|  |
| --- |
| Software Solutions Company |
| Software Project Proposal |
| American Video Game Company Customer Relation Management System Proposal |

|  |
| --- |
| Joshua James  10-26-2023  Version 1.0 |

Contents

[A. Introduction 3](#_Toc149226396)

[A.1. Purpose Statement 3](#_Toc149226397)

[A.2. Overview of the Problem 3](#_Toc149226398)

[A.3. Goals and Objectives 3](#_Toc149226399)

[A.4. Prerequisites 3](#_Toc149226400)

[A.5. Scope 3](#_Toc149226401)

[A.6. Environment 4](#_Toc149226402)

[B. Requirements 5](#_Toc149226403)

[B.1. Business Requirements 5](#_Toc149226404)

[B.2. User Requirements 5](#_Toc149226405)

[B.3. Functional Requirements 5](#_Toc149226406)

[B.4. NonFunctional Requirements 6](#_Toc149226407)

[C. Software Development Methodology 7](#_Toc149226408)

[C.1. Advantages of the Waterfall Method 7](#_Toc149226409)

[C.2. Disadvantages of the Waterfall Method 7](#_Toc149226410)

[C.3. Advantages of Sashimi Method 8](#_Toc149226411)

[C.4. Disadvantages of Sashimi Method 9](#_Toc149226412)

[C.5. Best Suited 9](#_Toc149226413)

[D. Design 10](#_Toc149226414)

[D.1. Storyboard (User Interface) 10](#_Toc149226415)

[D.2. GUI (Login and Main Menu) 11](#_Toc149226416)

[E. Testing 13](#_Toc149226417)

[E.1. Testing Type (Black Box Testing) 13](#_Toc149226418)

[E.1.1. Login Security Test 13](#_Toc149226419)

[E.1.2. Report Functionality Test 14](#_Toc149226420)

[E.1.3. Support Tickets Test 15](#_Toc149226421)

# Introduction

Software Solutions Company is excited to have the opportunity to propose a new Customer Relationship Management (CRM) system for the American Video Game Company (AVGC). The following proposal includes the custom development of the new CRM system while ensuring that the new system covers the key business requirements as described in the detailed CRM Requirements document. We are excited to have AVGC review our proposal and look forward to hearing from you soon.

# A.1. PUrpose Statement

The purpose of this document is to define the requirements for a new CRM solution for the American Video Game Company (AVGC).

# A.2. Overview of THE PROBLEM

AVGC has seen its sales grow by 42% over the last two years, leading to the company outgrowing its current CRM system. They require a new CRM system that provides better management of multiple disconnected manual and automated processes so that they can keep up with customer demand. The new CRM will need to be optimized to manage client contacts, reporting, activity management, and also perform sales tracking.

# A.3. Goals and Objectives

We wish to achieve the following goals and objectives upon creating the new CRM system for AVGC:

Manage all of the company’s contact and business information.

Based on the role and permission level of the company user, for both internal and remote access, access to features will be controlled and managed.

Sales tracking and reporting capabilities.

Robust security.

Be easy to use, intuitive, and user-friendly.

# A.4. Prerequisites

The following prerequisite is needed before the design, development, and implementation of the new CRM system project proposed in this document:

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Prerequisite | Description | Completion Date |
| 1 | Permission Levels and Roles | User permission levels and roles, and what each level and role can access. This is so that the CRM system features are controlled based on the user’s permission level and role. | 1 Week |

# A.5. Scope

The scope of our proposed solution for the new CRM system for AVG will cover the following:

* **Reporting:** Our proposed CRM system will allow for the production and saving of both predefined and custom reports on all data, including historical records. The reporting user interface (UI) will also be robust enough to allow a user to filter, format, query, and export data/reports.
* **Sales Tracking:** Our proposed CRM system will hold all data related to visits and meetings with stakeholders by AVGC.
* **Opportunity Management:** Our proposed CRM system will allow members of the sales teams to monitor sales processes, establish and manage pipelines, and perform various competitive analysis reports.
* **Historical and Versions of Records:** Our proposed CRM system will have the ability to have information saved and archived, including versions of records with auditing, workflow, and roll-back. This will also allow AVGC to be able to keep accurate historical records.
* **Soft and Hard Delete:** Our proposed CRM system will allow for data to be both soft and hard deleted based on the user's role and level of permissions.
* **Security:** Based on the level of a user’s permissions, a user’s access to data, workflow, and editorial controls, will be managed and controlled.

