Software Solution

American Video Game Company

CRM Solution Proposal

Overview of Solution and Business Requirements

Ryan Hildebrant – Student ID: 000709587

1-9-2020

[Version 1.0]

Western Governors University

TABLE OF CONTENTS

A. INTRODUCTION

The American Video game company has sent a document outlining their business vision to implement a new Customer Relationship Management (CRM) System. A Customer Relationship Management System is formally defined as an approach to managing a company's interaction about customers' history with a company to improve business relationships. More specifically, it focuses on customer retention, increased value, and ultimately drives sales growth. Such a system is useful to the American Video Game company because it offers a solution to organizing their process flows and making the most of customer interactions. This robust and scalable solution is required to match their rapid expansion while providing tools for improving the efficiency of data sharing, reporting, and business process flows. In the proceeding sections, we will explore the goals of a CRM, address the requirements of an effective solution, provide an overview of the CRM design, and address methods of user testing.

A.1. PURPOSE STATEMENT

This document provides an overview of our recommended software solution and outlines project management tools/methodologies required to achieve success with the software using a Software Development Lifecycle plan. Our software solution will be built inhouse based on our existing technologies and be called Reach. This CRM software is designed to provide a solution to the current and future business needs of The American Video Game Company.

A.2. OVERVIEW OF THE PROBLEM

The integrated software system will solve the problem the company is facing with rapid and continually company growth. Additionally, the software system will solve AVG's issue of having several disconnected manual and automated processes by integrating these components into the new system. Such systems include tools for managing custom contacts, tracking sales, generating sales reports, maintaining activity management, and producing related reports. The Reach CRM must also provide a robust security system that uses control access and connects with an existing Active Directory structure to enforce user access control. These integrations will try to mimic existing process flows as closely as possible but will also emphasis easy of use and operations clarity.

A.3. GOALS AND OBJECTIVES

The goals are outline as follows:

- To provide American Video Game Company with a CRM system which will meet the current business needs
    - Scale to current and any potential of future business needs
    - Take advantage of robust cloud-based software
    - Move all existing business processes into a uniform application
    - Implement a system that can mange data archives, record deletion, version control, auditing, and roll back
    - Implement data security and user access control for how removal and data flow is managed

  - Ensure all data is managed in house within the United States of America
  - Integrate into existing active directory hierarchies existing on current company servers
- Provide a system that consolidates all contact and business information
- Provide a system that is initiative to users and easy to use
- Enables restricted access to third party marketing companies under contract
- Provide on-going and easy to access maintenance teams
- Provide a robust reporting system that meets all the user requirements within the requirements document
- Provide a central repository for all the tools needed to provide sales tracking

A.4. PREREQUISITES

The project prerequisites are not extensive, but they are critical to the success of the CRM system.

Note: Below is an outline of the required prerequisites

Number	Prerequisite	Description	Completion Date
1	None	A requirements document must be provided that clearly outlines the goals and how those goals fit within the workflow. – This is the most important prerequisite	January 15th, 2020
2	1	The requirements document must be reviewed and accepted	January 20th, 2020
3	1,2	A budget for this project is allocated and obtained	January 20th, 2020
4	None	A complete overview of existing systems and their implementations is provided	January 20th, 2020

A.5. SCOPE

This document provides an overview of the business requirements (both functional and non-functional). These requirements must include functions stemming from the following areas: Contacts, Tickets, Data Control, Reports, Data Tracking, Quotes, Order Management, Support, and Forecasting. This project will not stem beyond the fields of customer management and should not be seen as a singular solution to all business related processes. Additionally, it provides a comparison of the proposed software development methodology and our suggested alternative. Lastly, this document outlines aspects of the design and provides a complete testing plan.

A.6. ENVIRONMENT

The software system is based on a cloud platform and must be compatible with all web browser applications in use by the company's staff and affiliated partners. These web browsers include

Google Chrome, Internet Explorer, Safari, Firefox, and Opera. Additionally, the must be compatible with the android and iOS versions of these browsers. User's should be able to login into their account through a provided website and view a complete graphical interface.

The interface is subject to remove certain features depending on the access control granted to the user account. Such access control can be accomplished by integrated the software with the existing active directory infrastructure the company is providing. Hosting of the cloud-based CRM software will be provided by our existing server infrastructure.

B. REQUIREMENTS

The American Video Game Company requires a new CRM system that can scale with their ongoing growth. They have increased their sales by 42% over the past two years and a new sales tracking system must be implemented. This tracking system must incorporate all their disconnected spreadsheets and integrate them into an adaptable environment. Additionally, the Reach CRM must include a set of tools for report management, client management, and data management. Lastly, the system is required to be user friendly and be thoroughly tested. Below you will find a more detailed outline of each major area of requirements.

B.1. BUSINESS REQUIREMENTS

The primary high-level goal of this project is to provide CRM software that is adaptable and can scale with future growth. At the time of this proposal, American Video Game Company has 2,000 and employees and it classifies 500 of these employees as active users during peak times. The system must scale to reach growth far beyond these numbers.

Additionally, our application will also address the following high-level goals:

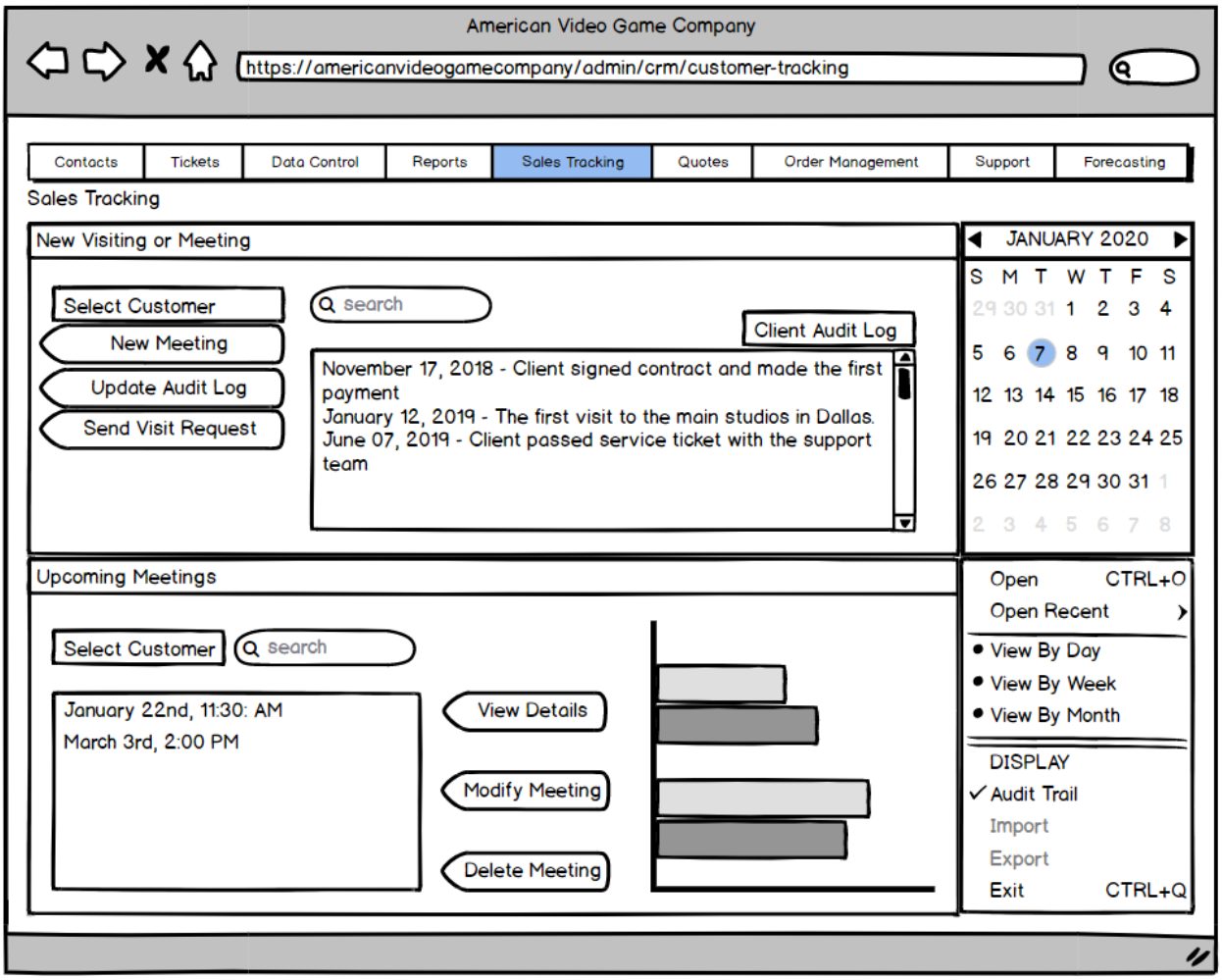
- Consolidate all business contacts and related information that is dispersed across the company
- Generate and manage reports of company's activities and interactions with contacts
- Integrate the company's existing activity directory system to allow user access control for both internal and external users – This will be further extended by allowing restricted user access to 3rd party marketing companies under contract
- Manage sales activities and generate reports to track different sales statistics
- Integrate existing systems to allow data-sharing and create an access point for future systems
- Promote a high level of security that is robust and compliant with best practices and current regulations
- Provide a roadmap for future enhances within a reasonable cost
- Provide a strong maintenance system
- Store data securely and ensure that it stays in the United States

B.2. USER REQUIREMENTS

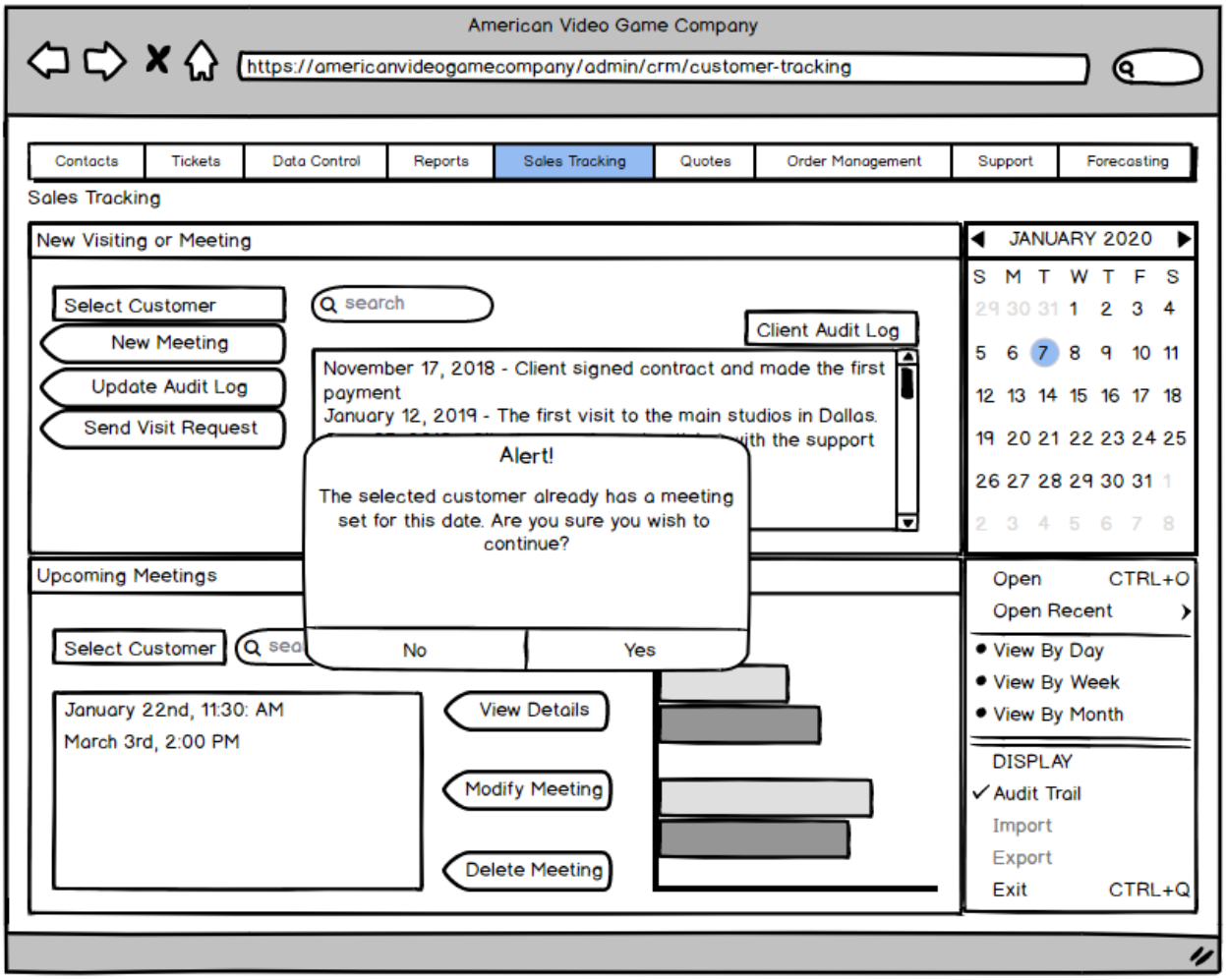
User requirements outline the how the application will be used by various sets of end-users, primarily the employees. More specifically, the employees with need to preform tasks to generate tickets, quotes, inquiry tags, orders, opportunity, tracking and create contracts. All these processes will also require the ability to generate and manage reports. Reports include current data

reporting, executive-level reporting, historical data reports, and customer activity (visits & meetings) reporting. Further details of these user requirements can be found in the next sub-section, Functional Requirements. Below we provide graphical example of the steps required to actively track customer visits and meetings. This example displays how to create a new customer appointment:

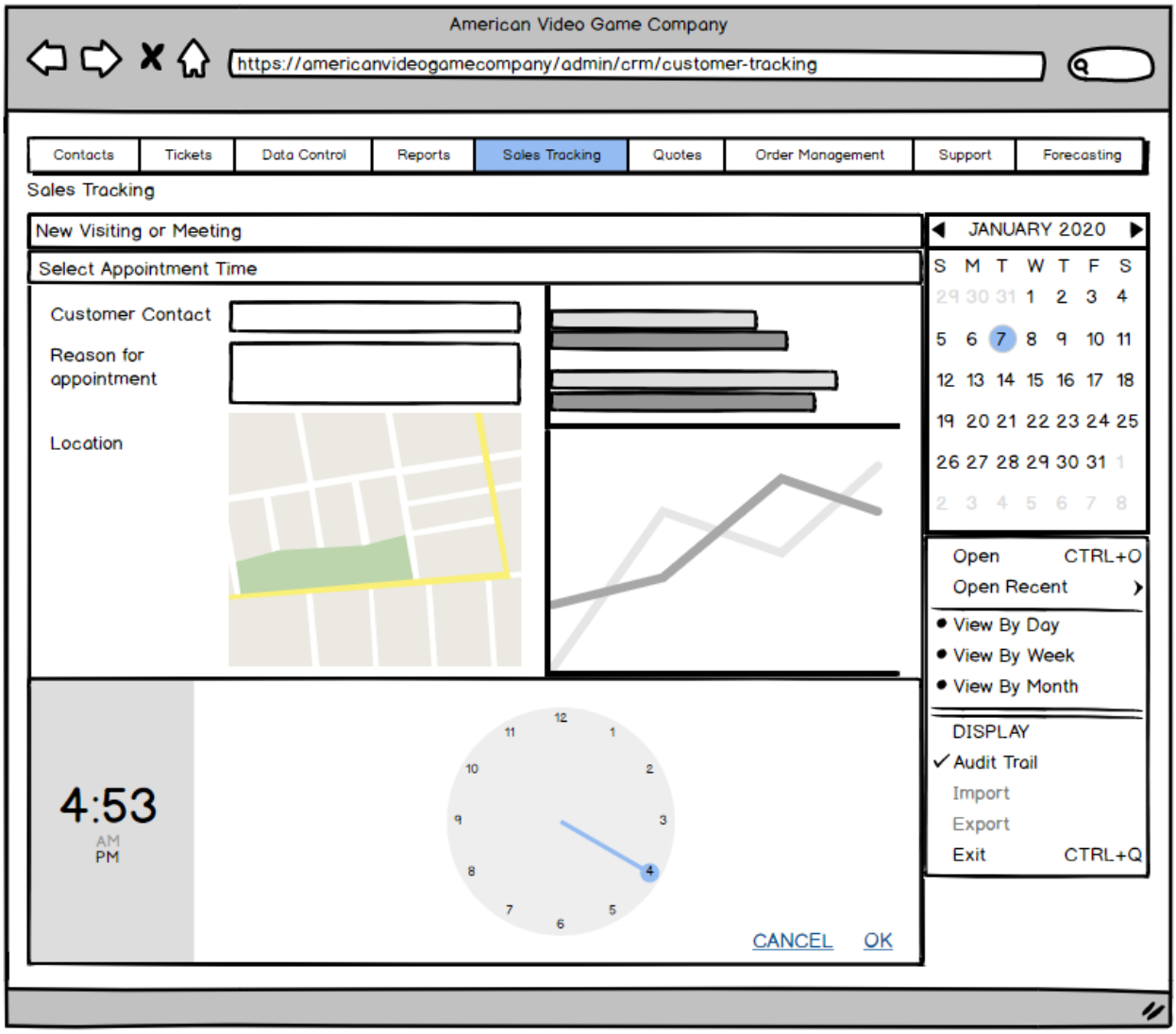
1.) Users must login to the application and select Sales Tracking from the available options at the top of the web browser. Once they are here, they will see the main menu for sales tracking. This menu allows the user to schedule a new visit or meeting by searching for a customer and selecting an appropriate date and time. Once a customer is selected, an audit log will appear will all notes from previous interactions with the customer. Below these menus is another tool that allows a user to lookup upcoming appointments for a customer. One can also select a format to view the appointments, display an audit log, or import the data to another application.



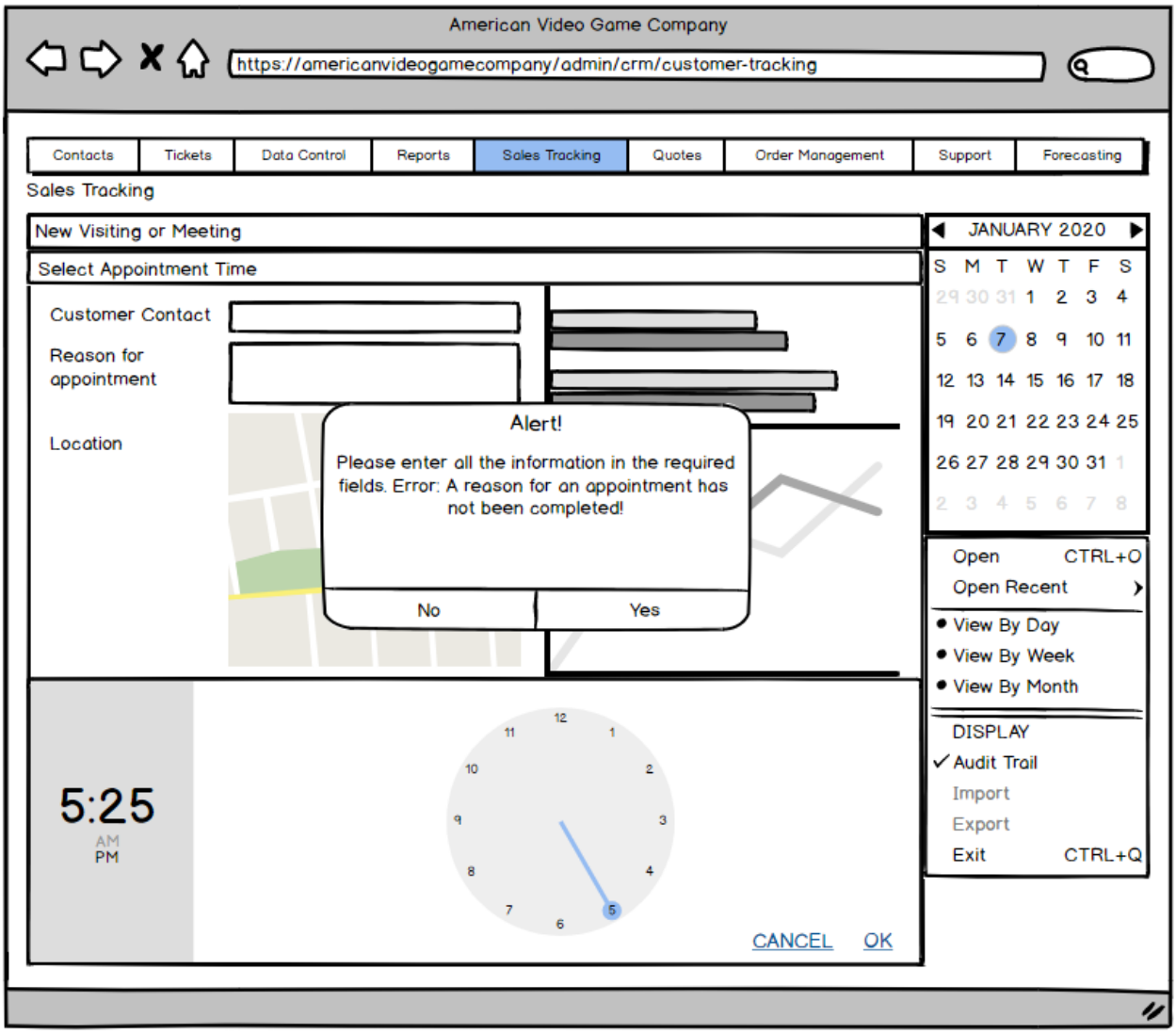
2.) The user searches a customer name and selects New Meeting. If a meeting or visit is already scheduled for that day, an alert will popup alerting the user that the customer already has a meeting. The user can choose to manually override this. If there are no appointments in place for the selected day than the user will be navigated to the next screen.



