|  |
| --- |
| American Video Game Company |
| CRM Solution |
| Customer Relationship Management System Proposal |

|  |
| --- |
| Therese M. Parks  3-2-2019  Version 1.0 |

Contents

[1. Introduction 3](#_Toc2449327)

[1.1. Purpose Statement 3](#_Toc2449328)

[1.2. Overview of the Problem 3](#_Toc2449329)

[1.3. Goals and Objectives 3](#_Toc2449330)

[1.4. Prerequisites 3](#_Toc2449331)

[1.5. Scope 4](#_Toc2449332)

[1.6. Environment 4](#_Toc2449333)

[2. Requirements 4](#_Toc2449334)

[2.1. Contact Management 5](#_Toc2449335)

[2.2. Data Types 5](#_Toc2449336)

[2.3. Ticketing System 5](#_Toc2449337)

[2.4 Activity Management 5](#_Toc2449338)

[2.5 Order Management 5](#_Toc2449339)

[1. SOFTWARE DEVELOPMENT METHODOLOGY 6](#_Toc2449340)

[1.1. Advantages/Disadvantages of the Waterfall methodology 6](#_Toc2449341)

[1.2. Advantages/Disadvantages of the Agile methodology 7](#_Toc2449342)

[The Agile Methodology is Best Suited 7](#_Toc2449343)

[2. Design 9](#_Toc2449344)

[Diagram for Order Management 9](#_Toc2449345)

[2.1. UML Diagram of Data Types 10](#_Toc2449346)

[3. Testing 11](#_Toc2449347)

[3.1. Testing Data Types 11](#_Toc2449348)

[3.1.1. Testing Data Types Requirement 11](#_Toc2449349)

[3.1.2. Testing Order Management Requirement 13](#_Toc2449350)

[3.1.3. Testing Activity Management 15](#_Toc2449351)

[1. Sources 18](#_Toc2449352)

# Introduction

The proposed solution is a new customer relationship management (CRM) system for the American Video Game Company. There are several disconnected manual and automated process which the new system will be able to manage in a unified way. The proposed system is a custom development and will be able to undergo changes and enhancements that may be required in the future, and it will be a scalable solution. Also, the proposed system will have the ability to integrate with other systems for the improvement of the efficiency of data sharing, reporting, and business process flows.

# Purpose Statement

The purpose of this document is to demonstrate how the proposed system, a new customer relationship management (CRM) system, can meet the requirements, objectives, and expectations of the American Video Game Company. The document provides detailed information for the review of the customer, so that the American Video Game Company can determine if the proposed system meets the needs of the organization.

# Overview of the Problem

The American Video Game Company is currently operating a system consisting of several disconnected manual and automated processes. Some issues springing from the current system may be the hinderance of efficiency, lack of communication, allowance for the unnecessary duplication of data, and time consumption. The new customer relationship management (CRM) system will solve this problem.

# Goals and Objectives

The goals of the new CRM are the unification, centralization, and security of data, increased efficiency, a user-friendly interface, a customer friendly interface. The solution will provide better management of several disconnected manual and automated processes, by meeting the requirements, expectations, and objectives of the American Video Game Company. The new system will be a crucial tool in being able to efficiently manage client contacts, perform sales tracking, maintain activity management, and manage reporting.

# Prerequisites

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Prerequisite | Description | Completion Date |
| 1. | Robust Server(s) | Based on the number of users, servers must be in place with enough Storage and Memory, Processors, the operating system, etc.. | 6/2/2019 |
| 2. | Database | A database must be selected to be installed on the server(s). | 6/2/2019 |
|  |  |  |  |

# Scope

The proposed solution will provide a newly created database that will allow for the migration of already existing data into a centralized, secure location. The proposed solution will provide in-house software for the employees of the American Video Game Company in all departments. The proposed solution will include an online customer portal which will be integrated into a dynamic webpage for optimal user performance. The proposed solution will not provide software to any individual or group that is not in the American Video Game Company. The proposed solution will not create a mobile application, but users can access a dynamic webpage using smart devices and still have an optimal experience.

