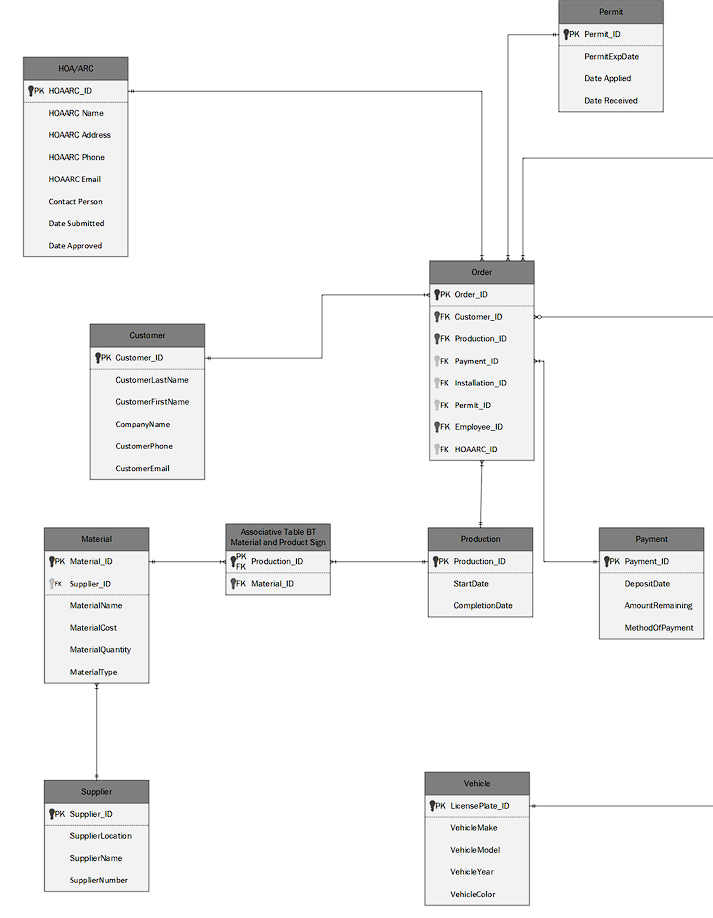
| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **ID referring to drivers** |
| **LicensePlate\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **Each driver uses a specific vehicle** |
| **LastName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Employee surname** |
| **FirstName** | **y** | **n** | **50** | **VARCHAR** | **n** | **Employee given name** |
| **Address** | **y** | **n** | **225** | **VARCHAR** | **n** | **Home address of employee** |
| **Phone** | **y** | **n** | **20** | **VARCHAR** | **n** | **Primary work phone number of employee** |
| **AssignedToJob** | **n** | **y** | **0 or 1** | **BIT** | **n** | **Has the employee been assigned to this specific job?** |

Order table is the primary table referring to each order. It provides a connection to most of the primary tables in the database.

