"Helping to light your world!"

James A Burks

University of Arkansas at Little Rock

Project Title: Young Electric Database

23 September 2020

Table of Contents

Introduction

For the past several years our company has been maintaining customer records utilizing the traditional paper and file cabinet. This has caused the company to misplace, lose and forget to add services rendered. The Young Electric Database (YED) project has been created to address and correct these issues and prevent further loss associated with the current record keeping practices. The project will integrate a database solution with the customers current website in order to establish a reliable database infrastructure.

System Analysis

- Young Electric has one branch located in Cabot, AR. The data held on each branch is the branch address made up of street, city, state, and zip, and the telephone number.

- Young Electric has a set allocated staff. The data held on a member of staff is his or her name, address, phone, email, dependents and salary.

- For each employee we need to track their hours and date for each project. These projects will require some material(equipment). The material should reflect the item, date, and company purchased from along with the contractor cost. Once installed for the customer we need to assign the cost for the customer.

- The data held on a generator is the model, Serial number, Install Date, Service Dates to include the employee performing the work. We should be able to display the generators that are currently in stock.

- Once the project is complete it is time for payment from the customer. We need to be able to track each payment from the customer. The information collected should be customer, form of payment (Credit Card, Card expiration, card code, Check # or cash), the date billed, and date payment received.

Goal

The objective of this project is to create a database to centrally handle the information of all customers, equipment and payment for Young Electric, and to provide access to this information with an easy-to-use web-based interface that can be accessed by any device with basic HTML rendering capabilities.

Requirements

Requirements for the system fall into three categories, those tending towards the usability of the system, those towards the maintenance and alteration of the system, and those towards the security of the system. For the first requirement, accessibility will be addressed by making the system accessible from the web via a standard web browser. The system will also be designed so that the users will be able to complete repeatable tasks in a streamlined manner in order to cut down on wasted time, and in a concise way to switch between tasks. To address the maintenance of the database, a modular design will be used. This should allow for bugs to easily be found and additional features can easily be added to the system. For the security part of the database, users will be required to run sessions over Hypertext Transfer Protocol over Secure Socket Layer (https) in addition to logon to the database with a username and password.

Summary Milestone Schedule – List key project milestones relative to project start.	
Project Milestone	**Target Date (mm/dd/yyyy)**
• Project Start	09/30/2020
• Complete Solution Design	12/07/2020
• Complete Solution Simulation	03/01/2021
• Complete Solution Simulation and Testing	04/01/2021
• Deploy Solution	05/01/2021
• Project Complete	05/15/2021

System Design

In order to achieve success on the YE project, the following objectives must be met within the designated time and budget allocations:

- Create an ER diagram within the next 30 days
- Create the tables used to store customers information within the next 60 days
- Develop web pages using HTML, JavaScript and CSS within the next 90 days
- Achieve a simulated solution which allows testing within the next 120 days
- Implement the solution across the organization within the next 180 days

Requirements

This project must meet the following list of requirements in order to achieve success.

- The design and functionality must meet the specific requirements of the project sponsor
- The solution must be tested prior to deployment
- Solution must be implemented without disruption to operations

Additional requirements may be added as necessary, with project sponsor approval, as the project moves forward.

Constraints

The following constraints pertain to the ISA project:

- Web application must be searchable and interactive
- The web application must display the customers information in a logical order
- Project mentors are available at any time to provide expert advice and resources for this project
- The project sponsor is wanting a fully functional front-end application

Assumptions

The following are a list of assumptions. Upon agreement and signature of this document, all parties acknowledge that these assumptions are true and correct:

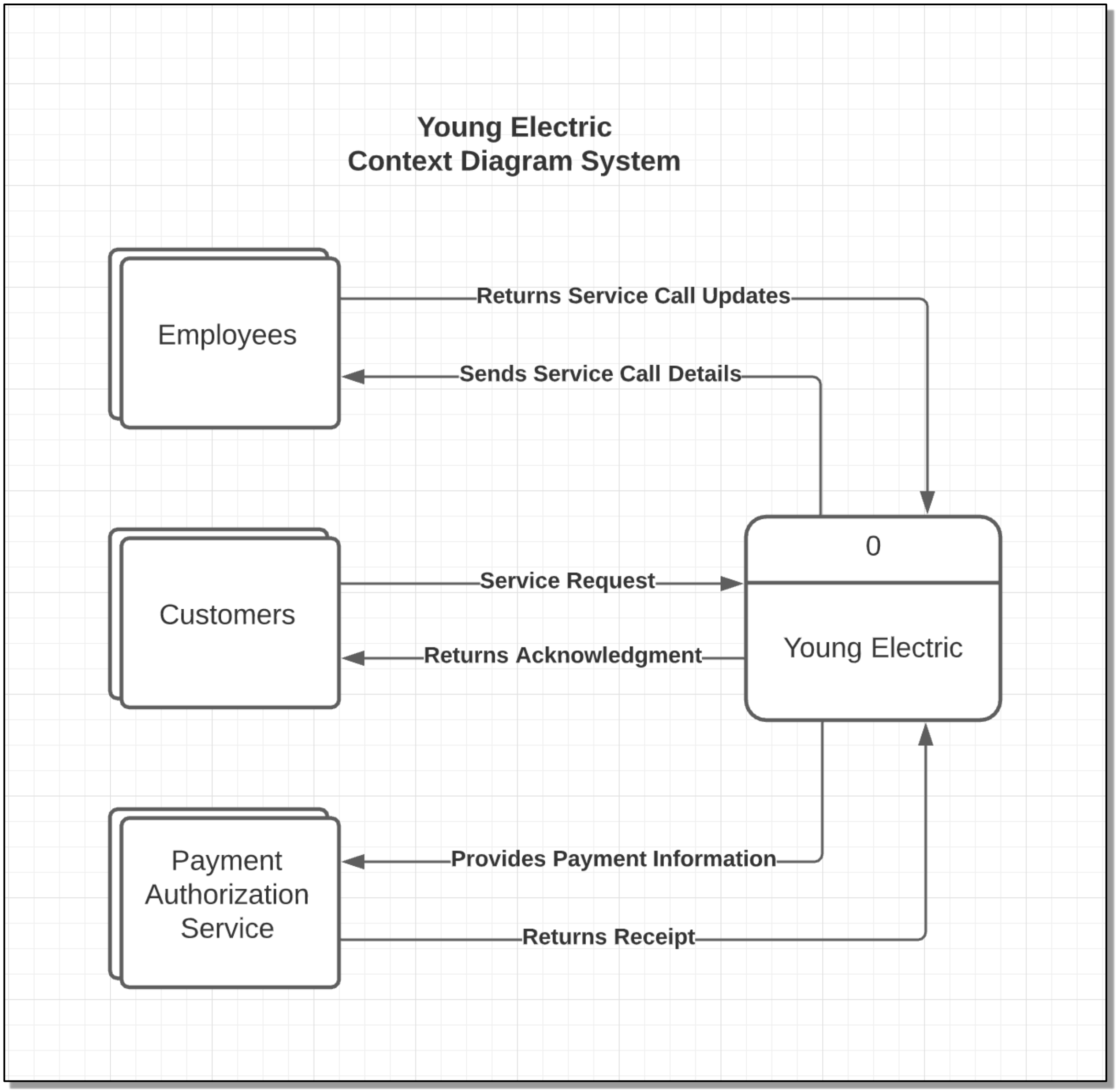
- This project has the full support of the project sponsor
- The purpose of this project will be communicated throughout the company
- The project sponsor will provide additional resources if necessary

