**C868 – Software Capstone Project Summary**

**Task 2 – Section A**



|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | Java Scheduling Desktop Application |
| **Student Name:** | Brandon Rider |

Contents

[**Business Problem** 3](#_Toc90665500)

[**The Customer** 3](#_Toc90665501)

[**Business Case** 3](#_Toc90665502)

[**Fulfillment** 3](#_Toc90665503)

[**Existing Gaps** 3](#_Toc90665504)

[**SDLC Methodology** 4](#_Toc90665505)

[**Deliverables** 4](#_Toc90665506)

[**Project Deliverables** 4](#_Toc90665507)

[**Product Deliverables** 4](#_Toc90665508)

[**Implementation** 5](#_Toc90665509)

[**Validation and Verification** 5](#_Toc90665510)

[**Environments and Costs** 6](#_Toc90665511)

[**Programming Environment** 6](#_Toc90665512)

[**Environment Costs** 6](#_Toc90665513)

[**Human Resource Requirements** 6](#_Toc90665514)

[**Project Timeline** 7](#_Toc90665515)

# **Business Problem**

**The Customer**

The customer AFS is a global consulting organization with personnel and provided services that span time multiple zones and languages. Their operation has four main offices in Phoenix, Arizona; White Plains, New York; Montreal, Canada; and London, England. With a staff of over 1,000 personnel who provide management and technology consulting to their clients within the three countries. Their mission is to help their clients transform their business segments into future-proof innovators. Long-term goals include opening management consulting services in new markets such as mainland Europe and Central America. This project will align with that goal by allowing scalability and informational assurance with future added markets. Growth has been explosive in the technology consulting group with future growth projected to outpace current growth as organizations seek help in transitioning to the Cloud.

## **Business Case**

For AFS, all business is based on building great relationships with potential and current clients. The problem AFS faces are missing client appointments due to the amount and complexity of their appointments. The current manual scheduling process creates many scheduling conflicts and furthers this issue. This of course is detrimental to their business relationships and impacts future revenue by jeopardizing future contracts. The need for a standalone desktop application that accesses a central database as the sole source of scheduling information is vital for AFS expansion into new markets.

## **Fulfillment**

This software application will provide multiple functions that remedy current issues. The 3 core needs that will be fulfilled by this custom application. First, having a central repository for all data lowers the chances of scheduling conflicts due to a lack of updated data. Second, the application will translate data according to the logged-in users' language preferences which prevents errors. Lastly, the application is highly scalable allowing for future use as the client expands their business where the current process cannot scale easily.

The system will integrate with a current MySQL database that is used by other client systems. The system will track user logins including failed logins. It will also allow customer records to be added, updated, and deleted from the database. These customer records will be associated with contacts and appointments. The functionality creates a smooth process to manage client appointments.

# **Existing Gaps**

There is no existing standalone software application that meets the functional and business requirements of this client. Currently, all scheduling data must be processed and stored through an error-prone method that utilizes manual data entry into an excel spreadsheet. This method is outdated and creates huge labor costs in maintaining and correcting errors. This existing gap of non-customizable and non-scalable data processing is a bottleneck in the client’s services. The only existing piece of this proposed software system is a MySQL database that is used for other systems by the client. It will be utilized with this software and the new custom software will be integrated appropriately to nullify potential conflicts with other current software systems.

# **SDLC Methodology**

This project will use the Waterfall Software Development Life Cycle due to its linear project management approach. This methodology allows for stakeholder and client input at the beginning of the project. Their input is used throughout the rest of the project as the phases are completed.

I choose the Waterfall methodology because it allows a strict linear order hence the term Waterfall. The phases of the Waterfall Model are Requirements, Design, Implementation, Verification, and Maintenance. The requirements phase is very important, and this approach creates the need to have firm requirements. After requirements are set the design phase begins which will be approved by the client. Following design, the implementation phase begins which will include initial critical coding, unit testing, and final coding. The code is then sent to the client for final verification including user acceptance testing. Lastly, ongoing maintenance is provided based on the service level agreement.

# **Deliverables**

The client has requested to be provided with both the project and product deliverables in order as each phase of the Waterfall Development Model is completed.

## **Project Deliverables**

These items are focused on the progress of the project and give insight to the client that work is being achieved.

* Scope Document
  + An outline of the entire project including deliverables, budgets, and deadlines.
* Business Requirements
  + A high-level overview of the stakeholder requirements is needed to guide the design of the software by stating the functional uses of the software.
* Test Case Scripts
  + Technical documentation that outlines the conditions (data and processes) needed to fully test the appropriate functionality of the software product.
* Maintenance Contract
  + This deliverable contains the Service Level Agreement for continued operation and upgrades of the delivered software application.

## **Product Deliverables**

These items are delivered to the client with the need of their approval before the next phase of the project can begin.