# Environment

There will be an overall server-client system in place for the new CRM. Data will be stored on the Company’s secure server(s) and American Video Game Company users will access the new system via client based virtual machines. Therefore, the proposed system will be stored centrally and securely on the server(s) and will execute locally on each user’s machine. This way users who are not located in the office, may access the system from anywhere through a virtual private network (VPN) connection, with proper credentials and permissions. This set up would be the best solution for maximum security, unity of data, and improvement of communication among employees of the American Video Game Company.

# Requirements

There are several different requirements that the proposed system will be able to meet. These include Contact Management, Data Types, Activity Management, Opportunity Management, and Order Management. Below are some introductions to the requirements which the proposed system intends to meet. All the requirements are self-consistent as well as consistent with each other (UCertify, 2019).

# Contact Management

Users will be contacted according to the category of users they belong to and users will have the ability to manage their own contact settings. Each category of users will be given permissions and privileges that are unique to each category. Specific requirements will be in place for how users’ information may be used, according to the user category that they belong to. There will be two main categories of users: business users and end users, under which there will be more specific subcategories, with more specific permissions and privileges.

# Data Types

The data types which are required to be handled by the customer relationship management system will be placed into three categories: Stakeholders, Businesses, and Contacts. Each entity will be given its own details and relationships between the entities will be defined. There will be room for growth as the company grows, for example, relationships can be changed or added, existing entities may be changed, and new entities may be created. The proposed system will have reporting capabilities as well.

# Ticketing System

The proposed system will have a ticketing system that will be able to enter and track every communication and inquiry for contacts. For example, there may be several interactions with one contact regarding unrelated inquires, and the ticketing system will be able to keep track of each inquiry in a separate ticket, so that neither the contact nor the representative from the company get confused. Therefore, it will include unique instances and the system will be able to track email replies along with an audit trail to facilitate the workflow.

# Activity Management

The proposed system will have the ability to hold all data for past and future visits and meetings that take place between stakeholders and any members of the company involved. The system will have a two-way exchange between MS Exchange/Outlook. A system will include a ticketing system to cover any interaction with the contact which will include an audit trail. The system will have the ability to export and re-import data, for example data can be exported to MS Exchange/Outlook for a member of the company to send data to a stakeholder regarding an upcoming meeting.

# 2.5 Order Management

The proposed system will have the ability to keep track of all orders. The system will have the ability to process a quote into an order to complete the sale. Therefore, the system will be able to take orders, keep track of orders, reorder with the same information from a previous order, and reorder specific parts from a previous order. Additionally, the customer will have the ability to complete self-serve orders through a customer portal.

# SOFTWARE DEVELOPMENT METHODOLOGY

Choosing which development methodology to use is one of the first decisions that needs to be made for the implementation of this project. Once a methodology is chosen, the project will be undertaken according to a certain organized plan. Although the Waterfall is a common methodology to follow, it is expedient to compare this methodology to another very popular one, in order to be fully aware of the pros and cons of each; and, in order to make a more confident decision about which approach is best suited to use for this project. Therefore, a comparison will be made between the Waterfall methodology, and the Agile methodology.

The Waterfall methodology is a linear approach to software development, which basically follows these steps from the top down: gather and document requirements, design, implementation, verification, deployment, and maintenance (UCertify, 2019). In general, the Waterfall methodology requires that each stage is completed before the next one can begin.

On the other hand, the Agile mythology is an iterative, team-based approach, which emphasizes the rapid delivery of an application in complete functional components, with tasks and schedules “time-boxed” into phases called “sprints” (Lotz, 2018). Deliverables are prioritized by business value as determined by the customer, and with a very high level of customer involvement being relied upon, sprints can be reprioritized. The Agile methodology does not require that one stage is completed before the other one begins. (Lotz, 2018).

# Advantages/Disadvantages of the Waterfall methodology

An advantage of the Waterfall methodology is that it requires customer involvement only at milestones. It also works well if the scope is known in advance or when changes are limited by contract terms (Lotz, 2018). Another advantage is that this methodology reduces risk in Firm Fixed Price contracts by getting an agreement up-front. This approach ensures that the customer gets everything they asked for. Since the Waterfall approach is Plan-Driven, it may reduce risk in face of certain constraints within a contract with a vendor and an external customer; for example, the government(Lotz, 2018).

