**Bright Idea Team Software**

|  |
| --- |
| Prepared for American Video Game Company |
| CRM Proposal |
| C188 – Software Engineering Performance Assessment |

|  |
| --- |
| Steven Bennett 003761827  11-21-2023  [Version 1.0] |

Contents

[A. Introduction 3](#_Toc150176002)

[A.1. Purpose 3](#_Toc150176003)

[A.2. Overview of the Problem 3](#_Toc150176004)

[A.3. Goals and Objectives 3](#_Toc150176005)

[A.4. Prerequisites 3](#_Toc150176006)

[A.5. Scope 3](#_Toc150176007)

[A.6. Environment 3](#_Toc150176008)

[B. Requirements 4](#_Toc150176009)

[B.1. Business Requirements 4](#_Toc150176010)

[B.2. User Requirements 4](#_Toc150176011)

[B.3. Functional Requirements 4](#_Toc150176012)

[C. Software Development Methodology 6](#_Toc150176013)

[C.1. Advantages of the Waterfall Method 6](#_Toc150176014)

[C.2. Disadvantages of the Waterfall Method 6](#_Toc150176015)

[C.3. Advantages of Agile 6](#_Toc150176016)

[C.4. Disadvantages of Agile 6](#_Toc150176017)

[C.5. Best Suited 6](#_Toc150176018)

[D. Design 8](#_Toc150176019)

[D.1. Storyboard or Flowchart (Change title to fit needs) 8](#_Toc150176020)

[D.2. UML Diagram (Change title to fit needs) 9](#_Toc150176021)

[D.3. GUI (Change title to fit needs) 10](#_Toc150176022)

[E. Testing 11](#_Toc150176023)

[E.1. Testing Type (change name to fit your needs) 11](#_Toc150176024)

[E.1.1. Test Name 1 11](#_Toc150176025)

[E.1.2. Test Name 2 12](#_Toc150176026)

[E.1.3. Test Name 3 13](#_Toc150176027)

[F. Sources 14](#_Toc150176028)

# Introduction

Provide a brief introduction to the proposed system. This section should be no longer than one paragraph.

# A.1. Purpose

Provide a brief overview of the purpose of this document.

# A.2. Overview of the Problem

Provide a brief overview of the problem that the proposed solution will solve.

# A.3. Goals and Objectives

Provide the goals and objectives for the project and solution.

# A.4. Prerequisites

Outline any aspects that need to be in place prior to the design, development, and implementation of the project proposed in this document. Be sure to be clear and concise for all listed prerequisites. Also, clearly outline why each prerequisite is needed.

*Note: If no prerequisites are needed, include a paragraph justifying why there are no prerequisites.*

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Prerequisite | Description | Completion Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# A.5. Scope

Provide a brief overview of what the proposed solution will cover and what the proposed solution will not cover. It is important to set clear boundaries for the project.

# A.6. Environment

Describe the IT and hardware environments that the solution will be deployed in.

# Requirements

Given the current parameters defined by AVGC, our proposal CRM system will meet the following 5 requirements listed in the CRM\_Requirements document:

1. Scalability to meet current and future needs.

2. Contacts, businesses, and stakeholders will each have their own datatype.

3. Soft and hard delete capabilities.

4. Individual user activity recording.

5. Up to date OS and Browser Support.

# Business Requirements

With 2000 total users accessing the system with 500 accessing during peak times, current systems are being outgrown as growth continues. Current numbers show a 42% increase over the past two years. With such rapid growth, scalability is a priority when developing a CRM system. AVGC needs a system to not only accommodate their current number of users, but also has the ability to scale and accommodate future needs as well. BIT Software proposes to solve scalability concerns by expanding in-house hosting capabilities with latest generation hardware capable of accommodating 12000 total users and 3000 during peak times upon launch with expansion capabilities as business demands. Hosting in-house results in no service provider connectivity issues, SLAs, or mandatory upgrades that are common with cloud solutions. The proposed hardware will handle current and future volumes based on projected growth for more than 5 years before expansion is required.

Recording and logging individual user activity provides incredibly functional tools that are valuable assets during auditing efforts. User activity such as logins, orders, sales, customer account creation and/or modification, will be logged and timestamped to a txt file which will only be accessible by privileged users.

# User Requirements

To increase portability and decrease the need for new or upgraded hardware and software, the BITS solution will offer OS support for Windows 10+ and Mac OS Big Sur+ as well as mobile support for iOS 11+ and Android 8.0+. Multiple browsers including mobile and tablet versions will also be supported including Safari, Microsoft Edge, Google Chrome, Mozilla Firefox, and Chromium.

