Chad Galloway

CST-250 Programming in C# II

Grand Canyon University

Oct. 26, 2025

Activity 1

Files

<https://github.com/CGalloway3/CST-250-Projects/tree/master/Activity%201>

Video

<https://www.loom.com/share/0b708cd1a5ec482eac7e5269918f2441>

**Part 1**

FLOW CHART

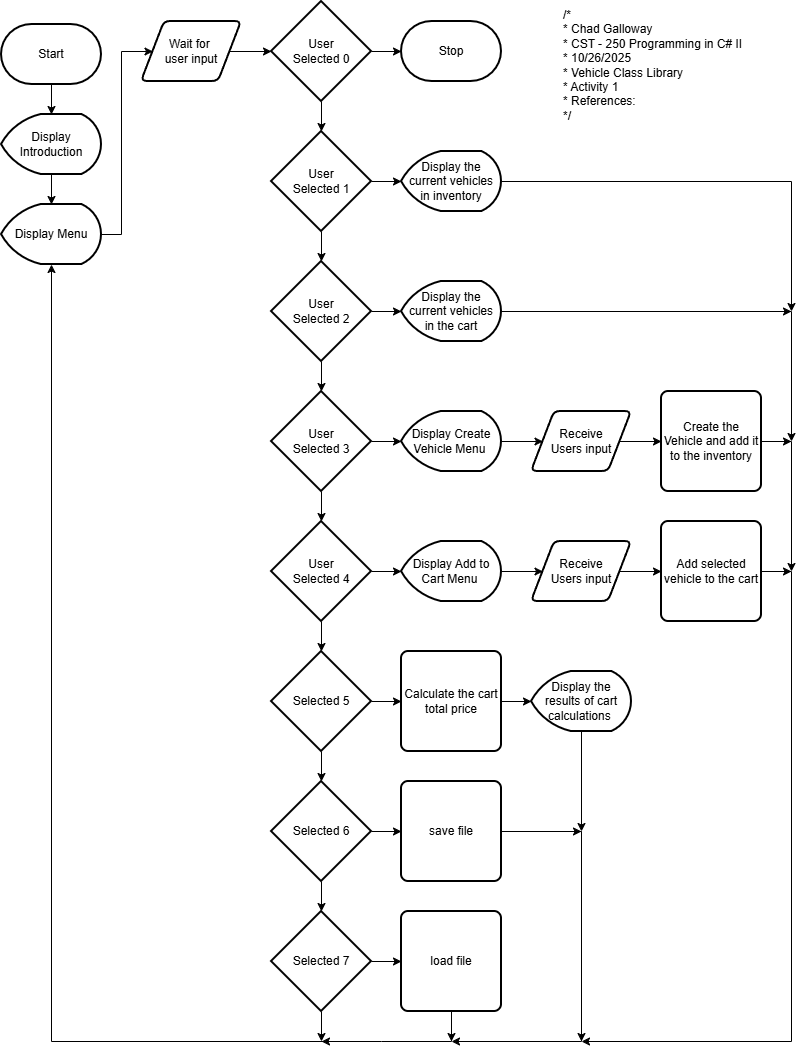


Figure X: Flow chart of Activity 1

UML Class Diagram