A disadvantage of the Waterfall approach is that it is difficult to be adaptable to change, since contract terms sometimes restrict it. In some cases, contract terms may not allow partial success and may require ‘do everything’ or ‘all or nothing’ conditions (Lotz, 2018). Another disadvantage is that high degrees of synchronization may not work, since contracts may be issued to separate vendors for different aspects of the project; and, team coordination/synchronization is limited to handoff points (Lotz, 2018). A major disadvantage is that all deliverables are based on documented requirements, and a “customer may not see what will be delivered until it is almost finished, making changes difficult and costly” (Lotz, 2018).

# Advantages/Disadvantages of the Agile methodology

An advantage of the Agile methodology is that the customer is highly involved in the project, having frequent and early opportunities to see the work being delivered, and can make decisions and changes as the project is being developed (Lotz, 2018). Also, a customer has a strong sense of ownership because of this level of involvement, and the overall Agile approach is very user-focused. This leads to the advantage that this high level of customer involvement greatly reduces risk. Another advantage is that Agile “can more quickly produce a basic version of working software which can be built upon in successive iterations”(Lotz, 2018).

On the other hand, the Agile methodology has some disadvantages as well. One disadvantage is that the high level of customer involvement may become an issue if the customer does not wish to participate is such a way because of lack of time or interest (Lotz, 2018). There can be issues with time, cost, and additional work. For example, when the customer wants additional features throughout the project, or when some items set for delivery are not completed within the timeframe allotted (Lotz, 2018). Another disadvantage is that “the interactive nature of Agile development may lead to frequent refactoring if the full scope of the system is not considered in the initial architecture and design”(Lotz, 2018). Refactoring is therefore necessary to prevent degradation of the system’s quality.

# The Agile Methodology is Best Suited

Although both the Waterfall and the Agile methodologies have their advantages and disadvantages, it seems that the Agile methodology is best suited for this project. Using the Agile method will ensure that a system is built that can be enhanced and scaled, easy to use, intuitive and user friendly. A high level of involvement in the project can be very important to the American Video Game Company, which already has a system in place and desires to seamlessly integrate this new system to replace the old one. As mentioned earlier, when using the Agile methodology, deliverables are prioritized by business value as determined by the American Video Game Company, and with a very high level of customer involvement being relied upon, sprints can be reprioritized. The Agile methodology does not require that one stage is completed before the other one begins (Lotz, 2018).

This means that several of the requirements can be developed at the same time, and one may be completed or reprioritized before another, independently. This means that the project is never at a standstill and can be developed at a more rapid pace. Parts of the new system can even be tested or used before the entire system is developed. For example, while the database is being developed, individual parts of the system for the different departments can be developed. For example, the Marketing department may begin testing an application that interfaces with the database, since the Marketing departments data is already in the database, however the Sales department’s data has not yet been entered into the database, and therefore is not at this stage yet. For example, the Sales departments application is going to be very similar to the Marketing department’s application, but it will require more detailed and robust features than the Marketing departments. Thus, having the marketing department’s application tested while the Sale’s department’s is still under development, can be important since the customer can have somewhat of an idea of how the Sale’s application will function, based on the Marketing application’s testing results. Thus, the American Video Game Company can make decisions for adjusting features etc.… early in the process and throughout the development cycle. These are some reasons why the Agile method is better suited for this project.

# Design

The following charts demonstrate the proposed solution’s ability to handle the required order management process and the data types requirements. The following charts are displaying preliminary, high-level information. They are to be viewed as examples rather than final solutions for the order management process, and for the categorization of data types, until a complete and comprehensive review is conducted to make certain that all details are identified.

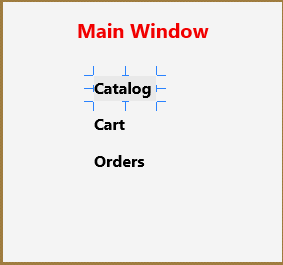
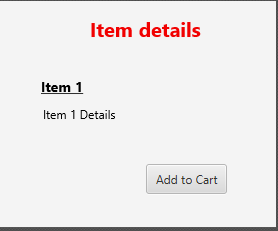
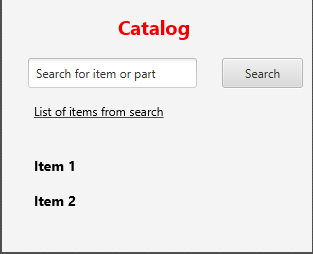
# Diagram for Order Management

The following diagram displays the high-level version of the Order Management process, which is the process of turning a quote into an order and completing a sale.

