**C868 – Software Capstone Project Summary**

**Task 2 – Section A**



|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | Box Builders Schedule Application |
| **Student Name:** | Jake Halley |

**Table of Contents**

Contents

[**Table of Contents** 2](#_Toc113451104)

[**Business Problem** 3](#_Toc113451105)

[**The Customer** 3](#_Toc113451106)

[**Business Case** 3](#_Toc113451107)

[**Fulfillment** 3](#_Toc113451108)

[**Existing Gaps** 3](#_Toc113451109)

[**SDLC Methodology** 3](#_Toc113451110)

[**Requirements Phase** 4](#_Toc113451111)

[**Design Phase** 4](#_Toc113451112)

[**Implementation Phase** 4](#_Toc113451113)

[**Testing Phase** 4](#_Toc113451114)

[**Delivery Phase** 4](#_Toc113451115)

[**Maintenance Phase** 4](#_Toc113451116)

[**Deliverables** 5](#_Toc113451117)

[**Project Deliverables** 5](#_Toc113451118)

[**Product Deliverables** 5](#_Toc113451119)

[**Implementation** 5](#_Toc113451120)

[**Validation and Verification** 5](#_Toc113451121)

[**Environments and Costs** 6](#_Toc113451122)

[**Programming Environment** 6](#_Toc113451123)

[**Environment Costs** 6](#_Toc113451124)

[**Human Resource Requirements** 6](#_Toc113451125)

[**Project Timeline** 6](#_Toc113451126)

# **Business Problem**

**The Customer**

Box Builders LLC is a growing producer of boxes used for a variety of applications, from shipping and storage to product boxes for store shelves. Although they produce boxes for general consumer use, they have recently had a lot of growth from smaller online sellers looking for custom boxes to ship their products in that both protect their products during shipping and works as an attractive display box. They want to capitalize on their recent growth as much as possible and maintain that growth for the foreseeable future.

## **Business Case**

Box Builders LLC’s current tracks appointment schedules with handwritten logs kept by each of the sales consultants. This has led to occasions where a consultant has scheduled overlapping appointments or management has called meetings when consultants have had appointments scheduled. The application to be developed will allow consultants to schedule meetings while preventing meetings from overlapping. Any user capable of logging into the application will be able to see a full schedule of meetings for the company as well as the schedule for individual consultants.

## **Fulfillment**

A stand-alone application will be developed to handle Box Builders LLC’s scheduling. A consultant will sign in with their username and password and be taken to a page to select what they want to see in the application. The selections on this page will include going to the full company schedule where appointments can be viewed and edited and new appointments can be made, viewing individual consultants schedule, viewing and editing the list of customers, and viewing a selection of reports. The appointment information, customer information, and user information will be stored in a connected MySQL database on the company’s existing server.

# **Existing Gaps**

The manual system used before left a lot of chance for easily preventable errors and much time wasted. With each consultant keeping their own manual log it is not unusual for a consultant to schedule overlapping appointments or contacting customers to make appointments when that customer already had an appointment with another consultant. It also means that the management team must contact and get the schedule of each consultant to plan meetings.

# **SDLC Methodology**

Waterfall methodology will be used to manage this project. The requirements of the project are already well defined and not likely to change once development starts. In the waterfall method, the customer requirements are defined at the start, the design is planned based on the requirements, the plan is then developed. Once the application has been developed, it is sent to the customer to review and ensure that it meets their requirements. Once it has been confirmed to meet the requirements, it gets deployed for regular use. If problems are found during regular use, reports are sent to the production team who fixes the identified problems.

**Requirements Phase**

This phase is to determine requirements for the application to be developed that will meet the needs of employees and management without becoming excessively costly to develop. Meetings with stakeholders will take place to define the specific set of requirements. The requirements will be subject to the approval of key stakeholders before continuing to the next phase of development.

