**1-5 Assignment: UML Diagrams**

Southern New Hampshire University

CS-230-J7537 Operating Platforms

Winnie Kwong

Professor Fredericks

May 6th, 2023

**UML Diagram**

Diagram

Description automatically generated

**OOPs Principles in the Class Diagram**

Object-oriented programming (OOP) is a programming paradigm that identifies objects containing data and code. Applying the OOPs principle enhances software development by breaking problems into smaller code units, making the project manageable.

The four major OOPs principles presented in the UML diagram are abstraction, encapsulation, inheritance, and polymorphism. Abstraction hides the complexity from the user and shows what is relevant information. Under the Bicycle class, the accessors are strictly visible to the end users and only provide data when the user calls it. Encapsulation restricts specific components of an object by preventing unauthorized parties to direct access to functions or methods within the class. Encapsulation is shown in the UML diagram by setting gears, cost, weight, and color as private, meaning it will only be accessible within the same class. Inheritance is a relationship between two objects to reuse a code from an existing superclass. In the UML diagram, Bicycle will inherit all the fields and methods of its parent class TwoWheeled, while TwoWheeled inherits all the fields and methods of Vehicle. Lastly, polymorphism uses method overloading to allow a function to do multiple tasks. For example, Bicycle() will be overloaded to utilize Bicycle() if there is no parameter when an object is created or Bicycle(String) if the user passes a string during object creation.

The OOPs principles are designed to improve overall quality and productivity, enhancing the software development experience. Utilizing classes and objects in code enables the reuse of code for future projects and optimizes the development process by reducing the risk of bugs.