**Introduction:**

For the database we chose to design, we decided to model it after a medical office. The reason why we chose a medical/doctor office as a database to create, is we believe designing a database for a medical office will give us a deeper understanding of such a business. This will allow us to understand the relationships between staff, and patients. Giving us an understanding of what it takes to be considered an actual doctor’s office. When it comes to the dataset for this project, a problem that we ran upon was confidentiality. No reasonable and legitimate medical office will just give out patient information to anyone, so to work around this most of the patient information such as address, name, ssn, etc. will be made up. Even though the data for the attributes are made up, the format and information added will be reasonable.

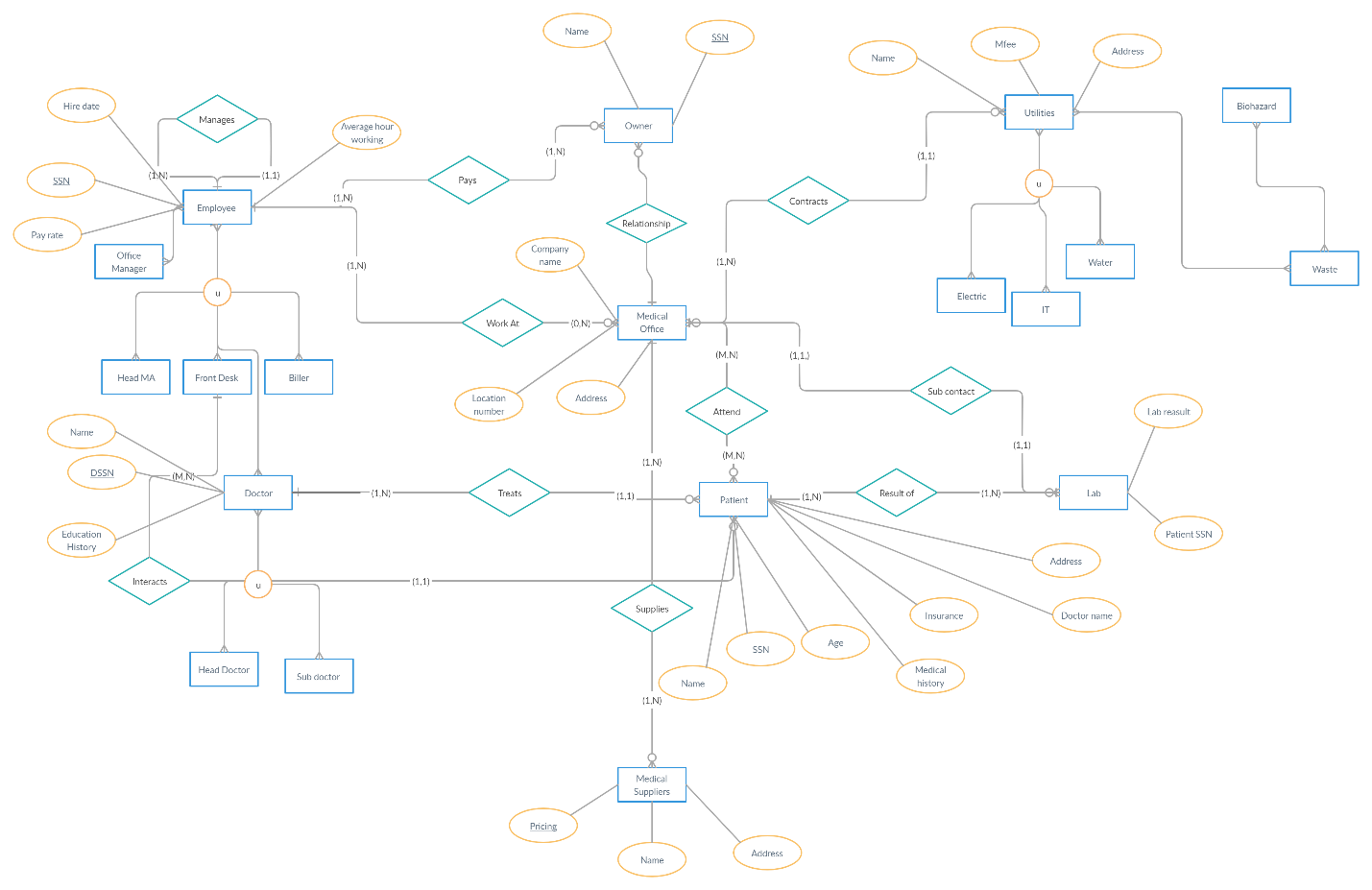
**Database Requirements Analysis:**

What we require our database to do after running queries, is to show the patients and their specific doctor. This could lead to multiple results because there are multiple doctors in a single office, we would like a query into the specific utilities at a medical office, and a query to show employees who makes more than others. Another query we would like to use is to gather which lab company is contracted at differing medical offices. We would also like a query into the doctor’s educational history. Another query we plan to use is gathering all employees who worked with a specific patient. A query where we show which employee or doctor interacted with whatever patient would be interesting to try and solve. What lab result ties to whatever patient is something we will investigate also. There’s plenty of queries that can be used to gather information from this database, and these specific queries listed might be subjected to change later but as for now this is what we’ll be working with.