Service Request Use Case

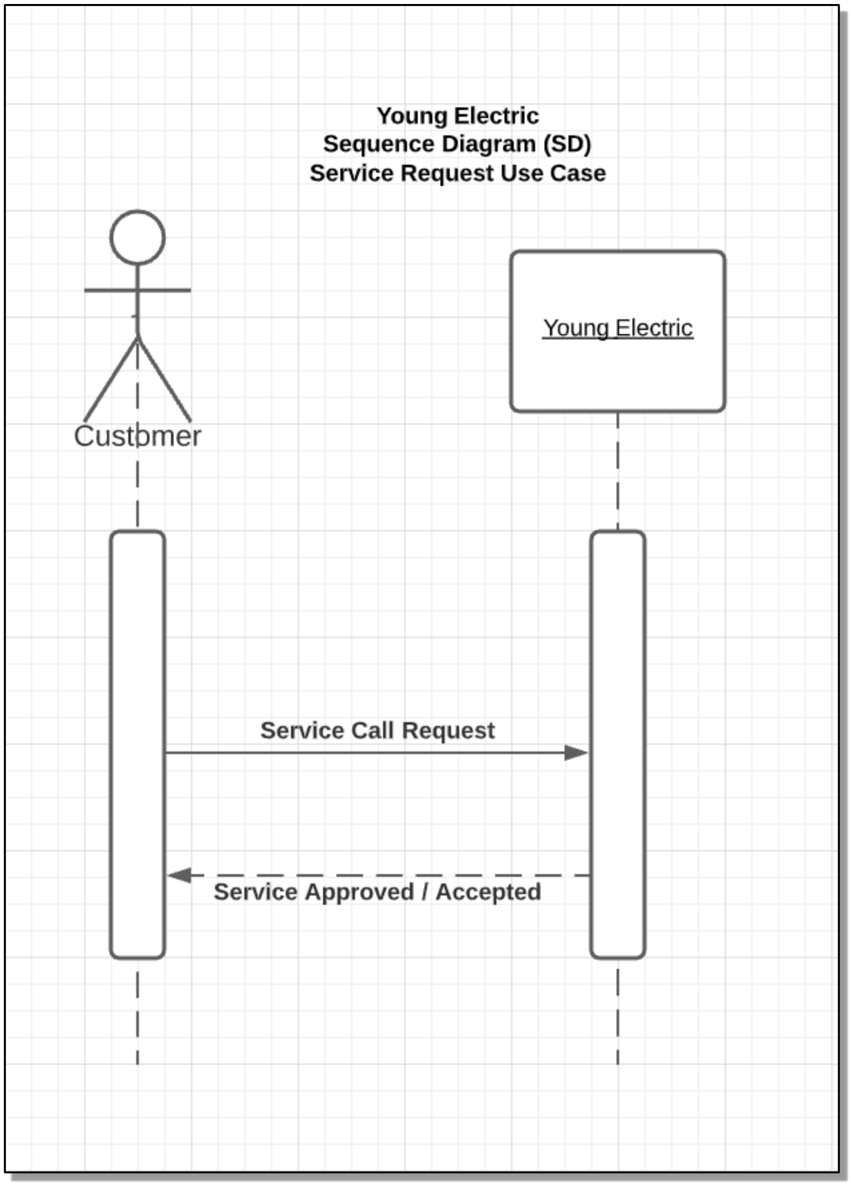
Young Electric Service Request Use Case

Name:	Service Request
Actor:	Customer / YE Employee / Young Electric
Description:	Describes the process used to request electrical service
Successful Completion:	1. Customer requests service with Young Electric (YE) 2. Young Electric creates Service Ticket 3. Young Electric coordinates availability 4. Customer confirms Service dates 5. YE Employee provides service to customer 6. YE generates customer invoice 7. Customer pays invoice
Alternative:	1. Customer requests service with Young Electric (YE) 2. Young Electric unavailable 3. Young Electric provides other Electricians Info 4. Student searches for other Electricians
Precondition:	Customer requests electrical service
Postcondition:	Customer received service and everything works

