**Design Template**

|  |
| --- |
| [James’ Software Solutions] |
| Software Project Template |
| [Project Optimize] |

|  |
| --- |
| James Blankenship  2-9-2022  [Version 1.1] |

Contents

[A. Introduction 3](#_Toc94271609)

[A.1. Purpose Statement 3](#_Toc94271610)

[A.2. Overview of the problem 3](#_Toc94271611)

[A.3. Goals and Objectives 3](#_Toc94271612)

[A.4. Prerequisites 3](#_Toc94271613)

[A.5. Scope 3](#_Toc94271614)

[A.6. Environment 3](#_Toc94271615)

[B. Requirements 4](#_Toc94271616)

[B.1. User Requirements 4](#_Toc94271617)

[B.2. Functional Requirements 4](#_Toc94271618)

[C. Software Development Methodology 5](#_Toc94271619)

[C.1. Advantages of the Waterfall method 5](#_Toc94271620)

[C.2. Disadvantages of the Waterfall method 5](#_Toc94271621)

[C.3. Advantages of the Agile method 5](#_Toc94271622)

[C.4. Disadvantages of the Agile method 5](#_Toc94271623)

[C.5. Best Suited 5](#_Toc94271624)

[D. Design 6](#_Toc94271625)

[D.1. Storyboard 6](#_Toc94271626)

[D.2. UML Class Diagrams 7](#_Toc94271627)

[E. Testing 8](#_Toc94271628)

[E.1. Testing Types 8](#_Toc94271629)

[E.1.1. Stress test 8](#_Toc94271630)

[E.1.2. White-Box Testing 9](#_Toc94271631)

[E.1.3. Black-Box Testing 10](#_Toc94271632)

# Introduction

Our company, James Software Solutions, is proposing this solution to AVG, and you can find a detailed overview of the requirements, methodologies, design, and testing in the following sections.

# A.1. Purpose Statement

The purpose of this document is to define our requirements and design to better AVG’s system.

# A.2. Overview of the problem

AVG is unable to meet customer demand, and their outdated system leave them vulnerable to security breaches. If the continue to proceed with their current operations, they will end up losing customers and might have to pay for lawsuits for security issues. Our solutions strives to fix these issues, so they are no longer worried about.

# A.3. Goals and Objectives

Here are the goals and objectives of the new system:

- has robust security

- can be enhanced and scaled

- manages activities and tracks sales

- consolidates all contact and business information

# A.4. Prerequisites

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Prerequisite | Description | Completion Date |
| 1 | Collect Data | Collect the needed data from AVG’s company | 1 week |
| 2 | Learn System | Learn about what system AVG is currently using. | 1 week |

# A.5. Scope

The items that will be in scope in our proposed solution are:

* + The new system must be able to support 2,000 users, as well as 500 concurrent users at one time.
  + The new system must turn a quote into an order and complete a sale. It may also include features, like order tracking, customer self-service, and more.
  + The new system must be compatible with a list of operating systems and browsers, like Chrome, Firefox, and Internet Explorer, to name a few.
  + The new system must allow entry and tracking of communication and inquiry for contacts. It must maintain and audit trail and facilitate a workflow.
  + The new system must be capable of delivering both predefined and customer reports on all the data within the system.

Some items which will be out of scope in our proposed solution will be:

* + Hosting
  + Soft delete
  + Hard delete

# A.6. Environment

The hardware environments that the solution will be deployed in consist of:

* + Latest Chrome and Chromium
  + Latest Firefox
  + I.E 9 and above
  + Safari 6.0
  + Mobile and tablet
  + iOS7 safari
  + iOS7 chrome
  + iOS7 Firefox
  + Android 4.0 Chrome

The database used in the proposed solution will be a relational database and it will be accessed

using SQL statements.

# Requirements

The five requirements our new system will be meeting will be Order Tracking, Users, OS and Browser Support, Ticketing System, and Reporting.

# User Requirements

The new system must be able to support 2,000 users, as well as 500 concurrent users at one time. The system will accomplish this requirement by obtaining a system large enough to support 2500 users. This will allow the system to work more efficiently while at the same time delivering information to the users quicker.

# Functional Requirements

The new system must turn a quote into an order and complete a sale. It may also include features, like order tracking, customer self-service, and more. The system will accomplish this requirement by implementing a script that can do just that. It will have methods that can be called to track orders, take orders, turn a quote into an order, and more.

The new system must be compatible with a list of operating systems and browsers, like Chrome, Firefox, and Internet Explorer, to name a few. The system will accomplish this requirement by using plenty of user testing through the agile methodology to ensure the system works on all the required platforms. We will invite users to participate in user testing for rewards, encouraging lots of feedback that we can use to better the system.