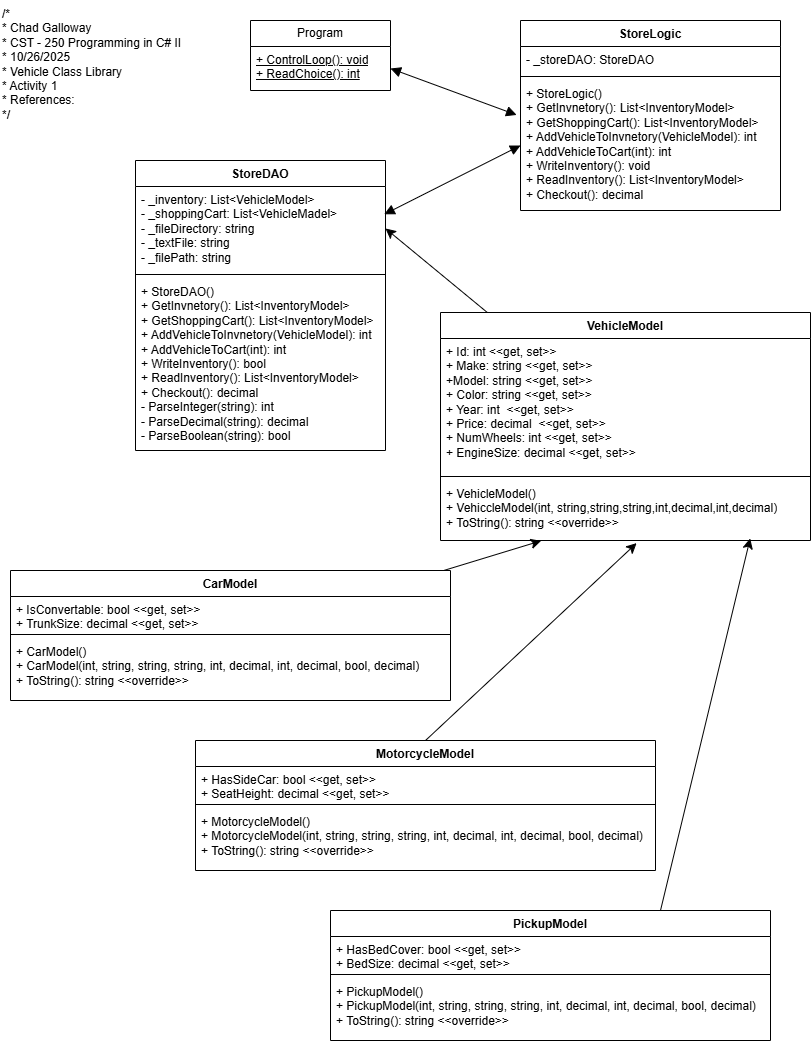


Figure X: UML Class Diagram

Screen Shots

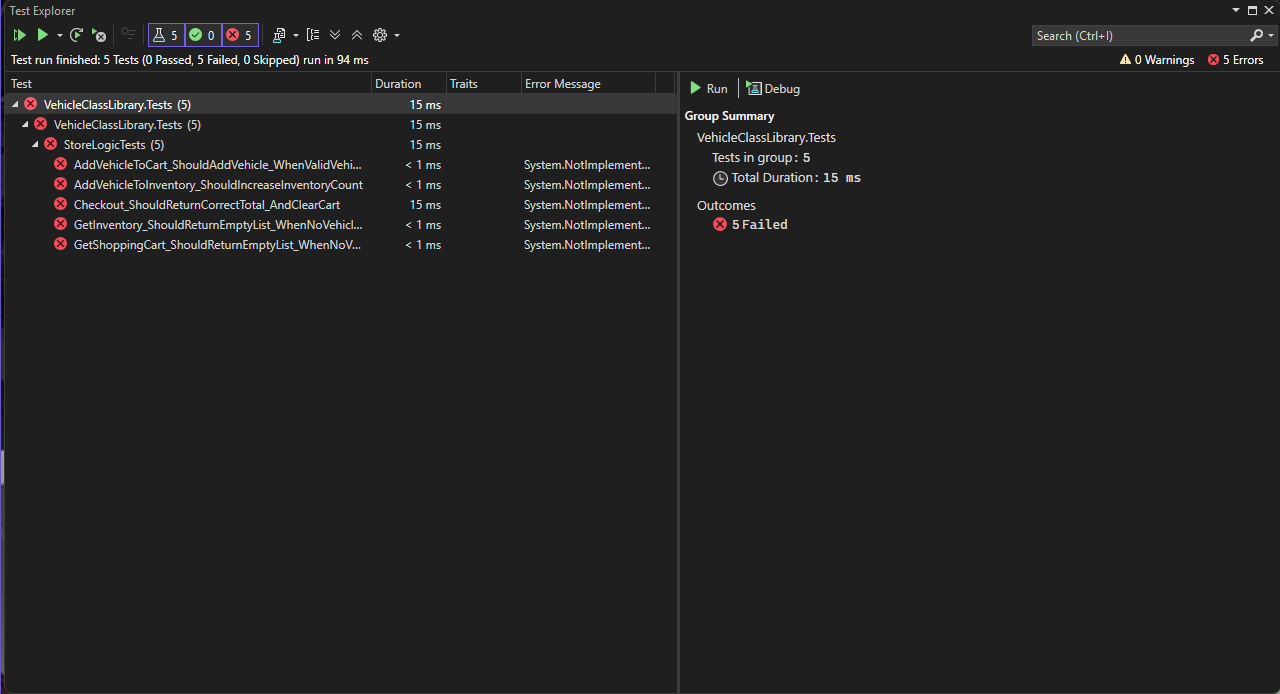


Figure 3: Screenshot of xUnit test

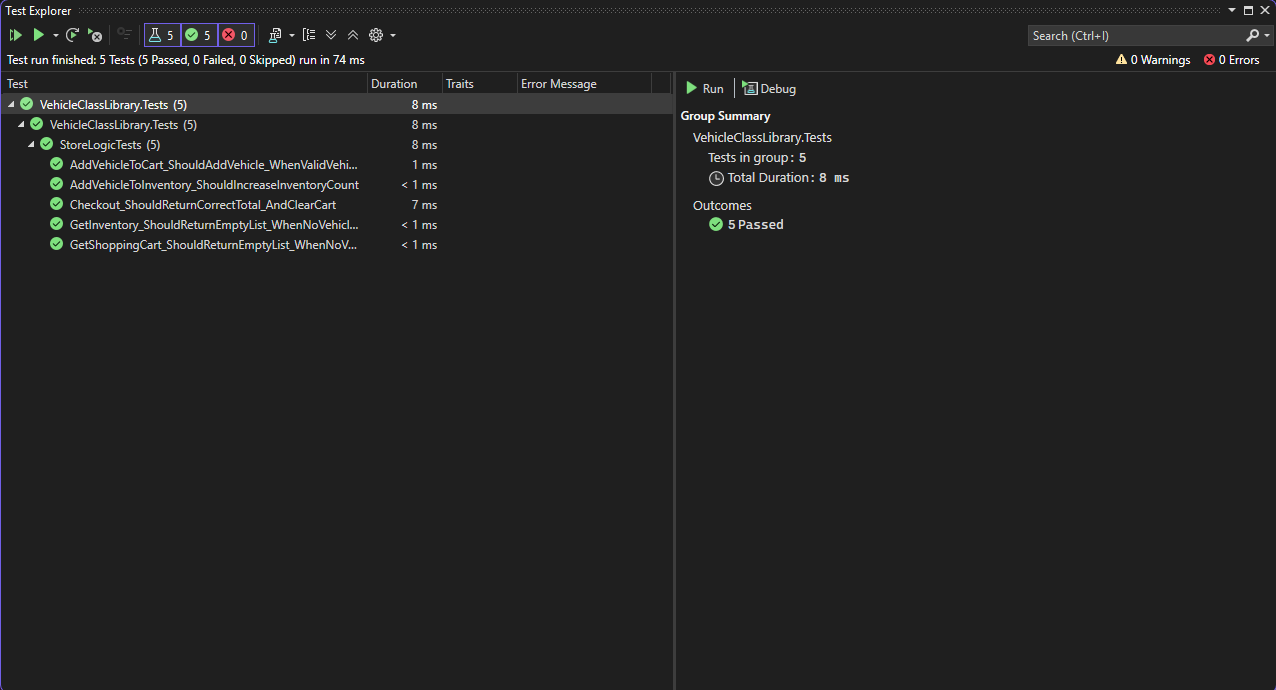


Figure 4: Screenshot of unit test

Figure 3 and 4 are screenshots of the unit tests. 3 is unsuccessful and 4 is successful completion of the tests.

