



"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	3
SYSTEM ANALYSIS.....	3
GOAL	4
REQUIREMENTS	4
SYSTEM DESIGN	5
REQUIREMENTS.....	5
CONSTRAINTS	5
ASSUMPTIONS.....	6
SERVICE REQUEST USE CASE	6
CONTEXT DIAGRAM SYSTEM	7
SERVICE REQUEST USE CASE DIAGRAM.....	8
SEQUENCE DIAGRAM (SD)	9
SYSTEM SEQUENCE DIAGRAM (SSD)	10
SERVICE DIAGRAM SYSTEM	11
SYSTEM CONSTRUCTION.....	12
DATABASE NORMALIZATION.....	12
TABLES	12
WEBSITE DESIGN	17
TABLE INPUT.....	19
SYSTEM TESTING.....	19

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 Branch number, street address, city, state, and zip, POC, phone number and eMail address.
- Young Electric has a set allocated staff. The data held on a member of staff is his or her name, SSN, street address, DOB, 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, SQL database using nuBuilder 4.5 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:

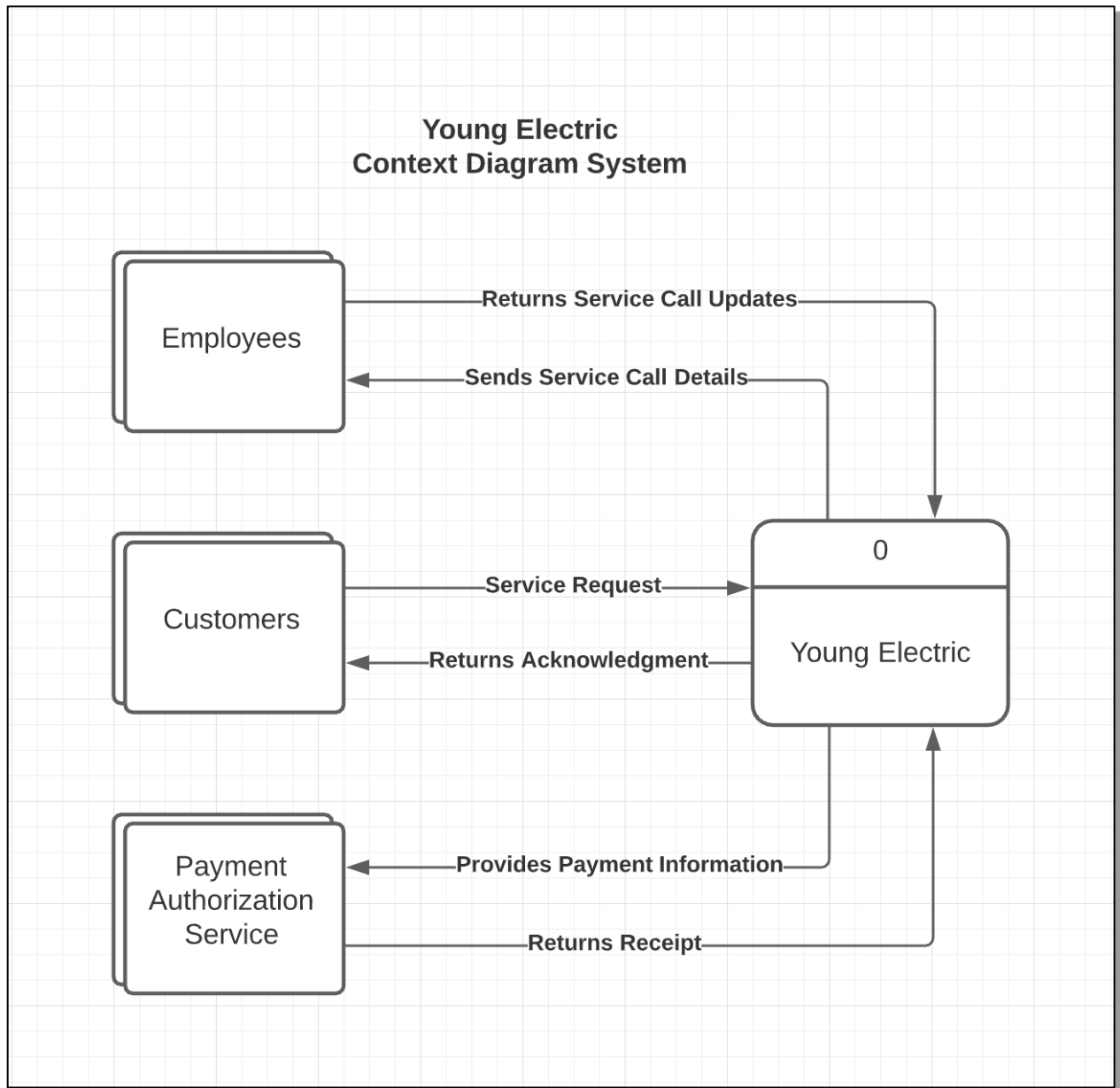
- This project has the full support of the project sponsor
- The purpose of this project will be communicated throughout the company
- The project sponsor will provide additional resources if necessary