The scope of our proposed solution for the new CRM system for AVG, will not cover the following:

* At this time, Software Solutions Company will not replace any AVGC computers, computer hardware (monitors, mice, CPU, hard drive, etc.), network cables, servers, or any other physical equipment.
* Software Solutions Company will not be involved in AVGC employee management, HR services, payroll services, or marketing and advertisement.

# A.6. Environment

The following is a list of the IT and hardware environments that the new CRM system solution will be deployed in:

* Amazon Web Services (AWS)
* Microsoft SQL 2022
* Latest Chrome: Version 118.0.5993.80
* Latest Firefox: Version 118.0.2
* Latest Microsoft Edge: Version 118.0.2088.57
* Safari 6.0 and above (Latest Version: 17.0)
* Current mobile and tablet:
  + iOS7 Safari and above (Latest Version: 17.0)
  + Android 4.0 Chrome and above (Latest Version: 118.0.5993.65)

Note: Versions listed above are the most recent version as of October 26, 2023.

# Requirements

American Video Game Company (AVGC), a premier developer and publisher of computer games, requires a new customer relationship management (CRM) system that allows for better management of multiple manual and automated processes that are currently disconnected. The new proposed system should not replace the current system, but rather it should enhance the current system by meeting the company's new requirements, and allow for future scalability and enhancements. Achieving these goals will allow AVGC to focus on continuing to rapidly grow their number of clients, without needing to be concerned about outgrowing their aging, existing systems.

# Business Requirements

The new proposed CRM system will meet the AVGC’s business requirements by addressing the company’s user requirements, functional requirements, and nonfunctional requirements.

User requirements will be used to ensure the CRM allows users to maintain their client records for reporting, analysis, and historical records, in a secure and reliable environment.

Functional requirements will be used to ensure the CRM system has all of the functionalities specified by AVGC, such as the ability to "soft" and "hard" delete data and being able to differentiate the data a user can see based on their level of permissions.

Nonfunctional requirements will be used to ensure the CRM system is secure, maintainable, reliable, and scalable, as well as performing at industry performance standards.

# User Requirements

The user requirements will be addressed by the new CRM system, by meeting the specific software needs and expectations set directly by AVGC. These requirements are crucial for users of the new CRM system to be able to perform various vital tasks. The new proposed CRM system will meet the following AVGC’s user requirements:

* **Reporting:** Our proposed CRM system will allow for the production and saving of both predefined and custom reports on all data, including historical records. The reporting user interface (UI) will also be robust enough to allow a user to filter, format, query, and export data/reports.
* **Sales Tracking:** Our proposed CRM system will hold all data related to visits and meetings with stakeholders by AVGC.
* **Opportunity Management:** Our proposed CRM system will allow members of the sales teams to monitor sales processes, establish and manage pipelines, and perform various competitive analysis reports.

# Functional Requirements

Functional requirements are the product features or functions specifically requested by AVGC that the CRM system must include. These functions enable users to be able to perform and complete important tasks. The new proposed CRM system will meet the following AVGC’s functional requirements:

* **Historical and Versions of Records:** Our proposed CRM system will have the ability to have information saved and archived, including versions of records with auditing, workflow, and roll-back. This will also allow AVGC to be able to keep accurate historical records.
* **Soft and Hard Delete:** Our proposed CRM system will allow for data to be both soft and hard deleted based on a user’s role and level of permissions.
* **Security:** Our proposed CRM system will manage and control the user's access to data, workflow, and editorial controls, based on the level of the user's permissions.

# NonFunctional Requirements

Instead of describing what the CRM system will specifically do, like a functional requirement, a non-functional requirement describes how the CRM system will perform. Types of non-functional requirements include scalability, reliability, maintainability, and security. The new proposed CRM system will meet the following AVGC’s non-functional requirements:

* The new CRM system will be scalable, reliable, maintainable, and secure, and will meet or exceed current industry performance standards.