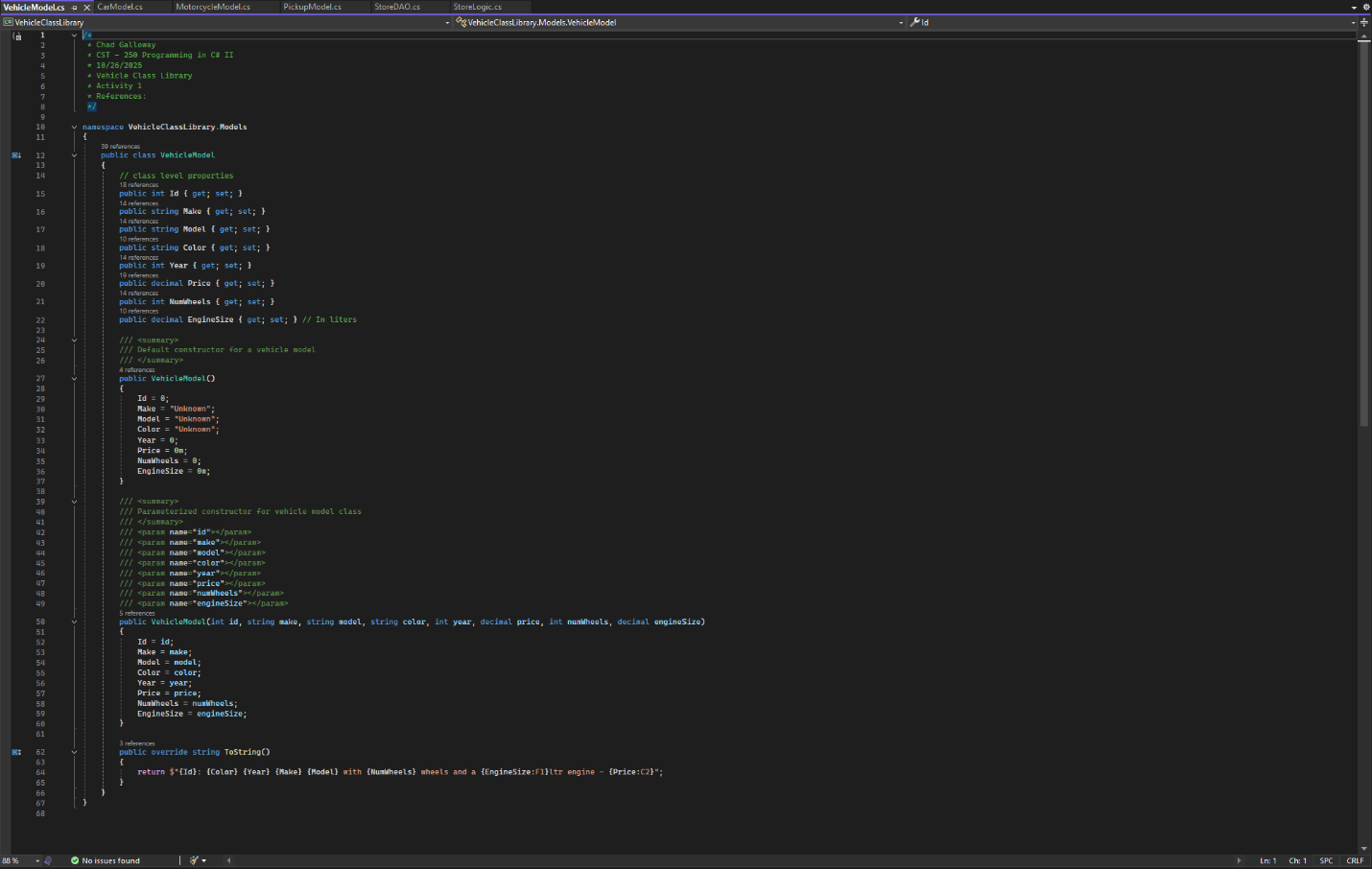


Figure 5: Vehicle Model

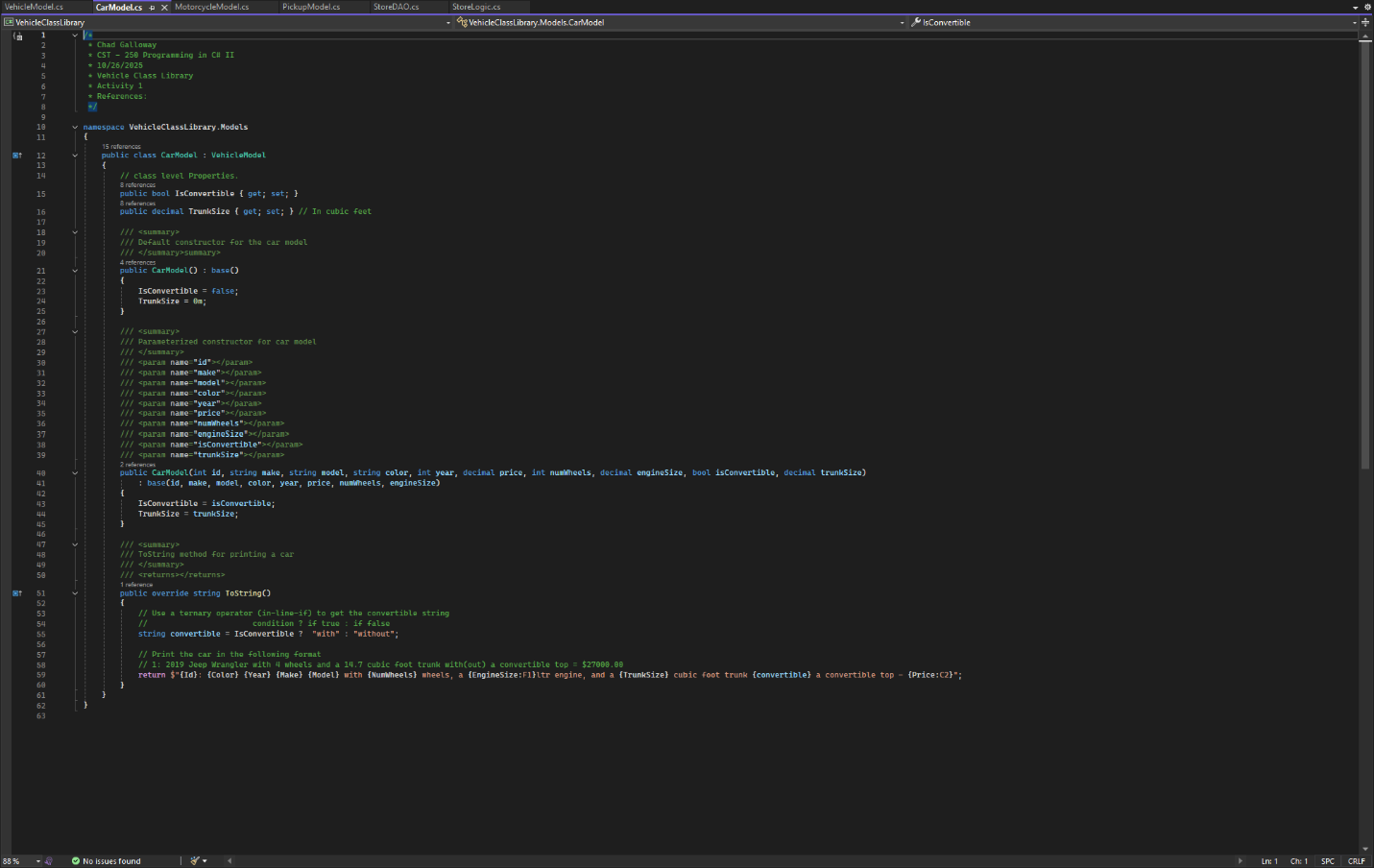


Figure 6: Car Model