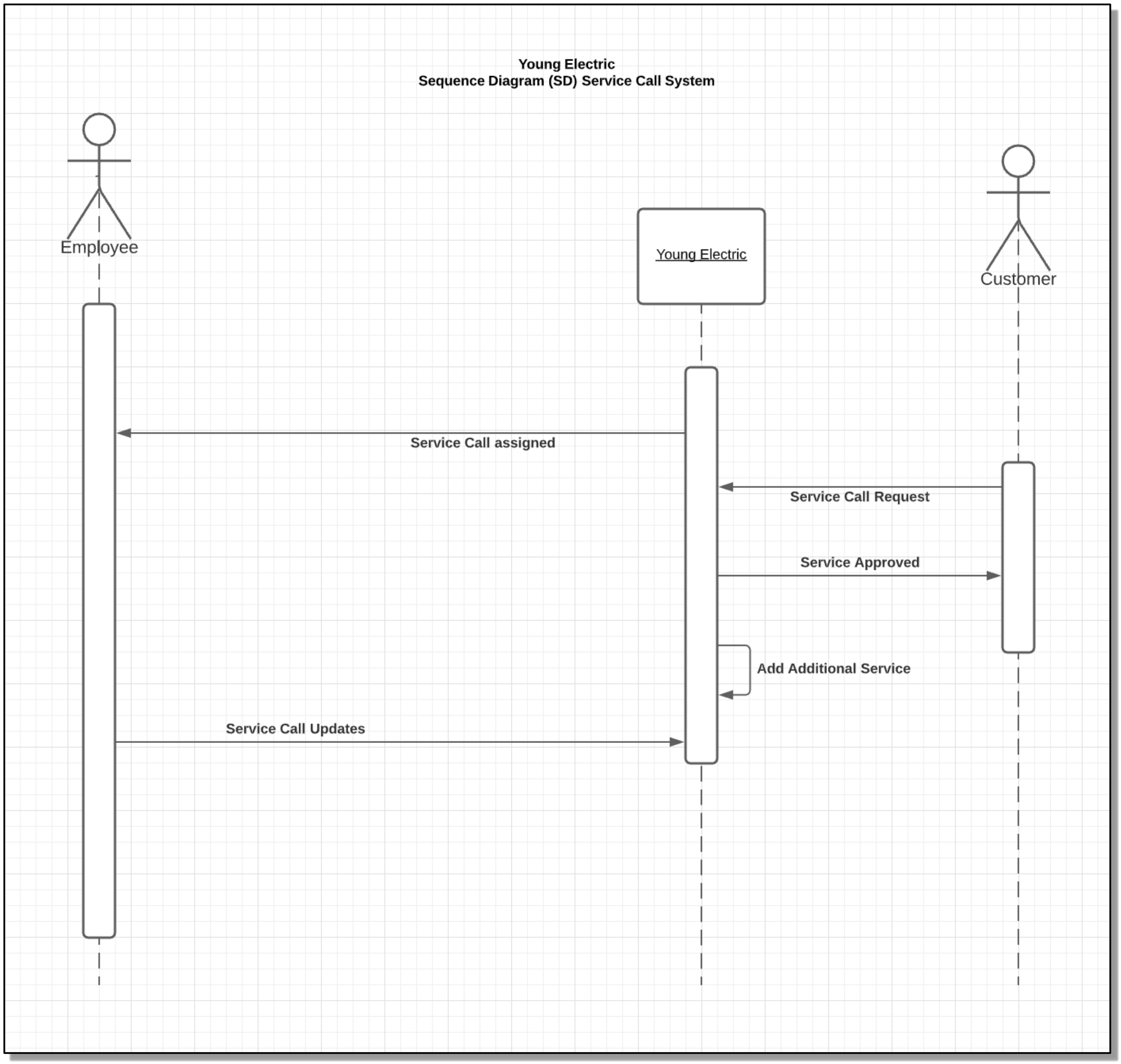
Context Diagram System



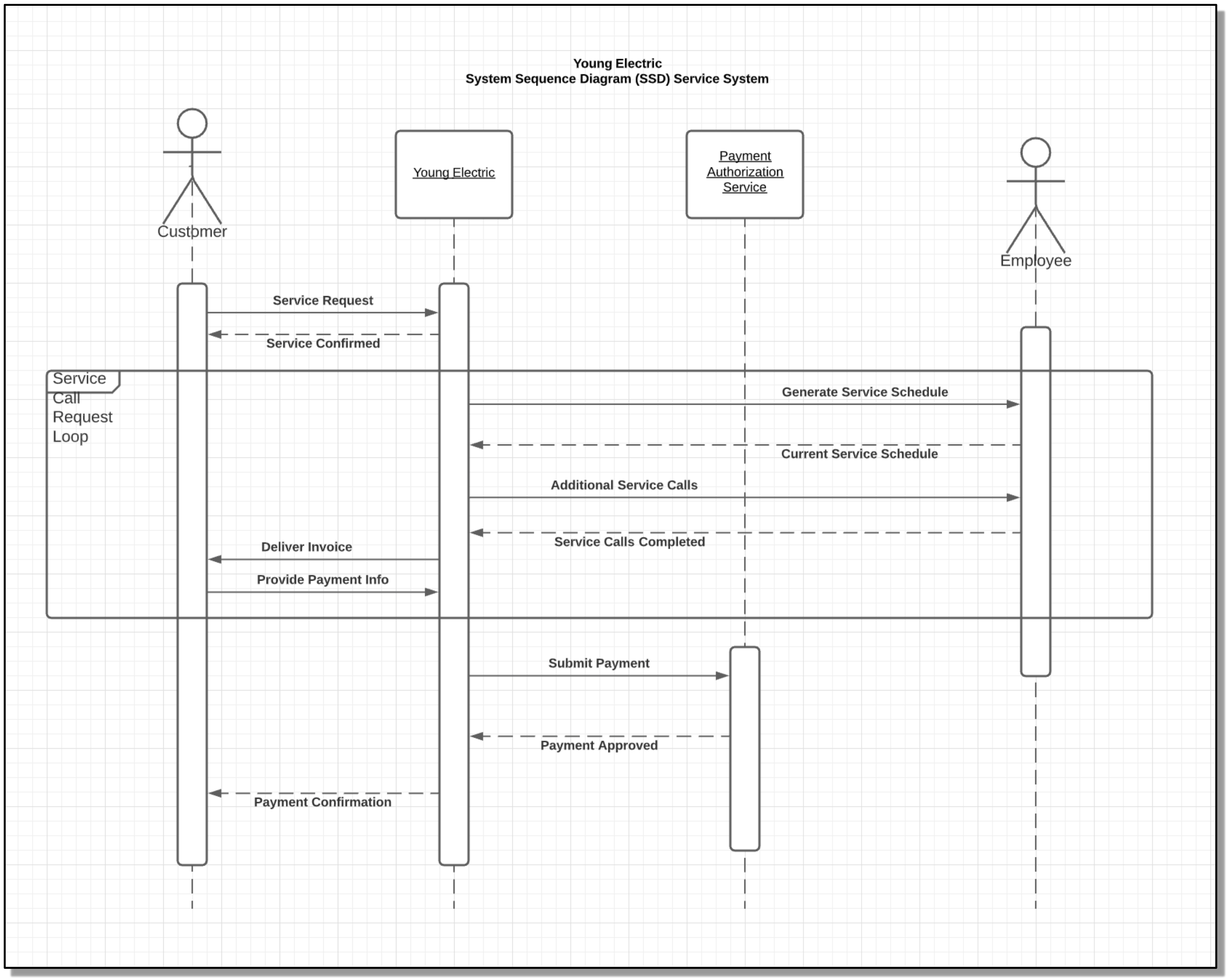
Service Request Use Case Diagram



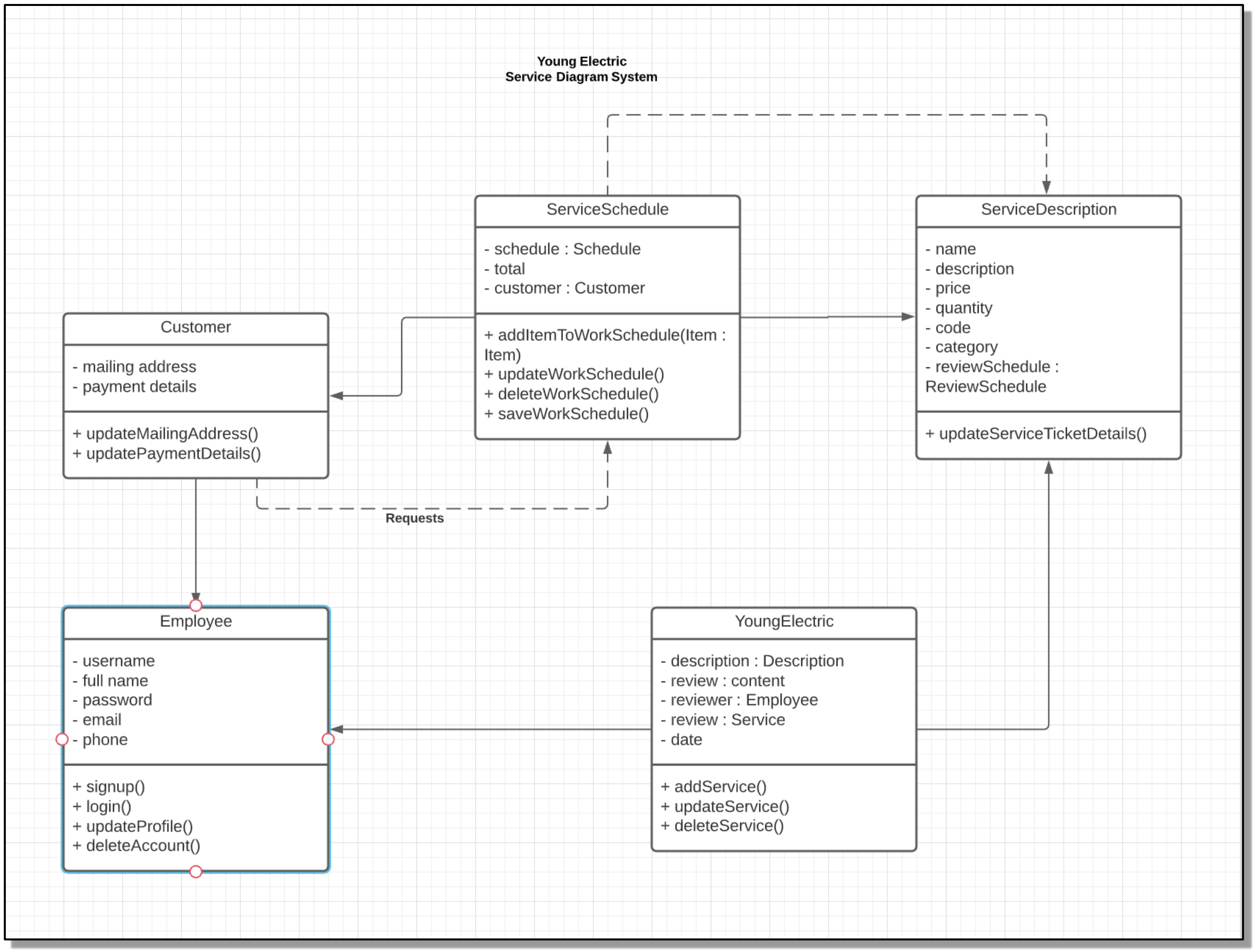
Sequence Diagram (SD)