The following is a list of the IT and hardware environments that the new CRM system solution will be deployed in:

* Amazon Web Services (AWS)
* Microsoft SQL 2022
* Latest Chrome: Version 118.0.5993.80
* Latest Firefox: Version 118.0.2
* Latest Microsoft Edge: Version 118.0.2088.57
* Safari 6.0 and above (Latest Version: 17.0)
* Current mobile and tablet:
  + iOS7 Safari and above (Latest Version: 17.0)
  + Android 4.0 Chrome and above (Latest Version: 118.0.5993.65)

Note: Versions listed above are the most recent version as of October 26, 2023.

# SOFTWARE DEVELOPMENT METHODOLOGY

Software development methodologies are typically a series of processes used to develop I.T. software solutions. The following section will review the advantages and disadvantages of both the Waterfall methodology and the Sashimi methodology. After reviewing each method, we will then provide our recommendation on which methodology should be used in the development of the new CRM system.

# Advantages of the waterfall method

The Waterfall methodology is a linear, sequential approach to software development. This approach usually has an established timeline, fixed requirements that are highly unlikely to be changed during development, an experienced team with both ample resources and sufficient technology and enough time to do everything in a linear, sequential order.

There are multiple advantages to using this methodology, some of which are listed below:

* **Clear and Intuitive Structure:** Using a Waterfall methodology means that every project has to go through the same sequence of six phases (requirements, design, implementation, verification, deployment, and maintenance), with each phase needing to be completed fully before moving on to the next phase. This creates a clear, sequential, and defined set of steps or phases for a project that a team can easily and intuitively understand. In addition, due to each phase needing to be completed fully before moving to the next phase, any problems that arise during development can be easily and quickly identified.
* **Early Determination of End Goal or Product**: With the Waterfall methodology, the end goal or product, in this case, a new CRM system, is determined during the initial planning phase. This allows for a team to have clear, defined goals from the start of the project, which helps to lessen the potential for a project to accidentally complete unnecessary and time-wasting tangents.
* **Ease of Management and Stability:** The Waterfall methodology is easy to manage and stable due to its linear, sequential approach, and the need to complete each phase fully before moving on to the next. This allows the methodology to be easier to track progress, identify bottlenecks or delays, and stay on an agreed upon timescale, for each step or phase of a project.
* **Fewer Financial Surprises:** By using the Waterfall methodology, it is easier to more accurately predict the financial cost of the project, due to the project and each phase, being defined during the initial planning phase.

# disAdvantages of the waterfall method

While there are multiple advantages to using the Waterfall methodology, there are also multiple disadvantages as well. Some of the disadvantages for this methodology include:

* **Inflexibility:** One of the advantages of the waterfall methodology is that it is linear and sequential, where each phase is completed before moving on to the next, however, this is also one of the disadvantages due to it being very inflexible. For example, if AVGC requested to make a change late in the new CRM system’s development, the development team may not be able to proceed using the already defined phases and may have to essentially start the project over again.
* **Excludes End-Users and Clients and Increased Financial Cost:** After the initial planning stage where the project’s requirements are defined and goals are set, the Waterfall methodology differs from other methodologies, in that it does not solicit feedback from the end-users and clients throughout the development process. If a change needs to occur once end-users and clients start using the product, it can be much more costly to go back and rework the project, rather than potentially catching the change during the development process by soliciting feedback throughout.
* **Testing Delay:** The Waterfall methodology delays system testing more than other software development methodologies. By delaying testing longer, more resources are used before testing, meaning if a problem does arise during testing, it will not only take longer to fix, but it will also likely be more finically costly as well due to needing to backtrack further than if testing were scheduled earlier.

# Advantages of THE SASHIMI Method

The Sashimi software development methodology is similar to the waterfall methodology, in that there are the same six steps (requirements, design, implementation, verification, deployment, and maintenance) that a project will go through. However, unlike the Waterfall methodology, when using the Sashimi methodology, the steps are allowed to overlap with each other. This means that for example, while the requirements phase is still being worked on, the design phase can start to be developed. Once the design has started to be developed, other team members can start the implementation phase and start coding the designs that have been developed. One way to help visualize the Sashimi method is to imagine how sashimi is typically served, in that raw slices of fish overlap with each other, similar to how the six steps in the Sashimi methodology overlap with each other.