The new system must allow entry and tracking of communication and inquiry for contacts. It must maintain and audit trail and facilitate a workflow. The system will accomplish this requirement by logging all the communication and inquiries together, creating an easy way for data entry and tracking. The system will have a script to allow users to navigate and search for the needed files easily.

The new system must be capable of delivering both predefined and customer reports on all the data within the system. It will have an interface to interact with the data. The system will accomplish this requirement by having a user interface which will allow users to export and query data. It will be able to access the entirety of the data in the system and the reports can be exported to multiple platforms (physical, digital, etc.).

# Software Development Methodology

The waterfall methodology is a big design up front method, meaning all the requirements are gathered first before moving onto the design. The design is then made, and the next sequential step starts when the current step has ended. You’re not able to go back to previous steps, therefore limiting the flexibility of the project.

# Advantages of the Waterfall method

The three top advantages of the waterfall methodology that I think will benefit this project are:

* It’s predictable
* Better cost-saving
* More stability

# Disadvantages of the Waterfall method

The three top disadvantages of the waterfall methodology that I think will hinder this project are:

* It’s inflexible
* Design is up front
* Release is prolonged

# Advantages of the Agile method

The three top advantages of the agile methodology that I think will benefit this project are:

* Continuous improvement
* A quality product
* Reduced risks

# Disadvantages of the Agile method

The three top disadvantages of the agile methodology that I think will hinder this project are:

* Limited documentation
* Hard to measure progress
* Scope creep

# Best Suited

I think the agile methodology is best for AVG because it will improve the overall product in iterations, using customer feedback to make the iterations value driven. The released product will better meet the users’ needs and raise user satisfaction of the project. This will reduce the risks that come with changing requirements in the waterfall methodology and will give users a chance to feel involved in the project.

# Design

I provided 2 diagrams in total, one that outlines the relations between the main window and the other windows, and one that uses UML to show the different classes, their functions, and data types needed to meet the requirements.

# Storyboard

The storyboard below is displaying a relation between the main window and the other windows that can be accessed through the main window. The storyboard supports the business solution because it displays how users will navigate through the product to perform needed tasks, like tracking and adding orders, or saving and printing a report.

Diagram

Description automatically generated

Figure 1: Sample Storyboard

# UML Class Diagrams

These class diagrams represent the classes, the classes data types, and the classes functions needed to fulfill the requirements. The diagrams support the business solution because it shows the developer(s) and team members of the product the classes needed to implement and meet the functional requirements.

Table

Description automatically generatedTable

Description automatically generated

Figure 2: Sample Class Diagrams

# Testing

# Testing Types

We at James’ Software Solutions, will be doing several different tests using multiple testing types. A stress test will be employed on the overall system, checking if it can meet requirements under stress. White-box testing will use a tester with knowledge of how the feature works so they can use their prior knowledge to influence their testing. A black-box testing method works just like a white-box one, expect the tester has no underlying knowledge of the feature so their testing isn’t influenced.

# Stress test

|  |
| --- |
| We will be stress testing our user’s requirement to validate that 500 concurrent users can use the system at once, and the systems performance doesn’t diminish. |
| There will need to be 500 users that will be ready to login at the same time. |
| 1. Allow 200 users to login to the system. 2. Check performance 3. Allow another 200 users to login to the system 4. Check performance 5. Allow the remaining 100 users login to the system 6. Check performance |
| All users will be able to login to the system at once, and the performance of the system hasn’t diminished. |
| Pass |

# White-Box Testing

|  |
| --- |
| We will be white-box testing our reporting requirement to ensure that our system will be capable of delivering predefined and custom reports on all the data in the system. |
| The developers must code the reporting capabilities in the system before it can be tested. This test must be done by a developer who worked on it, or someone who knows how the code functions. |
| 1. Boot up the system. 2. Login to the system. 3. Make and deliver a predefined report 4. Make a deliver a customer report 5. Log results into a test file. 6. Analyze results. |
| The system can deliver predefined and customer reports on all the data in the system. |
| Pass |

# Black-Box Testing

|  |
| --- |
| We will be black-box testing our system to ensure it supports different browsers and operating systems. |
| There will need to be a computer with all the different browsers downloaded, and a partitioned drive with the different operating systems installed. |
| 1. Boot up an operating system. 2. Login to the system. 3. Navigate to the browser. 4. Enter and test the website. 5. Turn off the system. 6. Repeat if needed with another operating system or browser. |
| The system works on all the required operating systems and browsers. |
| Pass |