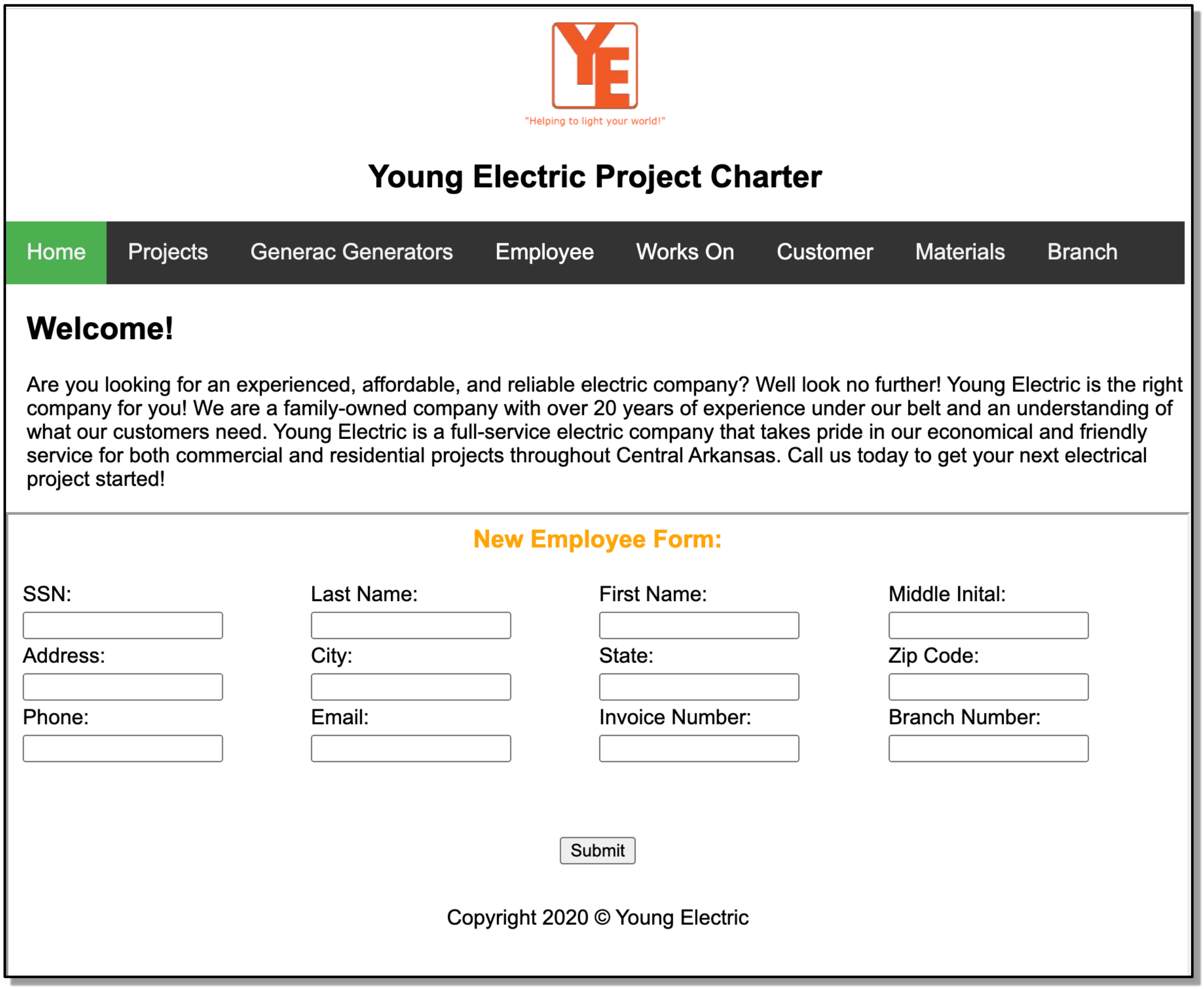
System Sequence Diagram (SSD)



Service Diagram System



Website Design



Young Electric Project Charter

Welcome!

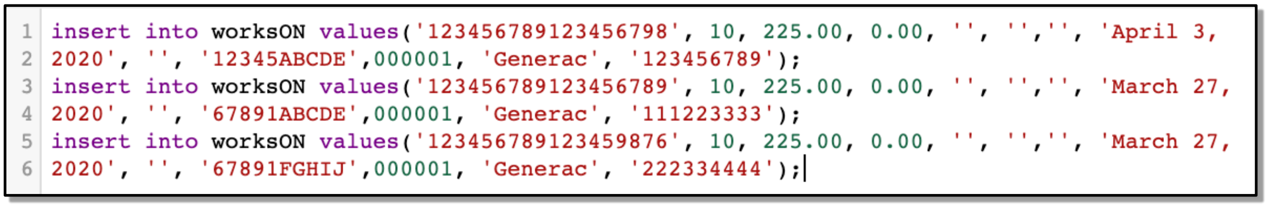
Are you looking for an experienced, affordable, and reliable electric company? Well look no further! Young Electric is the right company for you! We are a family-owned company with over 20 years of experience under our belt and an understanding of what our customers need. Young Electric is a full-service electric company that takes pride in our economical and friendly service for both commercial and residential projects throughout Central Arkansas. Call us today to get your next electrical project started!

New Employee Form:

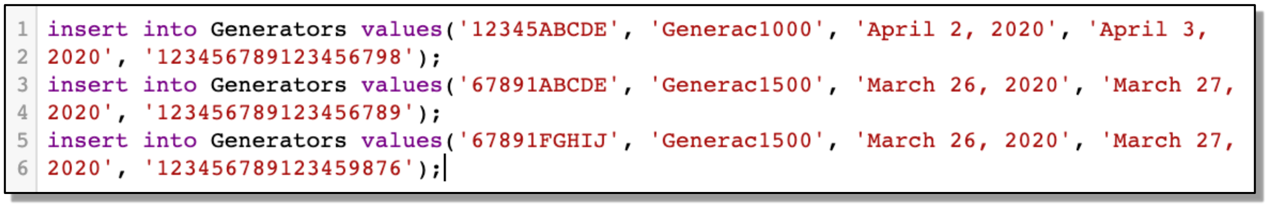
SSN:	Last Name:	First Name:	Middle Inital:
Address:	City:	State:	Zip Code:
Phone:	Email:	Invoice Number:	Branch Number:

Copyright 2020 © Young Electric

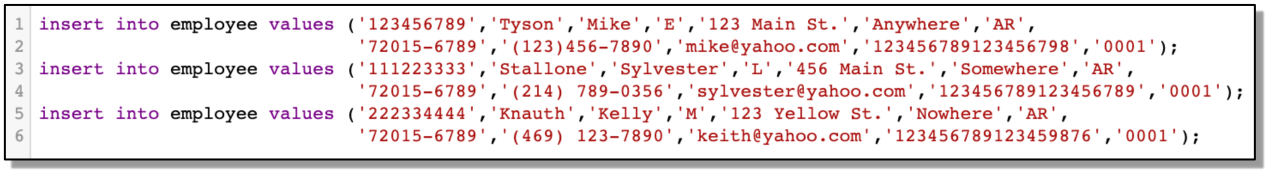
Table Input Format



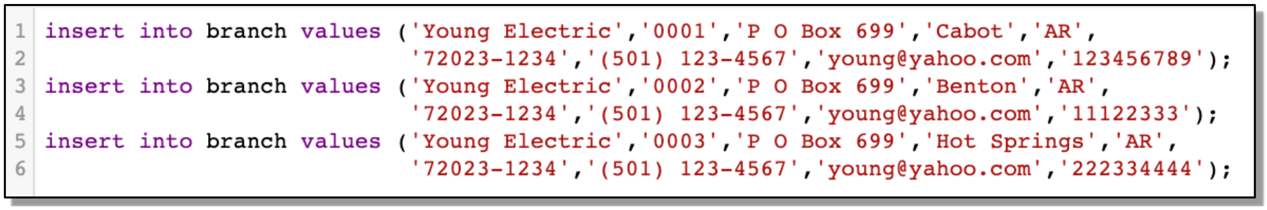
```
insert into worksON values('123456789123456798', 10, 225.00, 0.00, '', '','', 'April 3,
2020', '', '12345ABCDE',000001, 'Generac', '123456789');
insert into worksON values('123456789123456789', 10, 225.00, 0.00, '', '','', 'March 27,
2020', '', '67891ABCDE',000001, 'Generac', '111223333');
insert into worksON values('123456789123459876', 10, 225.00, 0.00, '', '','', 'March 27,
2020', '', '67891FGHIJ',000001, 'Generac', '222334444');
```



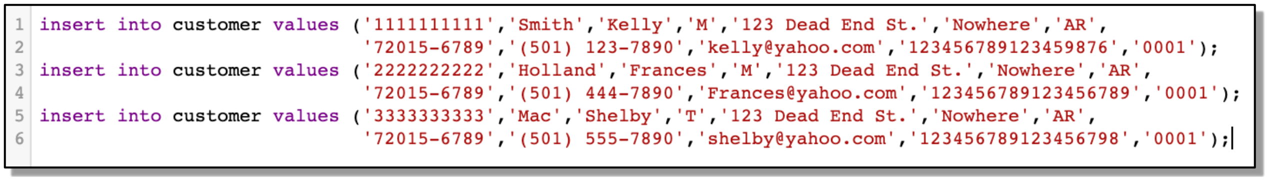
```
insert into Generators values('12345ABCDE', 'Generac1000', 'April 2, 2020', 'April 3,
2020', '123456789123456798');
insert into Generators values('67891ABCDE', 'Generac1500', 'March 26, 2020', 'March 27,
2020', '123456789123456789');
insert into Generators values('67891FGHIJ', 'Generac1500', 'March 26, 2020', 'March 27,
2020', '123456789123459876');
```



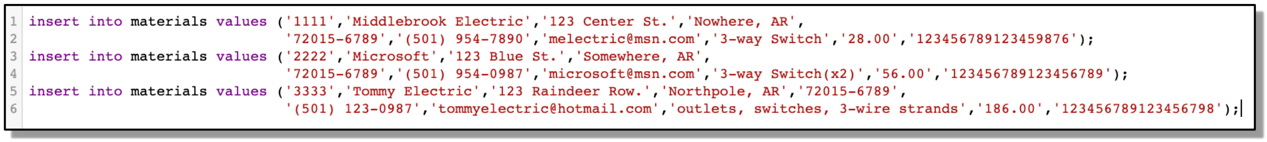
```
insert into employee values ('123456789','Tyson','Mike','E','123 Main St.','Anywhere','AR',
                             '72015-6789','(123)456-7890','mike@yahoo.com','1234567891234567 98','0001');
insert into employee values ('111223333','Stallone','Sylvester','L','456 Main St.','Somewhere','AR',
                             '72015-6789','(214) 789-0356','sylvester@yahoo.com','1234567891234567 89','0001');
insert into employee values ('222334444','Knauth','Kelly','M','123 Yellow St.','Nowhere','AR',
                             '72015-6789','(469) 123-7890','keith@yahoo.com','1234567891234598 76','0001');
```



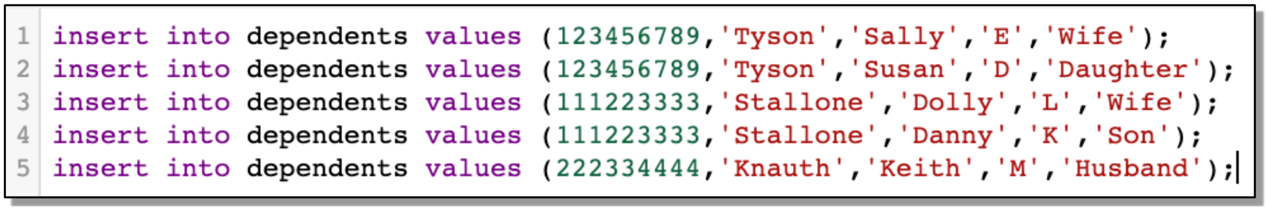
```
insert into branch values ('Young Electric','0001','P O Box 699','Cabot','AR',
                           '72023-1234','(501) 123-4567','young@yahoo.com','123456789');
insert into branch values ('Young Electric','0002','P O Box 699','Benton','AR',
                           '72023-1234','(501) 123-4567','young@yahoo.com','11122333');
insert into branch values ('Young Electric','0003','P O Box 699','Hot Springs','AR',
                           '72023-1234','(501) 123-4567','young@yahoo.com','222334444');
```



```
insert into customer values ('1111111111','Smith','Kelly','M','123 Dead End St.','Nowhere','AR',
                             '72015-6789','(501) 123-7890','kelly@yahoo.com','1234567891234598 76','0001');
insert into customer values ('2222222222','Holland','Frances','M','123 Dead End St.','Nowhere','AR',
                             '72015-6789','(501) 444-7890','Frances@yahoo.com','1234567891234567 89','0001');
insert into customer values ('3333333333','Mac','Shelby','T','123 Dead End St.','Nowhere','AR',
                             '72015-6789','(501) 555-7890','shelby@yahoo.com','1234567891234567 98','0001');
```



```
insert into materials values ('1111','Middlebrook Electric','123 Center St.','Nowhere, AR',
                              '72015-6789','(501) 954-7890','melectric@msn.com','3-way Switch','28.00','1234567891234598 76');
insert into materials values ('2222','Microsoft','123 Blue St.','Somewhere, AR',
                              '72015-6789','(501) 954-0987','microsoft@msn.com','3-way Switch(x2)','56.00','1234567891234567 89');
insert into materials values ('3333','Tommy Electric','123 Raindeer Row.','Northpole, AR','72015-6789',
                              '(501) 123-0987','tommyelectric@hotmail.com','outlets, switches, 3-wire strands','186.00','1234567891234567 98');
```



```
insert into dependents values (123456789,'Tyson','Sally','E','Wife');
insert into dependents values (123456789,'Tyson','Susan','D','Daughter');
insert into dependents values (111223333,'Stallone','Dolly','L','Wife');
insert into dependents values (111223333,'Stallone','Danny','K','Son');
insert into dependents values (222334444,'Knauth','Keith','M','Husband');
```