**For Catalog Window:** Type key word in the search bar. Click on an item to open the item in the list to open *the Item details window.*

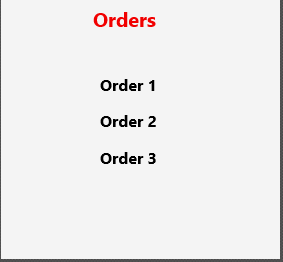
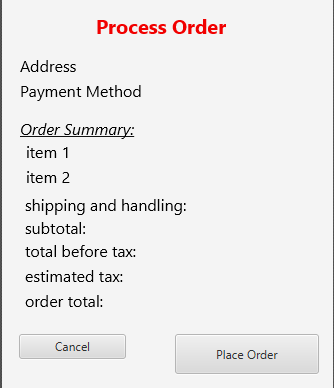
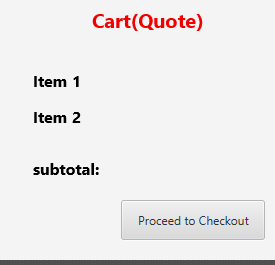
**For Main Window:** Click on Catalog link to open the *Catalog window*. Click on Cart link to open *Cart window*. Click on Orders link to open *Orders window*.

**For Item details Window:** Click on the button to add the item to the cart and to open the *Cart window.*

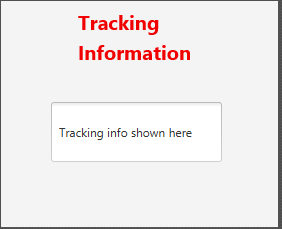
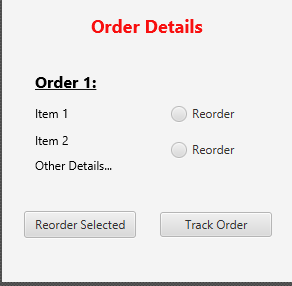


**For Orders Window:** Click on an order from the list to open the Order Details window.

**For Cart Window:** Cick on button to proceed to check and begin process to turn quote into order by opening the Process Order window.







**For Order Details Window:** Click on the track order button to get information about the current shipping location. Select one or many items or parts and click the Reorder button to open Process Order Window.

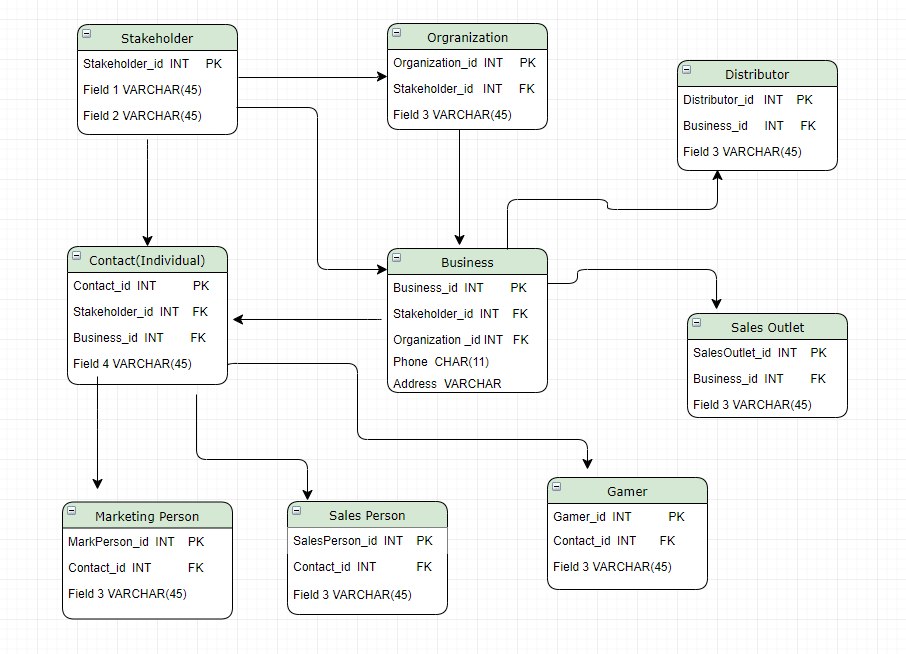
**For Tracking Information Window:** View tracking information. For example, from the carrier of the package in question.