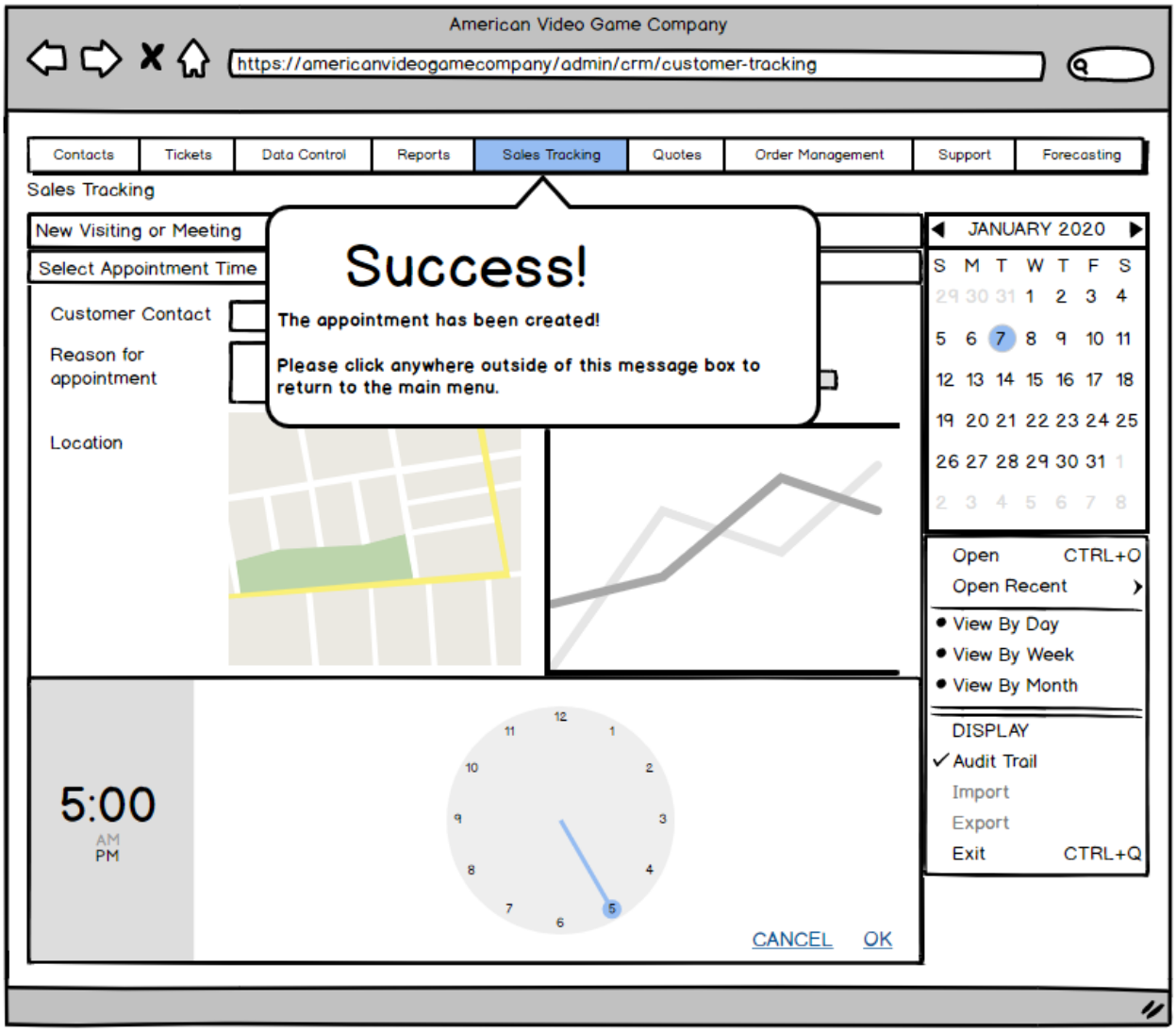
3.) The new meeting screen allows the user to input the following data for the selected customer: customer contact, reason for appointment, location of meeting, and start time of meeting. Additionally, the screen provides data on the customers current sales within the past 90 days.



4.) If the user does not fill out all the appointment screens correctly than an error message will appear alerting the user that they must fill in the missing required fields.



5.) Once the user fills out all the proper information than an overlay screen will appear alerting the user that the appointment has been successfully made. The user may click anywhere outside of this display message to return to the main menu.



B.3. FUNCTIONAL REQUIREMENTS

Below is a detailed outline of the functional requirements:

- Implement role and permission control based on access control in existing active directory system
  - Permissions must be controlled by an administrator
  - Extend access control to internal and remote users
  - Extend access to also allow restricted permissions to third party marketing companies