Database Normalization

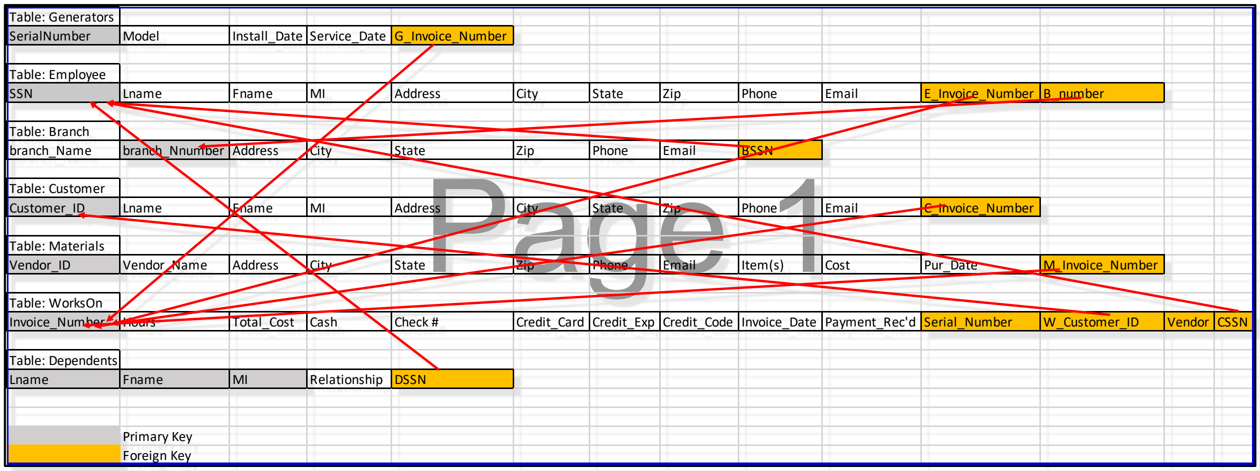
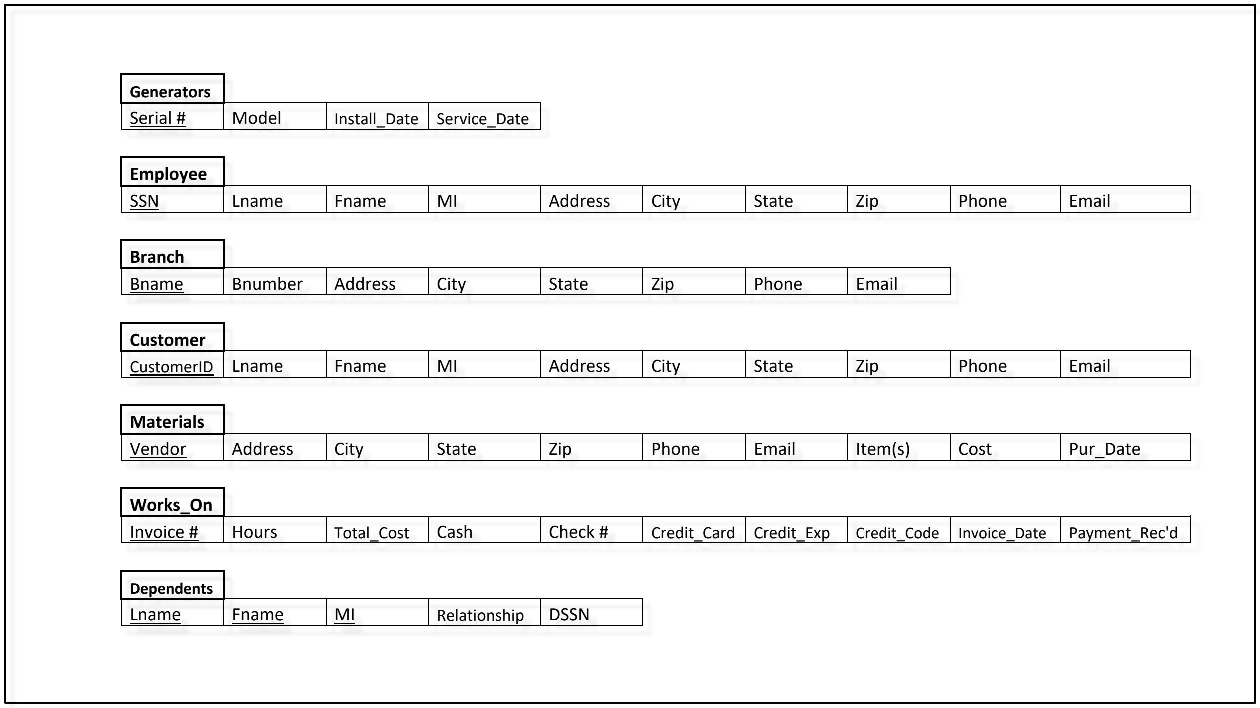


Table: Generators													
SerialNumber	Model	Install_Date	Service_Date	G_Invoice_Number									
Table: Employee													
SSN	Lname	Fname	MI	Address	City	State	Zip	Phone	Email	E_Invoice_Number	B_number		
Table: Branch													
branch_Name	branch_Nnumber	Address	City	State	Zip	Phone	Email	BSSN					
Table: Customer													
Customer_ID	Lname	Fname	MI	Address	City	State	Zip	Phone	Email	C_Invoice_Number			
Table: Materials													
Vendor_ID	Vendor_Name	Address	City	State	Zip	Phone	Email	Item(s)	Cost	Pur_Date	M_Invoice_Number		
Table: WorksOn													
Invoice_Number	Hours	Total_Cost	Cash	Check #	Credit_Card	Credit_Exp	Credit_Code	Invoice_Date	Payment_Rec'd	Serial_Number	W_Customer_ID	Vendor	CSSN
Table: Dependents													
Lname	Fname	MI	Relationship	DSSN									
▨	Primary Key												
🟧	Foreign Key												

System Construction

Tables



Generators			
Serial #	Model	Install_Date	Service_Date

Employee									
SSN	Lname	Fname	MI	Address	City	State	Zip	Phone	Email

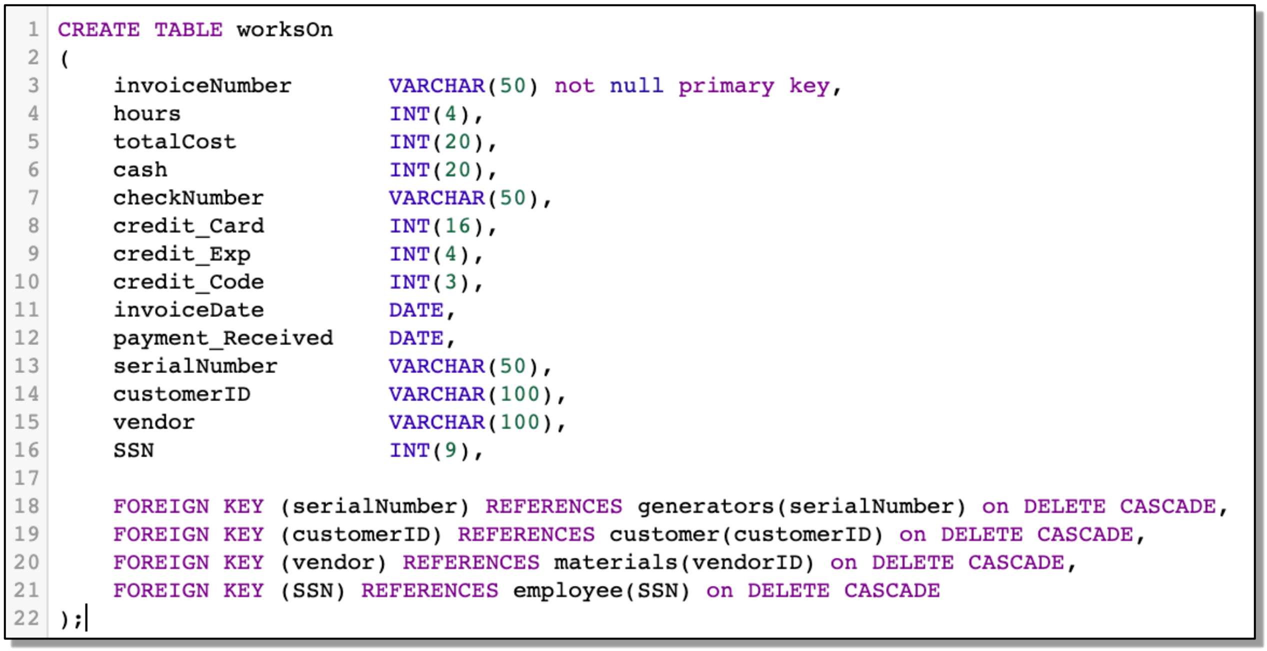
Branch							
Bname	Bnumber	Address	City	State	Zip	Phone	Email