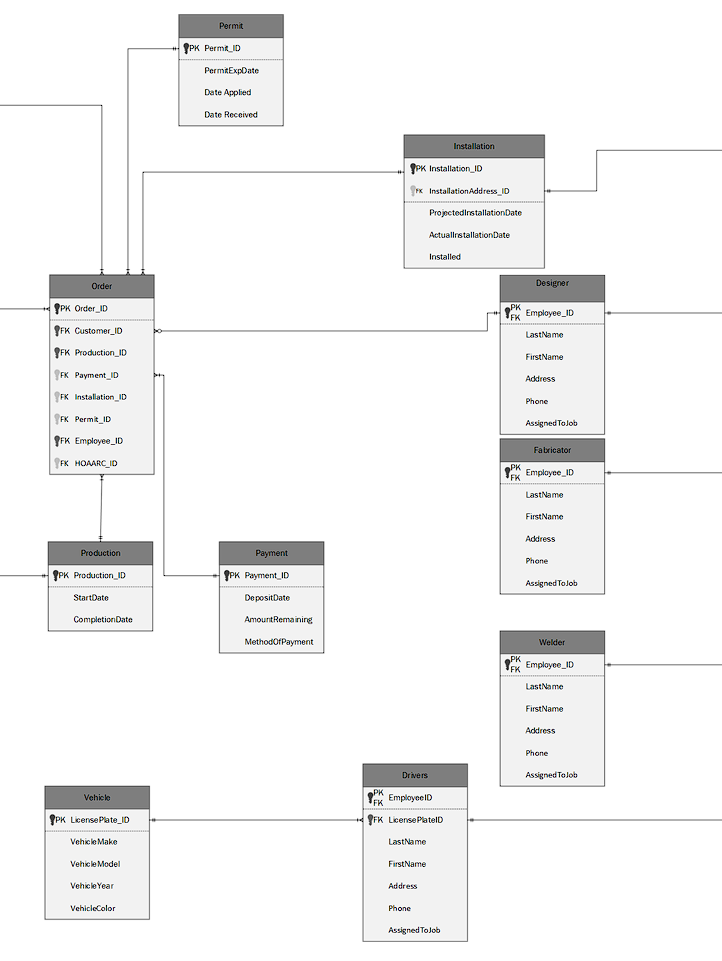
| **Attribute** | **Required?** | **Nullable?** | **Domain** | **Constraints** | **Cascade?** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Order\_ID** | **y** | **n** | **2^31-1** | **INT, PRIMARY KEY, UNIQUE** | **y** | **ID referring to each order** |
| **HOAARC\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **HOA of each order** |
| **Customer\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **Customer who placed each order** |
| **Permit\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **Permit information of each order** |
| **Installation\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **Installation information of each order** |
| **Employee\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY** | **y** | **Employee (designer) initially designed to each order** |
| **Production\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **Production state of each order** |
| **Payment\_ID** | **y** | **n** | **2^31-1** | **INT, FOREIGN KEY, UNIQUE** | **y** | **Payment state of each order** |

