**C868 – Software Capstone Project Summary**

**Task 2 – Section A**



|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | Appointment/Customer Database Custom Software |
| **Student Name:** | Micah Senn |

**Table of Contents**

[**Table of Contents** 2](#_Toc126867793)

[**Business Problem** 3](#_Toc126867794)

[**The Customer** 3](#_Toc126867795)

[**Business Case** 3](#_Toc126867796)

[**Fulfillment** 3](#_Toc126867797)

[**Existing Gaps** 3](#_Toc126867798)

[**SDLC Methodology** 4](#_Toc126867799)

[**Deliverables** 5](#_Toc126867800)

[**Project Deliverables** 5](#_Toc126867801)

[**Product Deliverables** 5](#_Toc126867802)

[**Implementation** 6](#_Toc126867803)

[**Validation and Verification** 6](#_Toc126867804)

[**Environments and Costs** 7](#_Toc126867805)

[**Programming Environment** 7](#_Toc126867806)

[**Environment Costs** 7](#_Toc126867807)

[**Human Resource Requirements** 7](#_Toc126867808)

[**Project Timeline** 8](#_Toc126867809)

# **Business Problem**

**The Customer**

Ace Automotive is a family-owned and operated mechanic shop in business for over 50 years. They are staffed by four people including the owner and his son. They have served much of the local community for their entire lives. Their mission is to provide good long-lasting repairs for all of their customers in a timely manner. The owner and his son handle the larger more intensive jobs and the other two employees will handle general maintenance such as oil changes and tire balancing. The only IT infrastructure they currently have in place is one desktop computer to run their billing.

## **Business Case**

Currently, Ace Automotive’s only way of tracking appointments is by keeping a written ledger. They also keep all of their client’s contact information in a paper file with no security in place but a locked drawer. This can lead to issues if the owner is not aware of an appointment that the son booked for a labor-intensive job that will take many hours and subsequently books another appointment at that same time. The son also has a concern about the security of their client’s information and would like a secure and easy-to-use method of adding and retaining their client’s information. They would like the ability to run various reports including by appointment type, by timeframe, by technician, and by customer.

## **Fulfillment**

A standalone desktop application will best fit the needs of Ace Automotive as they currently do not have internet access in their shop, and it will make use of their current desktop environment. We will develop an application that will be able to add, edit, delete, and view all appointments, view appointments by type, view appointments by timeframe, and view appointments by employee ID. We will include in the application the ability to add, edit, delete, and view all customers and their respective contact information as well. We will also add a secure login screen requiring a username and password that is unique to each user (the owner and the son). The application will run on windows forms for an easy-to-use interface and use MySQL server running on the user’s desktop computer for the application’s database.

# **Existing Gaps**

Ace Automotive uses an old handwritten ledger to keep track of all of their appointments and they keep track of customers’ information in an old file folder. In order to provide the best quality of service there needs to be a system in place to prevent appointments from overlapping and bottlenecking the operation. Currently, the owner may make an appointment that the son is unaware of, and he may not notice through the many pages of their ledger there is already an appointment made for that time. Through the development of our application, we will ensure that all appointments are clearly visible and will not allow overlapping appointments to be made unless authorized. There also is the issue of their client’s information security. Currently, a folder contains many files full of clients’ personal contact information. They have tried through the years to add security to this system by keeping the folder locked up but that still is not secure enough for them. We will add a secure way to keep, edit, add, and delete their client’s information all through the use of our desktop application.

# **SDLC Methodology**

Due to the nature of this particular application, we have chosen the waterfall methodology as our Software Development Life Cycle (SDLC) methodology. Waterfall methodology is a methodology that develops in sequence through each phase of the project. Each phase must be completed before moving on to the next phase. This methodology leads to a predictable and easy-to-budget project. The waterfall method will best suit this project as there is a well-defined scope and there are specific features to be implemented. Using the waterfall method, we will have several phases throughout the development of the application.

