# Coursework Report – 5COSC019C Object-Oriented Programming

Student Name: Rathusaigini Thavarajah

Student ID: 20221158/w1953841

|  |  |  |
| --- | --- | --- |
| Have you submitted the video with the demonstration of your system? | Yes | No |

If the video has been submitted specify where:

Blackboard

On the cloud (file shared from Google drive, or OneDrive or DropBox, etc), or YouTube. **Copy here the link: ……….**

**…….**

**Phase 1 – Design and class implementation**

|  |  |  |
| --- | --- | --- |
| **Task** | **Did you attempt the task?** | **Student’s comments** (To which extent you implemented the task? Have you encountered any problems or issue?) |
| Design a UML Use Case Diagram of your system (submitted in a separate file). | Yes  No | Submitted in a separate file. |
| Design a UML Class Diagram of your system (submitted in a separate file). | Yes  No | Submitted in a separate file. |
| Implementation Class Product | Yes  No | Implemented a Class, Product.  The class Product is abstract and includes appropriate get/set methods and holds information about the product ID  product name, number of available items, and price. |
| Implementation Class Electronics | Yes  No | Implemented a subclass, Electronics.  The Electronics subclass holds specific information and methods. add the brand and the warranty period as instance variables, constructors, and the relative get/set methods. |
| Implementation Class Clothing | Yes  No | Implemented a subclass, Clothing.  The Clothing subclass holds specific information and methods. add the size and colour as instance variables, constructors, and the relative get/set methods. |
| Implementation Class User | Yes  No | Implemented a Class, User.  implement a class User to represent the user account. The class holds information about the  username and password, constructors, and the relative get/set methods. |
| Implementation Class Shopping Cart | Yes  No | implement a class Shopping Cart to represent the user’s cart. The class should contain a list of products as instance variables, there should be methods to add, remove, and calculate the total cost |
| Implementation Interface WestminsterShoppingManager | Yes  No | implement a class called WestminsterShoppingManager, which implements the interface ShoppingManager  WestminsterShoppingManager maintains the list of the products in the system and  provides all the methods for the system manager defined in the console menu. |

**Phase 2 – Console menu implementation**

|  |  |  |
| --- | --- | --- |
| **Task** | **Did you attempt the task?** | **Student’s comments** (To which extent you implemented the task? Have you encountered any problems or issue?) |
| Add a product in the system with all the relative information (max 50 doctors) | Yes  No | Add a new product to the system. It should be possible to add either electronics or clothing, with all the relevant information. |
| Delete a product from the system by selecting the product ID. Display a message to confirm it has been removed and the total number of products in the system. | Yes  No | Delete a product from the system, inserting the product ID. Display a message with the information of the product.  (if it is electronics or clothing) that has been deleted and the total number of products left in the system. |
| Print on the screen the list of the products in the system with all the relative information. The list should be ordered alphabetically. | Yes  No | Print the list of the products in the system. For each product, print on the screen all the information and say if it is electronics or clothing. The list should be ordered alphabetically according to the product ID. |
| Save in a file entered by the user so far. The user should be able to load back the information running a new instance of the application. | Yes  No | Save in a file the list of products that have been added to the system, with all the relevant contributions. The next time the application starts, it should be able to read back all the information saved in the file and continue to use the  System. |

**Phase 3 – GUI Implementation**

|  |  |  |
| --- | --- | --- |
| **Task** | **Did you attempt the task?** | **Student’s comments** (To which extent you implemented the task? Have you encountered any problems or issue?) |
| The user can select the category through the drop-down menu | Yes  No | The user can select from a drop-down menu which type of product can be visualised (all, Electronics, or Clothes) |
| The GUI is open and a list of products with the information as per specification has been displayed | Yes  No | The GUI is open and a list of products with the information as per specification has been displayed |
| Items with low availability are highlighted in red | Yes  No | - |
| The user can select a product and all the details are displayed as per specification in the below panel | Yes  No | When the user selects the product, the product details (all the information related to the product) should appear in a panel below the table. |
| The user can add products to the shopping cart and all the information are displayed in a separate frame | Yes  No | The user can add the item to the shopping cart by clicking the relative button. The user can visualize  the shopping cart by clicking the “Shopping Cart” button. The user can select another item and keep  adding items to the shopping cart. |
| The final price is displayed correctly | Yes  No | - |
| The discounts, if applicable, are displayed as per specification and the final price updated accordingly | Yes  No | - |

**Phase 4 – Testing and system validation.**

|  |  |  |
| --- | --- | --- |
| **Task** | **Did you attempt the task?** | **Student’s comments** (To which extent you implemented the task? Have you encountered any problems or issue?) |
| Test plan. (Submitted in a separate file). | Yes  No | Submitted a separate file. |
| Implementation of an automated unit test for each scenario in the console menu. | Yes  No | Class, WestminsterShoppingManagerTest.java is submitted along with the code. |
| Error Handling across all the code, input validation and code quality. | Yes  No | I’ve included validations and exceptions wherever required. |