Design Document

CSC425 BSCS Integrated Project

Trident University International

Project: ServiceOrders

Jason Bergener

December 24, 2017

Contents

[Business Process Description 3](#_Toc501898597)

[Phase 1 – Customer Contact 3](#_Toc501898598)

[Decision Point 1 3](#_Toc501898599)

[Decision Point 2 3](#_Toc501898600)

[Phase 2 – Creating the Service Order 3](#_Toc501898601)

[Phase 3 – Receiving the Service Order 4](#_Toc501898602)

[Phase 4 – Updated the Service Order 4](#_Toc501898603)

[Business Process Diagram 4](#_Toc501898604)

[System Architecture 5](#_Toc501898605)

[Subsystems 5](#_Toc501898606)

[User Interface 5](#_Toc501898607)

[\*User Login 5](#_Toc501898608)

[Customer Query 5](#_Toc501898609)

[Customer Query Results 5](#_Toc501898610)

[Create Customer Profile 6](#_Toc501898611)

[View Customer Profile 6](#_Toc501898612)

[Edit Customer Profile 6](#_Toc501898613)

[Create Service Order 6](#_Toc501898614)

[View Service Order 6](#_Toc501898615)

[View Open Service Orders 6](#_Toc501898616)

[Database 6](#_Toc501898617)

[Customers 7](#_Toc501898618)

[Service Orders 7](#_Toc501898619)

[Entity Relationship Diagram 7](#_Toc501898620)

[\*Location Services 8](#_Toc501898621)

[Required Interfaces Between Subsystems 8](#_Toc501898622)

[Project Status 8](#_Toc501898623)

[References 9](#_Toc501898624)

# Business Process Description

The business process that is being addressed by this project is the handling of customer service requests. This business process is part of a lager overarching business process of providing requested services to customers, but does not include the actual service. The service itself is handled separately and is based on the requirements of the customer. Handling customer service requests is a four-phase process:

Phase 1 – Customer Contact: Customer contact with the service provider can be from numerous methods to include phone, online chat, email, or postal mail and can be initiated by the customer or the service provider. The method of contact is irrelevant to this process, but it is during this stage that two decision points are reached. The first step is to run a query to determine if the customer has an existing profile as this will affect the information required to proceed.

Decision Point 1: The first decision point is the determination if the information provided is enough to proceed with the service order or if additional information is required. This also may include information required to create a customer profile is this is a new customer.

Decision Point 2: The second decision point is based on if the customer has an existing customer profile or if a new profile needs to be created.

Once both decision points have been satisfied, the process can move on to the second phase.

Phase 2 – Creating the Service Order: A service order is created using the information obtained from the customer and made available to the appropriate employee(s).

Phase 3 – Receiving the Service Order: The employee or team responsible for providing the requested service(s) receives the service order and is able to locate the customer and provide the requested service(s) based on the information provided in the service order.

Phase 4 – Updated the Service Order: The employee or team updates the status of the service order based on services rendered. This could also be an update that includes information about why services were not rendered and/or any follow-on action(s) required.

Business Process Diagram:

Customer Contact

Customer Query

Existing Customer?

Enough Information?

Create Service Order

Receive Service Order

Provide Service(s)

Create Customer Profile

Update Service Order

Gather Additional Information

Yes

Yes

No

No

# System Architecture

The project is designed to be accessible as a web application from any client with a current web browser. This will require a server for the web application, a server for the database, and client devices (desktop, laptop, or mobile) for the employees. The web application and database servers will be hosted in the cloud using Microsoft Azure. Client devices are the responsibility of the service provider.

The project integrates location services provide by Google Maps. This requires the server providing this web service from Google is working for this subsystem to operate correctly.

# Subsystems

The project is broken down into three subsystems. Subsystems marked with an asterisk (\*) are pre-existing.

User Interface: The user interface is how the employees interact with the web application. The user interface will be divided into functional views based on the task at hand.

\*User Login: This view will consist of text block labels for email and password with corresponding text boxes for input and a button to submit the information to the web server. There will be an option to register for an account as well as reset a password if it has been forgotten.

