Course Number: 420-CT2-AS

Course Title: Object-Oriented Programming Concepts

Session: Winter 2022

**Final Project Report**

**My Bikes Factory**

**SUBMITTED**

**TO**

**TEACHER: GLAUBER TORRES**

**BY**

**Emiliano Ruiz Gomez**

**202130592**

**PROJECT DESCRIPTION**

My bikes Factory project is an interesting and important task I accomplish It is based on an application to select a bike of preference. The user creates a username and a password to login, and it is saved in a txt file. Then the next screen will pop up and the user will be able to select and edit the bike of choice. Some of the options are the type of the bike as Mountain or as Road Bike, the color, the model, the manufacturing year. By choosing the type of mountain the only option the user will have is suspension type and for the type of RoadBike it will be the tire type. Moreover, the application will also allow the user to save he created bicycle with every selected type into an XML file. The user will also be able to modify the saved bicycles, remove them from the list box, search for bicycles, sort the by displaying all. Displaying the mountain type or just the RoadBike and finally a clear button to remove everything from the list.

**PROJECT DOCUMENTATION**

**2.1**

MyBikeFactory.Business, MyBikesFactory.Data, and MyBikesFactory.UI are the three layers that make up this project.

The project's business layer is MyBikesFactory.Business. All of the classes that contain and manage the data that the programme will need throughout its execution may be found here. This layer specifies the many options available to the user when designing their bicycle, as well as the information required to sign in, and it provides a structure for the other levels to show and retain the data supplied by the user.

MyBikesFactory.Data is the data layer is made up of data. The user's data is saved in an XML file, which allows the software to readily access previous bicycles made and allows the user to make modifications to these saved bicycles without leaving the application. Simultaneously, this class is in charge of validating the sign-in procedure in order to prevent unwanted modifications to the existing files.

The user interface layer is MyBikesFactory.UI. This layer enables users to alter data without having to go through lengthy procedures. The major purpose of this layer is to provide the user with a simple application that allows them to design or change bicycles while yet providing all of the features they may require. All of the layers come together here to provide the user a seamless experience.

**2.2**

For this project, ten classes were needed:

- Enumerations are handled by three classes in the Enums folder:

o EColor: This Enum manages the many colour options available to the user when personalising their bicycle.

o ESuspensionType: This class Enum manages the various suspension types available to the user. If the bicycle is a mountain bike, it is only eligible.

o ETireType: This Enum is responsible for the tyre kinds that users can assign to the road bicycles they are designing. Only if the bicycle is a road bike is it qualified for this.

- There are two classes in the Interfaces and Delegates folder:

o Delegates/Notify: This class allows our interface to specify the message that will be displayed if the data entered does not meet the established parameters.

o Interfaces/IValidatable: This class allows the software to validate the data that will be used to create the object Bike and to warn the user if one of the parameters is not satisfied.

- Bike class: Regardless of the sort of bicycle the user is modifying, this class contains information that all bicycles will share. This information contains the serial number, model, colour, and year of manufacture.

o MountainBike class: This class is descended from Bike. The user may now add the suspension type to the bike item that they are configuring.

o The RoadBike class is descended from the Bike class. It has the responsibility of adding the tyre type to the Bike object.

Disclaimer: A bicycle can either be a mountain bike or a road bike; this can only be modified in the main form by upgrading the bicycle.

- User class: To avoid issues or defects, this class validates the information The user's login and password are placed in this class so that they can be validated, and the user may access the form where they can configure their bike.

- Validator class: To avoid issues or defects, this class validates the information the user has provided to the programme.

- BikeXMLData class: This class enables the software to store the bicycle that the user has generated as an XML file that authorised users can use, modify, and access at any time.

- UserSequeantialData class: This class enables the programme to access the Login file, which contains the permitted usernames and passwords. The application can only offer access to the main form with this information if the data submitted by the user is correct.

**2.3**

1. Enter your Username and Password to gain access to the main page by pressing on the login button, from where you may explore and register bicycles.

Graphical user interface

Description automatically generated

2. Once you've joined up. The primary form will be given to the user.

Graphical user interface

Description automatically generated

3. Choose a Bike type in the Main form, then provide an id as the serial number, a model, a manufacture year, a color, and, depending on the bike type, a suspension type or a tyre type to that bike.

4.After filling the available options press the “Add” button to add the bicycle.

Graphical user interface, application, Word

Description automatically generated

5. Following the creation of the bicycle, you will be offered many options: To update an existing bicycle, select it, adjust the desired parameter, and then click the "Update" button.

Graphical user interface, application

Description automatically generated

6. If there is a bicycle that needs to be removed, select it and proceed to click the “Remove” button and it will be deleted from the list and the file destined to store it.

Graphical user interface

Description automatically generated

7. We may search for any existing bicycle using the serial number by using the "Search" button.

Graphical user interface

Description automatically generated

8. The "Save" button will save the constructed bicycles to the "Bikes.xml" XML file.

Graphical user interface

Description automatically generated

9. The “Clear” button will erase the constructed bicycles listed in the list box.

Graphical user interface, application

Description automatically generated

10. Closing the programme by clicking the "Exit" button or the "X" in the upper left corner returns the user to the login page.

Graphical user interface

Description automatically generated

11. Use the checkboxes to sort the list and see all of the bicycles that have been built, or all of the bicycles that have been generated.

Graphical user interface

Description automatically generated

**CONCLUSION**

This project has helped me a lot understanding how a user can mess up your application and put me through difficult situations that made me grow and have a better insight into my own project. I am very excited that I was able to create the interface and add some of my own ideas into it. Furthermore, I know this project will really help me creating some other personal projects or even showcase it in my portfolio. And very glad of the result of my project and I am grateful for having such a supportive teacher.

**BIBLIOGRAPHY**

[regex101: build, test, and debug regex](https://regex101.com/)

https://flylib.com/books/en/2.654.1/form\_properties.html