Some of the advantages to using the Sashimi methodology include:

* **Shortens Development Time:** Compared to the Waterfall methodology, the Sashimi methodology shortens development time. This is due to the steps or phases being able to overlap with each other, meaning that people or teams with various skills, can start working sooner. For example, while the requirements team works to finish the requirements phase, the design phase team can start working, instead of needing to wait for the requirements phase to be fully completed like in the Waterfall methodology.
* **Greater Usage of Resources:** Similar to shortening development time, due to the Sashimi methodology allowing for phases to overlap each other, more resources can be used during the same time frame. This contrasts with the Waterfall methodology, where resources may have to wait to be used, due to an earlier phase not being fully completed yet.
* **Learning Deep-Dive/Spike:** The Sashimi methodology allows a team member to perform a learning deep-dive/spike into a specific topic. For example, if there were an experimental feature that was desired to be added to a project, a team member could perform a learning deep-dive/spike into the experimental feature and create a prototype. Based on the prototype, the decision whether or not to incorporate the feature into the project can be made.

# disAdvantages of THE SASHIMI Method

While there are multiple advantages to using the Sashimi methodology, there are also multiple disadvantages as well. Some of the disadvantages for this methodology include:

* **Lost Work Effort and Time:** While it can be advantageous to have different phases being worked on at the same time, this can also lead to a disadvantage in the case of a team or team member potentially working too far ahead. If the requirements are modified, then the extra work the team or team member has done may need to be reworked or removed, which leads to lost work effort and time.
* **Tracking Progress:** When using the Waterfall methodology, it is easy to track phase and project progress due to being linear and sequential. The Sashimi methodology has multiple phases being worked on in parallel, therefore being able to accurately track progress becomes more complicated and ambiguous.
* **Communication:** With multiple phases being worked on simultaneously, communication between teams or team members working on different phases, can be more complex than compared to the Waterfall methodology where each phase is completed fully and sequentially before moving on to the next phase.

# best SUITED

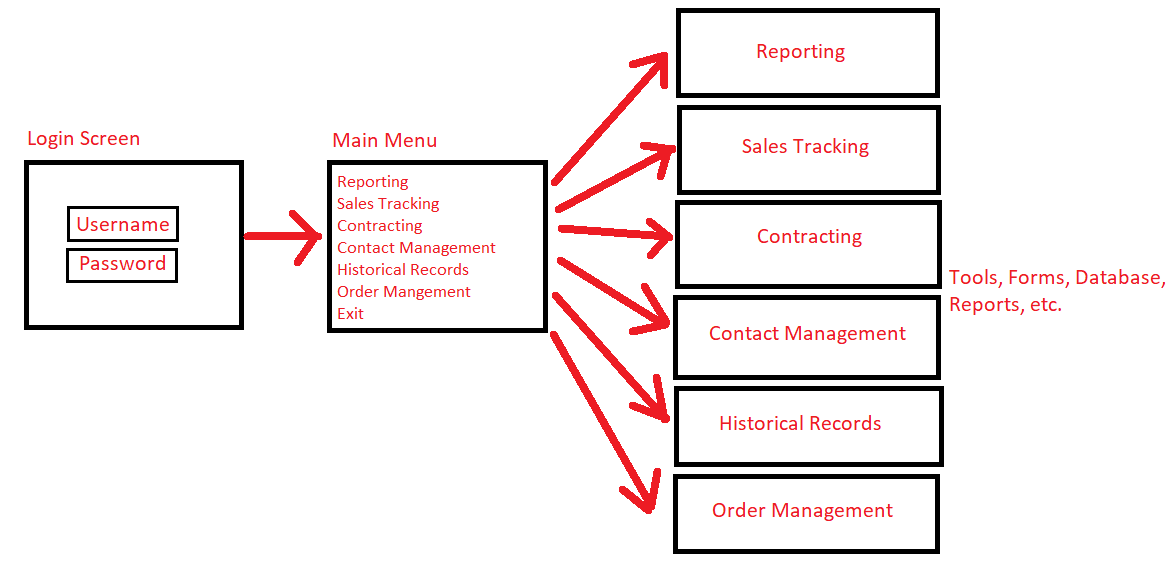
There are advantages and disadvantages to both the Waterfall and Sashimi methodologies. When considering the new CMS system proposal, we recommend that it is best suited to continue with AVGC’s current Waterfall software development methodology. The main reason why Software Solutions Company recommends this is due to the strong and detailed CRM requirements document AVGC provided, which specifically details the requirements the new CRM system will need. It is also anticipated that these requirements will not change during the development process. In addition, another reason why we recommend continuing with the Waterfall methodology is that the CRM requirements state that when it is possible, AVGC would prefer to keep existing business processes as much as possible. Finally, with the CMS requirements being well documented, clear, and unlikely to be modified, the Waterfall methodology can provide a more predictable and accurate estimation in regards to both the financial cost of the project, as well as how long the project should take to be completed. Based on these multiple reasons, Software Solutions Company recommends that the new CMS system be developed using the Waterfall methodology that AVGC already has in place.