**ERD with all attributes and relationships**

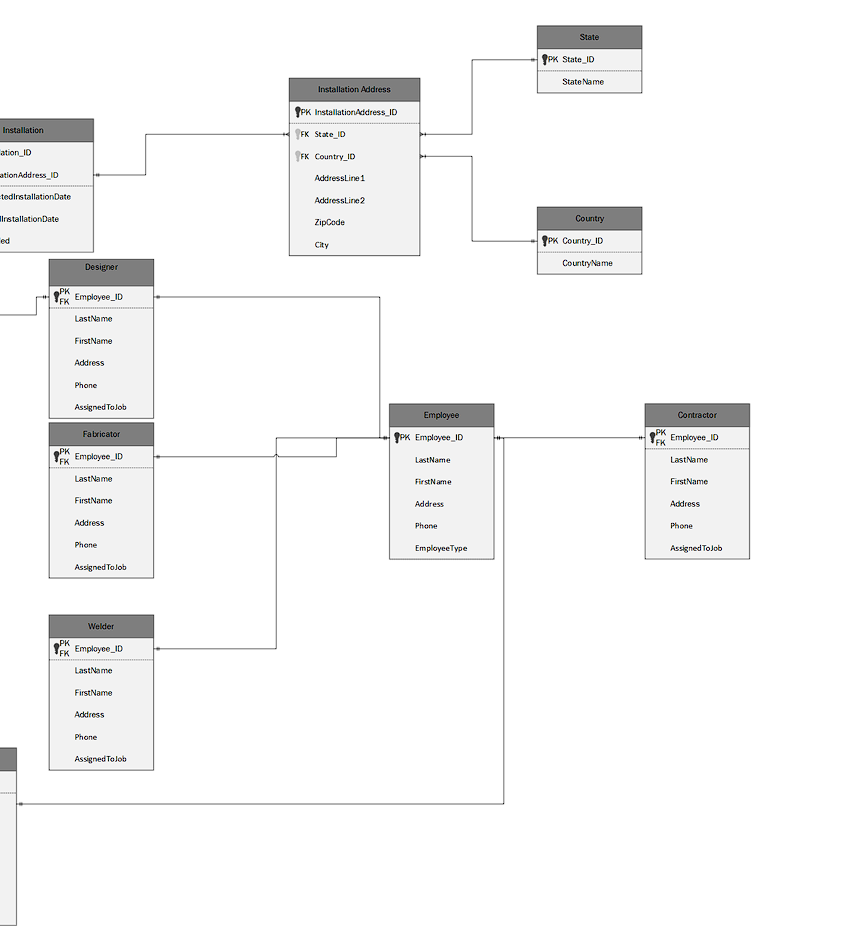
***LEFT SIDE***

****

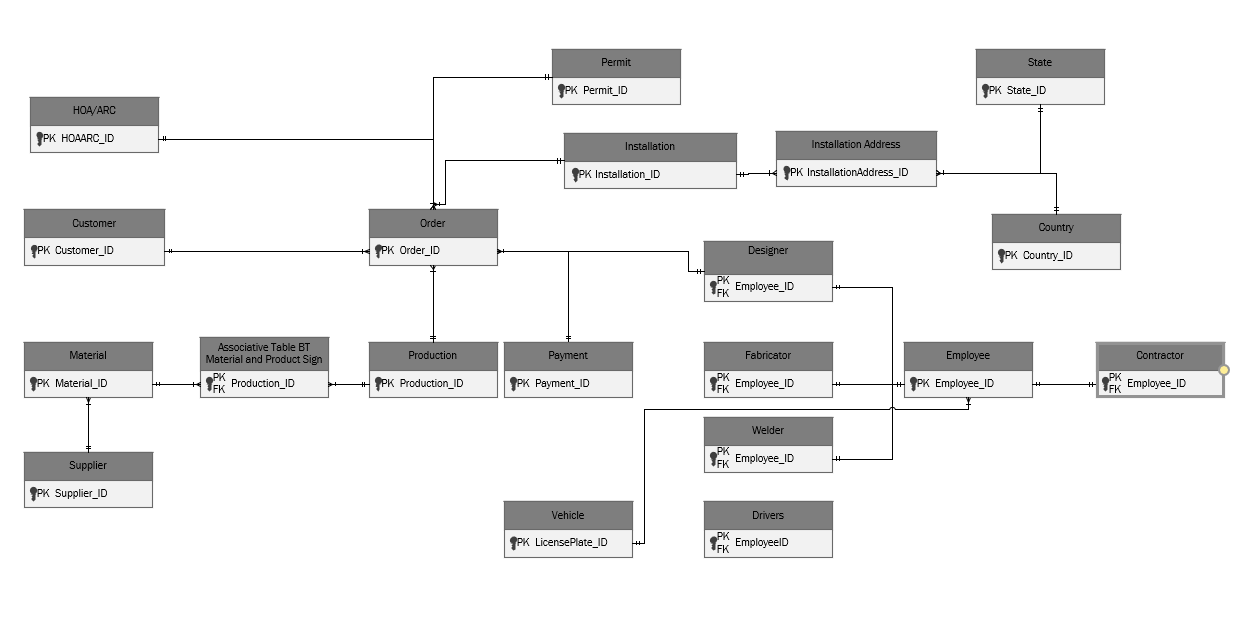
***MIDDLE***

****

***RIGHT SIDE***

****

**ERD WITH JUST PRIMARY KEYS AND RELATIONSHIPS**

****

**FINAL PROBLEMS AND REQUIREMENTS WITH SOLUTIONS**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**STATUS REPORTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2/18/2022 | Deliverable Description | Start Time | Hours Worked | Estimated Hours Remaining |
| Ahmed Rahman | Divided tasks between team members. Currently, figuring out a platform to build GUI upon, and overlooking progress of the documentation side of the project. Also, experimenting with GUI creation using python | 1/27/2022 | 7 | 50 |
| Shafin Momin | Has been assigned to overlook progress of the documentation side of the project. Finalizing ERD, data dictionary. | 1/27/2022 | 7 | 50 |
| Kameron Mason | Has been assigned to aid in finalizing the ERD, data dictionary | 1/27/2022 | 7 | 50 |
| Peter Nguyen | Has been assigned to overlook progress of the documentation side of the project. Finalizing ERD, data dictionary. | 1/27/2022 | 7 | 50 |
| Krishnali Patel | Has been assigned to aid in finalizing the ERD, data dictionary | 1/27/2022 | 7 | 50 |
| Andy Nguyen | Currently working on experimenting with the GUI creation, and is learning on how to connect the backend to the GUI using python | 1/27/2022 | 7 | 50 |
| Mohammad Ghafour | Currently working on experimenting with the GUI creation using python | 1/27/2022 | 7 | 50 |
| Taqi Syed | Currently aiding with finalizing the ERD, and data dictionary | 1/27/2022 | 7 | 50 |
| Schelling Hibbard | Gathering additional helpful resources for GUI, and SQL scripting. Also, experimenting with GUI creation using python, and SQL scripting by using access as the experiment platform | 1/27/2022 | 7 | 50 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3/4/2022 | Deliverable Description | Start Time | Hours Worked | Estimated Hours Remaining |
| Ahmed Rahman | Working on individual SQL and data reports. Also, experimenting with GUI creation using python | 1/27/2022 | 15 | 42 |
| Shafin Momin | Has been assigned to overlook the progress of the documentation. Working on the individual SQL and data reports, and re assessing the ERD. | 1/27/2022 | 15 | 42 |
| Kameron Mason | Re assessing the ERD, working on individual SQL and Data reports | 1/27/2022 | 15 | 42 |
| Peter Nguyen | Has been assigned to overlook progress of the documentation. Re assessing the ERD, working on the individual SQL and data reports | 1/27/2022 | 15 | 42 |
| Krishnali Patel | Aiding in re assessing the ERD, working on individual data and SQL reports | 1/27/2022 | 15 | 42 |
| Andy Nguyen | Working mostly on the GUI creation, and working on data and SQL reports. | 1/27/2022 | 15 | 42 |
| Mohammad Ghafour | Working on individual data and SQL reports | 1/27/2022 | 15 | 42 |