The first phase is the requirements phase. In the requirements phase, we will gather the requirements that the customer has for the application, establish the required resources, and develop a timeline for our project.

The second phase is the system design phase. In the system design phase, we will develop both the logical and physical design of the application. This phase will cover designing the class diagrams, designing test plans, and designing a low-fidelity and high-fidelity wireframe.

The third phase is the implementation phase. The implementation phase is where we take everything we have gathered from the previous two phases and begin the full development process. Through the completion of this phase, our team will have developed a fully functional piece of software that fulfills functionality requirements laid out in the requirements document but has yet to be fully tested.

The fourth phase is the testing phase. In this phase, we will have our quality assurance team use the test plans designed in the second phase to ensure that the application we are delivering is as error-free as possible before it is deployed by use of unit tests. We will also be running acceptance testing to ensure the application meets all requirements listed in the requirements document.

The fifth phase is the deployment phase. The deployment phase is the phase in which we will deliver the finalized software to the end-user, which in this case is Ace Automotive.

The sixth phase is the maintenance phase. The final phase in waterfall methodology is a continuous phase that will handle any bugs or errors that may come up after the delivery of the software product. While we strive to detect all bugs through the testing phase a bug-free software product is a software product that does not exist.

# **Deliverables**

Throughout each phase in the development of our application, we can find a well-defined set of deliverables. We will break down those deliverables into two distinct categories project deliverables and product deliverables.

## **Project Deliverables**

* Requirements Document
  + As stated in the previous section this document is gathered during the first phase of development. This document will outline all the requirements and lay out all functionalities the customer would expect from our application. We will compare the final delivered application with this document to determine whether the customer’s requirements have been met.
* Test Plans
  + Test plans are crafted in the design phase of development and are used to reduce the number of errors in our delivered software application. The test plans that we put in place will be implemented by our quality assurance team using both white and black-box testing.
* Project Timeline
  + The project timeline is generated in the requirements phase of development. The project timeline will set goals and estimate the time to complete each phase of development to keep the project within budget and on time.
* UML Diagram
  + The UML diagram is a class diagram that shows the design of the database and the relationships that exist between all database objects. This is imperative to the implementation of the database that will be used on our application.

## **Product Deliverables**

Product Deliverables represent what is produced to deliver to the customer.

* Wireframes
  + In the system design phase, we will deliver a low-fidelity wireframe to show the general UI design of the application. The low-fidelity wireframe is not a complete representation of the final design but instead is the blueprint for the navigational aspects of the application. After agreement on the design shown in the low-fidelity wireframe, we will design a high-fidelity wireframe as a more accurate and final representation of the UI design. The high-fidelity wireframe will have much more functionality and design elements shown than the previous wireframe and will give the customer an accurate depiction of the finished application.
* Finished Application
  + In the deployment phase, we will deliver the finished application to our customer. The finished application will fulfill all of the requirements laid out in the requirements document, it will match the design of the previously delivered wireframes, and it will have passed all unit tests by our quality assurance team.

# **Implementation**

The implementation of this application should be a simple one. This is a brand-new system being added in a single computer environment that should have no effect on business uptime. Due to the usage of the waterfall method there only needs to be customer involvement in the early and late stages of development. In the requirements phase of development, our project manager will meet with Ace Automotive to discuss the desired functionality of the application and discuss the project timeline and budget. This should not interfere with business and should only be a few short meetings. In the design phase, we will request approval of our wireframe designs with Ace Automotive to ensure acceptance of our user interface design. We will then go on through the subsequent phases of development until we reach the deployment phase. In the deployment phase, we will install the application on the business’s computer and provide training to all staff on how to operate the software. We will also be providing documentation on how to operate the application for future reference. We will also offer constant support through our support staff in case any needs arise after deployment.

# **Validation and Verification**

Our quality assurance team will design robust and thorough test plans to ensure that the application we are developing is both functional and meets all requirements given by our customer. We will test our code base using white-box testing to ensure that all best practices are followed and test the functionality of the application with black-box testing to be able to test the application from the perspective of a user.