**For Process Order Window:** Click on “Place Order” button to **complete the order.**

Note: The diagram above represents the order management process as it would work for both a customer representative, who is filling out an order for a customer; as well as, for a customer filling out an order in a self-service portal. The boxes represent the different windows or views pertaining to the information in association to the red titles, the plain textboxes give detailed information about the flow of the diagram, and the arrows represent a visual form of the flow of the diagram. In both cases, the process is relatively the same. Therefore, the proposed solution can meet these requirements.

# UML Diagram of Data Types

A universal database language, for example, SQL or a variation of SQL will provide a scalable and customizable solution for the creation, storage, and manipulation of data types. In the UML diagram below, each table represents an individual data type and relationships are displayed using foreign keys (FK) and the use of arrows.



# Testing

The following test cases are to be viewed as examples rather than final solutions for the for Data Types Requirement, the Order Management Requirement, and testing Activity Management Requirement, until a complete and comprehensive review is conducted to make certain that all details are identified.

# Testing Data Types

Before it can be determined whether the proposed solution can meet the Data Types Requirement, it is necessary that the Data Types Requirement be thoroughly tested, to prove that it can work seamlessly, free of any program errors. To do this the following test case must be run. The Data Types requirement will be working properly if the information can be added, updated, and deleted successfully.