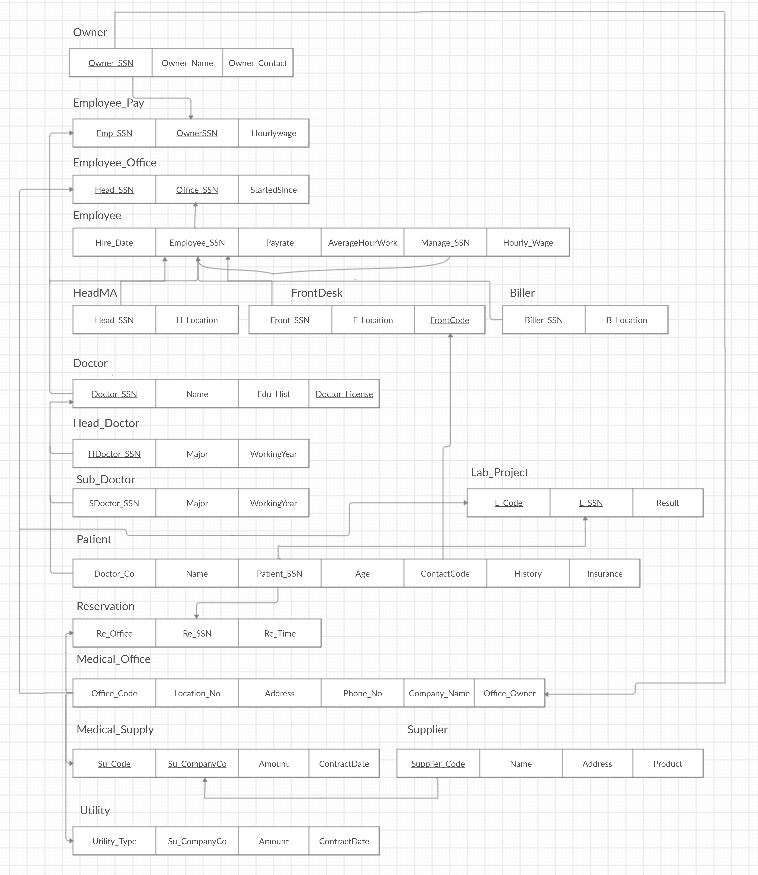
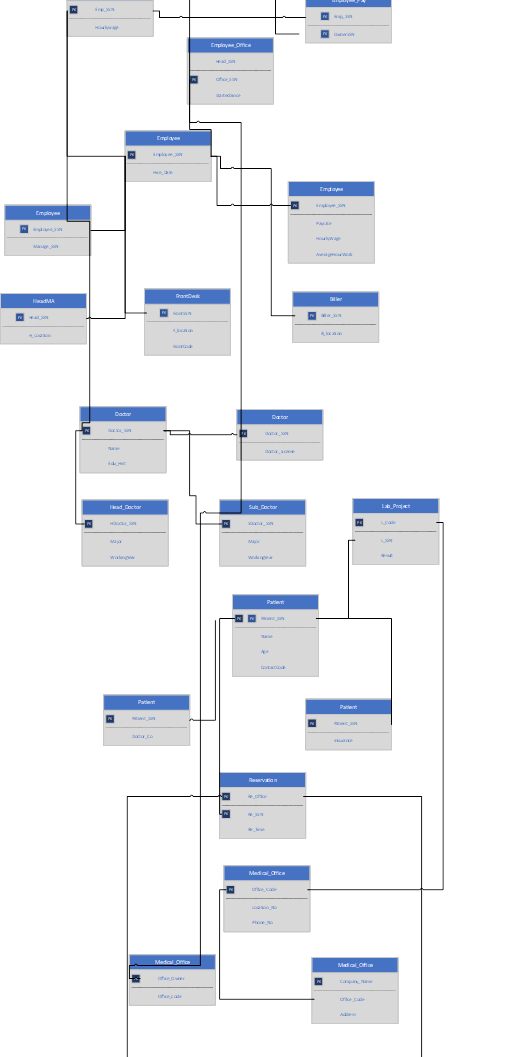
**EER Model:**

When it comes to the entities and relationships in our EER Model we have 19 entities and 11 relationships, with each entity having a specific relationship leaving none out. You have multiple patients who attend a medical office, utilities who contract under multiple medical offices and these utilities have a monthly fee, name, and address. Labs who have a contract to specific medical offices where they have a test result and patient ssn. Patient has the most attributes of all our entities which includes name, ssn, age, medical history, doctor’s name, etc. Employees who work at a medical office, and employees who manages other employees. Doctors treats patients and medical suppliers supply medicine to multiple medical offices. Lastly you have a specific employee who mainly interacts with the patients and that is the front desk staff. So, on our EER model we drew a relationship between front desk employee, and patient with the relation being interacts.

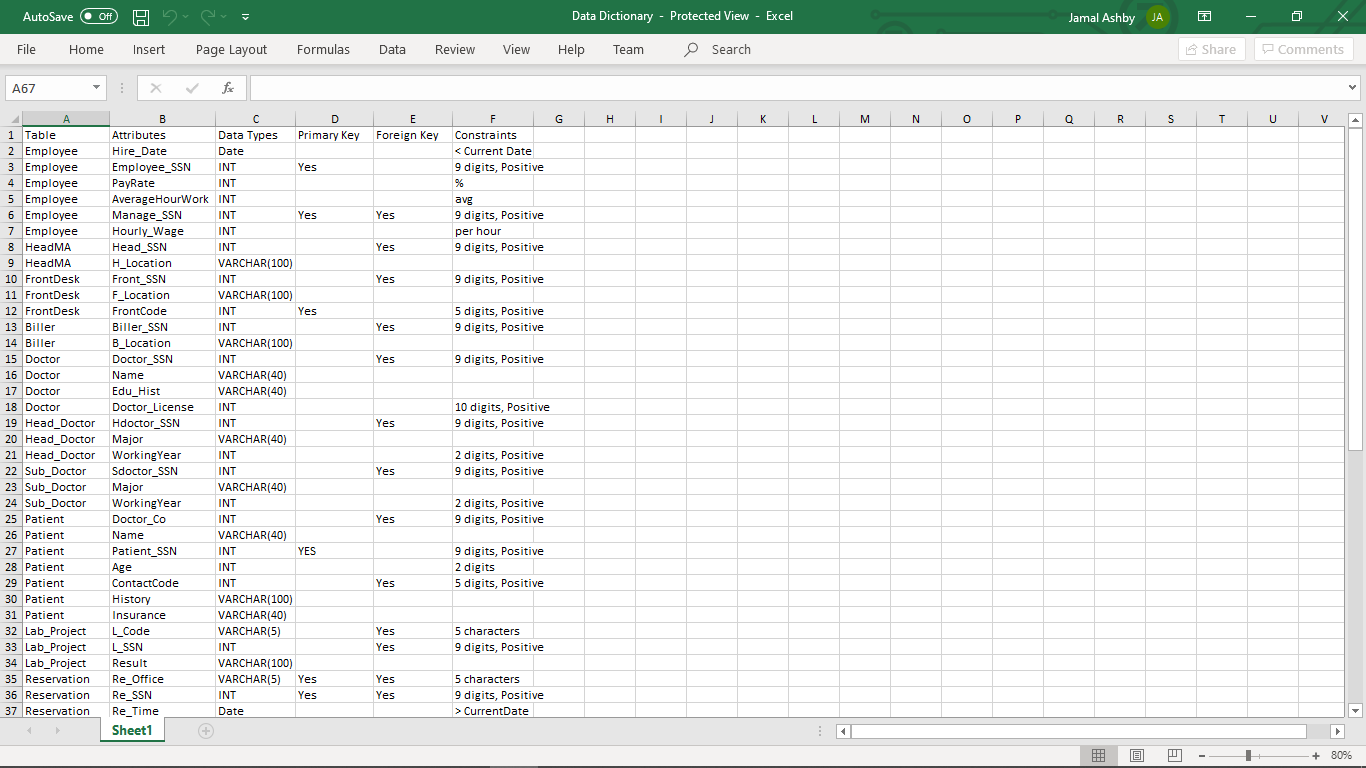
Below is the completed and finalized EER model, the only potential changes we’ll possibly make to it is more than likely just grammatical, such as fixing spelling errors.