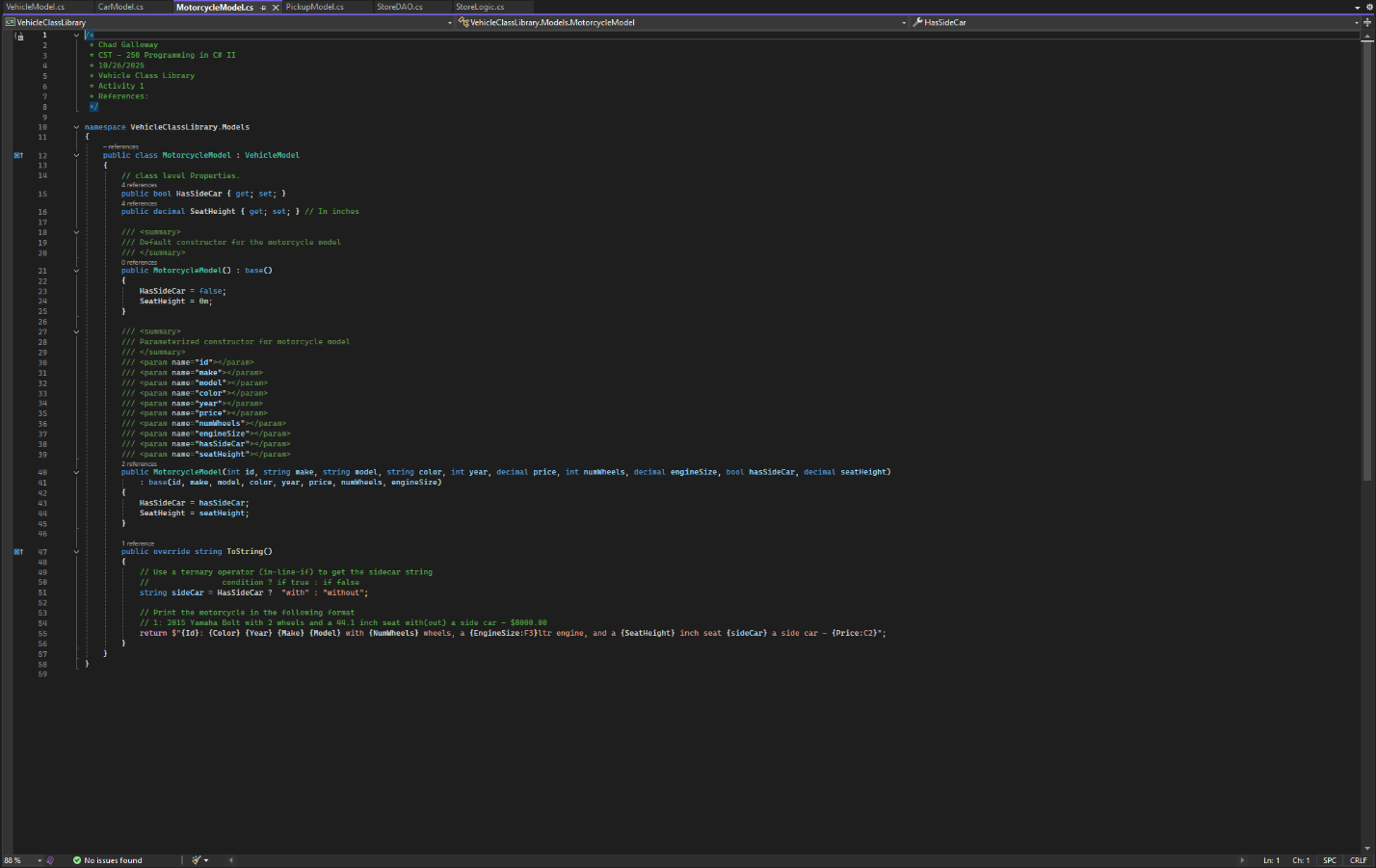


Figure 7: Motorcycle Model

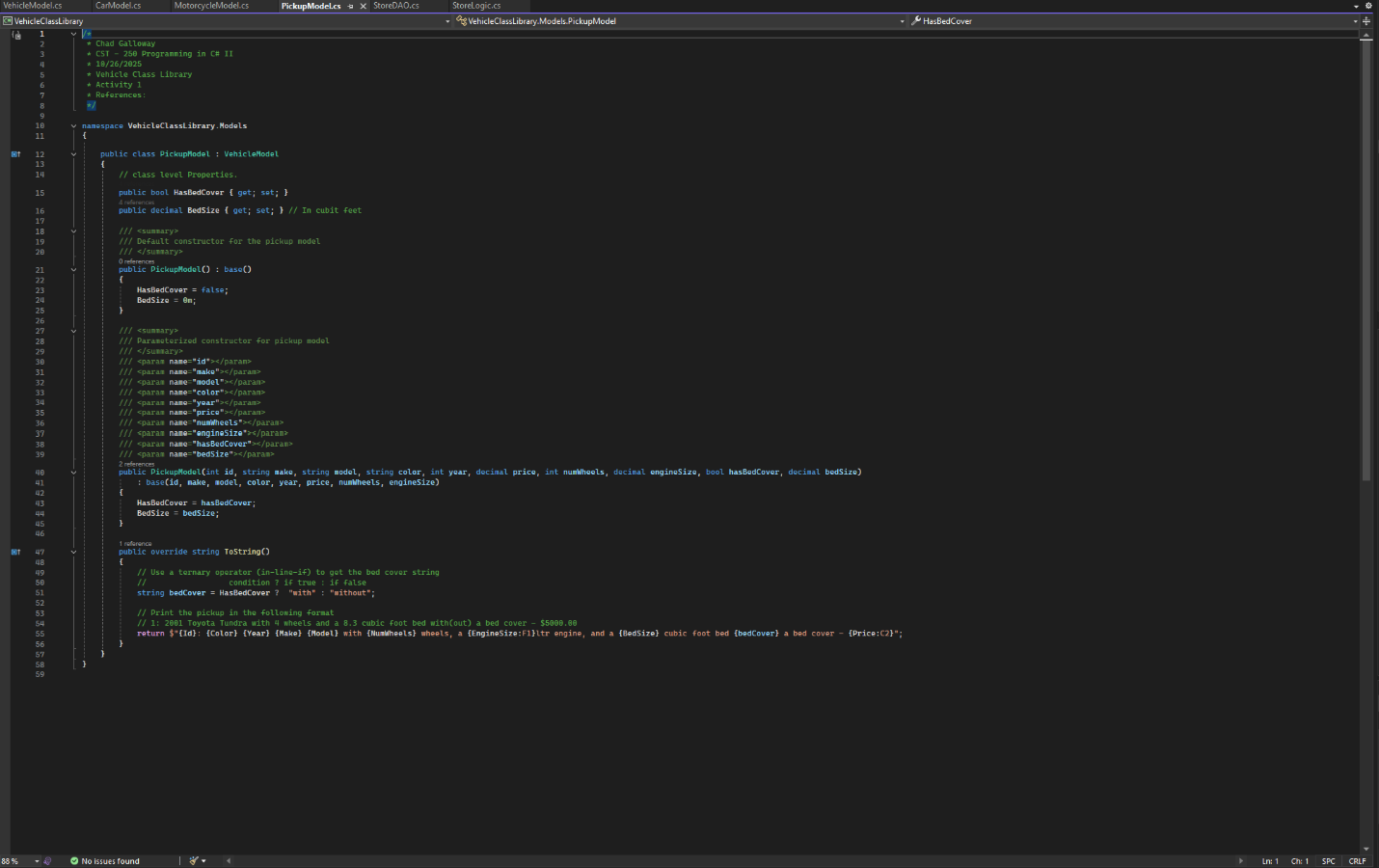


Figure 8: pickup model

Figures 5, 6, 7, and 8 are screen shots of the different models for vehicles in the application. Each screenshot covers the citations, constructors, and the overridden ToString() methods.