# Design

In the following section, you will see a mock storyboard and two mock Graphical User Interface (GUI) examples. Please note, that these are only mock examples. The new CMS system may appear differently, including additional functionalities.

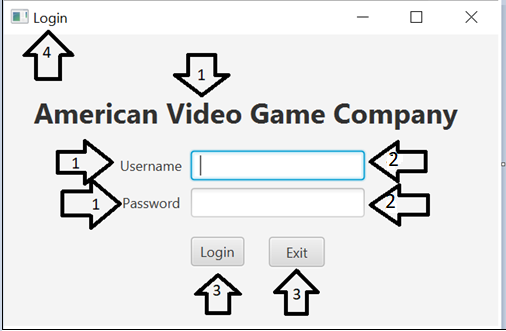
# Storyboard (User Interface)

Below is a high-level storyboard of a mock user interface for the new proposed AVGC CMS system. First, a user would log in using a valid username and password. Second, assuming their username and password were valid and matched a username and password from the Security database, they would be taken to the “Main Menu” in which they would be presented with multiple options, which may change depending on their role and permission level. Upon selecting an option from the “Main Menu”, they are then taken to that particular tool, form, database, report, etc.



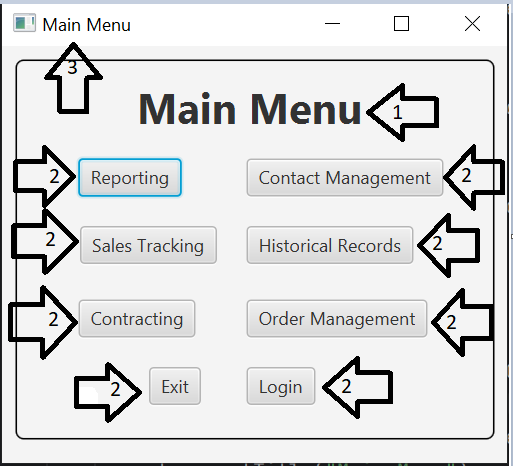
# GUI (Login and Main Menu)

The first mock GUI shows a login screen requiring a user to enter a username and password to log in to the American Video Game Company CMS system. As advised in the CMS Requirements document, AVGC is spread across multiple offices, has team members who work remotely, and wish to control access to features based on their role and permission level. Using a username and password system like this to log in users, will assist in ensuring the CMS system stays secure by limiting who can log in. In addition, once a user is logged in, they will only see what their role and permission level allow them to see.



|  |  |  |  |
| --- | --- | --- | --- |
| GUI Control Mapping | | | |
| ID | Control | Property | Data Source |
| 1 | Label | On the application open text = "American Video Game Company" | NA |
| 1 | Label | On application open text = “Username” | NA |
| 1 | Label | On application open text = “Password” | NA |
| 2 | Textbox | The user enters a username. | NA |
| 2 | Textbox | The user enters a password. | NA |
| 3 | Button | On click, log in to the application. | Internal Variable |
| 3 | Button | On click, exits the application. | Internal Variable |
| 4 | Form | Text= “Login” |  |

