**C868 – Software Capstone Project Summary**

**Task 2 – Section A**



|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | Co-Working Lots |
| **Student Name:** | Manuel Fuentes |

Contents

[**Business Problem** 3](#_Toc89266389)

[**The Customer** 3](#_Toc89266390)

[**Overview** 3](#_Toc89266391)

[**Business Case** 3](#_Toc89266392)

[**Fulfillment** 3](#_Toc89266393)

[**Existing Gaps** 4](#_Toc89266394)

[**SDLC Methodology** 4](#_Toc89266395)

[**Deliverables** 5](#_Toc89266396)

[**Project Deliverables** 5](#_Toc89266397)

[**Product Deliverables** 6](#_Toc89266398)

[**Implementation** 6](#_Toc89266399)

[**Validation and Verification** 6](#_Toc89266400)

[**Environments and Costs** 7](#_Toc89266401)

[**Programming Environment** 7](#_Toc89266402)

[**Environment Costs** 7](#_Toc89266403)

[**Human Resource Requirements** 7](#_Toc89266404)

[**Project Timeline** 8](#_Toc89266405)

[References 9](#_Toc89266406)

# **Business Problem**

**The Customer**

### **Overview**

Co-Working Lots Inc. (CWL) is an office space provider that serves working professionals by offering shared workspaces, private offices, and meeting rooms. CWL believes complex challenges will never be solved by one person or organization alone. Instead, people need to work together, which is why they set out to build a thriving, diverse community where people collaborate across sectors, networks, and neighborhoods.

The company is located in New York in a three-story building offering ten private offices, twenty dedicated desks, and fifteen conference rooms. CWL's in-house data center implements a virtual infrastructure but uses Amazon RDS (Relational Database Service) to host its current MySQL database

## **Business Case**

The Co-Working Lots building has a service desk that handles conference room bookings for its members. Currently, the service desk staff is using a Google Form to input the bookings. As its private offices and dedicated workstations reach full capacity, service desk staff have been experiencing a higher demand for conference room bookings. Having to manually check for conference room availability has proven to be time-consuming and fallible.

Co-Working Lots would like to develop software that connects to their MySQL database, allowing the call center employees to assign bookings only to their current members. The application will only allow booking for conference rooms that are not occupied during a selected date and time.

## **Fulfillment**

The proposed solution will be a stand-alone program written in java that service desk staff will use to create, update, or delete events that arise from customer requests. The events scheduled will contain the date, start/end times, the customer's name, type of event, and the conference room where the event will take place. The graphical user interface will allow the user to view events in either calendar or table format. In table format, events can be filtered by day, week, or month. Customers will be displayed in a table with search functionality to find specific customers. This booking application will prevent customers and conference rooms from having overlapping appointments or out-of-hours bookings. Users will be able to generate reports and save them locally on their computers.

# **Existing Gaps**

The existing system requires service staff to input the bookings using a Google Form. The form saves the entries in a Google Spreadsheet. Staff must manually check existing event entries for a given date to determine the availability of conference rooms. Manually checking conference room availability makes the booking process last longer and prone to overbookings.

Additionally, staff members have misspelled customer names when entering the host of an event. These events entries have resulted in conference rooms being empty but unavailable to members.

# **SDLC Methodology**

For this project, CWL would benefit most from a waterfall with feedback methodology. CWL's clear vision of their requirements and preference to apply the least possible changes to their infrastructure align best with a predictive model. The waterfall with feedback variation is divided into six sequential phases.

1. Requirements Phase.

All possible prerequisites of the framework to be created are caught in this stage and reported in a necessity particular document (Singh & Kaur, 2017).

1. Design Phase.

The requirement specifications from the first phase are studied in this phase, and the system design is prepared. This system design helps specify hardware and system requirements and helps define the overall system architecture (Singh & Kaur, 2017). This phase will include:

* 1. Establishing the main application features and how they will be implemented in the Graphical User Interface using mockups.
  2. Creating an Entity Relationship Diagram (ERD) of the database tables that the application will use.
  3. Creating a Unified Modeling Language (UML) class diagram detailing the different classes and interfaces the application will use along with their methods and properties.

1. Implementation Phase.

With inputs from framework design, the framework is initially grown in minor units, which are incorporated in the following stage. Every unit is created and tried for its usefulness, alluded to as unit testing (Singh & Kaur, 2017). This phase will include:

1. Creating a MySQL database.
2. Creating the project file structure in the IntelliJ IDE for the classes and interfaces in the UML class diagram.
3. Writing pseudo-code for the application's components using a top-down approach.
4. Creating the GUI pages that will display and allow interaction with the database records on the DBMS as well as any miscellaneous images that they may contain.
5. Converting pseudo code to java syntax.
6. Unit testing.
7. Verification Phase.

All the units that grew in the execution stage are coordinated into a framework after testing every unit. Then, the whole framework is tried for any issues and failures (Singh & Kaur, 2017). This phase will verify the product meets all requirements and is fully functional by performing a series of test cases.

1. Deployment Phase.

Once the functional and nonfunctional testing is done, the product is sent to the client environment or discharged into the business (Singh & Kaur, 2017).

1. Maintenance Phase

There are a few issues that come up in the customer environment. Patches are released to settle those issues. Additionally, to improve the item, some better forms are released. Maintenance is done to convey these adjustments in the client environment (Singh & Kaur, 2017).