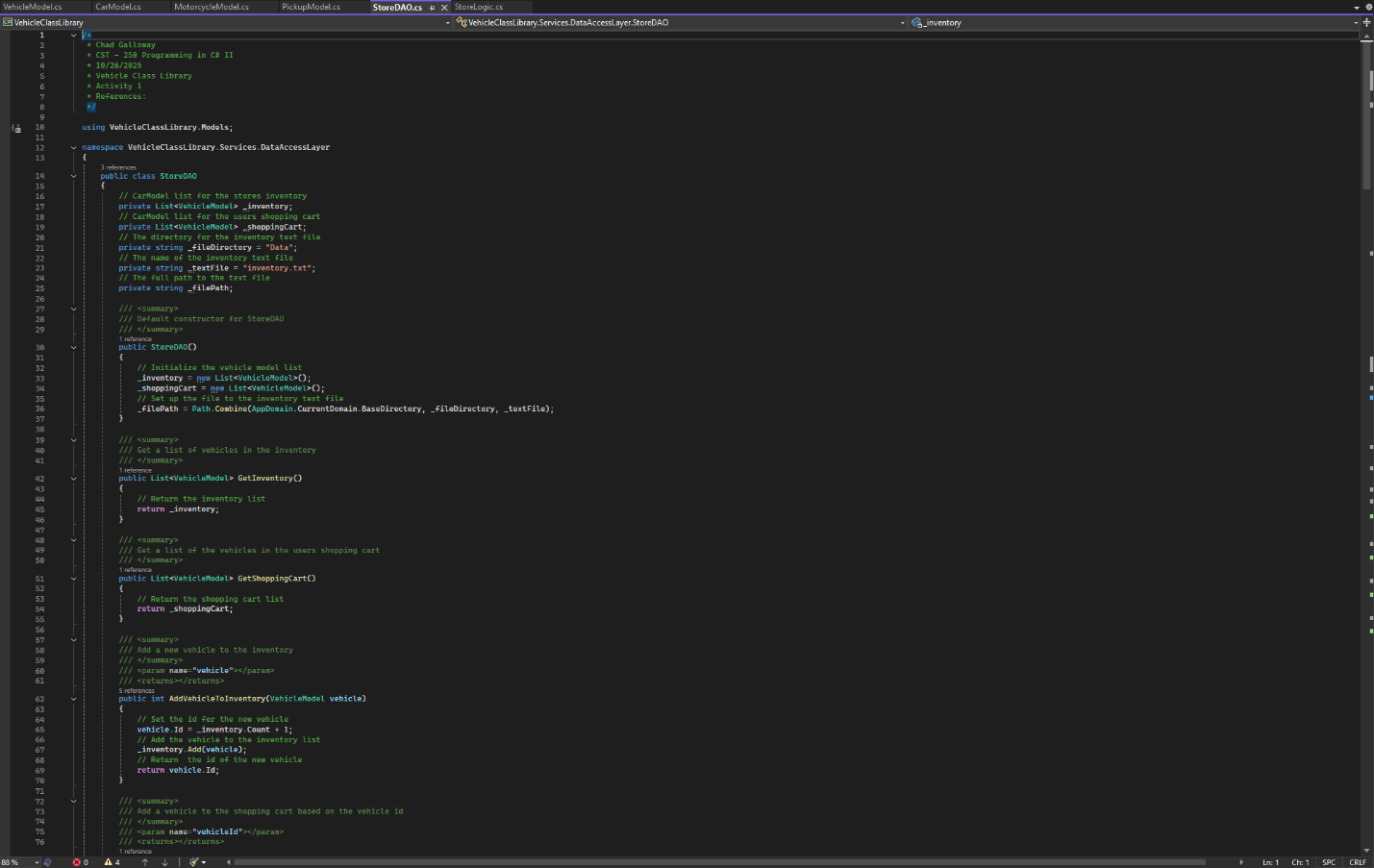


Figure 9: StoreDAO citations and constructors.