This second mock GUI shows the AVGC’s CMS system’s “Main Menu” once a user successfully logs in. Depending on their role and permission level, they may see more or fewer options as outlined in the CMS Requirements document. For this user, once they log in, they then are presented with the option of which tool they would like to access. For example, a user who needs to create a report would click on the "Reporting" button, or if a user wanted to check the status of an order, they would click the "Order Management" button.



|  |  |  |  |
| --- | --- | --- | --- |
| GUI Control Mapping | | | |
| ID | Control | Property | Data Source |
| 1 | Label | On the main menu scene open text = "Main Menu" | NA |
| 2 | Button | On click, go to “Reporting”. | Internal Variable |
| 2 | Button | On click, go to “Sales Tracking”. | Internal Variable |
| 2 | Button | On click, go to “Contracting”. | Internal Variable |
| 2 | Button | On click, go to “Contact Management”. | Internal Variable |
| 2 | Button | On click, go to “Historical Records”. | Internal Variable |
| 2 | Button | On click, go to “Order Management”. | Internal Variable |
| 2 | Button | On click, exit the application. | Internal Variable |
| 2 | Button | On click, return to the login menu. | Internal Variable |
| 3 | Form | Text= “Main Menu” |  |

# Testing

Software Solutions Company’s proposed new CRM system for AVGC will be tested using a black-box testing method. We will perform the following three black-box tests on the new CRM system:

* Login Security
* Report Functionality
* Support Tickets

# Testing Type (Black Box Testing)

With black-box testing, the tester knows what a method is supposed to do, but they do not have knowledge or details regarding how the method does it. We like to use this type of testing when creating software like the proposed CRM system, due to it most closely replicating the end user’s experience.

# Login Security Test

|  |
| --- |
| **The requirement to be tested:**  We will generate and use sample usernames and passwords to ensure the CRM system’s security system, specifically the login process, is working as intended and properly managing access to the system. |
| **Preconditions:**  Sample usernames and passwords must be generated and present in the security database so that when we test the login process, we will be able to validate that the system can compare and match the username and password entered versus the username and password in the database.  Generated usernames and passwords:  Username 1: AdminUser  Password 1: Admin123!  Username 2: TestUser  Password 2: Test123! |
| **Steps:**   1. The tester opens the new CMS system. 2. The tester will attempt to login using the following username and password:   Username: adminuser  Password: Admin123!   1. The tester will attempt to login using the following username and password:   Username: AdminUser  Password: Admin123   1. The tester will attempt to login using the following username and password:   Username: testuser  Password: Test123!   1. The tester will attempt to login using the following username and password:   Username: TestUser  Password: Test123   1. The tester will attempt to login using the following username and password:   Username: AdminUser  Password: Admin123!   1. The tester will need to log out of the CMS system and return to the login screen. 2. The tester will attempt to login using the following username and password:   Username: TestUser  Password: Test123!   1. Logout of CMS system. |
| **Expected results:**  We expect the following results to occur during the Login Security testing:   * Steps 2 and 4 are expected to fail to log in due to an incorrect username being entered. * Steps 3 and 5 are expected to fail due to login due to an incorrect password being entered. * Steps 6 and 8 are expected to successfully login due to both a correct username and password being entered that matches a username and password stored in the Security database. |
| **Pass/Fail:**  As expected, the Login Security test passed successfully. This is due to the CRM system not allowing a user to log in if they enter a username and password that does not match what is already stored in the security database. The only way for a user to log in successfully to the CRM system is to enter a username and password that matches the username and password in the database. By passing this test, we have demonstrated the CMS system’s Login Security functionality is working as intended. |