- Accurately control data flows throughout the system and implement checks to ensure that all data manipulation and access is done within the United States
    - Provide an application that restricts login access outside of the US to provide data security
    - Implement authorized employees to issue “soft” deletes on most data and only allow privileged users the ability to issue a “hard” delete
    - Correctly categorize data types and assign roles to all structured data. The data should be assigned one of the following data access roles: “Stakeholder”, “Business”, “Contact”
    - Control and track user data flow, data access, deletion of records, auditions, and user activity on types of data (this will be accomplished by using an audit log that is updated each time a user accesses the system with their UAC credentials)
- Ensure all functions of the application are testable and scalable
    - Provide a clear roadmap for future growth and additional features. Roadmap should be accessed through the web portal by users with manager level credentials
    - Thoroughly test and provide FAQ’s for all features within the application
    - Provide a 24/7 maintain team that can handle any tier 1 inquires and provide a tier 2 support team M-F
    - Support structure must be clear to all users and training modules will need to be created for different user levels. These modules can be further discussed in the design document
- Turn over a clearly identified licensing model that has defined ownership rights and a procedure path for custom development or change requests
- Provide a sales tracking system that provides activity management for customer visits and meetings
    - The Reach CRM should all communication protocols for accessing and retrieving information through MS Outlook and Exchange
    - A client ticket system will be provided to cover all interactions with client requests – this must also include an audit tracking system. Each ticket must be unique and provide contact details, current status of ticket, relevant details, and all communication with the client. Tickets that have not been responded to in over 3 days should be pushed as a notification to all team members that are actively involved with the project.
    - Provide an opportunity management tool that allows the sales team to track sale processes, manage pipelines, and do competitive analysis
    - Create a robust quoting system that follows state and federal regulations, stores an updated catalog, allows discounting structure, and electronic signature protection
    - Enhance the quoting system with an order management system that integrates existing systems to allows users to convert quotes to orders. The system should also include order tracking, reordering, part ordering systems.

- Include a program for forecasting that will assist in predicting upcoming sales and revenue based on empirical data collected. The Reach CRM will provide such functionality by incorporating data that is created by the other sales tracking reports. Additionally, the forecasting tool must be able to issue current adjustments (for domestic and foreign currencies), provide a baseline snapshot of forecasts for comparison, and be able to establish forecasting periods
- The forecasting application must also include managerial support for adjustments, product forecasting, upsides, and machine forecasting.
- Lastly, provides a contract tool that creates contracts and allows signing. The tool should also provide term tracking, termination tools, and approval process flows. Approval process should be only available to users with manager level or higher permissions.

- The hosting environment for the cloud-based CRM application will be setup on our internal servers. This will ensure we can monitor data security and provide increased resources as needed.

B.4. NONFUNCTIONAL REQUIREMENTS

The CRM System must perform as well or better than all existing tools used by the American Video Game Company. It must integrate with existing systems to share data and be able to include future programs. It must also be flexible enough to have a future development plan that maintains compliance with relevant use laws, regulation, and best practices. This integration should be based on the data provided from the existing active directory server and any be able to connect with external or client applications. It must also have a clear licensing model that has defined ownership rights and allow a system for custom development. Additionally, the application must stay within our hosting environment to ensure security and proper data backup. These standards will also follow our principle of no down time. If an application is not operational, we will try our best to get it live within 24 hours of the incidence report. Lastly, the application must be user friendly and be easy to adopt.

C. SOFTWARE DEVELOPMENT METHODOLOGY

The company has selected the waterfall software development methodology for this project. An outline of the waterfall method is provided below for users who may be unfamiliar with the methodology.

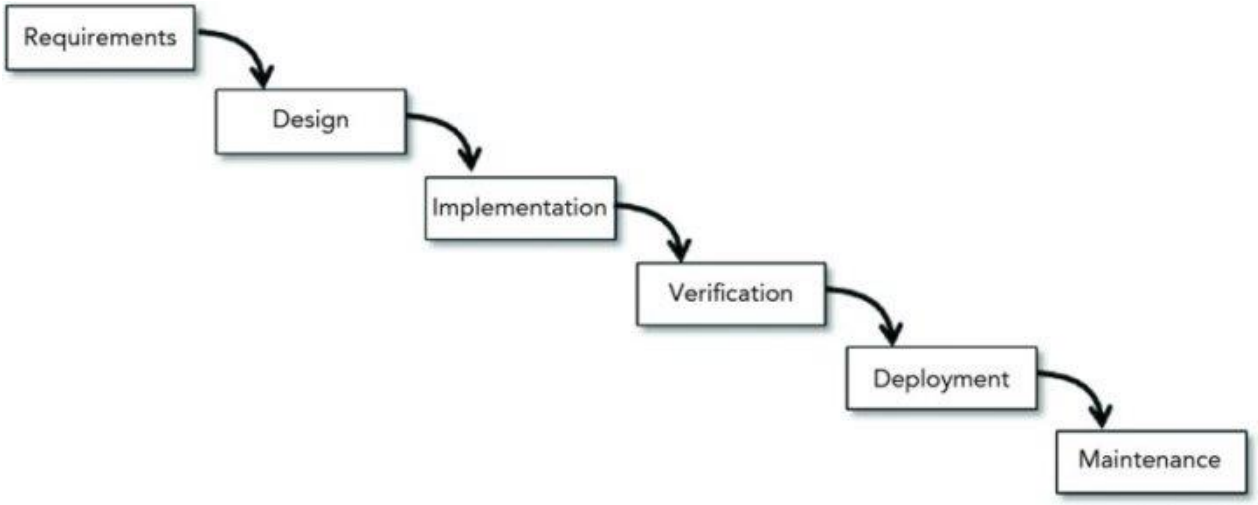


Figure 1: Example of Waterfall Method

According to the cited text (see section F, Sources), "*Waterfall* is the plain vanilla of the predictive model world. It assumes that you finish each step completely and thoroughly before you move on to the next step." The Waterfall method defines a clear path that is based on an iterative process of "stepping" through each process. The methodology outlines seven steps that can be further subdivided and are used in most variants of the Waterfall approach. They are as follows: Requirements, Design, Implementation, Verification, Deployment, and Maintenance. The model places a high importance on a predictable schedule and employing a team that has previous project experience. Additionally, Waterfall stresses that all requires are known in advance and there are no unresolved high-risk items. This approach does not allow for overlap and each stage acts like a bucket, it must overflow into the next stage. Unlike the traditional waterfall method, the Sashimi Waterfall approach allows for overlap in the process flows and reduces the penalty of having to go backwards in the process flow. Both methods are outline in further detail below.

C.1. ADVANTAGES OF THE WATERFALL METHOD

The advantages of the Waterfall method are outlined above but we will provide more detail here. The methodology allows you to step through each process completely and thoroughly before moving on. This allows for a clear approach that is predictable in nature and easy to track. All of the requires are outlined in advanced and makes the requirements document one of the most important parts of the entire project lifecycle. This document also attempts to eliminate all high-level risks to ensure that there are no unnecessary requirement changes. Lastly, the Waterfall approach encourages using a team that has previous experience with similar projects and allowing enough time to develop everything in a sequential fashion.

C.2. DISADVANTAGES OF THE WATERFALL METHOD

The Waterfall method has been around for quite some time and is not used in practice much these days. While the model does have the advantage of being predictable and well-studied, it does have some clear disadvantages. First, it is not flexible and restricts a team to using a heavy-prediction

framework. It means that the method may not be the best approach when working on projects that are being developed for the first time. Additionally, the Waterfall method does not provide a good process for dealing with unexpected risks. Instead, its focuses on preventing all risks and not allowing the project to move forward unless these risks are identified. The problem with this approach is that you cannot always predict every associated risk. In fact, there may be non-associated risks that can stall the project and push back well thought out development cycles. Lastly, the requirements are mostly unable to change, and this property becomes strong as you get later in the development process. This can be a big concern if the team is face with an unexpected hurdle.