Our quality assurance team will also test the application in comparison with the requirements document to ensure that all required functionalities are deployed to the customer.

Our software team will develop and implement the use of unit tests for every section of code developed and used in the application. This is an autonomous and low-level form of testing that can be run during the development of a code base to ensure certain inputs give expected outputs.

Our project manager will also meet with the stakeholders to run acceptance testing. Acceptance testing is an end-to-end testing method that ensures that software is able to handle real-time business scenarios and meets the customer’s requirements as laid out in the requirements document. These tests will be the only tests to involve the customer and will take place directly before deployment.

# **Environments and Costs**

## **Programming Environment**

The application will be developed using windows forms. Most development will be done using the C# coding language, but we will also be using SQL for the implementation of the database. The database will be hosted on MySQL Server 8.0 locally on the client’s system. Due to the client’s lack of internet access and single desktop setup, our custom application will be developed with that in mind. Our application will be designed to run on Windows 10 and be usable within the specifications of the client’s desktop computer.

## **Environment Costs**

Due to the nature of the application and the environment we are developing it for there should be little to no recurring costs. There will be no web server used or cloud storage that would require monthly fees. We will also be using MySQL Server which is open source and has no fees.

## **Human Resource Requirements**

This project has several hands involved in product development. We have the project manager, design engineer, database engineer, quality assurance engineer, and software engineer. The project manager will most be involved in the early and late stages of development. The project manager is expected to put 20 hours into the production of this application, their rate is $30/hr which equates to $600. Our design engineer will be heavily involved in the system design phase of this project but not involved at all in other phases of development. Their rate is $40/hr and they expect to put 20 hours in equating to $800. Our database engineer will be working alongside the design engineer to develop the database schema and also working alongside our software engineer to ensure proper implementation of the database. The database engineer’s rate is $45/hr and they expect to put in 15 hours equating to $675. Our quality assurance engineer will only be involved in the testing phase of development, but they will be heavily involved in that phase. Our quality assurance engineer’s rate is $30/hr and they expect to put 20 hours in equating to $600. Lastly, we have our software engineer which will be involved in many phases in some capacity but will put the most time into the implementation phase. Their rate is $45/hr and they expect to put in 50 hours into the development of the application totaling. The final cost for human resources for the development of this custom software application is estimated to be $4,925

# **Project Timeline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Milestone/Task | Deliverable | Description | Dates |
| Requirements | Address goals and complete requirements document and project timeline | Requirements Document,  Project Timeline | Meet with Ace Automotive to gather functionality requirements, budget, and timeline requirements | 2/1/2023 –  2/5/2023 |
| System Design | Create low-fidelity wireframe and high-fidelity mockup | Low-fidelity Wireframe,  High-fidelity Mockup | Create the UI that relates the look and feel of the project | 2/6/2023 – 2/13/2023 |
| System Design | Create database schema | UML Diagram | Database engineers will create tables and develop relationships between them using a UML diagram | 2/14/2023 –  2/20/2023 |
| System Design | Create test plans | Test Plan | QA engineers will create full coverage test plans | 2/21/2023 –  3/1/2023 |
| Implementation | Create fully functional application | Application Ready for Testing Phase | Software Engineers will develop an application that matches the approved design and meets functionality requirements | 3/2/2023 –  3/29/2023 |
| Testing | Run unit tests to ensure functionality | Functional Application | Unit tests designed by our quality assurance team will ensure full functionality of the application | 3/30/2023 –  4/3/2023 |
| Testing | Acceptance testing | Ready to Deploy Application | Our Project manager will meet with the customer to run acceptance testing to ensure that the application fulfills all stakeholder needs | 4/4/2023 –  4/8/2023 |
| Deployment | Deploy finished application for business use | Finished Application | Our team will install the application and train the customer on proper usage | 4/9/2023 –  4/11/2023 |
| Maintenance | Execute maintenance plan | Customer Support | Our customer support team will offer continuous support as needed throughout the life of the application | 4/12/2023  -- |