# Functional Requirements

The BITS solution leverages the power of object-oriented programming to efficiently manage and organize AVGC provided data. By implementing OOP, we will create a robust and flexible system that will house AVGC data in three distinct data types: Stakeholders, Businesses, and Contacts. Each respective data type will be represented as objects with their own AVGC predefined attributes and methods. This ensures a structured and scalable approach to data management while enabling BITS to model each datatype after the real-world entities involved with AVGC. This approach allows for seamless data retrieval, manipulation, and reporting, improving overall efficiency and functionality of their software solution.

Also incorporated into the BITS solution are methods to perform “hard” and “soft” deletions based on user privilege settings. Unprivileged users will be able to perform soft deletions of the Stakeholders, Businesses, and Contacts datatypes. Performing a soft delete changes the status to Archived which filters that entry from being displayed or considered for transactions. A soft deleted entry remains in the database accessible only by a privileged user. Privileged access to the database provides hard delete and restore capabilities to each entry. Hard deleting an entry removes that entry from the database completely. The restore function changes the status of a soft deleted entry from Archived to Active, returning the entry to full functionality.

# Software Development Methodology

For development of this solution, the agile and waterfall software development methods were considered. Each method is advantageous in its own way, but ultimately the waterfall method was selected as the best fit. Advantage and disadvantages are discussed in further detail followed by selection justification in subsections C.1 through C.4 below (Hoory & Bottorff, 2022):

# Advantages of the Waterfall Method

Waterfall advantages:

1. Provides a concrete plan of the project from start to finish
2. Project requirements are defined and agreed upon early on, which can save time
3. Each phase of the project requires a deliverable to progress to the next phase, making the workflow more structured.

# Disadvantages of the Waterfall Method

Waterfall disadvantages:

1. Because each project phase needs to be completed before progressing to the next stage, the process can take longer.
2. Stakeholders may not see the product until the end and have little to no involvement after completion of the early stages. As a result, stakeholders are not able to offer feedback until verification. Any late changes or revisions would be a very costly and timely endeavor.
3. Waterfall methodology focuses on being proactive in risk identification and management but is less adaptable in dealing with risk occurrences outside of those predictions. Dealing with an unplanned event or mistake could result in a delay until completion and added expenses.

# Advantages of Agile

Agile advantages:

1. Agile adds flexibility and allows changes throughout development. Approved change requests are incorporated into the following iteration.
2. Increased stakeholder involvement results in increased feedback incorporated into the process, which can lead to more refined deliverables.
3. Short term deadlines encourage productivity and efficiency.

# Disadvantages of Agile

Agile Disadvantages:

1. Agile lacks a linear view of progression. Deliverables are not required to progress to the next phase, often muddying the measure of progress.
2. Increased stakeholder involvement can result in increased change requests, which can prolong the project timeline and increase costs.
3. Agile team members work on multiple phases at a time, creating a potential for overlap or unnecessary effort spent on later stages if an early phase needs to be modified.

# Best Suited

The BITS solution incorporates the Waterfall method (Figure 1.1) for several reasons:

1. Clearly defined phases with quantifiable progress promote accountability. Progression to the next phase is only possible after completion of the current phase.
2. Well defined deliverables with a thoroughly documented end goal will help stakeholders maintain cohesive efforts and maximize efficiency reaching the end goal.
3. Concrete deadlines are established and agreed upon early. This minimizes the chances of scope creep and provides a clear end date for project completion.

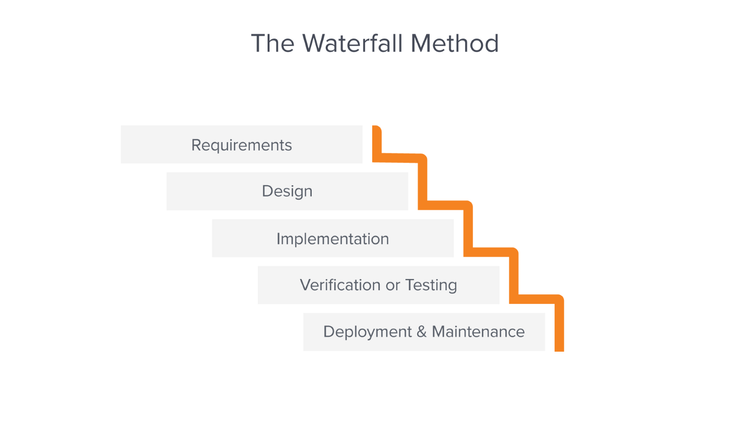


Figure 1.1: Waterfall methodology — a complete guide (Adobe Communications Team, 2022)

# Design

Provide a brief overview of the proposed design.