Service Request Use Case

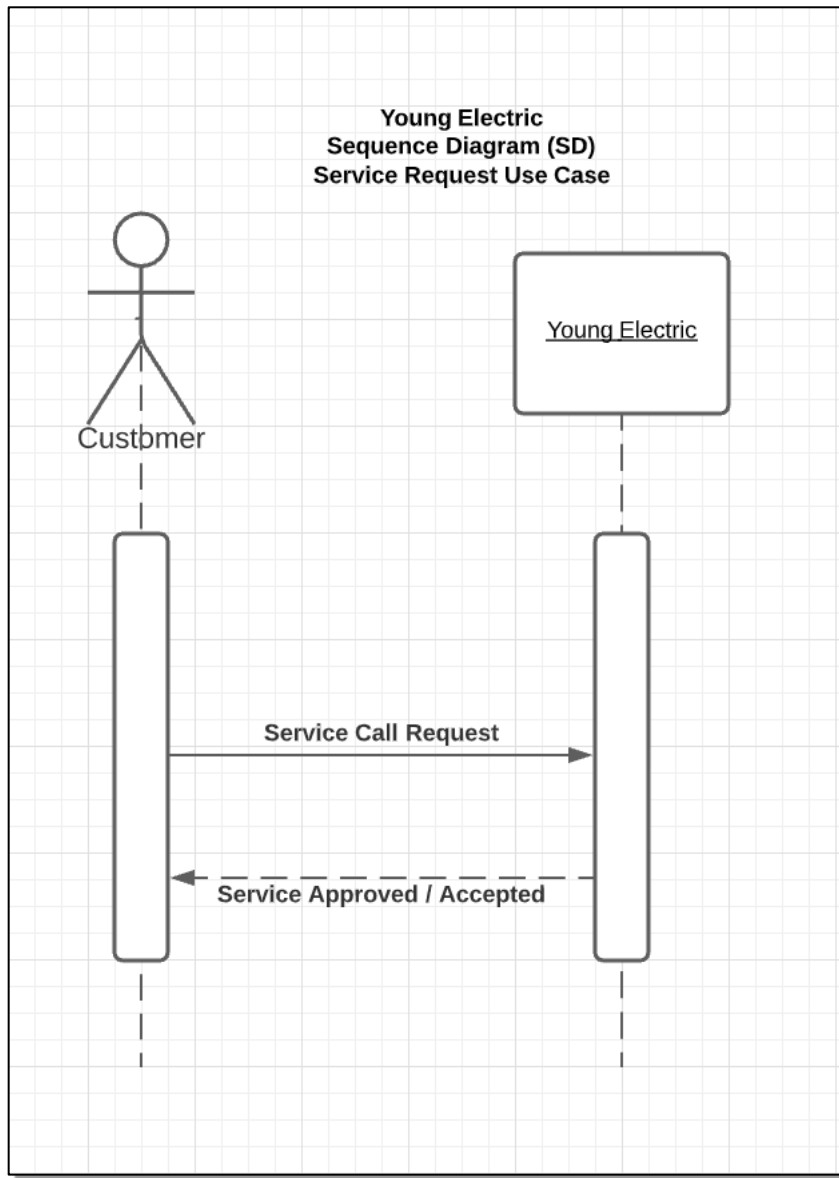
Young Electric Service Request Use Case

Name:	Service Request
Actor:	Customer / YE Employee / Young Electric
Description:	Describes the process used to request electrical service
Successful Completion:	<ol style="list-style-type: none">1. Customer requests service with Young Electric (YE)2. Young Electric creates Service Ticket3. Young Electric coordinates availability4. Customer confirms Service dates5. YE Employee provides service to customer6. YE generates customer invoice7. Customer pays invoice
Alternative:	<ol style="list-style-type: none">1. Customer requests service with Young Electric (YE)2. Young Electric unavailable3. Young Electric provides other Electricians Info4. Customer searches for other Electricians
Precondition:	Customer requests electrical service
Postcondition:	Customer received service and everything works