# Testing Data Types Requirement

|  |
| --- |
| Requirement to be tested:  Data Types |
| Before the Data Types requirement can be tested it is necessary that the data base is already in place containing the required data types. The Data Types requirement would be working properly when the information can be added, updated, and deleted successfully without program errors. |
| Steps: The steps the tester must execute to test the data types.  Test a Stakeholder:   1. Create a new Stakeholder with new id and associated fields. 2. Retrieve an existing Stakeholder with id and associated information. 3. Update an existing Stakeholder’s information. 4. Retrieve the Stakeholder's updated information. 5. Delete an existing Stakeholder along with associated information. 6. Be unable to retrieve a Stakeholder along with associated information, once it has been deleted.   Test an Organization:   1. Create a new Organization with new id and associated fields. 2. Specify whether the Organization is also a Stakeholder. 3. Retrieve an existing Organization with id and associated information. 4. Update an existing Organization’s information. 5. Retrieve the Organization’s updated information. 6. Delete an existing Organization along with associated information. 7. Be unable to retrieve an Organization along with associated information, once it has been deleted.   Test a Business:   1. Create a new Business with new id and associated fields. 2. Specify whether the Business is also a Stakeholder. 3. Specify whether the Business is also an Organization. 4. Retrieve an existing Business with id and associated information. 5. Update an existing Business information. 6. Retrieve the Business’s updated information. 7. Delete an existing Business along with associated information. 8. Be unable to retrieve a Business along with associated information, once it has been deleted.   Test a Contact:   1. Create a new Contact with new id and associated fields. 2. Specify whether the Contact is also a Stakeholder. 3. Specify whether the Contact is also a Business. 4. Retrieve an existing Contact with id and associated information. 5. Update an existing Contact information. 6. Retrieve the Contact’s updated information. 7. Delete an existing Contact along with associated information. 8. Be unable to retrieve a Contact along with associated information, once it has been deleted.   Test a Distributor:   1. Create a new Distributor with new id and associated fields. 2. Specify whether the Distributor is also a Business. 3. Retrieve an existing Distributor with id and associated information. 4. Update an existing Distributor information. 5. Retrieve the Distributor’s updated information. 6. Delete an existing Distributor along with associated information. 7. Be unable to retrieve a Distributor along with associated information, once it has been deleted   Test a Sales Outlet:   1. Create a new Sales Outlet with new id and associated fields. 2. Specify whether the Sales Outlet is also a Business. 3. Retrieve an existing Sales Outlet with id and associated information. 4. Update an existing Sales Outlet information. 5. Retrieve the Sales Outlet’s updated information. 6. Delete an existing Sales Outlet along with associated information. 7. Be unable to retrieve a Sales Outlet along with associated information, once it has been deleted   Test a Marketing Person:   1. Create a new Marketing Person with new id and associated fields. 2. Specify whether the Marketing Person is also a Contact. 3. Retrieve an existing Marketing Person with id and associated information. 4. Update an existing Marketing Person’s information. 5. Retrieve the Marketing Person’s updated information. 6. Delete an existing Marketing Person along with associated information. 7. Be unable to retrieve a Marketing Person along with associated information, once it has been deleted   Test a Sales Person:   1. Create a new Sales Person with new id and associated fields. 2. Specify whether the Sales Person is also a Contact. 3. Retrieve an existing Sales Person with id and associated information. 4. Update an existing Sales Person’s information. 5. Retrieve the Sales Person’s updated information. 6. Delete an existing Sales Person along with associated information. 7. Be unable to retrieve a Sales Person along with associated information, once it has been deleted   Test a Gamer:   1. Create a new Gamer with new id and associated fields. 2. Specify whether the Gamer is also a Contact. 3. Retrieve an existing Gamer with id and associated information. 4. Update an existing Gamer’s information. 5. Retrieve the Gamer’s updated information. 6. Delete an existing Gamer along with associated information. 7. Be unable to retrieve a Gamer along with associated information, once it has been deleted |
| Expected results:  It is expected that a data type such as a Stakeholder, an Organization, a Distributor, a Contact, a Business, a Sales Outlet, a Marketing Person, a Sales Person, and a Gamer should all be added, updated, and deleted, without any program related errors. |
| Pass:  The criteria for this test passing are if all the expected results are yielded. In more detail:  In this case, a data type is any one of the following: A Stakeholder, an Organization, a Distributor, a Contact, a Business, a Sales Outlet, A Marketing Person, a Sales Person, or a Gamer.   * A new data type, along with its associated information, can be added without any errors. * An existing data type along with its associated information, can be updated without any errors. * An existing data type along with its associated information can be retrieved without any errors. * An existing data type along with its associated information can be deleted without any errors. * A deleted data type along with its associated information can no longer be retrieved once it has been deleted.   Since all the expected results are yielded, this test case passes. |

# Testing Order Management Requirement

Before it can be determined whether the proposed solution can meet the Order Management Requirement, it is necessary that the Order Management Requirement be thoroughly tested, to prove that it can work seamlessly, free of any program errors. To do this the following test case must be run.