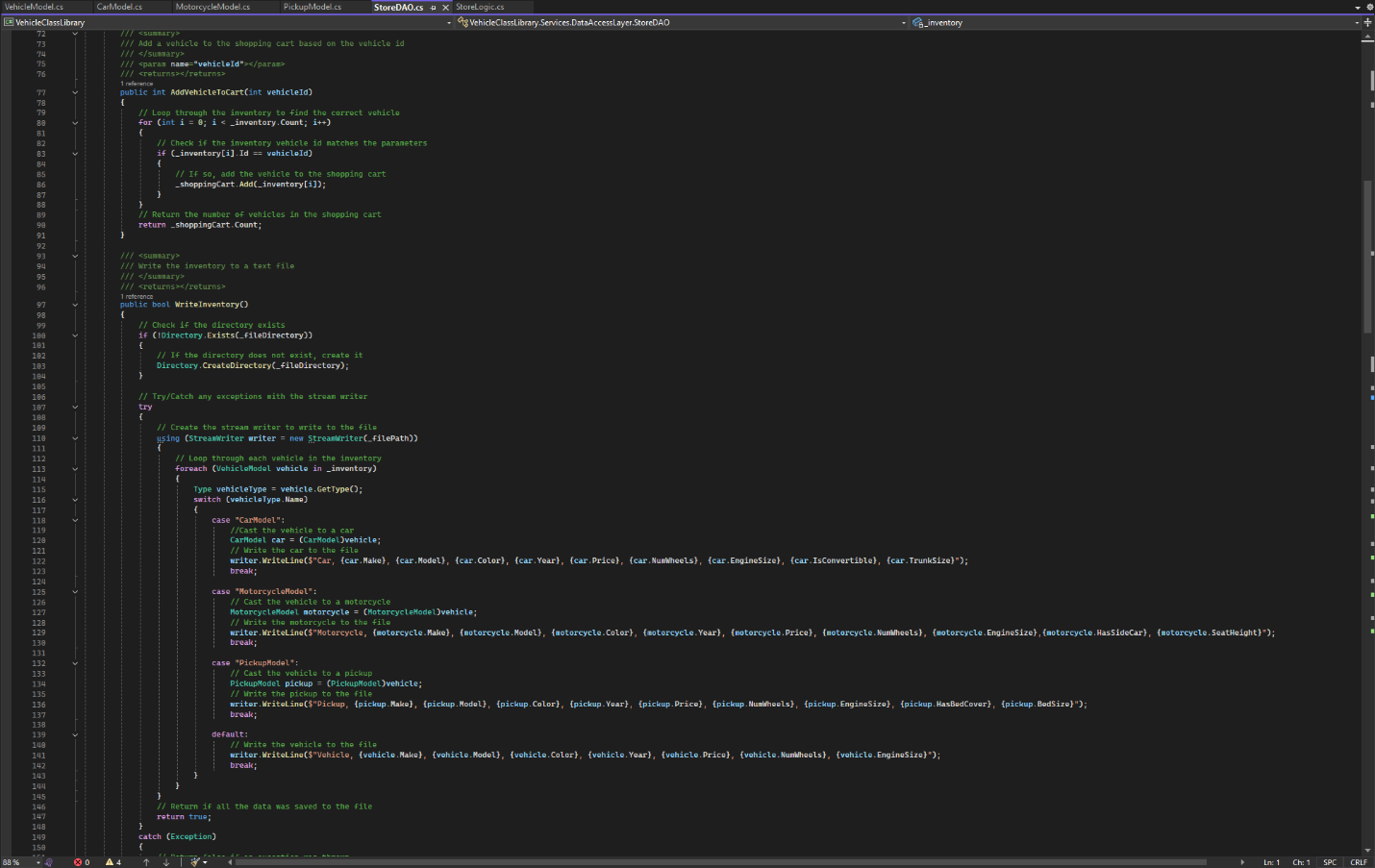


Figure 10: continuation of StoreDAO

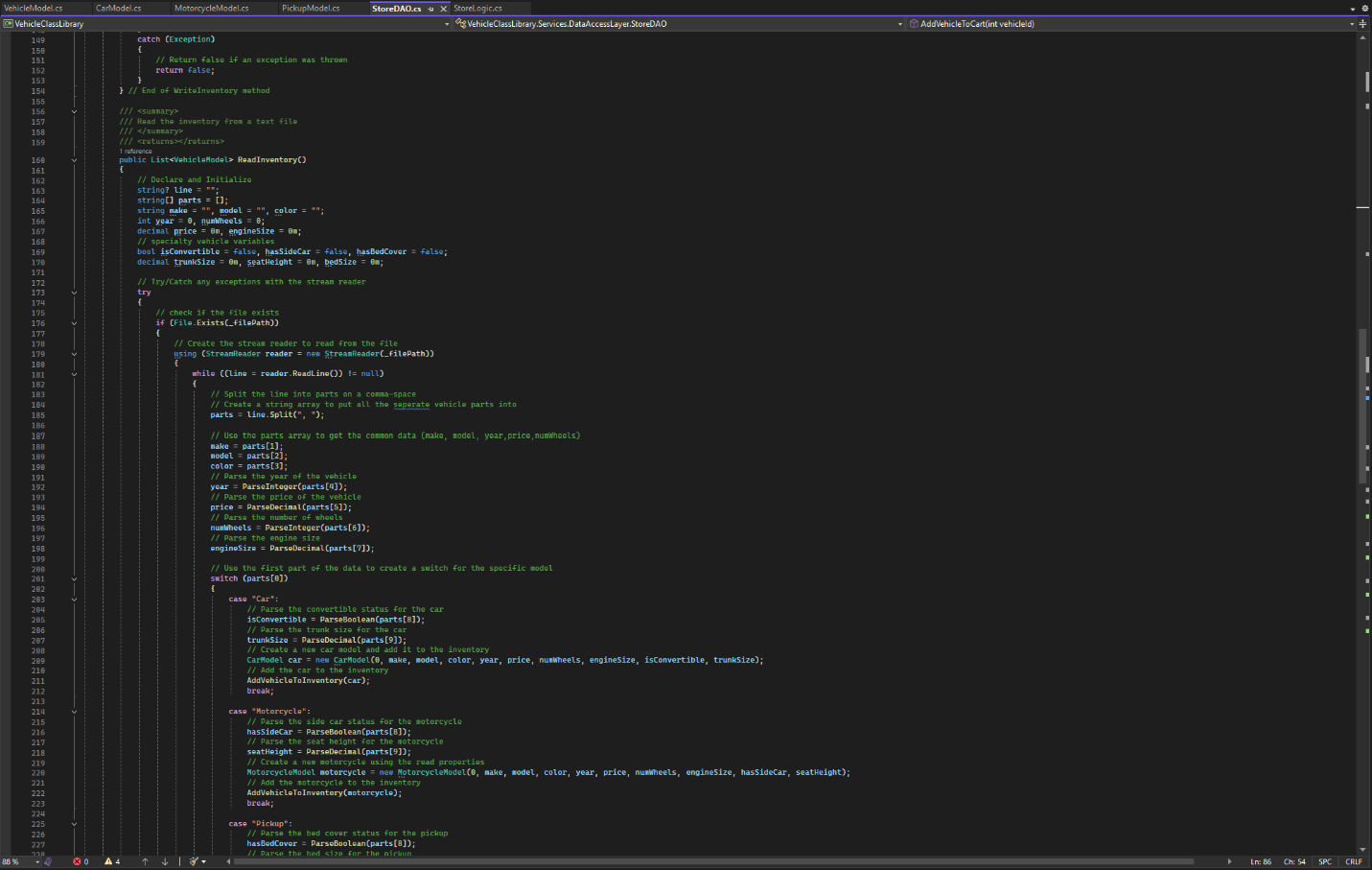


Figure 11: continuation of StoreDAO

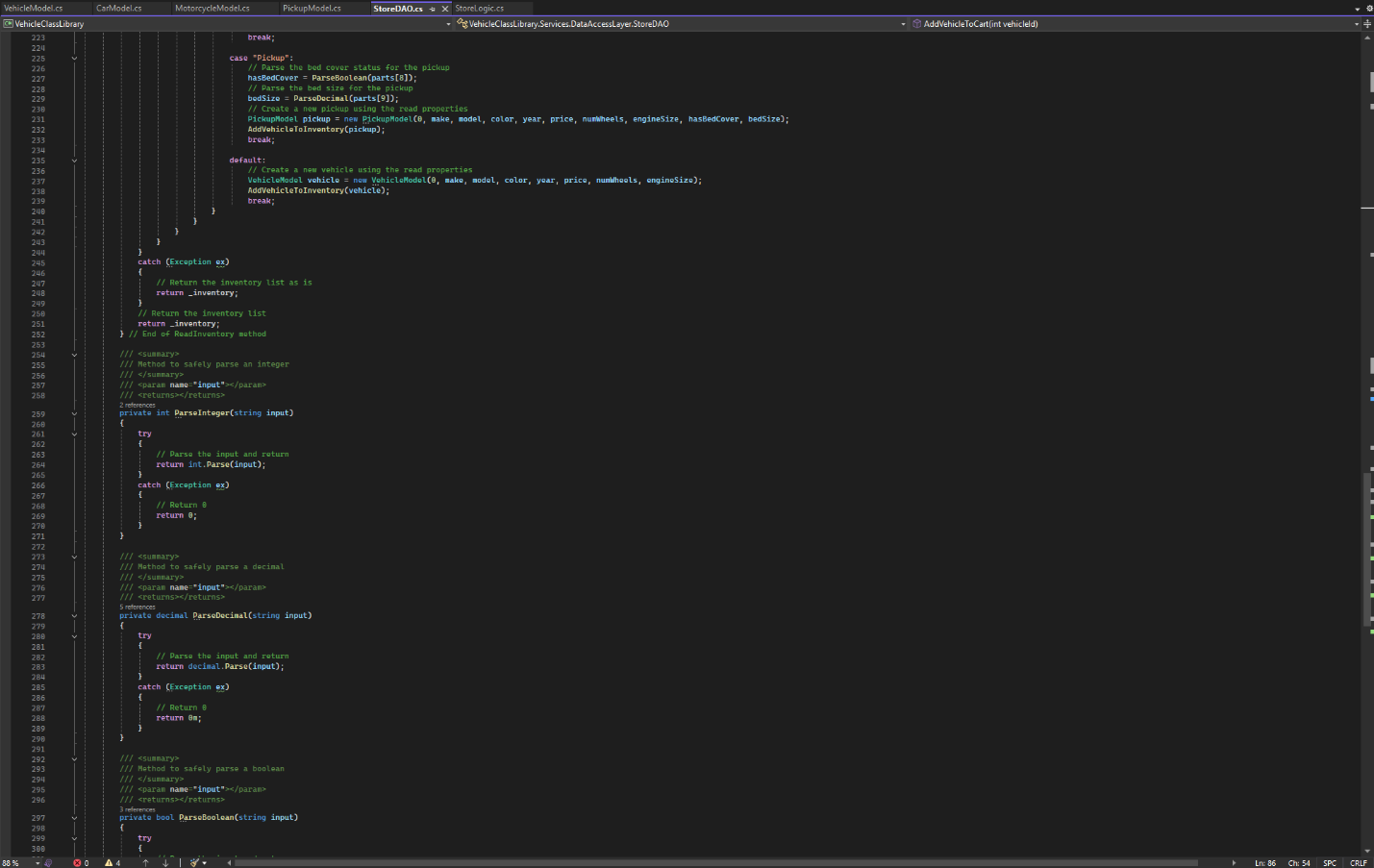