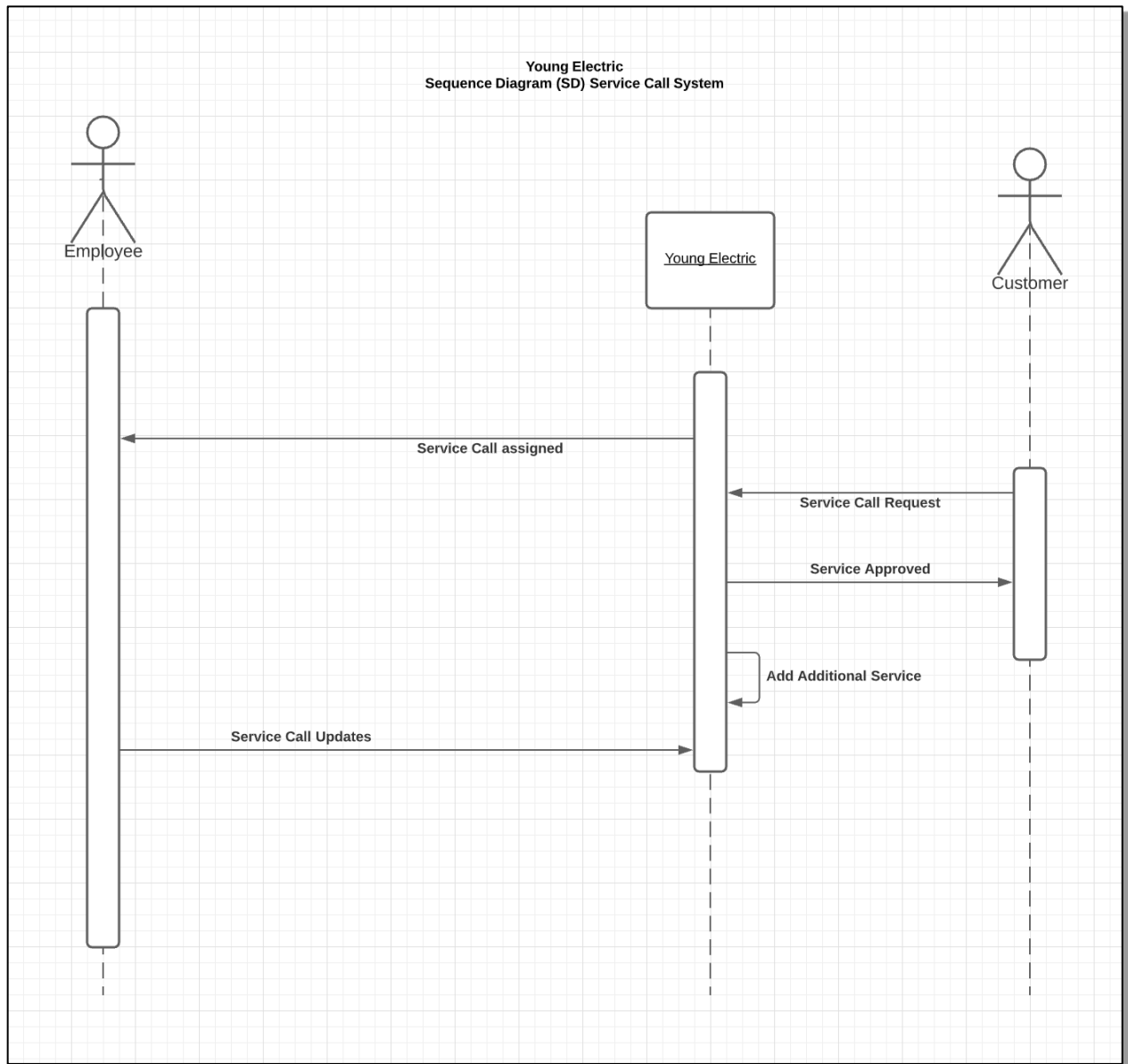
Context Diagram System