Customer									
CustomerID	Lname	Fname	MI	Address	City	State	Zip	Phone	Email

Materials									
Vendor	Address	City	State	Zip	Phone	Email	Item(s)	Cost	Pur_Date

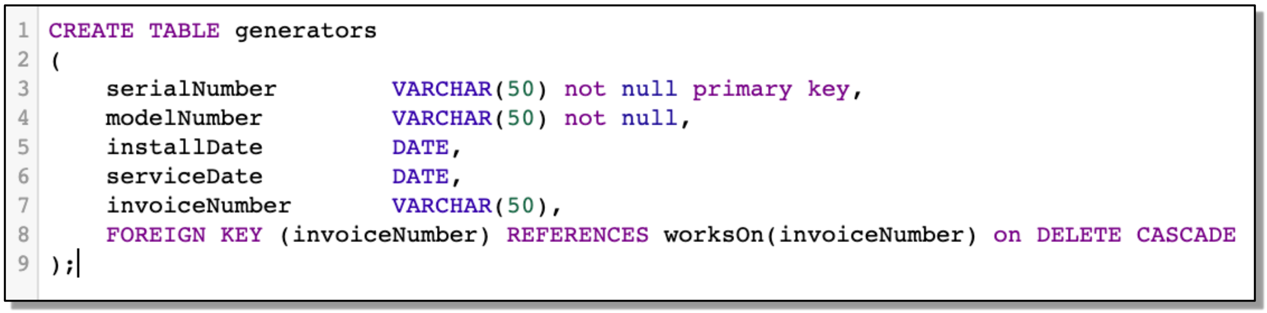
Works_On									
Invoice #	Hours	Total_Cost	Cash	Check #	Credit_Card	Credit_Exp	Credit_Code	Invoice_Date	Payment_Rec'd

Dependents				
Lname	Fname	MI	Relationship	DSSN

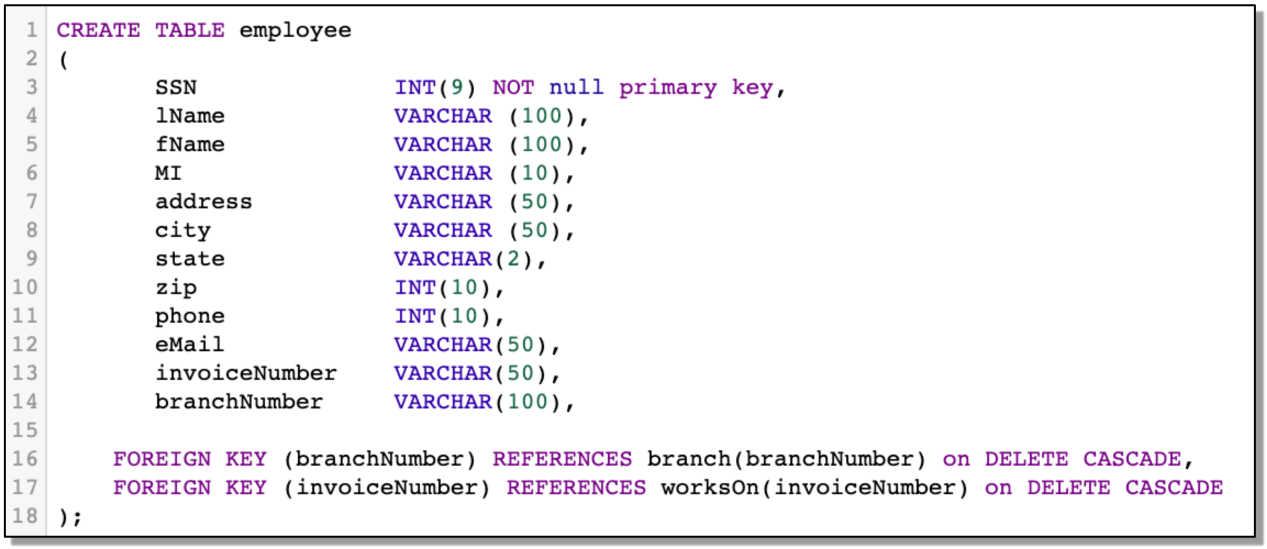


```
CREATE TABLE worksOn
(
    invoiceNumber       VARCHAR(50) not null primary key,
    hours               INT(4),
    totalCost           INT(20),
    cash                INT(20),
    checkNumber         VARCHAR(50),
    credit_Card         INT(16),
    credit_Exp          INT(4),
    credit_Code         INT(3),
    invoiceDate         DATE,
    payment_Received    DATE,
    serialNumber        VARCHAR(50),
    customerID          VARCHAR(100),
    vendor              VARCHAR(100),
    SSN                 INT(9),

    FOREIGN KEY (serialNumber) REFERENCES generators(serialNumber) on DELETE CASCADE,
    FOREIGN KEY (customerID) REFERENCES customer(customerID) on DELETE CASCADE,
    FOREIGN KEY (vendor) REFERENCES materials(vendorID) on DELETE CASCADE,
    FOREIGN KEY (SSN) REFERENCES employee(SSN) on DELETE CASCADE
);
```

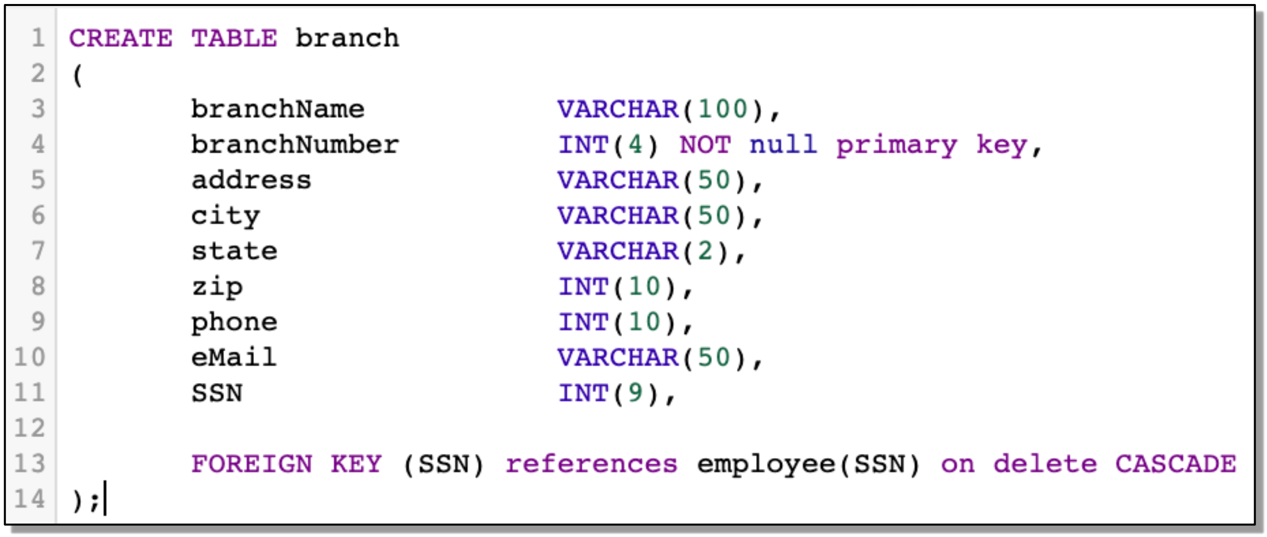


```
CREATE TABLE generators
(
    serialNumber        VARCHAR(50) not null primary key,
    modelNumber         VARCHAR(50) not null,
    installDate        DATE,
    serviceDate        DATE,
    invoiceNumber       VARCHAR(50),
    FOREIGN KEY (invoiceNumber) REFERENCES worksOn(invoiceNumber) on DELETE CASCADE
);
```



