**D424 – Software Engineering**

**Task 3**

|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  Bike Shop Inventory Desktop Application |
| **Student Name:** | Fabio Tran |

**Table of Contents**

*Create a professional looking Table of contents that includes your main and subheadings and the related page numbers. Use the automatic TOC generating function of Word or other word processing packages to make the process easier.*

*The headings that follow are only examples of what might be included. You will need to create headings that are appropriate for your application and process.*

***Remember that this needs to be a professionally formatted document with detailed information about your project that is easily accessible.***

**Task 3 Design Document**

# Application Design and Testing

## Class Design

The Bike Shop Inventory Desktop Application is made leveraging the .NET MAUI Blazor framework. This design adheres to a well-organized and modular class structure, offering a systematic and efficient approach to managing inventory-related functionalities. The class design emphasizes encapsulation, inheritance, and polymorphism to ensure a cohesive and manageable codebase. This design ensures scalability, maintainability, and flexibility, adhering to industry-standard practices for robust software development.

1. The Bike Spares Inventory System manages bike spare parts efficiently via core classes: Inventory, User, and Activity Logs. Inventory tracks parts, User handles authentication, and Activity Logs capture system actions for auditing, ensuring a well-structured system.
2. Using .NET MAUI Blazor, the Bike Spares Inventory System applies advanced OOP principles for scalability and maintainability. The class diagram illustrates relationships and methods, promoting code reusability and facilitating the integration of new features to adapt to evolving business needs.

## UI Design

Include images here of your user interface design. You may include both low and high fidelity. Include an introductory paragraph that describes what’s provided.

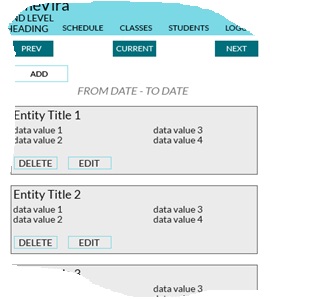
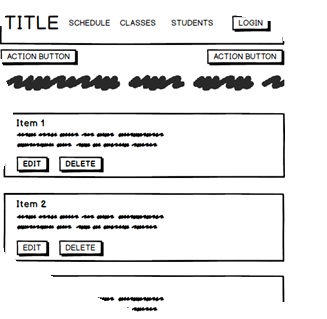


Figure 1: Low Fidelity

Figure 2: High Fidelity

# Unit Test Plan

## Introduction

### Purpose

Provide a brief description of the testing method(s) that you used and what the results it yielded. Also, what remediation was required if necessary and how it would be performed.

### Overview

Here you go into more detail about the test(s) and how it related to the overall project. You should include if a similar method was used in other parts of the application or why this was unique for a certain aspect of the code. Then, go into detail about what functions were tested, how the tests were conducted, and how errors were dealt with.

## Test Plan

### Items

What is required to complete the test(s)?

### Features

List the function/features that are part of each test.

### Deliverables

List what the test(s) would produce. For example, documentation or code notations.

### Tasks

List the tasks required to complete the testing and provide the outcomes you identified.

### Needs

Describe what was needed to be running or what support items had to be in place to perform the test? Specify versions if appropriate and other technical requirements. If a testing package and/or library was employed, be sure to identify it/them.

### Pass/Fail Criteria

Describe the criteria you used to determine the success of the test and what the protocol was for a positive result. Also describe what the recourse was if the test failed including remediation strategies and documentation requirements.

## Specifications

Provide sample code that represents what testing code was used. Screenshots are acceptable.



## Procedures

Provide a detailed list of the steps you used to complete the testing process. Be sure to mention if iterations were/are part of the process used and when pass/fail results were provided.

## Results

Here you will describe and provide examples of the testing results. If you were using a testing package include a screenshot of the interface. Screenshot work best.



*C2. Provide a link to where the web app is hosted with HTML code (if applicable).*

*C3. Provide a link to the GitLab repository of the code indicating the version included in this submission.*

C4. User guide for setting up and running the application for maintenance purposes.

C5.  User guide for running the application from a user perspective

. **User Guide**

## Introduction

Provide a description of the content you’re providing in the User Guide. This guide will include how to install, log into, sign up, and use all of the functions of the application. The steps need to be clearly defined and fully tested so the process works flawlessly for the evaluator.

## Installation and Using the Application

This procedural information should follow the basic rules of such technical references. While some procedures may provide for personal judgment yours should be clear and concise. Here are other rules to remember:

* Provide step-by-step sequences in the correct order.
* Follow the timing and sequencing of the actual operations.
* Provide visual stepping stones by using bullets or labeling steps.
* Strive to be concise. Avoid lengthy paragraphs but include enough detail so false assumptions are not made.
* Use common terms and jargon appropriate for the audience (someone with basic IT background).
* Explain why steps are completed or what they will yield as well as "How to" instructions.
* Test the instructions to ensure they match the actual product.
* Format the material for ease of reading and use graphic aids to clarify point/steps.
* Write in the present tense and the active voice.

## *Login and Signup (An example*)

1. *Click the "Log in" button in the top right corner of the app.*

**

1. *If you already have an account, log in with your account name and password. If you need an account, click on the link below that states “Need an account?”*
2. *If you need to create an account, choose a unique username and password. By default, the password requires at least 6 characters. This function could be changed to address new password requirements.*

## *Classes*

### *Create a New Class*

1. *Once logged in, click on the link at the top labeled “Classes”. This will enable you to create a new class of students.*

**

1. *Click on “+ Add Class”.*

**

1. *Enter a class name and its description. The class name must be unique.*
2. *Click “Add Class” to add the class, otherwise click “Cancel” or outside of the modal to cancel adding the class.*



## *Reports*

1. *To access the reporting feature, from the Schedule module, click on “Generate Report” near the top right of the page.*

**

1. *By default, all events are generated and displayed.*