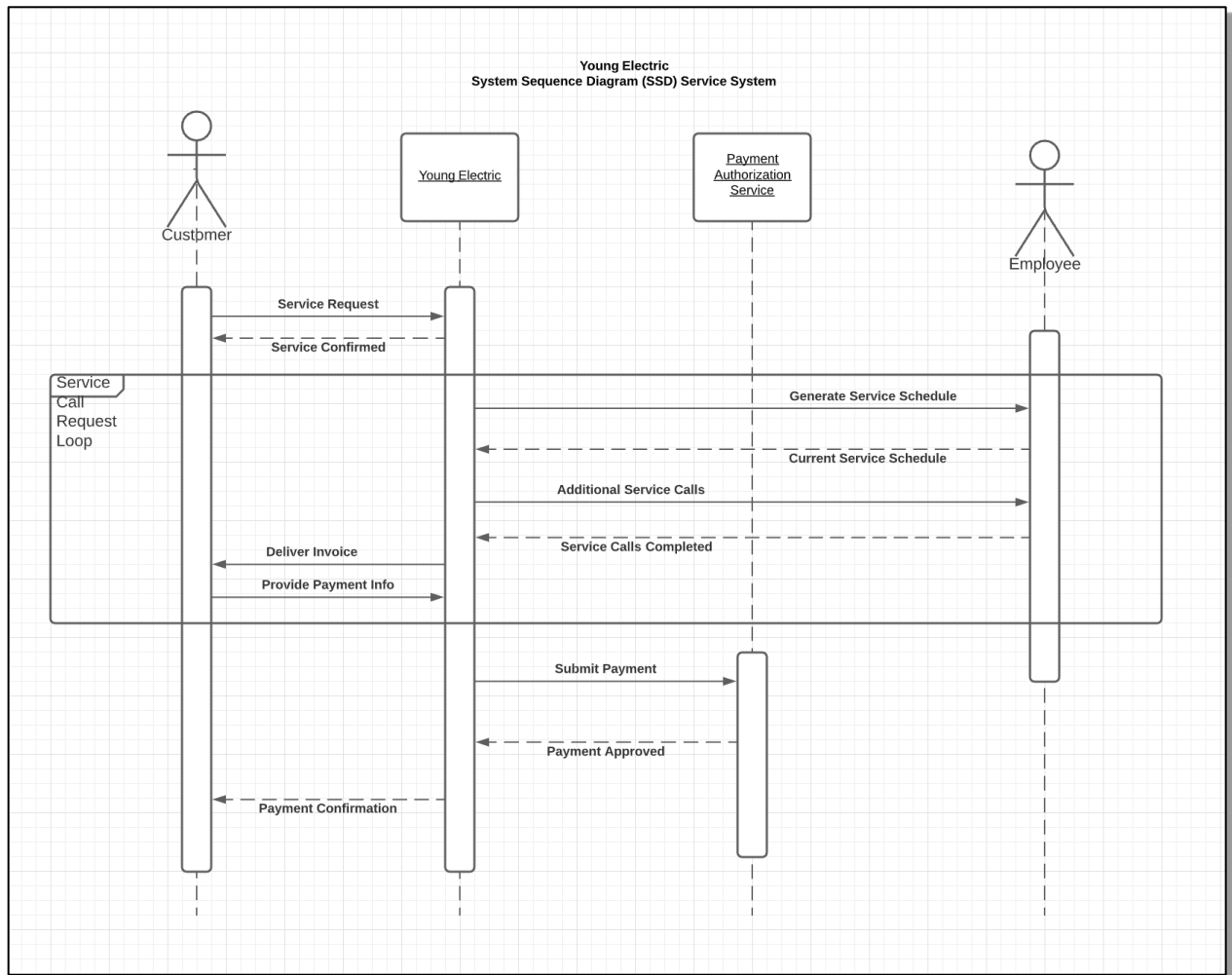
Service Request Use Case Diagram



