The University of Texas at El Paso
Department of Computer Science
CS 3331 – Advanced Object-Oriented Programming
Instructor: Dr. Bhanukiran Gurijala
Spring 2024

Project Part 1

Academic Integrity Statement:

This work is to be done as a team. It is not permitted to share, reproduce, or alter any part of this assignment for any purpose. Students are not permitted to share code, upload this assignment online in any form, or view/receive/modify code written from anyone else. This assignment is part of an academic course at The University of Texas at El Paso and a grade will be assigned for the work produced individually by the student.

Instructions:

Your code must be written in Java. In the comment heading of your source code, you should write your name(s), date, course, instructor, programming assignment 1, lab description, and honesty statement. The honesty statement must state that you completed this work entirely on your own without any outside sources including peers, experts, online sources, or the like. Only assistance from the instructor, TA, or IA will be permitted. Generate Javadoc for your complete code.

Scenario:

You will be part of the creation of a new car dealership Mine Cars. Your dealership will offer both brand new and used cars. The system shall provide a wide variety of different models with varying prices and mileage to satisfy customers. Customers will have a budget to purchase cars and can also opt for a membership to receive discounts or a better interest rate. Additionally, your system will ensure security by allowing users and managers to sign in and gather all their information from the database.

Part A:

Read the requirements described in Part B to complete Part A. Part A must be completed before implementing the requirements in Part B

1. Write a UML Use Case Diagram (Level I) for your system. With at least the following:
    a. 2 actors
    b. 3 Use Cases
2. Write a UML Class Diagram to structure your code using the classes, requirements, and concepts described in Part B

Part B

1. Create the following classes (Note: some may be abstract)
    a. User
    b. Person
    c. Admin
    d. Car
        1 . Sedan
        2 . SUV
        3 . Hatchback
        4 . Pickup
    e. Log
    f. RunShop (Where you have your Main Method)
    g. Any other additional Classes that you believe will be beneficial to help with successfully implementing this program.
    h. All Classes should have appropriate methods.
        ii. All Classes should have appropriate fields.
2. Read files with information and store the information appropriately.
    a. Pick an appropriate data structure.
        i. Consider the time complexity.
        ii. Consider space complexity.
    b. Consider the use of objects and how your objects will interact with each other.
3. Your system should be menu-driven with the following options:
    a. Login
        i User
            1 Display all the cars.
            2 Filter by Used or New

3 Buy
a Check if the user can buy it.
b Give a ticket with car information.
i Tickets should include car type, model, year, and color.

ii Admin

b. Exit

Example :

User's perspective:

Username: *******
Password: ********

Welcome (username)

User's perspective:

1. Display all cars.
2. Filter Cars (used / new)
   1)New
   (Print info)
   2)Used
   (Print info)
   3)Go back
3. Purchase a car
4. View Tickets
5. Sign out

4. Your system should be able to handle the following:
   a. Read a User file to verify that customers are on their profile and their information
   b. Customers will be able to browse all the cars with their characteristics.

    c. Purchase a car **Only** if their cars are available and if the users have the budget.
    d. You should be able to update the user's CSV with the new user budget
    e. You should be able to update the car's CSV with the new quantity of cars available.
    f. Log all activity being done on the system.
        i. User logs
        ii. User actions
        iii. Examples of log messages.
            1. "10:00:10 10/10/23 - User Name logged in
            2. "11:57:10 10/13/23 - User Name print all cars
            3. "12:01:10 10/13/23 - User Name filtered cars by brand

5. The user can exit the program by writing "EXIT" or creating an Exit value while in the main menu only. When the user exits the program
    a. Write a new updated Users sheet (similar to the original input, except with the new values such new budget)
    b. Write a new updated Cars sheet (similar to the original input, except with the new quantity of cars available)
6. Handle all exceptions appropriately. (Your system should not crash)
7. Write Javadoc for your system.
8. Write a lab report describing your work (template provided)
    a. Any assumptions made should be precisely mentioned in the source code and described in the lab report.
    b. The lab report should contain sample screenshots of the program being run in different circumstances, including successful and failing changes.
9. Schedule a demo with TA/IA
10. **If submission is past the deadline** Your report must have an additional section entitled "Why I submitted late". In that section, explain the reason why your submission was late. (Note: you will still be penalized the typical late penalty)

Deadlines:

March 29, 2024, by 11:59 pm:

1. UML Class Diagram Progress (.pdf)
2. UML Use Case Diagram Progress (.pdf)
3. Current Progress Source Code and files (.java/.txt/.csv) – Commit current progress up to this point.

For each item (1-3)

a. Does not have to be complete. Ensure there are no compilation errors.
b. There should be a significant amount of work done (as determined by the instructional team).
c. TA/IA will review for progress only.

April 5, 2024, by 11:59 pm:

1. UML Class Diagram (.pdf)
2. UML Use Case Diagram (.pdf)
3. Source code (.java files)
4. Lab report (.pdf file)
5. Javadoc (entire doc folder)
6. Updated Users Sheet (.csv)
7. Updated Cars sheet (.csv)
8. Log (.txt)