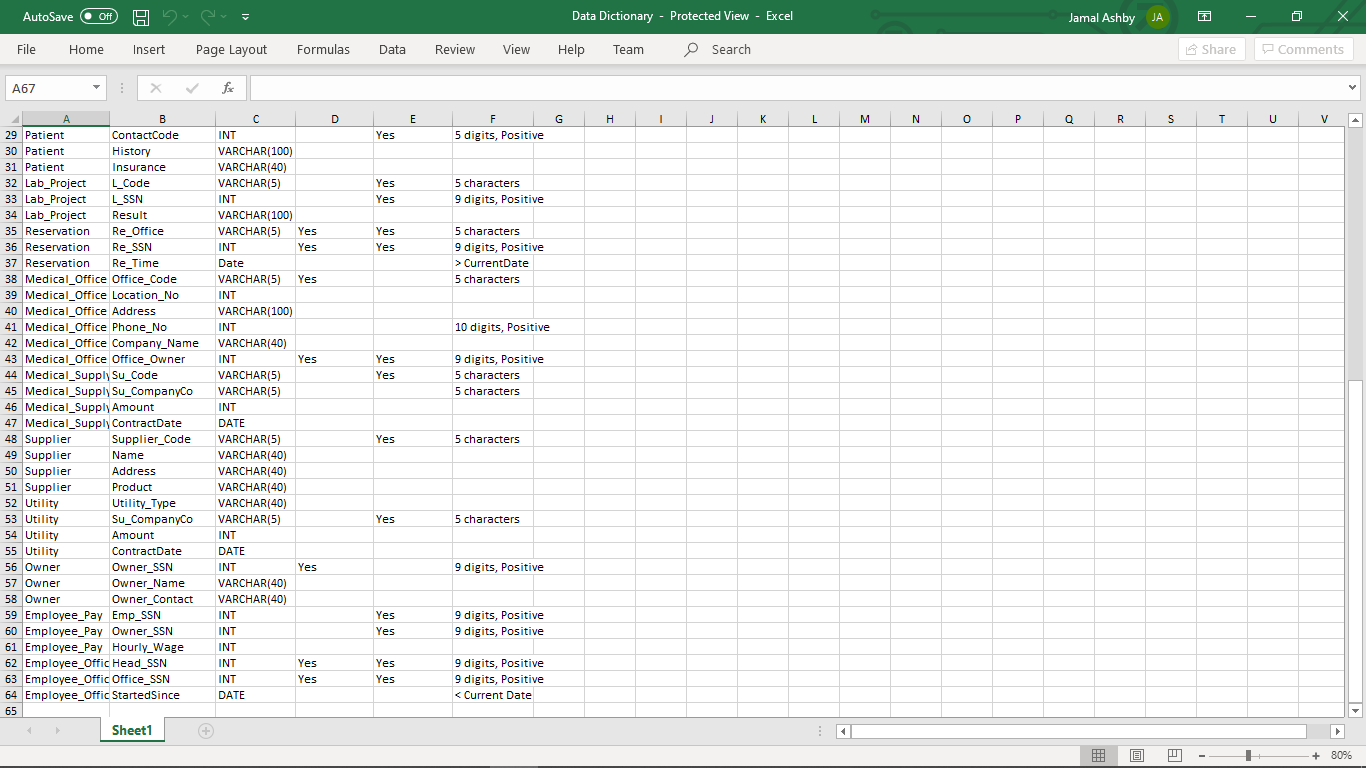
**Relational Model(Mapping & Normalization):**

Below is our schema, the first picture shows the relational model before normalization and the second picture shows the normalized relational model. Some parts of the normalized schema might be cut off, so I made sure to submit the original pdf file that contains it also. :