# Report Functionality test

|  |
| --- |
| **The requirement to be tested:**  This test will ensure the new CRM system’s reporting functionality is working as intended and can successfully generate accurate reports. |
| **Preconditions:**  AVGC sales data for the previous month must be in the sales database so that the reporting system can use the data when generating a report. In addition, a username and password will need to be provided to the tester at the time of testing. |
| **Steps:**   1. The tester opens the CMS system. 2. The tester logs into the CMS system using a provided username and password. Note: Username and password will be given to the tester at the time of testing. 3. On the “Main Menu” screen, select the “Reporting” button. 4. On the “Reporting” screen, select the “Run Report” button. 5. On the “Run Report” form, select “Sales” as the “Report Type”, and “Last Month” as the “Report Timeframe”, then select “Run”. 6. The report is generated. 7. Save the report with the name "Test Report 1". 8. Log out of the CMS system. 9. The development team will compare the saved generated report with the sales database data to ensure the report is accurate. |
| **Expected results:**  We expect that the tester will be able to successfully load and log in to the CMS system, navigate to the "Reporting" screen from the "Main Menu", and generate a report for sales that occurred last month. |
| **Pass/Fail:**  As expected, the tester was successfully able to generate a report for the sales that occurred last month. After the tester generated and saved the report, the development team was successful in testing the report’s accuracy, by comparing the report to the data in the sales database. By passing this test, we have demonstrated the new CMS system’s reporting functionality is working as intended. |

# Support Tickets Test

|  |
| --- |
| **The requirement to be tested:**  First, we will test that a tester can successfully submit a support ticket and view the status of the ticket. Second, we will test that a tester can successfully retrieve the submitted support ticket, and close it. Finally, we will test that a tester can see that the ticket has been closed. |
| **Preconditions:**  Two sets of usernames and passwords will need to be generated and provided to the tester at the time of testing. The first username and password set needs to have permissions set to allow for the submission of a support ticket. The second username and password set needs to have permissions set to allow for the ability to close a support ticket. |
| **Steps:**   1. The tester opens the CMS system. 2. The tester logs into the CMS system using the first provided username and password. Note: This username and password will be provided at the time of testing. 3. From the “Main Menu”, select the “Support Tickets” button. 4. From the "Support Tickets" screen, select the "New Support Ticket" button. 5. Enter the following information in the “New Support Ticket” form:   **Username:** Enter the provided username.  **Issue Type:** Select “Equipment” from the drop-down combo box.  **Issue Description:** Support Ticket Test 1.0 Equipment   1. Select “Submit Ticket”. 2. Verify “Your support ticket has been successfully submitted. Your ticker number is \_\_\_\_\_\_\_\_\_.” dialogue window appears. Please record the ticket number, then click the "OK" button to close the dialogue window. 3. Logout of CMS system. 4. The tester logs back into the CMS system using the second provided username and password. Note: This username and password will be provided at the time of testing. 5. From the “Main Menu”, select the “Support Tickets” button. 6. From the "Support Tickets" screen, select the "Open Tickets" button. 7. Verify that the ticket created during step 6 is “open” with the same ticket number as displayed in step 7. 8. Select the ticket to open the support ticket form. 9. On the support ticket form, enter the following information into the "Support Ticket Resolution" field:   **Support Ticket Resolution:** Support Ticket Test 1.0 Equipment Closed Successfully.   1. Select the “Close Ticket” button. 2. Verify the ticket no longer appears in “Open Tickets”. 3. Logout of CMS system. 4. The tester logs back into the CMS system using the first username and password again. 5. From the “Main Menu”, the tester will select the “Support Tickets” button. 6. From the "Support Tickets" screen, the tester will select the "View Support Ticket Status" button. 7. The tester will enter the support ticket number from step 7. 8. The tester will verify the support ticket opens, and that they can see the “Support Ticket Resolution” notes. The notes should state that the ticket is “Closed”. 9. Logout of CMS system. |
| **Expected results:**  First, we expect that the tester will be able to successfully log in to the CMS system, navigate to "Support Tickets", and be able to submit a new support ticket.  Second, we expect that the tester will be able to successfully log in to the CMS system, navigate to "Support Tickets", find the open ticket, notate, and close it.  Finally, we expect that the tester will be able to successfully log in to the CMS system, navigate to "Support Tickets", and view the status of their ticket, which will be "closed". |
| **Pass/Fail:**  As expected, the support tickets test was successfully passed. Through this test, we were able to demonstrate a user being able to submit a new ticket, for that ticket to be found and closed, and that the user was able to view the status of the ticket and see that it was closed. By successfully passing this test, we have shown that the support tickets functionality in the new CMS system is working as intended. |