|  |
| --- |
| Requirement to be tested:  Order Management |
| Preconditions:  It is expected that an application is already in place that provides a GUI interface between the user and the database. A customer must have an account to be able to place an order on his/her own through the self-serve portal, or with a sales representative’s assistance. |
| Steps: The steps the tester must execute to test the feature.  Test the Main Window:   1. Open Main window of application 2. Click on “Catalog” link to open Catalog 3. Click on “Cart” link to open Cart window 4. Click on “Orders” link to open Orders window   Test Catalog Window:   1. Search Catalog for items and parts. 2. View items and parts from the search in a list on the same window 3. Click on item in list to open Item Details Window   Test Item Details Window:   1. Click on button to add item to cart and to open the Cart (Quote) window   For Cart Window:   1. View list of items and read subtotal as a quote before the sale. 2. Click on button to proceed to open Process Order window   For Process Order Window:   1. Click on Place order button to complete the order.   For Orders Window:   1. View list of past orders 2. Click on an order to open Order Details Window   For Order Details Window:   1. View items and other order details from the order 2. Select item(s) and click on “Reorder Selected” to reorder the entire order or just a part of it. This action will open the Process Order Window. 3. Click on the “Track Order” button to open the Tracking Information window.   Tracking Information Window:   1. View Tracking information for the selected order. |
| Expected results:  It is expected that the user who tests the Order Management requirement can complete each step, providing he/she has the correct information, e.g. credit card information from the customer, and the carrier with whom the package is shipped, is able to provide tracking information in a timely manner. There should be no system related errors that hinder the flow of the order management process. |
| Pass:  The criteria for this test passing are if all the expected results are yielded. In more detail:  Main Window:   * The Main window of the application opens; the Catalog window opens when the “Catalog” button is clicked; the Cart window is opened when the “Cart” button is clicked; and the Orders window opens when the “Orders” window is clicked, without any program errors.   Catalog Window:   * The user can search the Catalog for items and parts, view items and parts from the search in a list of the same window and click on an item in the list to open the Item Details Window, without any program errors.   Item Details Window:   * The Cart window is opened when the “add to cart” button is clicked, without any errors.   Cart Window:   * The user can read a list of items and a subtotal as a quote before a sale, and the Process Order window is opened when the button is clicked, without any program errors.   For Process Order Window:   * The order is completed when the “Place order” button is pressed, without any program errors.   For Orders Window:   * The user can view past orders and click on an order in the list to open the Order Details window, without any program errors.   For Order Details Window:   * The user can view items and other details from an order, select item(s) and click on the “Reorder Selected” button to open the Process Order Window, and click on the “Track Order” button to open the Tracking Information window, without any program related errors.   Tracking Information Window:   * The user can view tracking information for the selected order, without any program related errors.   Since all the expected results are yielded, this test case passes. |

# Testing Activity Management

|  |
| --- |
| Requirement to be tested:  Activity Management |
| It is expected that the database is already in place. It is expected that an application is already in place that provides a GUI interface between the user and the database. The user must be signed into his/her account for the system to track his/her activity. |
| Steps: The steps the tester must execute for the Activity Management Requirement  Change a customer’s personal information:   1. The system will automatically validate permissions, record the representative’s information, and any updates made to an existing Customers personal information. 2. The system will automatically validate permissions, record, and automatically send an email to a customer when his/her personal information has been updated.   Create or Change a ticket:   1. The system will automatically validate permissions and record any adjustments made to a Customers ticket. 2. The system will automatically validate permissions, record, and send an email to a customer when his/her ticket has been created. 3. The system will automatically validate permissions, record, and send an email to a customer when his/her ticket has been completed.   Correspond with a customer via email:   1. The system will automatically record the representative’s information and any emails sent to a customer. 2. The system will automatically record the representative’s information and any emails received from a customer.   View/Audit   1. Open an Activity Management main screen to view all information on activities from the audit trail. |
| Expected results:  Every step from the test case should be written to an audit file that can be later viewed all at once from an Activity Management main screen. Any emails should be sent and received to and from customers, without any program related errors. |
| Pass:  The criteria for this test passing are if all the expected results are yielded. In more detail:  Change a customer’s personal information:   * The system automatically validates permissions, records the representative’s information, and any updates made to an existing Customers personal information, and this action is written to an audit file without any program related errors. * The system automatically validates permissions, records, and automatically sends an email to a customer when his/her personal information has been updated, and this action is written to an audit file without any program related errors.   Create or Change a ticket:   * The system automatically validates permissions and records any adjustments made to a Customers ticket, and this action is written to an audit file without any program related errors. * The system automatically validates permissions, records, and sends an email to a customer when his/her ticket has been created, and this action is written to an audit file without any program related errors. * The system automatically validates permissions, records, and sends an email to a customer when his/her ticket has been completed, and this action is written to an audit file without any program related errors.   Correspond with a customer via email:   * The system automatically records the representative’s information and any emails sent to a customer, and this action is written to an audit file without any program related errors. * The system automatically records the representative’s information and any emails received from a customer, and this action is written to an audit file without any program related errors.   View/Audit   * The user can open the Activity Management main screen to view all information on activities from the audit trail, without any program related errors.   Since all the expected results are yielded, this test case passes. |

# Sources

UCertify. (2019). Retrieved from https://wgu.ucertify.com/?func=ebook&chapter\_no=1#top

Lotz, M. (2018, July 5). Waterfall vs. Agile: Which is the right development methodology for your project. Retrieved from https://www.seguetech.com/waterfall-vs-agile-methodology/