****

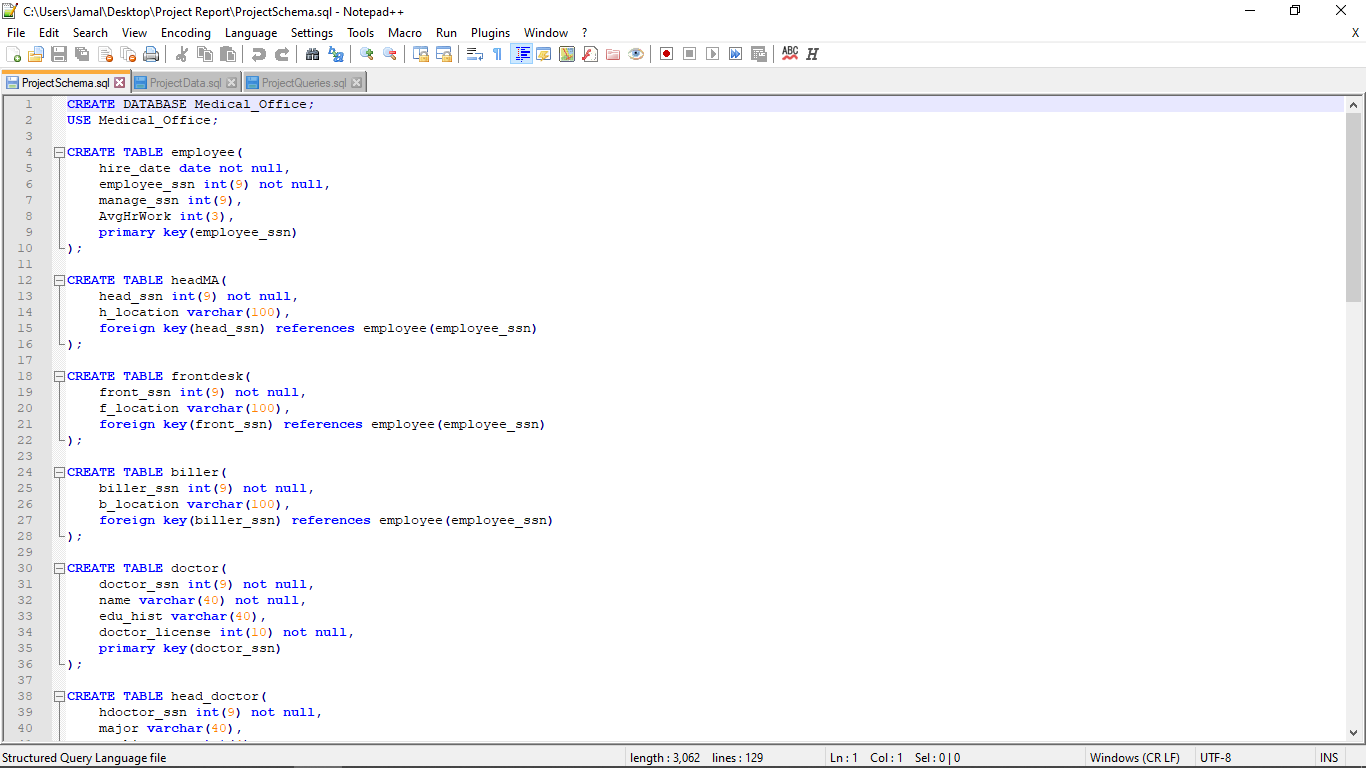
**Data Dictionary:**

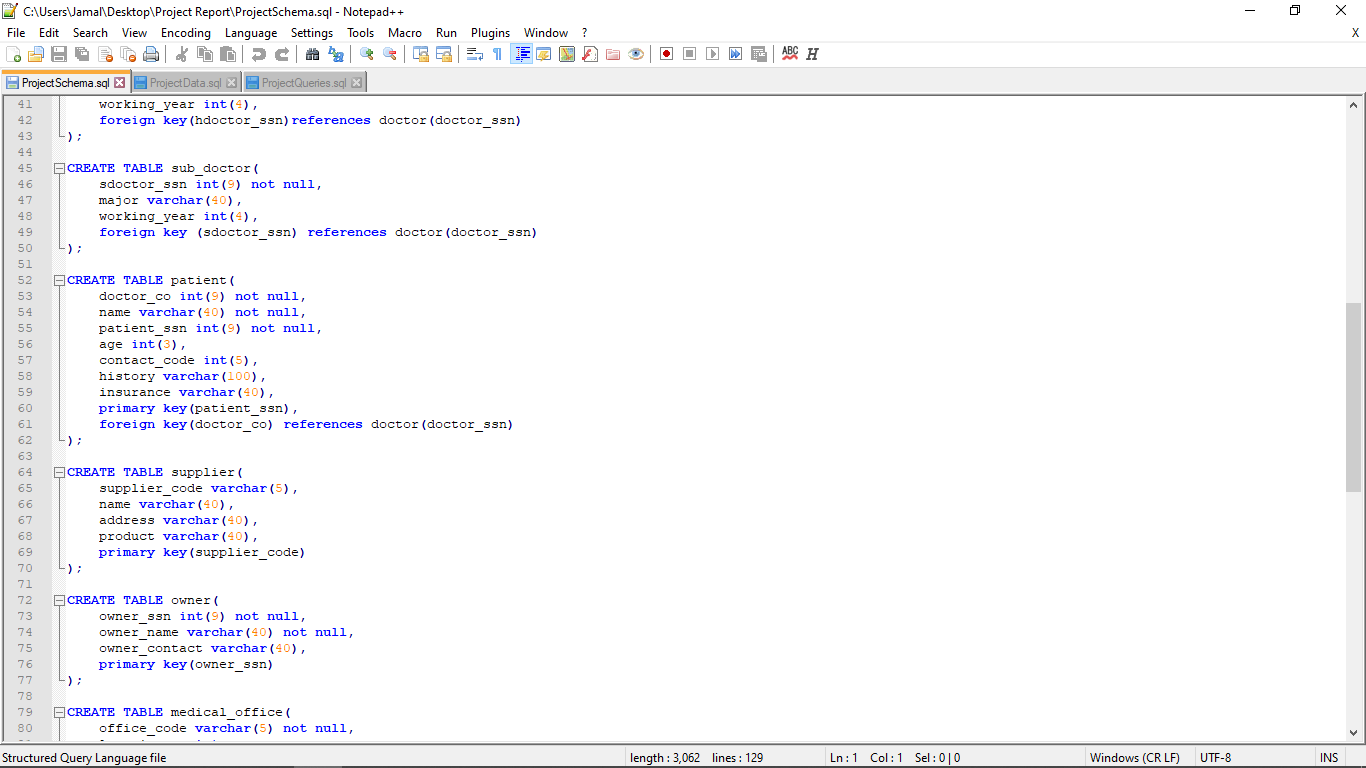


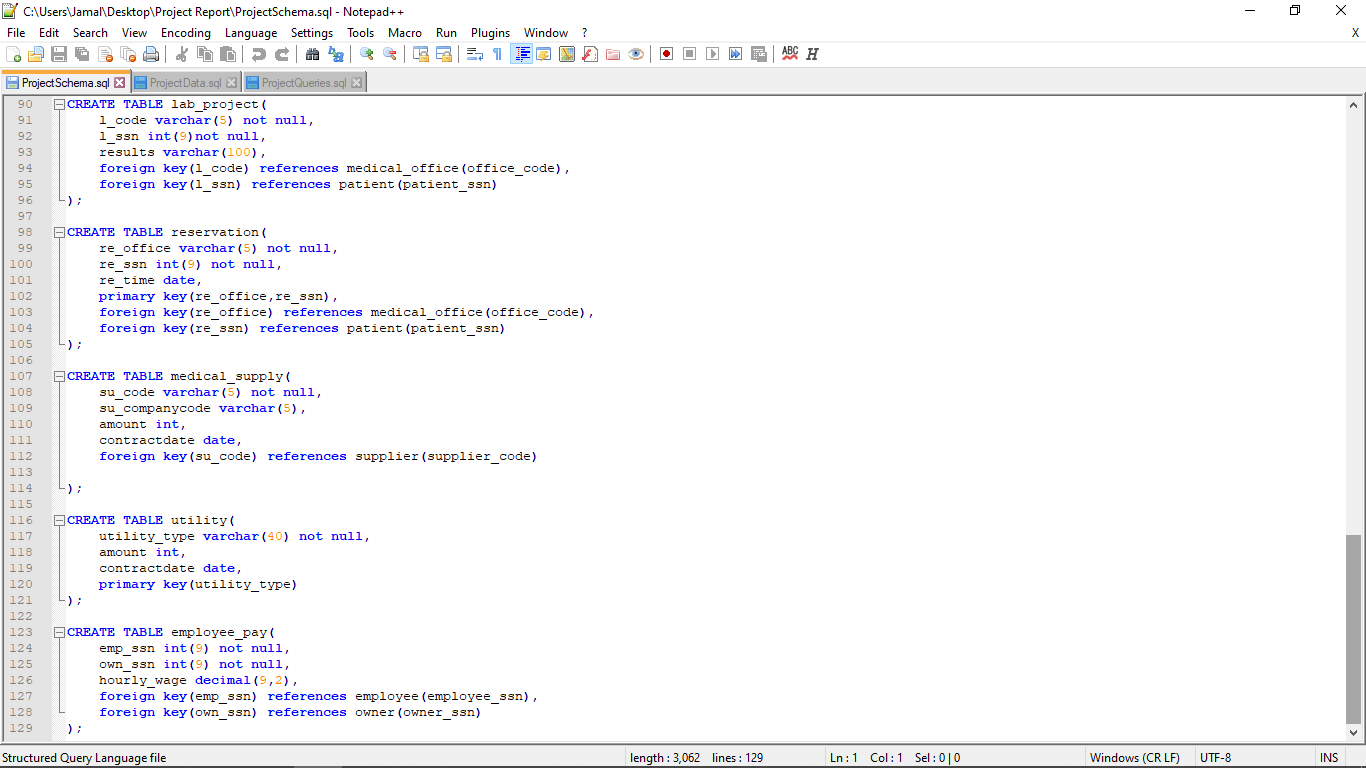