* Wireframe
  + A design layout of the software application that outlines the structure of the application.
* Database Design
  + This deliverable includes a data flow diagram (DFD) that shows where the data flows and how it is mapped to the associated tables.
* Functional Prototype
  + Includes any design revisions based on the wireframes but also has simple functionality to increase the overall understanding of the coding process and confirm the system design.
* Critical Code Package
  + This first coding package includes the critical functionalities that are vital for the software to achieve its core business usefulness.
* Completed Code Package
  + The second code package contains revised code based on the product testing and completes the less important features as appropriate for the go-live date deadline.
* UAT Test Document
  + This final deliverable to the client confirms the system functions appropriately based on the client's high-level testing.

# **Implementation**

This standalone software will have a standard implementation. Before implementation the Operational Support Team will create a basic user guide to demonstrate the initial download, extracting, and setup of the software system. This document will be delivered to the appropriate personnel with a go-live date. This team will serve as technical support and be made available to new users as they initially set up the program.

From a technical perspective, the first task will be to create the AWS database instance and verify its availability which will be performed by the Software Engineering Team. The team will then need to update the database connection information in the code package for the new AWS database instance. Their role will include debugging any unexpected issues.

The completed code package will then be compressed by the same Engineering team and uploaded into the shared drive listed in the user guide. The client users will simply follow the guide to download, extract and run the new standalone application via the created executable. After the package is saved and executed to the desktop there will be an initial compile time for the application to build. Once built, the implementation is complete and subsequent executions will have an increased load time.

# **Validation and Verification**

To provide a valid product to the client the process begins with expert requirement gathering and analysis. These requirements will form the basis of the test cases and coding development. Multiple testing techniques will be used to confirm the correctness of the software application.

The first validation will come from the Software Engineering Team and will include Unit Testing. As the developers write code, they will also test the specific functionality of the code as each component is completed. This process does not test the entire code base but ensures a basic level of validation.

The Quality Assurance Team will be responsible for writing Product Test Cases, which will be approved by the client but executed by the QA Team. The execution will be started once the critical code package is delivered by the Software Engineering Team. Any errors found will be communicated with the Software Engineering Team and retested once the code has been revised.

Lastly, the client will verify the software system through the User Acceptance Testing process by executing specific tests for themselves. This is completed after the final coding package is delivered to the client and the written test cases can be based on the results from the Product Tests already conducted.

# **Environments and Costs**

## **Programming Environment**

Software Requirements:

* Mysql-connector-java-8.0.26jar
* IntelliJ IDEA Community 11.0.6+8-b765.25 amd64
* JDK: v11.0.11
* JavaFX: v11.0.0

Hardware Requirements:

* 2 GB of free RAM
* 1 GB Disk Space
* Monitor Resolution: 1024 x 768

## **Environment Costs**

The technical environment costs associated with this standalone application are very minimal. The application is written in Java and every Operating System in use today has a JVM able to compile and run the code. The main costs will be for hosting and maintaining a database. The suggested database hosting is Amazon Web Services (AWS) for their ease of use. Based on the current client usage estimates a MySQL compatible database through AWS with 100 GB of storage will cost around $150 annually. This figure can increase if the application load is increased based on demand. This includes a serverless option or an additional server added cost of $50 annually. Upgrades and security improvements are built into the annual costs already provided.

## **Human Resource Requirements**

The major cost for this project would be the cost of labor per individual. A typical human requirement breakdown for large-scale applications contains the following percentages of hours which correlates directly with the project costs.

Software Engineering Team – 70% or $149,100   
Quality Assurance Team – 20% or $42,600  
Functional/Business Team – 10% or $21,300  
Operational Support Team – 5% or $10,650  
Project Management Team – 5% or $10,650

These percentages and costs will vary depending on which phase of the project the software is currently involved in. The associated costs will dramatically increase if over-time hours are needed to meet difficult deadlines or if deadlines are extended leading to paid downtime for certain teams.

# **Project Timeline**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Milestone/Task | Deliverable | Description | Costs | Dates |
| Requirements | Task 1 | Scope Document | Defined scope from client and SME | $2,000 | 06/15/2021-06/30/2021 |
| Requirements | Task 2 | Business Requirements | Meeting with client and business needs review | $4,000 | 07/05/2021-07/14/2021 |
| Design | Task 1 | Wireframe and Database Design | Create the wireframe of the project and data model | $4,500 | 7/15/2021 – 7/30/2021 |
| Design | Task 2 | Functional Prototype | Functional simple version of the software | $25,000 | 08/05/2021-09/15/2021 |
| Implementation | Task 1 | Critical Code Package | Critical components completed with 75% of code | $135,000 | 09/15/2021-10/15/2015 |
| Implementation | Task 2 | Test Case Scripts | The results of the quality assurance testing | $25,000 | 10/16/2021-10/30/2021 |
| Implementation | Task 3 | Finished Code Package | Final Code 100% of code | $15,000 | 11/05/2021-12/05/2021 |
| Verification | Task 1 | UAT Testing Document | User acceptance testing of the final code | $2,000 | 12/05/2021-12/10/2021 |
| Maintenance | Task 1 | Maintenance Contract | Continued maintenance | $5,000 | 12/10/2021-12/15/2021 |