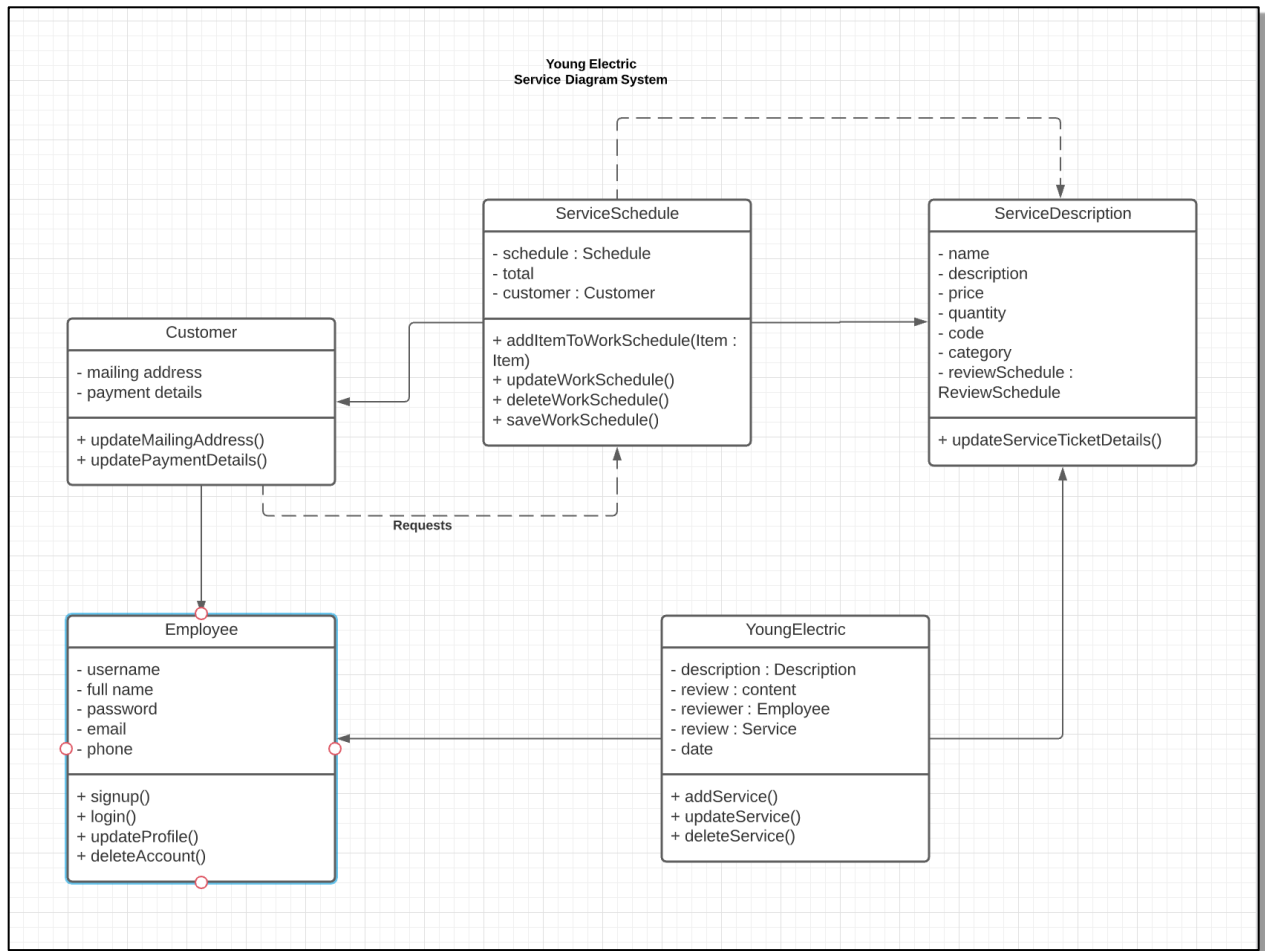
Sequence Diagram (SD)