C.3. ADVANTAGES OF THE SASHIMI WATERFALL METHOD

Next, we will go over an alternative of the traditional Waterfall method, the Sashimi approach. This approach is also called the Waterfall with overlapping phases method because it allows overlap. The model still follows elements of the traditional Waterfall approach and uses the seven incremental steps, but it allows to move between stages before completing all necessary requirements (see figure below for more detail).

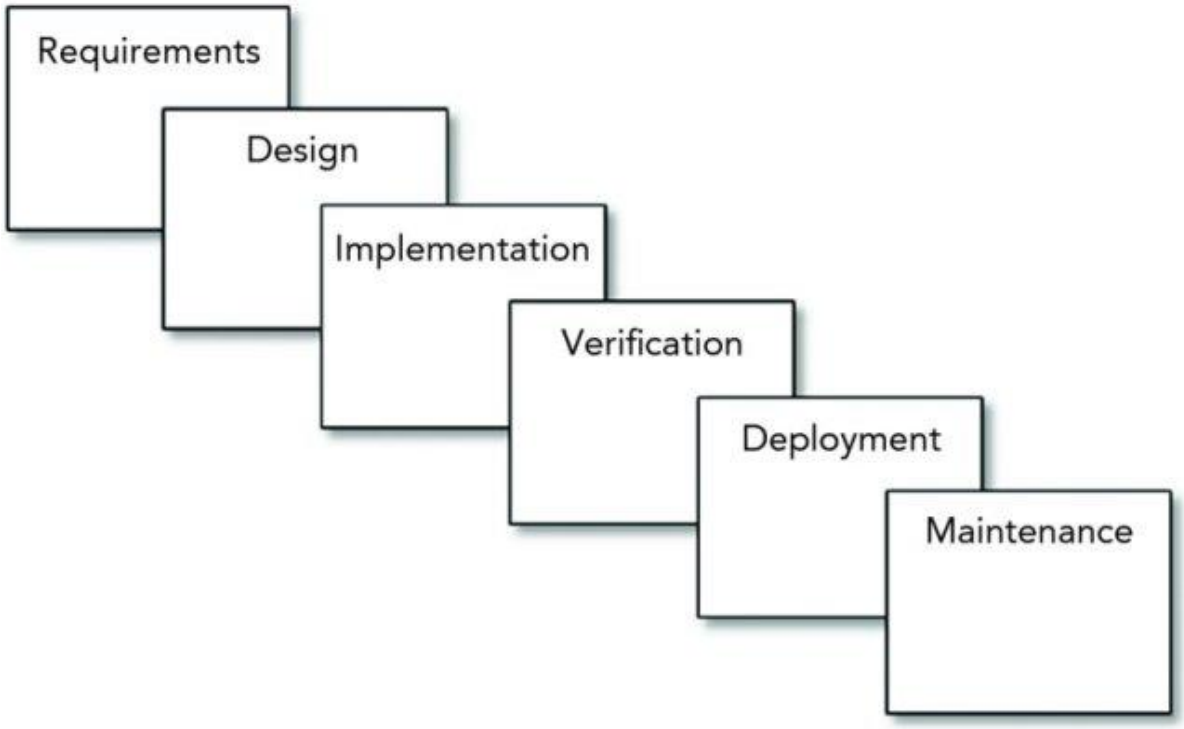


Figure 2: Example of Sashimi Waterfall Method

The model may not look all that different, however this property of overlap can address a lot of the problems discussed in section C.2. Firstly, the approach is more flexible because of the overlap. It is easier to move between phases inherently so the penalty for going backwards is less severe overall. This can also extend to dealing with unexpected risks. For example, the requirements phase may still be in process when the design phase starts. If the designers discover something critical missing from their work, they can tell the requirements team and it can be added in. One of the most

important advantages is our ability to perform a "deep dive" on a topic. If we are unsure about how a feature of the Reach CRM will integrate with the American Video Game Company's internal structure, we can test it. Creating prototypes or concepts of purpose will allow the users to test the integration and provide us with feedback to adapt the feature as needed. Lastly, these two properties allow the model to implement changes more freely. Because of these advantages we believe the model is a much stronger fit for the CRM deployment.

C.4. DISADVANTAGES OF THE SASHIMI WATERFALL METHOD

While we recommend the Sashimi method, we should also be clear on some disadvantages that may be present with this approach. Firstly, there is a limit to how much overlap we allow. While the process of designing with overlap can see as an advantage overall, it should not be used as a crutch. The Sashimi method is still a variant of the traditional Waterfall approach and must therefore follow the rules of predictability and be somewhat inflexible. This is also true of our "deep dive" abilities. We cannot afford to take an in-depth look with every feature based on customer request. Therefore, we reserve the right to implement this change control process. Lastly, the process must still have elements of an incremental approach. While we want to encourage overlap in the development phases, it is important that we do not allow one process team to get too far ahead. This is important to prevent excess changes if a major change is approved.

C.5. BEST SUITED

Because of the large scope of the project and the need to having a working application we quickly, we believe the Sashimi Waterfall method will be best. It suits the needs of the project because it allows us to use our experience to build a predictively model and provide prototypes. We want to ensure that we can deploy a CRM system that meets all user requirements and is easy to operate. Having an overlapping approach will allow us to use the full resource power our team can offer as well. Additionally, we will present a working application in a reduce timeline. If any additional changes need to be made, we can go back and polish the application within the same timeline as a sequential approach.

D. DESIGN

The high-level design goals of the Reach CRM should be described as a clean interface that is easy to use across any of the supported web interfaces. It should also contain a navigation bar at the top of every screen to allow quick navigation to all the primary features. Another great feature of the navigation bar is that it will be adjusted to include/exclude menu's depending on the level of user access. There will be an alert system that monitors user interaction and displays an error message if a user commits an illegal operation. Below we explore a flowchart, UML, diagram, and login screen GUI to get a better feel for the design across the entire application.

D.1. FLOWCHART THAT DESCRIBES PROCESS TO CREATE NEW BUSINESS RECORD

For the application to scale with future business growth, users must have a way to add and update the database. One such process involves creating a new business account record. The user will access the Reach CRM System than navigate to the Contacts menu. From there they can use the flow chart to add a new business and associated primary contact. Note: If a user tries to add a business that already exists, they will receive an error alert providing them with the business

account number for the existing record. All accounts are search either from a manual name search or by using the associated business account number.

