

C868 – Software Capstone Project Summary

Task 2 – Section A



Capstone Proposal Project Name: _____ Parts and Products Manager Improvement _____

Student Name: _____ Jonathan Kleve _____

Table of Contents

Contents

Table of Contents	2
Business Problem	3
The Customer	3
Business Case.....	3
Fulfillment	4
Existing Gaps.....	4
SDLC Methodology	5
Deliverables	6
Project Deliverables	6
Product Deliverables.....	7
Implementation	7
Validation and Verification.....	8
Environments and Costs.....	8
Programming Environment	8
Environment Costs.....	9
Human Resource Requirements	10
Project Timeline	10

Business Problem

The Customer

The customer is a bicycle repair shop owner who is currently the sole employee of his own small business. Their business is growing, so they are looking to hire additional help shortly. Their current inventory management software lacks the features they will need to expand their business.

Business Case

The current inventory management software has an interface that allows users to enter, edit, delete, and search information associated with parts and products. However, that information is only stored locally as an object as part of the program. The owner fears the current software will become too cumbersome and slow as the business expands. Due to the increased sales, the owner has trouble keeping track of the inventory on hand for various parts and products. In addition, the part and product information can be accessed, modified, and deleted by anyone who can access the computer it is currently on. With greater foot traffic in the store and potential new employees, the owner wants to add a means to secure access to the software as well as track changes made. Therefore, three major enhancements to the software are proposed. The first major enhancement is storing the part and product information in a database for scalability, data integrity, and security. The second major enhancement is the ability to generate reports about the parts and products to help the owner monitor inventory levels and otherwise manage their business. The third major enhancement is adding a login system with tracking for enhanced security and accountability.

Fulfillment

The first improvement, the change to database storage, will allow vastly greater storage capacity for entries of different parts and products. Databases also enable efficient queries of this information, allowing the owner to search for the parts or products he needs without wasting his or the customer's time. The database also stores the information more securely regarding access controls, integrity, and availability. In brief, this means the information stored in a database is harder for malicious parties to access when configured correctly, is much less likely to be corrupted or otherwise compromised due to a computer error, and can be accessed from other locations and devices if they take the step to run the database on a server.

The second improvement, the ability to generate reports, will allow the business owner to quickly obtain information about his current inventory of parts and products to know what to order more of if they did not notice when they made the sale. The third improvement, the addition of the login system, will enable the business owner to control who has access to their inventory management software. It will also enable them to check what user accessed the system and when if something about the information in the system is incorrect. That way, they can tell if, for example, their new employee forgot to update the inventory of an item after selling one.

Existing Gaps

As stated in the business case section, the current inventory management software has an interface which allows users to enter, edit, delete, and search information associated with parts and products. However, that information is only stored locally as an object as part of the program. This fact means the information crucial to the business's operation is vulnerable to corruption and otherwise being compromised due to computer errors. It also has no means to

control access to the software that a login system allows and enhances security. Furthermore, checking information about multiple parts or products would require the user to search or scroll through each entire list on the main menu. In contrast, a report-generating function can create a list of all parts and products with a current inventory of zero with a few clicks on some buttons.

SDLC Methodology

The waterfall software development life cycle methodology will be used because the requirements are explicitly defined and are unambiguous. The bicycle shop owner wishes to proceed with the major improvements proposed as their main priority. They also do not have the time to review iterations throughout development and are okay with providing feedback towards future adjustments at a later time. The bicycle shop owner would also like the enhancements completed as soon as possible.

The six phases of the waterfall software development methodology life cycle are requirements gathering and analysis, system design, implementation, testing, deployment, and maintenance. Requirements gathering and analysis is the process of collecting information to ensure a clear understanding of all of the requirements the software product must fulfill. In this scenario, requirements gathering and analysis will be accomplished simply by communicating with the bicycle shop owner. System design is the process of mapping the requirements to specifications for the software product. In this scenario, the initial proposal has already accomplished this goal as it contains concrete aspects that fulfill the business owner's desire to improve their software for usability, scalability, and security. Implementation is the actual software development. This is the process whereby design specifications are translated into working code. Testing is verifying and validating that the software meets the requirements and is

free from bugs or other defects. Deployment makes the software available to the end user on their system. Maintenance is providing ongoing support and making necessary changes to the software after deployment.

Deliverables

Two types of deliverables are associated with the Waterfall SDLC. They are project and product deliverables.

Project Deliverables

These are created to manage and control the project itself.

