**C868 – Software Capstone Project Summary**

**Task 2 – Section A**



|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | Ticket Management App: Project Themis |
| **Student Name:** | Michael Tillotson |

**Table of Contents**

Contents

[**Table of Contents** 2](#_Toc60212503)

[**Business Problem** 3](#_Toc60212504)

[**Customer** 3](#_Toc60212505)

[**Problem** 3](#_Toc60212507)

[**Case** 3](#_Toc60212508)

[**Solution** 3](#_Toc60212509)

[**Methodology** 4](#_Toc60212510)

[**Deliverables** 4](#_Toc60212511)

[**Project Deliverables** 5](#_Toc60212512)

[**Product Deliverables** 5](#_Toc60212513)

[**Deployment** 5](#_Toc60212514)

[**Testing Methods** 5](#_Toc60212515)

[**Resource Requirements** 6](#_Toc60212516)

[**Programming Environment** 6](#_Toc60212517)

[**Environment Costs** 6](#_Toc60212518)

[**Human Resource Requirements** 6](#_Toc60212519)

[**Timeline** 7](#_Toc60212520)

# **Business Problem**

**Customer**

The customer is Temple Forestry, a small lumber manufacturing company that has been able to expand significantly in recent years due to strong stewardship from its management, sustainable long-term forestry practices, and recent upswings in lumber pricing. These driving factors have enabled Temple Forestry to purchase a second sawmill and manufacturing plant, as well as the purchase of an additional two hundred thousand acres of timber. These purchases allowed Temple Forestry to double their operational capacity but have also led to significant increases in headcount and a need for investment in more scalable infrastructure to accommodate their growth. They currently have roughly two hundred employees, half of whom utilize or have utilized IT services in the past.

**Problem**

Currently within the IT department tickets or requests for technical support are handled on an individual basis. Due to the relatively small size of the company, and the proportionately small size of the IT department, support issues are handled between four or five different people who communicate as necessary to serve the needs of their users. While this process served adequately in a small environment, it was already becoming problematic in certain situations such as when a senior technician took a multi-week vacation and was unavailable for most of that time. Tickets known only to that technician went unworked on for the entirety of that time. This method of ticket handling creates information silos, where only certain people know about certain issues, and thus only they can handle those issues.

## **Case**

Temple Forestry deals with on average between five and ten support requests a day. These vary from simple technical fixes, requests for the installation of software or hardware, or assistance in business projects that require technical expertise. Up until the expansion this was a functional if not efficient method of handling the technical needs of the business. When the acquisition occurred, it was rapidly determined that this would no longer be a viable method of handling user support. There is a need for a centralized repository application to manage the technical requests and issues of the company’s users. By developing this application, you create a single point of reference where any technician can log in, view what tickets are active, what user needs help, and what their issue is.

## **Solution**

Themis will be a standalone workstation application written in Java, utilizing a MySQL database. A technician will log in using standard username/password credentials, which will be validated against a username table within the database. The username will be cached in session so that when the technician finishes logging in, the main application dashboard will display a table view of all tickets that have been assigned to them. From there they can step into individual tickets to update or delete them as needed. There will also be menu buttons leading to the other pieces of the application. The user will be able to generate reports based off specific criteria, including total number of tickets, high priority tickets, tickets without owners, or tickets that have been active for over a certain length of time. The user will also be able to search by Ticket ID or technician name to find specific tickets.

# **Methodology**

Due to this project being the first of its kind for Temple Forestry, it is likely that the first iteration will be unable to capture everything that the sponsor and client are envisioning in the long term. There is a vital need for this application to be implemented as soon as possible; to prevent further decay of the reputation of the IT department within the company due to delayed or lost support requests. For this reason, we recommend developing under the Agile model. This will allow the department to get the project functional as quickly as possible, and allow for the implementation of less critical pieces of the application to be developed in future iterations.

Agile, like most SDLC methods, follows the six stages within each sprint:

* Planning
* Design
* Coding/Development
* Testing
* Deployment
* Review

Each of the six Agile stages will be fully utilized in developing this application. During the Planning phase, developers will interview and develop stories from the perspectives of at least three of the people using this application. Review of these stories can determine what the application needs, vs what would be nice to have, but can be included in later iterations. There will be some overlap with the design phase here, as when planning what to include developers must communicate with key power users within the company who can best confirm their findings when the application requirements are dialed in using the Top-Down approach.