*Note: These subsections may be copied, rearranged, and modified to fit the needs of the solution. At least two visual representations of your design need to be present.*

# Storyboard or Flowchart (Change title to fit needs)

Provide a storyboard or flowchart of the application.

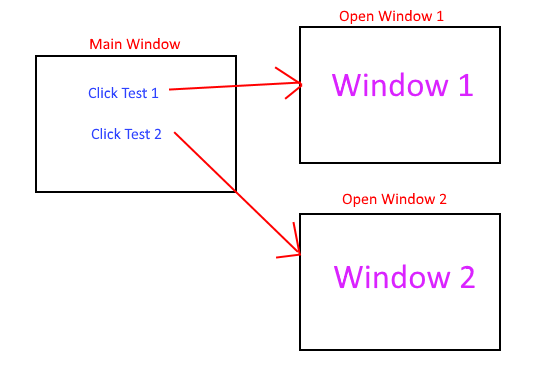


Figure 2: Sample Storyboard

# UML Diagram (Change title to fit needs)

Provide a set of UML diagrams that cover the proposed solution. This can include but is not limited to class diagrams, database diagrams, and use case diagrams. Also, ensure that all diagrams are clearly discussed and noted.

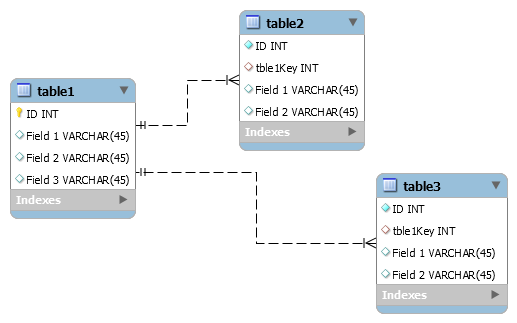


Figure 3: Sample Database

# GUI (Change title to fit needs)

Provide a mock-up of the proposed GUI forms that will be used in the proposed solution. Also, clearly indicate where the GUI components point inside the application.

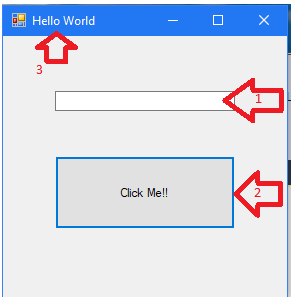


Figure 4: Sample GUI Mock-up

|  |  |  |  |
| --- | --- | --- | --- |
| GUI Control Mapping | | | |
| ID | Control | Property | Data Source |
| 1 | Textbox | On application open text = “” or null | NA |
| 1 | Textbox | On click of button text = “Hello World” | Internal Variable |
| 2 | Button | On click change text of textbox 1 to “Hello World” | Internal Variable |
| 3 | Form | Text= “Hello World” |  |

# Testing

Provide a brief introduction to the proposed testing solution. The tests need to be from 3 completely different functionality aspects. Testing the same aspect with slightly different criteria is not acceptable.

\*\*Note: *Add and remove subsections as needed to cover all the testing needs.*

# Testing Type (change name to fit your needs)

Provide a brief introduction paragraph.

# Test Name 1

|  |
| --- |
| Requirement to be tested |
| Preconditions: Conditions that must be present before test case can successfully run |
| Steps: The steps the tester must execute to test the feature. |
| Expected results: Expected results and any side effects such as updating a database, writing to a file, etc. |
| 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. |

# Test Name 2

|  |
| --- |
| Requirement to be tested |
| Preconditions: Conditions that must be present before test case can successfully run |
| Steps: The steps the tester must execute to test the feature. |
| Expected results: Expected results and any side effects such as updating a database, writing to a file, etc. |
| 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. |

# Test Name 3

|  |
| --- |
| Requirement to be tested |
| Preconditions: Conditions that must be present before test case can successfully run |
| Steps: The steps the tester must execute to test the feature. |
| Expected results: Expected results and any side effects such as updating a database, writing to a file, etc. |
| 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. |

# Sources

Place the sources that you used here.

*Note: See the sources section in the requirements and rubric. If you did not use any outside sources, you may delete this section.*

Hoory, L., & Bottorff, C. (2022, August 10). Agile vs. Waterfall Methodology. Forbes. <https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/>

Adobe Communications Team. (2022, March 18). Waterfall methodology — a complete guide. Adobe. [https://business.adobe.com/blog/basics/waterfall#:~:text=The%20waterfall%20methodology%20%E2%80%94%20also%20known,before%20the%20next%20phase%20begins](https://business.adobe.com/blog/basics/waterfall%23:~:text=The%20waterfall%20methodology%20%E2%80%94%20also%20known,before%20the%20next%20phase%20begins)