C868 – Software Capstone Project Summary

Task 2 – Section A

Capstone Proposal Project Name: SapbaShop

Student Name: Christopher Smith

Table of Contents

Business Problem 3

The Customer 3

Business Case 3

Fulfillment 3

Existing Gaps 3

SDLC Methodology 3

Deliverables 4

Project Deliverables 4

Product Deliverables 4

Implementation 5

Validation and Verification 5

Environments and Costs 5

Programming Environment 5

Environment Costs 6

Human Resource Requirements 6

Project Timeline 7

Business Problem

The Customer

The client is SapbaShop, a small marketplace that sells local produce and locally made products. It is owned and operated by a group of friends that want to improve their community. They have a storefront location that is doing well in Chiang Mai, Thailand, and looking to purchase another. Outside the store, they provide consultations for organic farming and lifestyle improvement. They hope to improve people's lives by promoting locally grown organic foods and natural products.

Business Case

SapbaShop conducts daily operations and inventory management through the sole office computer utilizing a commercial desktop application. Having all the inventory data on one computer provides better security; however, it does not allow flexibility. Flexibility is what the company needs to grow. The client wants to expand its footprint and needs to be able to track inventory across multiple locations. They are looking for an application that is easy to use, shows inventory reports at all locations, and is not limited to a single computer. In addition, it needs to be secure, allowing only authorized users to modify the inventory. They also want to establish an online presence, allowing customers to view their products, search and see if a product is available at a store nearby.

Fulfillment

A web application would be the best solution for managing inventory across multiple locations for the client. An easy-to-use interface would be accessible from any device using a web browser. All store's inventory data would be kept synchronized utilizing a central server. The client will have the flexibility of using an on-premise server or cloud service provider to host the application. The web application also would allow customers to view their inventory through the store's web page. After authentication, the same page could enable the client to modify inventory and product details.

Existing Gaps

The current system is a commercial desktop application aimed at small businesses. It has been able to perform all the client's daily business operations and inventory management needs. The desktop application, however, limits the client to using only one computer containing all their business data. Therefore, the application has already been a bottleneck in inputting transactions within the current store. Additionally, their online footprint has been minimal as they have only used social media to post information about their store and products. The social platform has been chiefly a means of advertising, as any sales come at the cost of using it.

SDLC Methodology

This project will use the Waterfall methodology and later transition to an agile scrum during the maintenance phase. The main reasons for choosing this methodology are that all the requirements are presented, well defined, and will not change. Additionally, the project has a short deadline due to the client working to open another storefront. With the short deadline, the focus will be placed on functionality rather than design or user experience to allow enough time for each stage and ensure it meets all requirements. The Waterfall methodology consists of six phases:

1. Requirements Gathering

2. Design
3. Implementation
4. Validation
5. Deployment
6. Maintenance

During the Maintenance phase, the project will shift to agile scrum methodology for bug fixes and follow-on projects. Upon acceptance of the proposal, this will enable continuous integration and delivery of new features and upgrades.

Deliverables

For the Waterfall methodology, there are two types of deliverables, Project, and Product.

Project Deliverables

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

- Requirements Documentation
  - A collection of the application requirements that meet stakeholder objectives. They are completed during the requirements gathering.
- Project Scope Statement
  - The document sets the project's boundaries, describing what will and will not be included in the application. It also is completed during the requirements gathering phase.
- Project Schedule
  - Will show when tasks will begin and end with associated deliverables. The last deliverable for the requirements gathering phase.
- Test Plans
  - The test plan will be the way to verify that the application meets the requirements outlined in the requirements documentation.
- Source Code / Final Project
  - The complete application for the client.

Product Deliverables

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

- User Interface Wireframe
  - The wireframe will be low fidelity, rough representation of the user interface as seen from a web browser.
- Class Diagrams
  - The class diagrams are designs with no functionality that show the objects used in the application, their attributes, behaviors, and how they relate to each other.
- User Manual
  - The user manual guides how to use the application and its features.