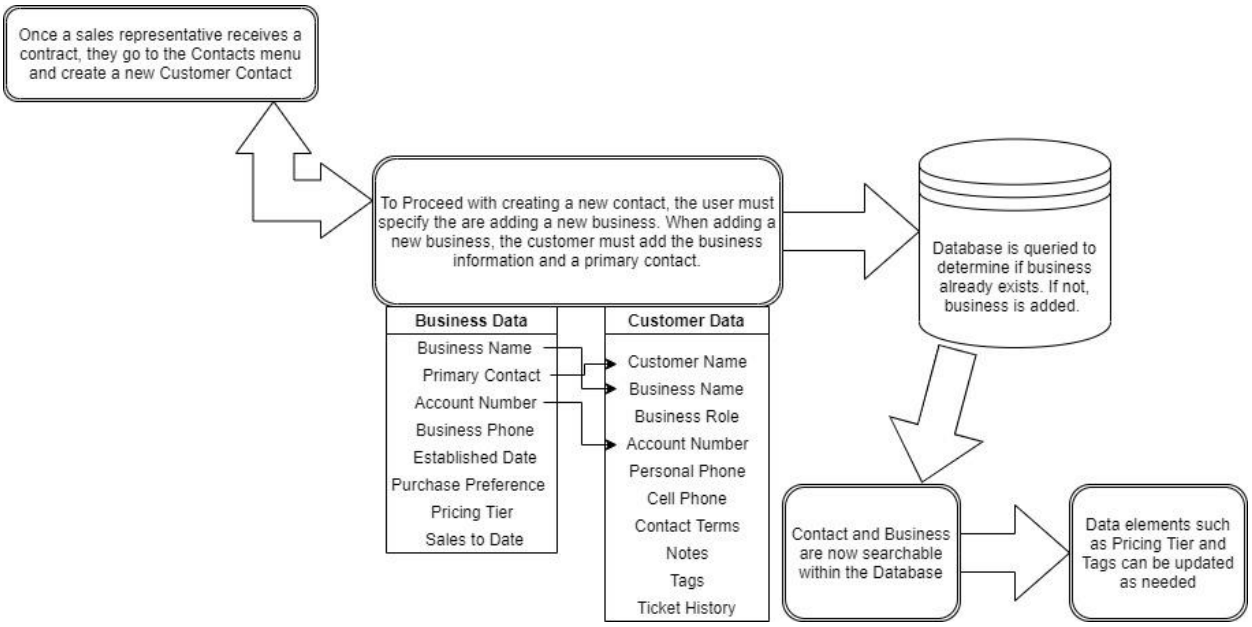


Figure 3: Sample Flowchart

D.2. GUI OF USER LOGIN SCREEN

Provided below is the login screen as viewed from a desktop browser. Anytime a user goes to the website where the CRM System is hosted, they are prompted with this screen. If invalid credentials are provided, an alert window will appear letting them know they have entered incorrect information. Below the GUI Mock-up is a diagram that further explains the low-level design.

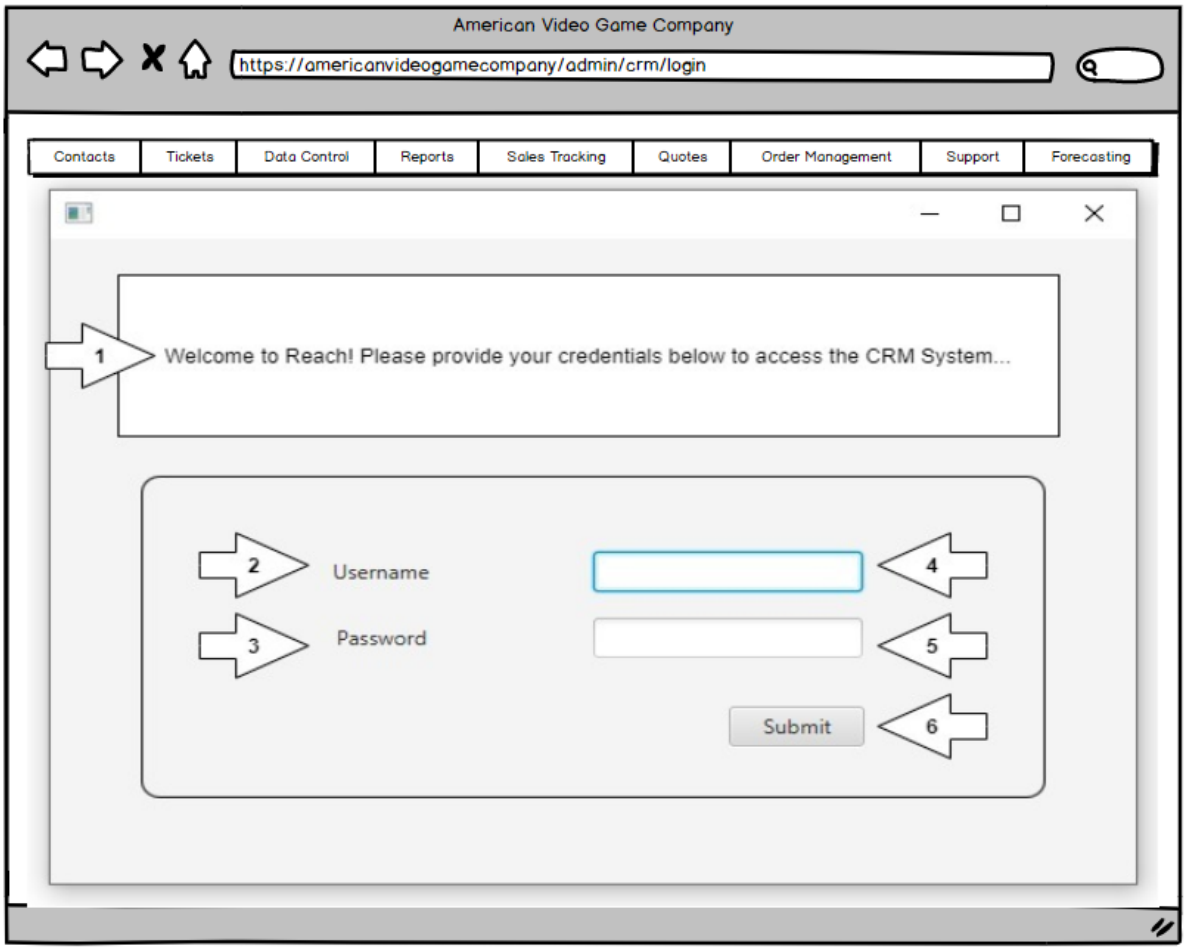


Figure 4: Sample GUI Mock-up

GUI Control Mapping			
ID	Control	Property	Data Source
1	Form	Provides a welcome message to the user and prompts them to login	NA
2	Text	Displays “Username”	NA
3	Text	Displays “Password”	NA
4	Textbox	The user provides their username here	Internal Variable
5	Textbox	The user provides their password here	Internal Variable
6	Button	Prompts the database search when clicked to confirm user credentials.	Database Search

E. TESTING

Because we are basing the Reach CRM on existing software we have developed, many non-functional aspects have already been thoroughly tested. For this reason, the application testing will focus primarily on the functional properties of the CRM. This includes report generation, tag management, contact

management, user access control, sales tools test, and much more. Below we outline 3 different tests that ensure critical requirements of the initial deployment are thoroughly tested and working as intended. Our goal in testing these specific scenarios is to demonstrate program functions that would be frequently used and require a large amount of resources to execute.