Each phase of the waterfall methodology depends on the deliverables of the previous one, but movement between phases is allowed if some modification needs to occur (Stephens, 2015).

# **Deliverables**

There are two types of deliverables that are associated with the Waterfall SDLC that the customer has requested. They are project and product deliverables.

## **Project Deliverables**

These consist of items that are part of the Project Manager's realm of responsibilities.

* Requirements Document
  + A detailed list of the features the application must provide. The requirements should be explicit, but they should not contain technical details on how to execute the solution.
* Project Schedule
  + A summary of each task and their expected completion date.
* Test Plan
  + Test cases to validate functionality and fulfillment of the requirements document.

## **Product Deliverables**

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

* Database Entity Relationship Diagram
  + A graphical representation of the relationships between new entities/tables and the existing database schema.
* UML Class Diagram
  + A schematic representation of the classes that make up the system, their properties and methods, and their relationships.
* Wireframes
  + Low fidelity, rough representations of the application's graphical user interface
* GUI Mockups
  + High fidelity designs of the graphical user interface
* Java Application

# **Implementation**

The Project Manager will oversee the implementation process involving two CWL departments in a variety of capacities.

IT Department

CWL's existing virtual infrastructure will simplify the deployment process. The IT staff will first access the MySQL Server's virtual machine to import the new schema. Then the application will be installed on all service desk virtual machines. This process will be performed during non-operating hours to avoid service disruption.

Service Desk Department

The service desk staff will take part in a 30-minute training prior to deployment. Upon training completion, staff will use the java application to input conference room bookings.

# **Validation and Verification**

Unit tests will be performed by the Software Quality Assurance Specialist using a combination of white and black box testing. Once the application is validated through unit testing, it will proceed to acceptance testing. In addition, the customer will perform comprehensive acceptance testing before taking ownership of the application, thus ensuring that the application meets the designed requirements.

# **Environments and Costs**

## **Programming Environment**

CWL's virtual infrastructure will provide the necessary resources to complete the project. The software requirements for the new virtual machines are:

* MySQL Community Server 8.0.23 (running a copy of the existing database schema)
* MySQL Workbench 8.0.23
* IntelliJ IDEA Community Edition 2021.2.2
* Java SDK 17.0.1

## **Environment Costs**

Environment costs are relatively minimal. Given CWL's existing infrastructure, no additional hardware is required. However, the changes made to their existing database schema will increase its size significantly, and the Amazon RDS MySQL instance will need to be upgraded. The changes amount to an estimated $500 annual increase.

## **Human Resource Requirements**

The larger share of human resources is by the developers, followed by the project manager.

* Developers consume approximately 80% of the hours and 53% of the budget.
* The Project manager consumes 100% of the hours and 22% of the budget.
* The Software Quality Assurance Specialist consumes 44% of the hours and 14% of the budget.
* The UX Designer consumes 40% of the hours and 11% of the budget.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Role | Quantity | Rate Per Hour | Time / Perce | Total |
| Project Manager | 1 | $50.00 | 200 hours | $10,000 |
| UX Designer | 1 | $60.00 | 80 hours | $4,800 |
| Developer | 2 | $75.00 | 160 hours | $24,000 |
| Quality Assurance | 1 | $70.00 | 88 hours | $6,160 |
| Total Costs |  |  |  | **$44,960** |

# **Project Timeline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Milestone/Task | Deliverable | Description | Dates |
| Requirements | Task 1 / Discovery | Requirements document | Meet with the customer and stakeholders to discuss requirements. | 10/4/2021 – 10/8/2021 |
| Design | Task 2 / Wireframe and Mockups | Low-Fidelity Wireframe  High Fidelity Mockup | UX Designer creates the UI that relates to the look and feel of the project. | 10/11/2021 – 10/15/2021 |
| Design | Task 3 / Database Design | ERD Diagram | Developers create a visual representation of changes made to the existing database schema. | 10/11/2021 – 10/13/2021 |
| Design | Task 4 / Class Design | UML Class Diagram | Developers create a visual representation of the data classes that will be used during the development phase. | 10/14/2021 – 10/15/2021 |
| Design | Task 5 / Testing Plan | Testing Plan | Quality Assurance Specialist creates a plan for functional and user acceptance testing. | 10/11/2021 – 10/13/2021 |
| Implementation | Task 6 /  Application | Database Schema  Java Application  Test Source Code | Developers write the application's source code while taking into consideration the testing plan. | 10/18/2021 – 10/29/2021 |
| Verification | Task 7 /  Unit Testing | Unit Test Results | Quality Assurance Specialist evaluates unit test results. | 11/1/2021 – 11/1/2021 |
| Verification | Task 8 /  Acceptance Testing | User Acceptance Test | Quality Assurance Specialist evaluates acceptance test results | 11/2/2021 – 11/3/2021 |
| Deployment | Task 9 / Application Deployment | Application Deployed | The client's IT staff follows the deployment steps produced in the requirements document | 11/4/2021 – 11/5/2021 |
| Maintenance | Task 10 / Maintenance Plan | Maintenance contract | Terms of the established service level agreement are followed. | 11/5/2021 |

# References

Singh, A., & Kaur, P. J. (2017). A Simulation Model for Incremental Software Development Life Cycle Model. *International Journal of Advanced Research in Computer Science, 8*(7), 126-132. doi:10.26483/ijarcs.v8i7.4136

Stephens, R. (2015). *Beginning Software Engineering* (1 ed.). Wrox.