**Deliverables:** Document that outlines the requirements approved by key stakeholders.

**Design Phase**

Once the requirements are finalized, low-fidelity wireframes will be designed to show the general design of the application and the flow between screens of the application. These wireframes will be presented to key-stakeholder whose feedback will be used to refine the wireframes into higher fidelity wireframes. The final version of the wireframe will guide the development of an entity diagram that will represent the tables for the database.

**Deliverables:** Wireframes of the software and an entity diagram

**Implementation Phase**

This phase will be used to develop the project planed in the previous stages. The development environment will be established and prepared and the remote database will be set up. The entity diagram will be used to design the database tables and the wireframes will be used to design the program.

**Deliverables:** An application that adheres to the requirements and wireframes, and a database that adheres to the entity diagram.

**Testing Phase**

The testing phase is required after the software has been built to find any issues and ensure that it meets the initial requirements. Quality assurance tests performed on the software will validate that each screen matches the design of the wireframe and that the flow between screens does not have any bugs. Issues and bugs detected in the quality assurance test will be documented. A unit test that can quickly test portions of the application will also be used in this phase.

**Deliverables:** A fully tested application, reports for any bugs found, and an automated unit test.

**Delivery Phase**

Once the critical issues turned up in testing have been solved the software will be ready for distribution to company computers. With the assistance of the companies IT department, the software will be packaged and installed on company computers. Andy configuration to the software to ensure it can access the remote database will also be done now.

**Deliverables:** Delivery and installation of software to all company computers and a packaged version of the software for any future installation needs.

**Maintenance Phase**

At this point the software will be in general use around the company. It is likely that issues will be found with the software during daily use that were not found in testing. These issues will be documented as they are reported. Each issue will be categorized and given priority by severity before the report is sent to the development team to fix.

**Deliverables:** General maintenance, bug fixes, and patches as needed

# **Deliverables**

The previous section included deliverables for each part of the project. This section describes these deliverables in more detail. The deliverables can be broken up into 2 different types, outlined below.

## **Project Deliverables**

Project deliverables are anything that the project manager is ultimately responsible for.

* Project Requirements Document
  + This document defines what the application to be developed should ultimately be able to do. Each requirement will need to be approved by the stakeholders before the next phase of development can begin.
* Project Schedule
  + The schedule is a breakdown of the planed timeline for the project. Each requirement will be turned into a set of tasks. Each task will have an estimated time to complete that will be placed on the schedule.
* Wireframes
  + A mockup of the application used to visualize the basic outline and different possible flows of the application. There will be a wireframe for each screen that describes the basic functionality. All wireframes will require approval before moving on to higher fidelity mockups.
* Prototype
  + Higher fidelity mockup based on the wireframe. The prototype is used to test design elements and give stakeholder clearer pictures of the final software.
* Database diagrams
  + This diagram shows the database tables and their relationship to one another.
* Test Plans
  + The testing plan will describe the procedures to be used to validate that the software is free of bugs. Bugs include both problems during runtime of the code and failure to adhere to the requirements described in the requirements document.

## **Product Deliverables**

Product deliverables are anything that will be delivered to the customer.

* Functioning application
  + The final application will include a graphical interface that will allow access to customer information, appointment information, and several reports. It will adhere to the documented and approved requirements.
* Database
  + The database hosted in a remote location and set up according to the entity diagram will be accessible by the application.
* Account Access
  + User accounts to access the application will be provided to employees

# **Implementation**

During implementation of the software, the project lead will meet with key stakeholders on a weekly basis to report on and discuss the project. Each requirement for the project will be broken down into smaller tasks, allowing for progress to be easily tracked.

Once the software is close to being completed, the Quality Assurance team will prepare to test the software. This will include setting up their workstations to install the software and connected to the database. Once the completed software is handed off to them, they will begin thorough testing each part of the program. Any bugs discovered by the QA team will be documented and reported to the development team. The development team can then prioritize and fix critical bugs. The weekly meetings will include reports from the QA team during this phase.

Once the QA team is satisfied that the application is nearly ready, the weekly meetings will include demonstrations of the software. This will help keep stakeholders engaged with the project and allow for feedback to be documented for future development.