**Implementation:**

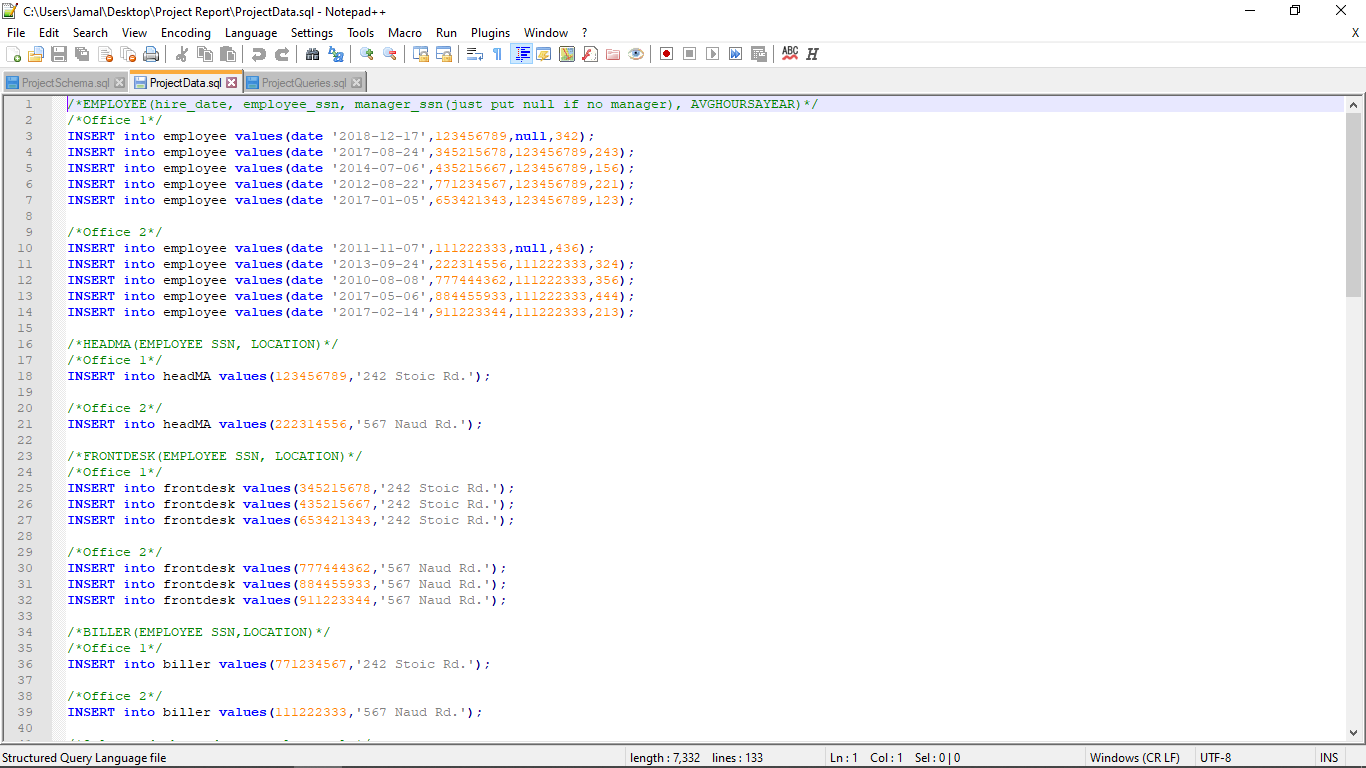
The first three pictures contains the project schema:

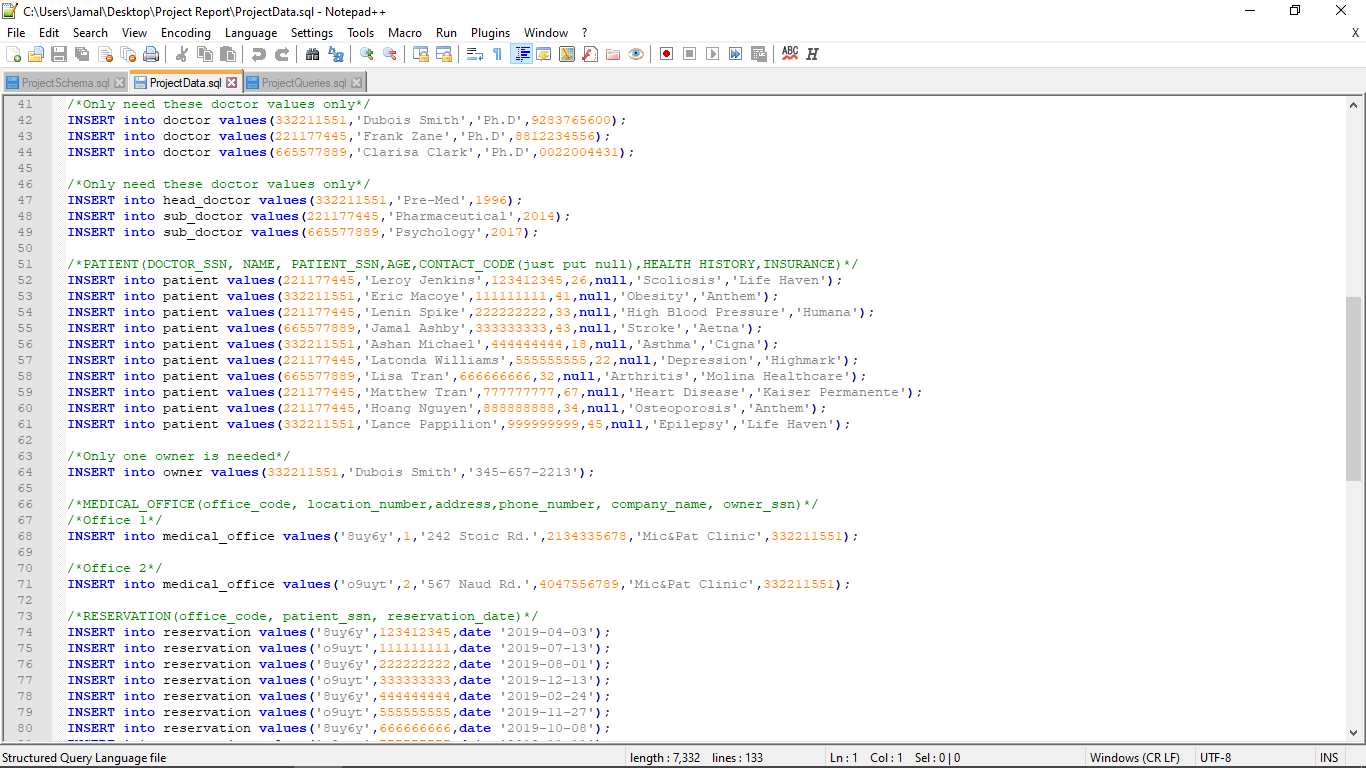


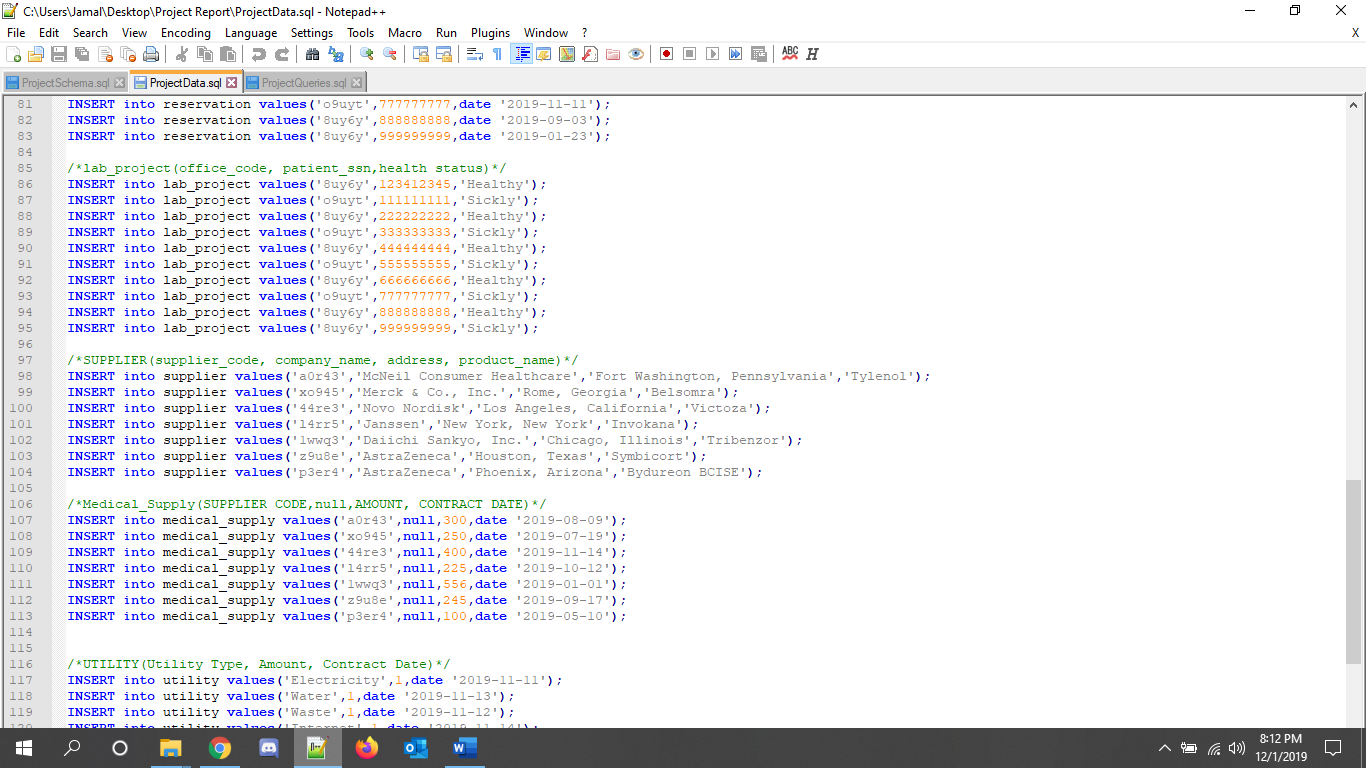