E.1. TESTING TYPE: BLACKBOX

We have chosen to execute these tests under a Blackbox method for two reasons. Frist, we want the key users for each function in the organization to use the system as normal. This will help us determine how much processing power is needed for each super user and will help us ultimately determine the server resources that need to be initially allocated to the Reach CRM System. Secondly, this process allows us to remove any bias from the tester because we are depending on user reported results. If the user did not receive the results, they were expecting than we can further investigate to determine what went wrong.

E.1.1. TEST 1: SUCCESSFUL CONTACT INTEGRATION

Requirement to be tested For our first test we can going to take a batch of customer data that is existing within the American Video Game company's existing database and import it. Once the data has been imported, we will ask the users to search for all the imported account numbers and determine if any data discrepancies exist. Estimated time to complete is two days.
Preconditions: Conditions that must be present before test case can successfully run The contact(s) must be present in the existing database and not contain any data violations. We must use a minimum of 250 customer contacts in the batch.
Steps: The steps the tester must execute to test the feature. 1. Once the super users give us the green light, we will push the data from the database into the new system. 2. These users will login into the Reach CRM using their provided credentials 3. They will then be provided a list of account names that were ported over. Each user will be assigned 50 names to lookup in the new system. The users will also be assigned a test sheet to write down notes for each assigned account 4. The users will Navigate to the Contact's Main Menu by clicking on the "Contacts" button located on the far left-hand side of the screen 5. From the main menu, users can go to the search bar for press alt + F4 to perform a quick search. They will type the account number into the search bar and press enter 6. This will populate the customer account and all associated data as outlined in Section D.1. 7. If the information is correct than the user will mark a C next to the client's name on their test sheet. If data is missing or incorrect, they will provide additional details in the notes section and mark the customer as NC.

Expected results: Expected results and any side effects such as updating a database, writing to a file, etc. We are expecting that the super users will produce a total result of over 95% C's on the first iteration. We will allow for a small tolerance gap because not all user data may be correct to begin with. If the results are not as expected than we will examine each case and determine what is incorrect. Our developers will try to determine any patterns in the incorrect data exchange and make necessary changes. Once this is complete than we will run the test again.
Pass/Fail: Mark whether the test case passed or failed. The results can be compiled and used to determine if the application is ready for delivery/release. Pass: The system passed on the first integration with only 2 of the 250 accounts containing errors. To ensure correctness, we brought in a second group of super users and had them preform the same tests. They confirmed our results with 2 errors. Those errors were investigated, and it was determined that the dataset contained the error. These entries were corrected, and the test results were sent for approval.

E.1.2. TEST 2: ADDING & REMOVING UAC ON AN EMPLOYEE ACCOUNT

Requirement to be tested Our second test will involve having an admin import the existing user accounts from their Active Directory System. Next, we will create 3 user accounts and change their privileges. Finally, these created user accounts will than be deleted. Once they are deleted, we must confirm their absence from the user directory and ensure that the logs display the delete action. Estimated time to complete is two-three weeks.
Preconditions: Conditions that must be present before test case can successfully run We must have admin level access to the existing active directory trees. For this specific test we can be restricted to only have access to user accounts Company admins must be available for testing during the entire testing period The log tracking system must be actively working and tested

Steps: The steps the tester must execute to test the feature.

1. We will begin my porting over all the user accounts from the existing active directory domain
2. The system admins at the American Video Game Company will than begin going through the user accounts and confirming that all active accounts exist and have the proper permissions. This can be down quickly due to our "sort by role" feature
3. Next, we will add 3 user accounts with the lowest level of permissions. We will than incrementally elevate their user privileges and try to access more resistive file data.
4. After the privilege level of these user accounts reaches that of an admin, we will test downranking their privilege levels.
5. Once each level of user privileges has been thoroughly tested, we delete all 3 of the fictious user accounts.
6. Lastly, we will check the audit logs to determine these changes were successfully tracked

Expected results: Expected results and any side effects such as updating a database, writing to a file, etc.

We expect to have all the users imported properly. We also expect that the user escalation and demotion will reflect the proper privileges. Additionally, we expect that the audit log will work as intended and display these changes. We have strong confidence in these functions because the software is based on an existing system we have developed and tested.

Pass/Fail: Mark whether the test case passed or failed. The results can be compiled and used to determine if the application is ready for delivery/release.

Pass: The transition of user accounts was completed successfully. Each active user was found and confirmed that they had the proper permission level in place. Our users confirmed that each level of user access was working as intended. After deleting the users, we also determined that the audit log was working correctly.

E.1.3. TEST 3: SENDING OUT EMAILS THROUGH OUTLOOK BASED ON TAGS

Our last email will be demonstrating a common task that the sales team will use. It involves sending out mass promotional emails through outlook based on a customer's tags. The sales team will receive new promotional media from the marketing team. They will than send out a mass email with the material to all receipts that fit the associated tag. It is critical that the proper clients receive these emails as we do not want another client having access to pricing or information that is not related to their business plan.

Preconditions: Conditions that must be present before test case can successfully run

The first test must be completed so that we can use 50 of the 250 imported client contacts

These clients must be notified that they will receive a series of test emails and are asked to respond.

Steps: The steps the tester must execute to test the feature. 1. We will begin my creating a test named “test” which will be applied to 50 accounts that have previously been communicated with 2. The selected set of super users will than access their outlook mail program and compose a new email. From here they will type in the following in the “To” line: Reach:pull:where(tag)=’test’ 3. The email application will than change the receipt addresses to any primary customer contact that is associated with the tagged account 4. The super users will than reach out to these contacts to confirm receipt of the test email
Expected results: Expected results and any side effects such as updating a database, writing to a file, etc. We expect this will work because it is also based on previous development. If there any issues with customers receiving the emails than we must first check to ensure the information is correct in the database. If all information is correct and an error still ours, then our development team will need to investigate.
Pass/Fail: Mark whether the test case passed or failed. The results can be compiled and used to determine if the application is ready for delivery/release. Pass: We confirmed that the tag was added correctly to all 50 user accounts and that they received the emails. These tests will continual incrementally until all user tags are updated and current. We recommend assigning an employee with the role of tag management to ensure all tags remain correct with future operations.

F. SOURCES

Figure 1: Image of Example of Waterfall Method from the provided course materials, Beginning Software Engineering, Ch. 12, Section 4. Online material accessed on January 8, 2020

https://wgu.ucertify.com/?func=ebook&chapter_no=14#02TfW

Figure 2: Image of Example of Sashimi Waterfall Method from the provided course materials, Beginning Software Engineering, Ch. 12, Section 6. Online material accessed on January 8, 2020

https://wgu.ucertify.com/?func=ebook&chapter_no=14#02TfX