Once the QA team has approved the application as being ready for release the project lead will work with the IT department of the company and the development team to develop a release plan. This plan will include packaging the application along with any necessary configurations needed to access the database then shipping it to company computers. The plan should use delivery software to make installation easy and reproducible. Once the plan is fully complete, the IT department will begin doing the installation on each computer.

A log in account will be created for each user to log into the application. Each user will have to enter the data from their personal logs manually. If their personal log was in the form of a spreadsheet, they will be able give the log to the IT department who will work with the development team to create a script to extract the data and add it to the database while preventing the duplication of entries in the database.

# **Validation and Verification**

Testing will be handled by the Quality Assurance team. They will go through each part of the application to find any bugs that may exist. They will also verify that the application meets all requirements outlined in the requirements document.

The testing will involve going to each screen of the application and testing various data entry, both entries that should return errors and entries that should go through without errors. This will help expose any bugs and potential security risks. The QA team will make a report for any bug they find and meet with the development team to discuss these bugs.

The application will also go through tests with the key stakeholders and end-users. The team leader will give users access to the software and observe how they attempt to use the software. If the users have issues using the software, either bugs the QA team did not find or confusion on how to use the application, the project lead will document these problems and work with the development team to fix the critical issues.

The product will be considered validated and ready for release once there are no glaring issues from user tests and the QA team has signed off.

# **Environments and Costs**

## **Programming Environment**

The application will be built in the following environment:

For example:

* Windows 10
* C# Programing Language
* MySQL server version 8.0.25 hosted on company server

## **Environment Costs**

Environment costs will be minimal for this project. The application is a standalone application that runs on each of the users’ computers. As the users will only need access to the database in the office, will be hosted on the company’s internal server. The internal server is currently primarily used for consultants to share mockups of customers desired boxes with the design team and for the design team to share design diagrams for consultants to go over with customers. If more storage is needed than what the server already has installed, a 10 TB server hard drive can be obtained for a one-time cost of about $300 and will cover storage needed for the database and many company projects for the foreseeable future.

## **Human Resource Requirements**

The team required to complete this project along with their projected hourly rate, an estimate of the time required, and the total amount paid for their work in this project is outlined in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Rate** | **Estimated Time** | **Total** |
| Project Lead | $60/hour | 50 hours | 3000 |
| Designer | $60/hour | 20 hours | 1200 |
| Developer | $75/hour | 80 hours | 6000 |
| QA engineer | $50/hour | 20 hours | 1000 |

# **Project Timeline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Milestone/Task | Deliverable | Description | Dates |
| Pre-development | Task 1 | Requirements Document | Meeting with stakeholders to define the requirements for the project. | 9/1/2022 – 9/5/2022 |
| Design | Task 2 / Wireframes and Prototype | Wireframes and  Prototype | Design the interface of the application | 9/6/2022 – 9/9/2022 |
| Design | Task 3/ Entity Diagram | Entity diagram of planed database | An entity diagram will be created that will be used to design the database schema | 9/10/2022 |
| Design | Task 4/ Testing Plan | Testing Plan | A testing plan to be used in the testing phase will be developed | 9/11/2022 |
| Implementation | Task 5/ Application | Functional Application | An application that adheres to the requirements outlined in the requirements document will be developed. | 9/12/2022 – 9/18/2022 |
| Testing | Task 6/ QA Testing | QA tests and approves for release, bug reports for developer to fix | The QA engineer will follow the testing plan and ensure the application meets the requirements | 9/19/2022-9/20/2022 |
| Testing | Task 7/ User Testing | User Testing is performed, reports the developer can use to refine user experience | The project lead will give the application to initial users and observe their use of its functionality | 9/21/2022 |
| Delivery | Task 8/ Deployment | The application is deployed onto company computers | The MySql database is set up and the application is distributed to users. | 9/22/2022 |
| Maintenance | Task 9/ Maintenance | Maintenance Plan and Contract | A plan and contract for ongoing and future maintenance is discussed | 9/23/2022+ |