System Sequence Diagram (SSD)



Service Diagram System



System Construction

Database Normalization

invoice_id	invoice_customer_id	invoice_employee_id	invoice_inventory_id	invoice_branch_id
customer_id	cus_invoice_id	cus_branch_id		
employee_id	emp_invoice_id	emp_branch_id		
dependents_id	dep_employee_id			
vendor_id				
inventory_id	inv_vendor_id	inv_item_id		
item_id	ite_invoice_id	ite_inventory_id		
generator_id	gen_customer_id	gen_employee_id	gen_vendor_id	
branch_id				
Primary Key		Foreign Key		

Tables

Invoice																
invoice_id	invoice_customer_id	invoice_employee_id	invoice_inventory_id	invoice_branch_id	Purchase Order	Service Date	Customer Last Name	Customer First	Hours worked	Total	Tax	Grand Total	Invoice Created	Date Posted		
Customer																
customer_id	cus_invoice_id	cus_branch_id	Last Name	First Name	MI	Address	City	State	Zip	Phone	eMail					
Employee																
employee_id	emp_invoice_id	emp_branch_id	Last Name	First Name	MI	SSN	Address	City	State	Zip	DOB	Phone	eMail	Hire Date		
Dependents																
dependents_id	dep_employee_id	Last Name	First Name	MI	DOB	Relationship	Sex									
Vendor																
vendor_id	Name	POC	Address	City	State	Zip	Phone	eMail								
Inventory																
inventory_id	inv_vendor_id	inv_item_id	Manufacturer	Manufacturer #	Description	Type	Order Date	Qty Ordered	Item Cost	Customer %	Customer Cost	Item Total	Qty on Hand	On Hand Total	Qty Used	Adj on Hand
Item																
item_id	ite_invoice_id	ite_inventory_id	Description	Qty	Cost	Total										
Generator																
generator_id	gen_customer_id	gen_employee_id	gen_vendor_id	Last Name	First Name	Serial Number	Model Number	Instll Date	Service Date							
Branch																
branch_id	Branch Number	POC	Address	City	State	Zip	Phone	eMail								

```

1 CREATE TABLE invoice
2 (
3     invoice_id                varchar(25) NOT null PRIMARY KEY,
4     invoice_customer_id       varchar(25),
5     invoice_employee_id       varchar(25),
6     invoice_inventory_id      varchar(25),
7     invoice_branch_id        varchar(25),
8     invoice_purchase_order    varchar(1000),
9     invoice_service_date      date,
10    invoice_cus_last_name     varchar(1000),
11    invoice_cus_first_name    varchar(1000),
12    invoice_emp_last_name     varchar(25),
13    invoice_emp_first_name    varchar(1000),
14    invoice_hours_worked      int(11),
15    invoice_date_posted       date,
16    invoice_total              decimal(12,2),
17    invoice_tax                decimal(12,2),
18    invoice_grand_total        decimal(12,2),
19    invoice_created            date,
20
21    FOREIGN KEY (invoice_customer_id) REFERENCES customer(customer_id) ON DELETE CASCADE,
22    FOREIGN KEY (invoice_employee_id) REFERENCES employee(employee_id) ON DELETE CASCADE,
23    FOREIGN KEY (invoice_inventory_id) REFERENCES inventory(inventory_id) ON DELETE CASCADE,
24    FOREIGN KEY (invoice_branch_id) REFERENCES branch(branch_id) ON DELETE CASCADE
25 );