Implementation

Once the application is completed, we will help install it on the client's server of choice. The application can be installed from source code or container services such as docker or podman. For simplicity, ease of management, and security, a rootless container service, podman, will be used. It will need to be paired with an instance of MongoDB with a container volume for the NoSQL data storage. Once the Sapbashop app and MongoDB images are pulled from the registry, a simple podman CLI run command or YAML file with the client's configuration and the application is running. The rest is to ensure that networking is set up for the client to access the application port. Networking will vary depending on the client's chosen infrastructure.

Validation and Verification

Testing will first run through our testing framework, ensuring the application logic functions as required. The framework will consist of unit and endpoint tests. The client and their employees will conduct testing during final validation and acceptance testing. As agreed upon, the focus of the testing will be more concerned with functionality than the user experience due to the project's timeline. Each tester will be given a checklist of tasks based on the application requirements. After all testing rounds are completed and data complied, the client will decide to accept the application or return for further work.

Environments and Costs

Programming Environment

The environment is flexible and can be configured in multiple ways. The below configurations will show two possible routes, minimal self-hosted and cloud-based.

Self-hosted:

- Internet Service
- Router with port-forwarding
- 64-bit Computer with:
  - Ethernet port or Wi-Fi, *Recommend Gigabit Ethernet port
  - Linux x64 Operating System Installed (Debian or Fedora Server)
  - (Optional) Docker or Podman to run applications in containers
- MongoDB v5 (From Package manager or Container Image)
- SapbaShop application (From Source Code or Container Image)
- Client machine: Web Browser to view User Interface

Cloud-Based:

- Virtual Server with Linux x64 OS at a Platform as a Service provider (i.e., AWS, Digital Ocean, etc.)
- MongoDB v5 (From Package Manager or Container Image)
- SapbaShop application (From Source Code or Container Image)
- Client machine: Web Browser to view User Interface

Environment Costs

The self-hosting option will be a higher upfront cost due to the price of purchasing the hardware required. Router and Server hardware with storage can vary widely, but refurbished enterprise-grade hardware can be purchased for less than $1000. In the long term, it may be cheaper if the company or customer demand does not outgrow the hardware. This option is recommended if the client has a System Administrator to keep the systems updated and verify upgrades before deploying live.

The cloud-based option startup costs will be lower, and packages will vary between service providers. This option may not require the client to maintain the services as most providers offer packages to manage the server for additional costs. If the client forecasts the fast growth of the company, then this option is recommended, as most cloud services are expandable to fit added demand. For example, AWS services can provide almost a fully managed server using Elastic Beanstalk. It combines their EC2 and S3 storage and can be paired with DocumentDB, which is compatible with MongoDB. A quick estimate of a minimal EC2 and DocumentDB instance based in Singapore, the closest to Thailand, will cost around USD 1309 annually. *This may not be an accurate estimation as there are free services to take advantage of, and storage requirements may differ.

Human Resource Requirements

The time estimated to complete the application is 39 days, excluding the maintenance phase. Below is the breakdown by Job title of the small team. Person-hours overlap phases depending on the position.

Position	Hours	Rate	# Members	Totals
Project Manager	920	65	1	$ 59,800
UI/UX Designer	120	28	1	$ 3,360
Full-Stack Developer	1280	48	3	$ 184,320
QA Specialist	360	42	1	$ 15,120
Totals	**2680**		**6**	**$262,600**

Project Timeline

Phase	Milestones	Deliverable	Description	Days	Dates
Requirements Gathering & Planning	The client accepts the contract and project plan.	• Requirements Document • Project Scope Statement • Project Schedule	Meeting with the client, gathering all requirements, and developing the project plan.	3	6/27/2022 – 6/29/2022
Design	All Design deliverables are complete.	• Class Diagrams • Wireframe • Test Plans	Create visualizations of the application logic, interactivity, and user interface.	3	6/30/2022 – 7/4/2022
Implementation	Application Complete		Developing the application	21	7/5/2022 – 7/26/2022
Validation	Acceptance Testing Successful		Testing the application according to the test plan to ensure it meets the client's requirements.	9	7/27/2022 – 8/5/2022
Deployment	The application is live on the client's server.	• User Manual • Source Code	Ensuring the client has the application and is running as intended.	3	8/8/2022 – 8/10/2022
Maintenance	Plan for CI/CD accepted.	• Product Backlog • Sprint Backlog	Bug fixes and the next project planning begins.	–	–