Once Planning and Design is dialed in the development process begins. This is where the actual coding of the application is performed, as well as the development of the User Interface (UI). This stage will overlap with the early stages of Testing, as unit tests will be deployed as soon as feasible, to further minimize total development time. When all major pieces have been individually tested and combined, full Application testing must be done, first by quality assurance, then by real users in a live beta. Once the live beta is completed, and all necessary fixes implemented, the application will be deployed to Temple Forestry technicians in the native environment. Feedback and post-mortem will be delivered, and the second iteration of the Agile process can begin. This serves as one of the main benefits of the Agile model for this project. The software can be prepared and delivered in a minimal amount of time, and more cosmetic, less critical features can be delivered in future iterations as the company has the bandwidth and time to do so.

# **Deliverables**

There are 2 types of deliverables that are associated with the Agile SDLC. Project Deliverables and Product Deliverables which are broken down below.

## **Project Deliverables**

Items that fall within the Project Managers bailiwick of responsibilities to manage:

* Timeframe
  + The schedule of the project, what needs to be done, and roughly outlines for how long it will take to complete.
* Wireframe/UI Documents
  + Rough outline of the applications user interface
* Product backlog
  + Full list of product requirements both in scope and out of scope for the current cycle of development. This is the master document which will be referenced when determining what will be done during each iteration of development.
* Sprint Backlog
  + List of product requirements within scope for this cycle of development. This document will be limited to what the development team is currently working on.

## **Product Deliverables**

What will be delivered to Temple Forestry on completion of the project.

* GUI fully functional
  + A fully developed UI that allows the user to navigate within the application.
* Security access
  + A guarantee of user security using standard login protocols.
* Custom functional database
  + The database will reside on the company servers, to be accessed primarily through the Themis application.

# **Deployment**

Deployment of this software will need to be coordinated between the IT Department manager as well as the heads of the two teams within the IT department. As the department itself is small, numbering only fourteen people, the pre-requisites for deploying the software will be basic. The application must be installed and functional on all endpoints and maintain a solid network connection to access the database stored on the company servers. The support of senior technicians will be invaluable in the deployment of this software as they are the users most often contacted for assistance. Initial training should be done with these users to ensure their comfort and knowledge of the new application. The entire deployment process should take no more than four business days including training sessions.

# **Testing Methods**

This application is primarily a data entry and storage application. Other features will be developed in time, but its core functionality is the ability to create, update, store, and delete records. To that end the most critical testing should be around these functions. There are three types of records. Users/Technicians, Employees, and Tickets. The function of adding, updating, saving, and deleting each type of record must be tested, as well as the ability to search and select records for viewing. Unit testing of each segment as it is added to the application will be most effective at ensuring record management is functional. Once the application has been assembled, end-to-end application testing should be implemented with dummy data provided. Black box testing should be the last stage, done by the Power users during training. This will also double as requirements gathering for future development cycles.

# **Resource Requirements**

## **Programming Environment**

This application is only designed to run within a Windows environment and will require a dedicated server for database storage. Software requirements detailed below:

* Windows 10 Server Enterprise
* MySQL Custom Database
* Microsoft Windows 10 (any version)

## **Environment Costs**

Environment costs are projected to be minimal for the development of the standalone application. The infrastructure needed is largely in place already as users have Windows 10 machines, and existing servers in place used for database management. There will be a $2000 per year licensing fee to use MySQL in an enterprise environment, the only fee not currently included in yearly budgeting.

## **Human Resource Requirements**

The largest human resources requirement for this project will be the developers. It is estimated that 85% of the budget hours and money will be consumed by the development team. The remaining 15% will come from the power users involved in testing and administrative activities.

# 

# **Timeline**

Below is an overview of the projects proposed timelines. All dates are tentative and subject to change based on requirements gathering or project need.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Phase | Milestone/Task | Deliverable | Description | Dates |
| Planning | Requirements gathering/initial planning | Requirements | Client meeting, user stories gathered | 1/1/2021-1/14/21 |
| Design | Core Design | Low fidelity wireframe  High fidelity mockup | Wireframe mockup, UI prototypes of wireframe | 1/17/21-1/28/21 |
| Development | Complete Application 1st iteration | Functional application with live data | Development Work, creation of application based on requirements gathering and using UI prototypes | 1/31/21-2/21/21 |
| Testing | Unit and Application testing | Test results, and hit list for future bug fixes | Testing will begin during the Development process and will run concurrently to shorten development time | 2/6/21-2/24/21 |
| Deployment | Application Deployment | Application Deployment | Application will be deployed first to Power Users, then several days later to regular users after training sessions | 2/26/21-3/3/21 |
| Review | Post-mortem | Review of process and future goals | A review of the development process, determining areas that need focus in future Agile cycles. Will form the basis of requirements gathering for next stage of project | 3/6/21 |