| Taqi Syed | Aiding in re assessing the ERD, also working on individual data and SQL report. | 1/27/2022 | 15 | 42 |
| Schelling Hibbard | Gathering additional helpful resources for GUI, and SQL scripting. Working on the individual data and SQL report | 1/27/2022 | 15 | 42 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3/11/2022 | Deliverable Description | Start Time | Hours Worked | Estimated Hours Remaining |
| Ahmed Rahman | Working on individual SQL and data reports. Also, experimenting with GUI creation using python | 1/27/2022 | 17 | 40 |
| Shafin Momin | Has been assigned to overlook the progress of the documentation. Working on the individual SQL and data reports, and re assessing the ERD. | 1/27/2022 | 17 | 40 |
| Kameron Mason | Re assessing the ERD, working on individual SQL and Data reports | 1/27/2022 | 17 | 40 |
| Peter Nguyen | Has been assigned to overlook progress of the documentation. Re assessing the ERD, working on the individual SQL and data reports | 1/27/2022 | 17 | 40 |
| Krishnali Patel | Aiding in re assessing the ERD, working on individual data and SQL reports | 1/27/2022 | 17 | 40 |
| Andy Nguyen | Working mostly on the GUI creation, and working on data and SQL reports. | 1/27/2022 | 17 | 40 |
| Mohammad Ghafour | Working on individual data and SQL reports | 1/27/2022 | 17 | 40 |
| Taqi Syed | Aiding in re assessing the ERD, also working on individual data and SQL report. | 1/27/2022 | 17 | 40 |
| Schelling Hibbard | Gathering additional helpful resources for GUI, and SQL scripting. Working on the individual data and SQL report | 1/27/2022 | 17 | 40 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3/25/2022 | Deliverable Description | Start Time | Hours Worked | Estimated Hours Remaining |
| Ahmed Rahman | Finalizing SQL scripts and entering them into the server as instructed. Also, experimenting with GUI | 1/27/2022 | 32 | 25 |
| Shafin Momin | Finalizing ERD along with Peter. Aiding with SQL scripts insertion into server | 1/27/2022 | 32 | 25 |
| Kameron Mason | Aiding with SQL script insertion | 1/27/2022 | 32 | 25 |
| Peter Nguyen | Finalizing ERD along with Shafin. Aiding with SQL scripts insertion into server | 1/27/2022 | 32 | 25 |
| Krishnali Patel | I’m not sure, have not been able to get anything from her. | 1/27/2022 | 17 | 40 |
| Andy Nguyen | Working mostly on the GUI creation. | 1/27/2022 | 32 | 25 |
| Mohammad Ghafour | Aiding with documentation and creation of presentations | 1/27/2022 | 32 | 25 |
| Taqi Syed | Aiding with documentation and creation of presentations | 1/27/2022 | 32 | 25 |
| Schelling Hibbard | Working on the GUI creation and connection alongside Andy | 1/27/2022 | 32 | 25 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4/1/2022 | Deliverable Description | Start Time | Hours Worked | Estimated Hours Remaining |
| Ahmed Rahman | Working on the GUI such as setting up different pages, iterating through each different page, loading in data on each page, adding/removing data | 1/27/2022 | 42 | 15 |