- Hardware and Software Requirements Specification: Lists necessary tools for development and deployment.
- Code Documentation (Comments, Style Guides): Internal documentation for developers.
- Test Plan: Strategy for testing the product.
- Test Logs and Execution Reports: Records of testing activities and results.
- Defect Reports (Bug Reports): Documentation of identified issues.
- Deployment Plan: Strategy for releasing the software.
- Maintenance Logs: Records of post-deployment activities.
- Change Requests: Formal requests for modifications.
- Bug Fixes and Patches: Solutions to identified problems.
- Requirements Traceability Matrix (RTM)

Product Deliverables

These are produced for the customer to review or utilize.

- Requirements Specification Document (or Software Requirements Specification - SRS):
Defines what the product will do.
- Business Requirements Document (BRD): Describes the business needs the product addresses.
- Use Case Specifications: Detail how users will interact with the system.
- High-Level Design (HLD) Document: Outlines the system's architecture.
- Low-Level Design (LLD) Document: Provides detailed specifications for components.
- Interface Design Specifications (Mockups, Wireframes): Visual representations of the user interface.
- Database Design Document (Schema): Structure of the product's data storage.
- Developed Software Codebase: The actual working software.
- Unit Test Cases: Tests for individual code components.
- Deployed Software in the Target Environment: The final, working product.
- Installation Guide: Instructions for setting up the software.
- User Guide: Help users understand how to utilize the product.
- Release Notes: Describe changes and updates in new versions.

Implementation

The implementation of this project will be straightforward. The improved software and MySQL Workbench will be installed on the device the bicycle shop owner is already utilizing. The database will be set up to be hosted locally on the same device. Then, the improved software

will be run to ensure it is working correctly. Last, the bicycle shop owner will be educated on how to operate the software.

Validation and Verification

Given the low complexity of this software's technical aspects and the waterfall methodology, the validation and verification will boil down to two main points. First, the developer will test the software internally on their own device, ensuring that all aspects operate properly in that environment before attempting to deploy the software on the customer's device. Second, the software will be deployed on the customer's device, and the customer, the bicycle shop owner, will be able to operate the program and confirm whether it meets their needs. The developer will be on hand to guide the customer on utilizing the software and troubleshooting or at least witnessing any issues they encounter. The customer will be able to perform this acceptance testing before finalizing the software purchase.

Environments and Costs

Programming Environment

The following hardware and software will be required for the development of this project:

- Windows 10/11
- MySQL Workbench 8.0 CE
- IntelliJ IDEA CE 2024.2.1
- Gluon Scene Builder
- A personal computer capable of running all the above software

Environment Costs

The IntelliJ Idea, Scene Builder, and MySQL Workbench applications have no cost associated with using them for development. The Windows license is around \$100, but this cost has already been covered as the necessary licenses have already been purchased for the developer's and business owners' computers. There are also no necessary hardware costs at this time, as the business owner already has a computer that runs the inventory management program. This computer can also operate the database for the time being. However, if the business owner wants to transition to operating the database on a server, they would need to cover the cost of the necessary allocation space on a rented server or their own dedicated server. The pricing of the server space varies and largely depends on the size of the database. The number of parts and products to be tracked would need to be discussed with the bicycle shop owner to estimate the necessary size.

Human Resource Requirements

For human resources, the project will take an estimated 30 working hours of design and planning, 100 hours of development, and 20 hours of documentation to complete. The cost for this time will be billed at a rate of \$20 per hour, a normal rate for a junior software developer, meaning the total estimated labor cost will be \$3,000.

Project Timeline

Phase	Milestone/Task	Deliverable	Description	Dates
Planning	Task 1	Requirements, Use case specifications	Meet with the customer and understand their needs	05/05/2025 – 05/06/2025
Design	Task 2 Section A	LLD Documents, HLD Documents, Mockups, Wireframes, Database Design Schema,	Create all design documents outlining the software	05/07/2025 – 05/18/2025
Development	Task 2 Section B	Developed Software Codebase	Adding the database connectivity, report generation, and login system features to the software	05/19/2025 – 06/01/2025
Testing	Task 2 Section C	Test Logs and Reports,	Test the software to ensure it is free from defects	06/02/2025 – 06/08/2025
Documentation	Task 2 Section C	Installation Guide, User Guide, Code Documentation, Release Notes	Documenting code, creating user and installation guides, and writing up release notes	06/09/2025 – 06/15/2025