```

```

1 CREATE TABLE generators
2 (
3     generator_id              varchar(25) NOT null PRIMARY KEY,
4     gen_customer_id           varchar(25),
5     gen_employee_id           varchar(25),
6     gen_vendor_id             varchar(25),
7     gen_emp_last_name         varchar(1000),
8     gen_emp_first_name        varchar(1000),
9     gen_serial                 varchar(25),
10    gen_model                  varchar(1000),
11    gen_install_date           date,
12    gen_service_date           date,
13
14    FOREIGN KEY (gen_customer_id) REFERENCES customer(customer_id) ON DELETE CASCADE,
15    FOREIGN KEY (gen_employee_id) REFERENCES employee(employee_id) ON DELETE CASCADE,
16    FOREIGN KEY (gen_vendor_id) REFERENCES vendor(vendor_id) ON DELETE CASCADE
17 );

```

```

1 CREATE TABLE employee
2 (
3     employee_id          varchar(25) NOT null PRIMARY KEY,
4     emp_invoice_id       varchar(25),
5     emp_branch_id        varchar(25),
6     emp_last_name        varchar(1000),
7     emp_first_name       varchar(1000),
8     emp_MI               varchar(1000),
9     emp_SSN              varchar(1000),
10    emp_address           varchar(1000),
11    emp_city              varchar(1000),
12    emp_state             varchar(1000),
13    emp_zip               varchar(1000),
14    emp_DOB               date,
15    emp_phone             varchar(1000),
16    emp_eMail             varchar(1000),
17    emp_hire_date         date,
18
19    FOREIGN KEY (emp_invoice_id) REFERENCES invoice(invoice_id) ON DELETE CASCADE,
20    FOREIGN KEY (emp_branch_id) REFERENCES branch(branch_id) ON DELETE CASCADE
21 );

```

```

1 CREATE TABLE branch
2 (
3     branch_id            varchar(25) NOT null PRIMARY KEY,
4     bra_number            varchar(25),
5     bra_POC              varchar(1000),
6     bra_address           varchar(1000),
7     bra_city             varchar(1000),
8     bra_state            varchar(1000),
9     bra_zip              varchar(1000),
10    bra_phone             varchar(1000),
11    bra_eMail             varchar(1000)
12 );

```

```

1 CREATE TABLE customer
2 (
3     customer_id          varchar(25) Not null PRIMARY KEY,
4     cus_invoice_id       varchar(25),
5     cus_branch_id        varchar(25),
6     cus_last_name        varchar(1000),
7     cus_first_name       varchar(1000),
8     cus_MI               varchar(1000),
9     cus_address          varchar(1000),
10    cus_city              varchar(1000),
11    cus_state             varchar(1000),
12    cus_zip               varchar(1000),
13    cus_phone             varchar(1000),
14    cus_eMail             varchar(1000),
15
16    FOREIGN KEY (cus_invoice_id) REFERENCES invoice(invoice_id) ON DELETE CASCADE,
17    FOREIGN KEY (cus_branch_id) REFERENCES branch(branch_id) ON DELETE CASCADE
18 );