| Shafin Momin | Working on documentation such as powerpoint presentation, and binder | 1/27/2022 | 42 | 15 |
| Kameron Mason | Working on documentation such as powerpoint presentation, and binder | 1/27/2022 | 32 | 25 |
| Peter Nguyen | Working on documentation such as powerpoint presentation, and binder | 1/27/2022 | 42 | 15 |
| Krishnali Patel | I’m not sure, have not been able to get anything from her. | 1/27/2022 | 20 | ?? |
| Andy Nguyen | Working on the GUI such as setting up different pages, iterating through each different page, loading in data on each page, adding/removing data | 1/27/2022 | 42 | 15 |
| Mohammad Ghafour | Aiding with documentation and creation of presentations | 1/27/2022 | 35 | 5 |
| Taqi Syed | Aiding with documentation and creation of presentations | 1/27/2022 | 35 | 5 |
| Schelling Hibbard | Overseeing the assignment of tasks on documentation side, as well as setting up the presentation itself. | 1/27/2022 | 42 | 15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4/11/2022 | Deliverable Description | Start Time | Hours Worked | Estimated Hours Remaining |
| Ahmed Rahman | The GUI was approved by the professor and was connected to the remote server. Now, overseeing the completion of the rest of the project and individual reports | 1/27/2022 | 45 | 5 |
| Shafin Momin | Working on documentation such as powerpoint presentation, and binder. Also working on individual report | 1/27/2022 | 45 | 5 |
| Kameron Mason | Working on documentation such as powerpoint presentation, and binder. Also Working on individual report | 1/27/2022 | 45 | 5 |
| Peter Nguyen | Working on documentation such as powerpoint presentation, and binder. Also working on individual report | 1/27/2022 | 45 | 5 |
| Krishnali Patel | Working on individual project report and helping format the binder for the documentation part of the project | 1/27/2022 | 45 | 5 |
| Andy Nguyen | Helping with the creation of more test data. Will need to create 50 rows of data for each table. Also working on individual report | 1/27/2022 | 45 | 5 |
| Mohammad Ghafour | Aiding with documentation and creation of presentations. Also working with individual report | 1/27/2022 | 45 | 5 |
| Taqi Syed | Aiding with documentation and creation of presentations. Also working on individual report | 1/27/2022 | 45 | 5 |
| Schelling Hibbard | Overseeing the assignment of tasks on documentation side, as well as setting up the presentation itself. Also working on individual report | 1/27/2022 | 45 | 5 |

|  |  |
| --- | --- |
| TEAM MEMBER | OVERALL NUMBER OF HOURS |
| Ahmed Rahman | 50 hours |
| Shafin Momin | 50 hours |
| Kameron Mason | 50 hours |
| Peter Nguyen | 50 hours |
| Krishnali Patel | 50 hours |
| Andy Nguyen | 50 hours |
| Mohammad Ghafour | 50 hours |
| Taqi Syed | 50 hours |
| Schelling Hibbard | 50 hours |
|  | TOTAL: 450 hours |