Customer Query: This view will consist of text block labels for first and last name with corresponding text boxes for input and a button to submit the information to the web server.

Customer Query Results: This view will list customers found with matching information in the database and will include the ability to select any existing customer or create a new customer profile.

Create Customer Profile: This view will consist of text block labels for customer information with corresponding text boxes for input and a button to submit the information to the web server. Submission will automatically re-direct to the View Customer Profile for the new customer.

View Customer Profile: This view will consist of text block labels for customer information with corresponding text blocks displaying the information and a button to edit the profile. There will be a table showing previous and current service orders with the ability to view them as well as a button to create a new service order.

Edit Customer Profile: This view will mimic the appearance of the Create Customer Profile view but with the current information filled in each textbox.

Create Service Order: This view will display customer information and include a text box for describing the requested service(s). The physical address will automatically populate with information from the customer’s mailing address, but can be changed to specify another location for the service(s). There will be a button to submit the form to the web server.

View Service Order: This view will display customer information, description of service(s) requested, status of the service order, and a text box for annotating any notes (service rendered, reasons services could not be rendered, next steps, etc.). There will be a button to submit the form to the web server.

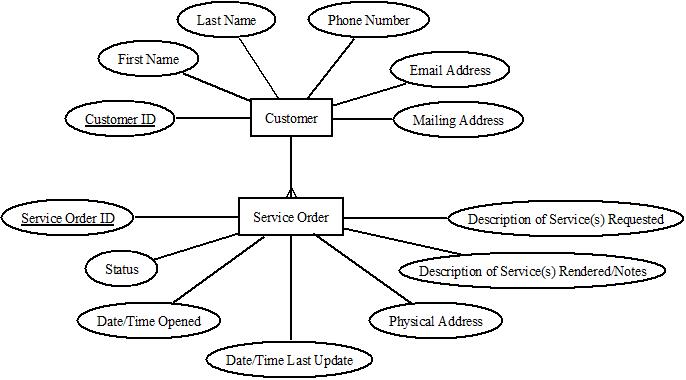
View Open Service Orders: This view will display a list of all service orders with a status of open.

Database: The database for the project is simplified due to the small scale and consists of three tables. The customers table contains customer information, the service orders table contains information regarding each service order, and the tables with user information are generated by Visual Studio during project creation.

Customers: The customers table will have an integer customer ID as the primary key and contain columns for first name, last name, phone number, email address, and mailing address. There is a one-to-many relationship between customers and service orders.

Service Orders: The service orders table will have an integer service order ID as a primary key and contain columns for status, date/time the service order was opened, date/time the service order was last updated, physical address, description of service(s) requested, description of service(s) rendered/notes, and a foreign key reference to the customer ID.

Entity Relationship Diagram:



\*Location Services: Location Services will be provided using an embedded Google Map API. The map is rendered using a simple HTML iframe with a string that determines the information provided by the map (Google, n.d.).

# Required Interfaces Between Subsystems

The user interface and database will interface using Entity Framework. “Entity Framework is an object-relational mapper (O/RM) that enables .NET developers to work with a database using .NET objects. It eliminates the need for most of the data-access code that developers usually need to write” (Microsoft, n.d.).

The interface between the user interface and the Location Services is the source string in the iframe. The string will be dynamically generated using a combination of Entity Framework to access the physical location from the database and Razor to create the string with that location.

# Project Status

The project is progressing nicely. The project is currently on schedule with the initial framework established, the database established, and the Location Services functioning. There are currently no roadblocks to development. Next steps include refining the user interface and restricting access to the site with login credentials.

# References

Google. (n.d.). *Google Maps Embed API*. Retrieved December 23, 2017, from Google Developers: https://developers.google.com/maps/documentation/embed/?authuser=1

Microsoft. (n.d.). *Entity Framework*. Retrieved December 23, 2017, from Microsoft Docs: https://docs.microsoft.com/en-us/ef/#pivot=entityfmwk&panel=entityfmwk1