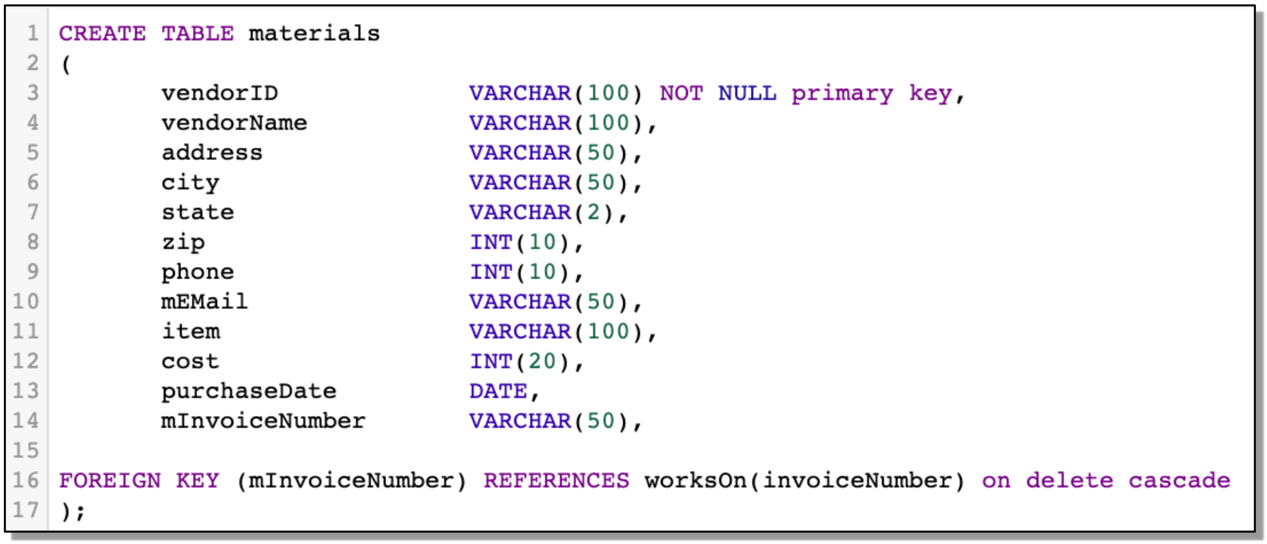
```
CREATE TABLE employee
(
       SSN              INT(9) NOT null primary key,
       lName            VARCHAR (100),
       fName            VARCHAR (100),
       MI               VARCHAR (10),
       address          VARCHAR (50),
       city             VARCHAR (50),
       state            VARCHAR(2),
       zip              INT(10),
       phone            INT(10),
       eMail            VARCHAR(50),
       invoiceNumber    VARCHAR(50),
       branchNumber     VARCHAR(100),

    FOREIGN KEY (branchNumber) REFERENCES branch(branchNumber) on DELETE CASCADE,
    FOREIGN KEY (invoiceNumber) REFERENCES worksOn(invoiceNumber) on DELETE CASCADE
);
```



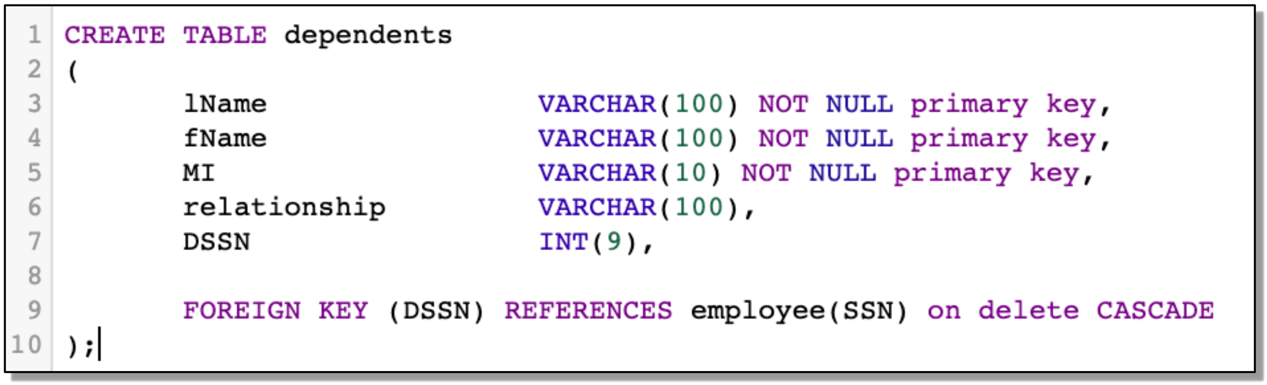
```
CREATE TABLE branch
(
       branchName         VARCHAR(100),
       branchNumber       INT(4) NOT null primary key,
       address            VARCHAR(50),
       city               VARCHAR(50),
       state              VARCHAR(2),
       zip                INT(10),
       phone              INT(10),
       eMail              VARCHAR(50),
       SSN                INT(9),

       FOREIGN KEY (SSN) references employee(SSN) on delete CASCADE
);
```



```sql
CREATE TABLE materials
(
        vendorID                VARCHAR(100) NOT NULL primary key,
        vendorName              VARCHAR(100),
        address                 VARCHAR(50),
        city                    VARCHAR(50),
        state                   VARCHAR(2),
        zip                     INT(10),
        phone                   INT(10),
        mEMail                  VARCHAR(50),
        item                    VARCHAR(100),
        cost                    INT(20),
        purchaseDate            DATE,
        mInvoiceNumber          VARCHAR(50),

FOREIGN KEY (mInvoiceNumber) REFERENCES worksOn(invoiceNumber) on delete cascade
);
```



```sql
CREATE TABLE dependents
(
        lName                   VARCHAR(100) NOT NULL primary key,
        fName                   VARCHAR(100) NOT NULL primary key,
        MI                      VARCHAR(10) NOT NULL primary key,
        relationship            VARCHAR(100),
        DSSN                    INT(9),

        FOREIGN KEY (DSSN) REFERENCES employee(SSN) on delete CASCADE
);
```

System Testing

Success for the YED project will be achieved when a fully tested database solution, and all technical documentation, is fully deployed throughout the company within the time and cost constraints indicated in this charter. Additionally, this measure of success must include a recommendation list for future database configuration considerations as we fully anticipate the necessity of this solution to evolve in order to provide a more robust solution. Success will be determined by the Project Sponsor, Mr. Jack Young, who will also authorize completion of the project.