```

```

1 CREATE TABLE inventory
2 (
3     inventory_id          varchar(25) NOT null PRIMARY KEY,
4     inv_vendor_id         varchar(25),
5     inv_item_id           varchar(25),
6     inv_manufacturer      varchar(1000),
7     inv_manufacturer_number varchar(1000),
8     inv_description        varchar(1000),
9     inv_type              varchar(1000),
10    inv_order_date         date,
11    inv_qty_ordered        decimal(12,2),
12    inv_item_cost          decimal(12,2),
13    inv_cus_percentage     varchar(1000),
14    inv_cus_cost           decimal(12,2),
15    inv_item_total         decimal(12,2),
16    inv_qty_on_hand        decimal(12,0),
17    inv_on_hand_total      decimal(12,0),
18    inv_qty_used           decimal(12,2),
19    inv_adj_on_hand        decimal(12,2),
20
21    FOREIGN KEY (inv_vendor_id) REFERENCES vendor(vendor_id) ON DELETE CASCADE,
22    FOREIGN KEY (inv_item_id) REFERENCES item(item_id) ON DELETE CASCADE
23 );

```

```

1 CREATE TABLE dependents
2 (
3     dependents_id          varchar(25) NOT null PRIMARY KEY,
4     dep_employee_id        varchar(25),
5     dep_last_name          varchar(1000),
6     dep_first_name         varchar(1000),
7     dep_MI                 varchar(1000),
8     dep_DOB               date,
9     dep_relationship       varchar(1000),
10    dep_sex                varchar(1000),
11
12    FOREIGN KEY (dep_employee_id) REFERENCES employee(employee_id) ON DELETE CASCADE
13 );|

```

```

1 CREATE TABLE vendor
2 (
3     vendor_id              varchar(25) Not null PRIMARY KEY,
4     ven_name               varchar(1000),
5     ven_POC               varchar(1000),
6     ven_address            varchar(1000),
7     ven_city              varchar(1000),
8     ven_state              varchar(1000),
9     ven_zip               varchar(1000),
10    ven_phone              varchar(1000),
11    ven_eMail              varchar(1000)
12 );|

```

```

1 CREATE TABLE item
2 (
3     item_id                varchar(25) Not null PRIMARY KEY,
4     ite_invoice_id         varchar(25),
5     ite_inventory_id       varchar(25),
6     ite_description        varchar(1000),
7     ite_qty               decimal(12,2),
8     ite_cost              decimal(12,2),
9     ite_total             decimal(12,2),
10
11    FOREIGN KEY (item_invoice_id) REFERENCES invoice(invoice_id) ON DELETE CASCADE,
12    FOREIGN KEY (item_inventory_id) REFERENCES inventory(inventory_id) ON DELETE CASCADE
13 );|

```


Website Design

Save Delete Clone Print Invoice Back

Main

Invoice Number

Branch

Customer

First Name

Employee Assigned

First Name

Service Date

Hours Worked

Items

	Item Name	Quantity	Cost	Total	
	<input type="text" value="Legrand"/>	<input type="text" value="2807W"/>	<input type="text" value="2"/>	<input type="text" value="\$ 4.95"/>	<input type="text" value="\$ 9.90"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>

Invoice Created

Date Posted

Total

Tax

Grand Total



Invoice Date :	Invoice #
2021-06-28	Bell_06_28_2021
REMIT TO :	Page #
YOUNG ELECTRIC COMPANY P.O. Box 699 Cabot, AR 72023	1 of 1

BILL TO :

Bell Tinker
123 Neverland Cr.
Neverland AL
78934 987.654.2468

SHIP TO :

Bell Tinker
123 Neverland Cr.
Neverland AL
78934 987.654.2468

Invoice Number	Customer PO Number	Ordered By :	Branch #		
Bell_06_28_2021		Bell	001		
Invoice Created By :	Terms :	Service Provided By :	Service Date :		
			2021-06-28		
Manufacturer :		Manufacturer Number :	Qty :	Item Cost :	Item Total :
Legrand		2807W	2	\$ 4.95	\$ 9.90

Invoice is due by : 14-12- 2121

Past due invoices are subject to 1.5% late charge.

Tax Jurisdiction ARLOCB

Young Electric is Licensed, Bonded and Insured

Total	\$ 9.90
State Tax	\$ 0.99
Grand Total	\$ 10.89

Printed : 14-12-2121

Table Input

Input to the database will take place during the testing and modification phase.

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.