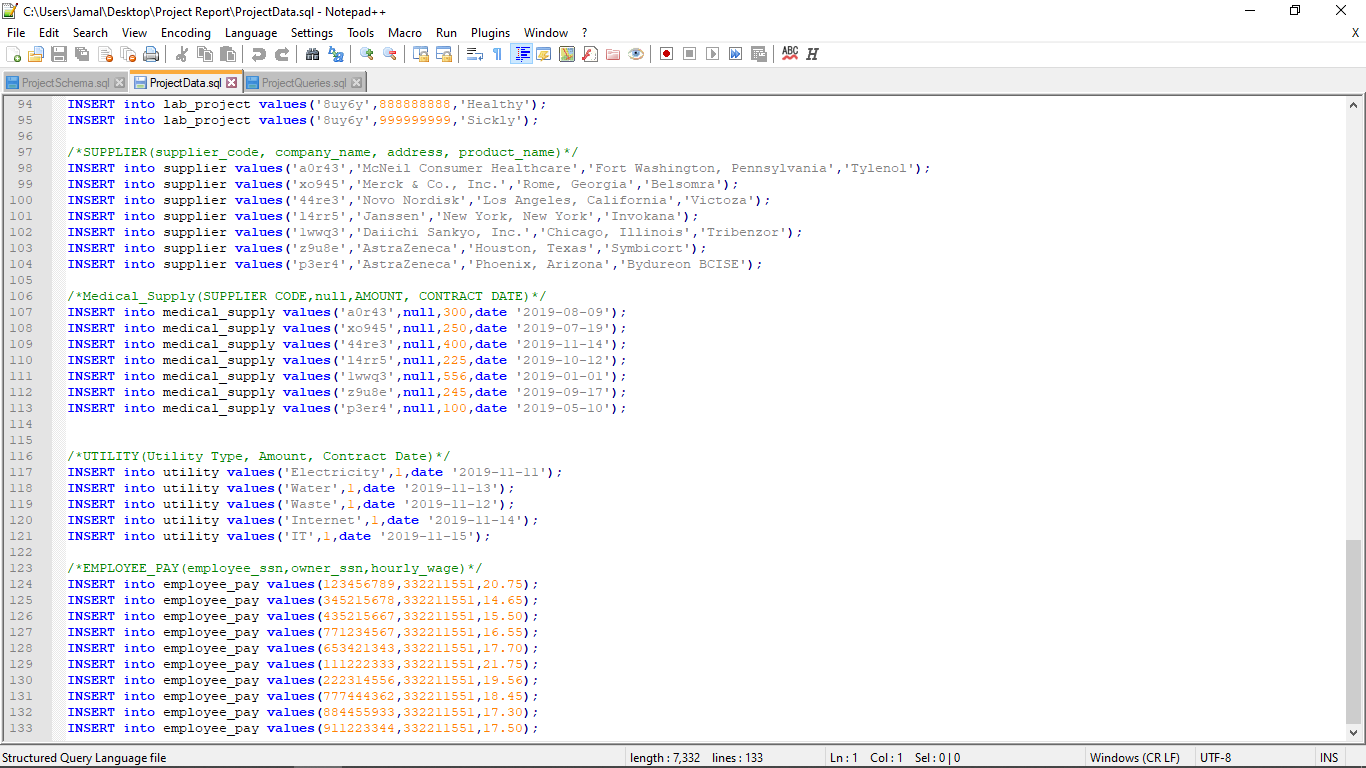


The next four pictures contain the insertion of the data:

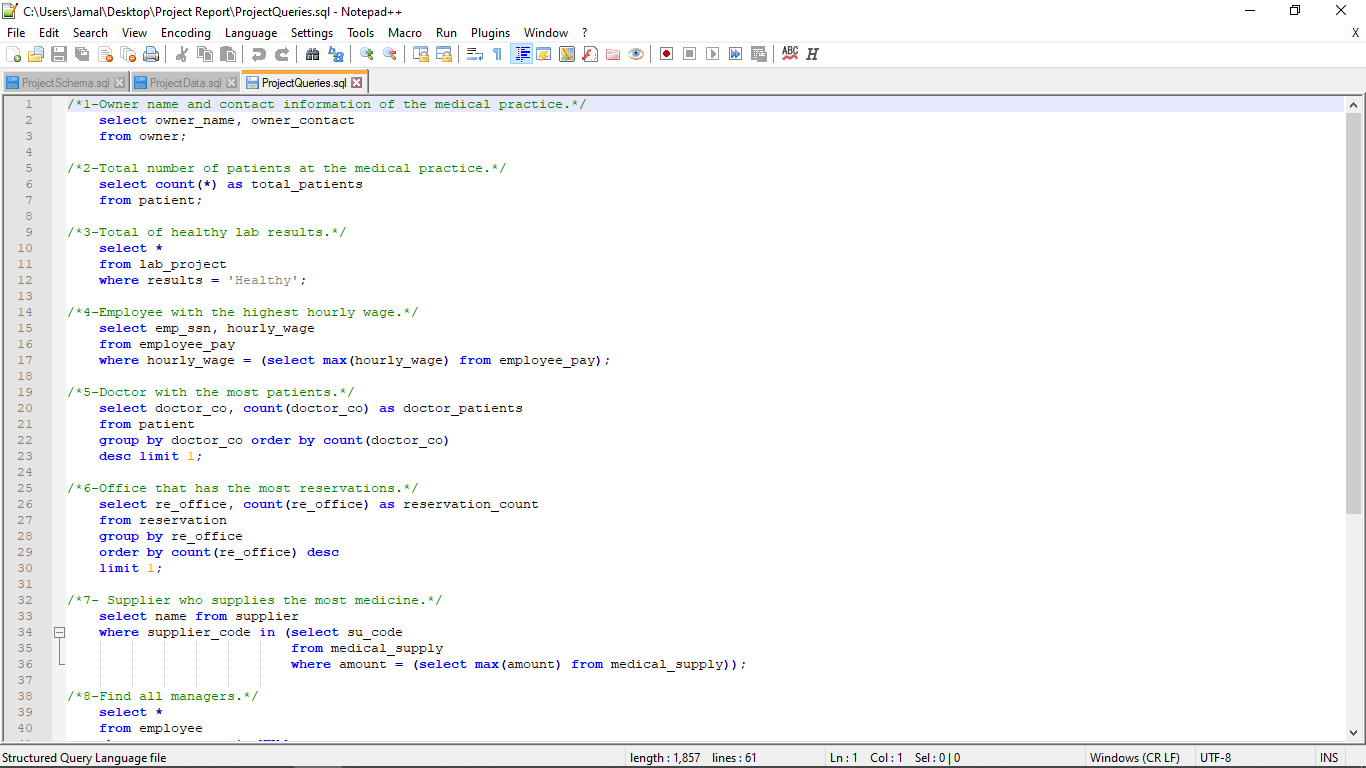


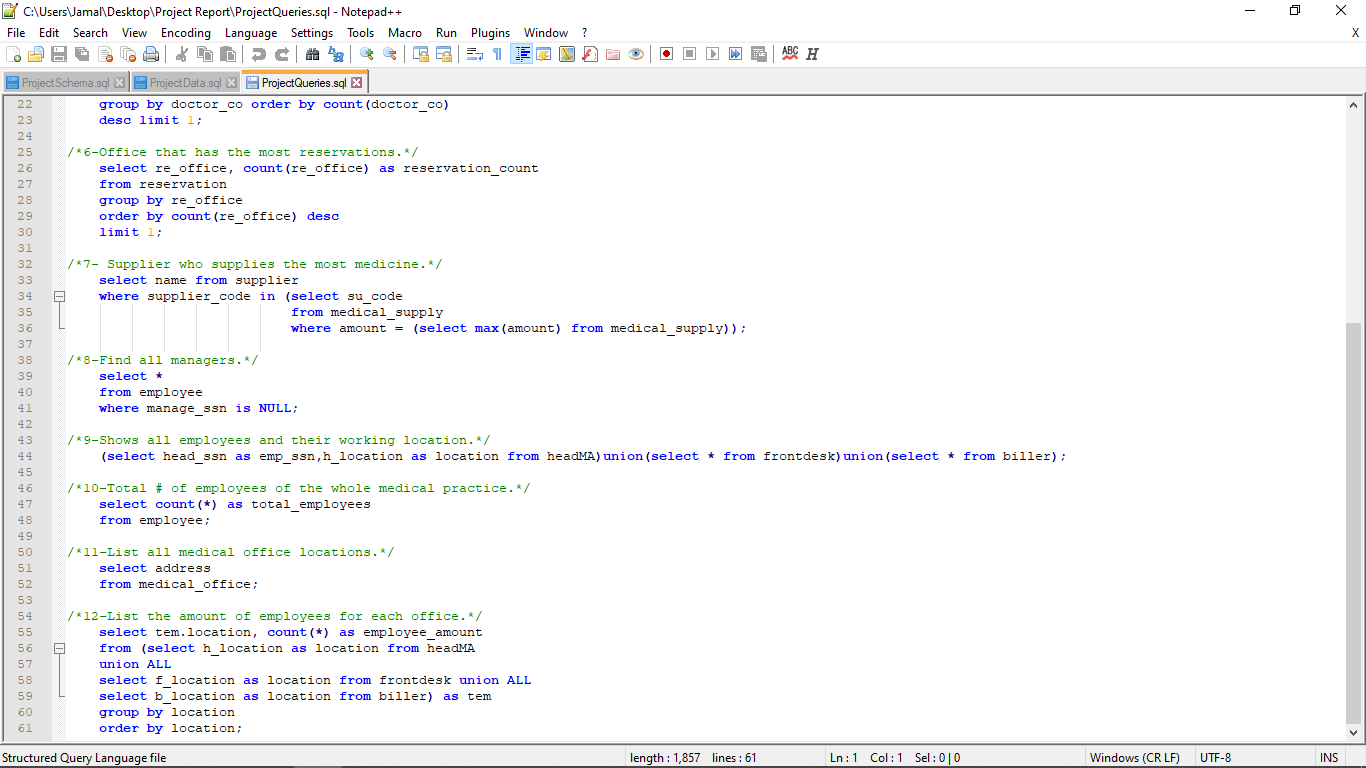






The last two pictures contains my queries:





**Teamwork:**

When it comes to teamwork, all group members were helpful, and supportive when it comes to designing the database. Jamal Ashby’s task was to work on the introduction, report, and create a rough draft of the EER Model. Abrham Dejene’s task was to create the final version of the EER Model. Minho Song’s task was to do the mapping part of the database. During time in class we got together to talk about how to split up these tasks evenly, so everyone gets a chance to do something without feeling left out. We also exchanged information and stay in contact out of class when it comes to the project. Minho also took care of the Data Dictionary, and Abrham finished the projects Normalization. Jamal’s task was to do the implementation and finish the final report.

|  |  |
| --- | --- |
| **Member** | **Task** |
| Ashby, Jamal | Introduction, Report, Implementation |
| Dejene, Abrham | EER Model, Normalization |
| Song, Minho | Mapping, Data Dictionary |

**Summary:**

In conclusion, this project was great in helping us learn how to design a database that is ready to be used. Besides the SQL coding parts, we can take the time to appreciate how much work is put into designing a database and the different techniques that can be used to finalize your specific database. When it came to our functional dependencies, we found that we were able to get our queries to look up everything that was needed. The only true problem we ran into was during the implementation process, where there was a foreign key error and it wouldn’t create a table. The problem was quickly fixed, and we were able to get everything working properly. Other than that, it was a pretty informative project and we can see ourselves using this knowledge in the future when going into the workforce or for future classes such as Web Design or Software Engineering.