Figure 12: continuation of StoreDAO

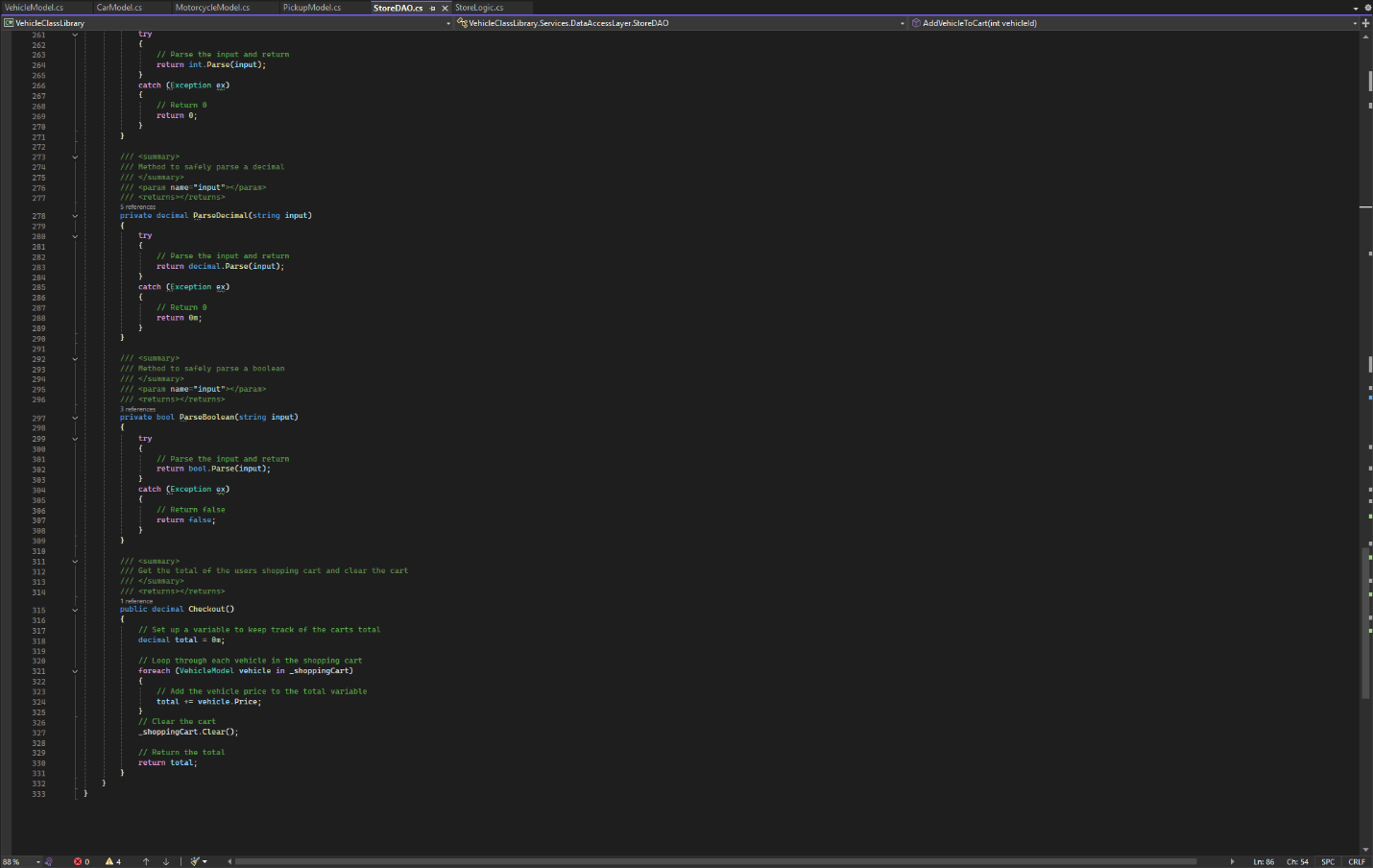


Figure 13: Final screenshot of StoreDAO

Figures 9-13 are screen shots of the StoreDAO class in its entirety. Every method is represented in the screen shots.

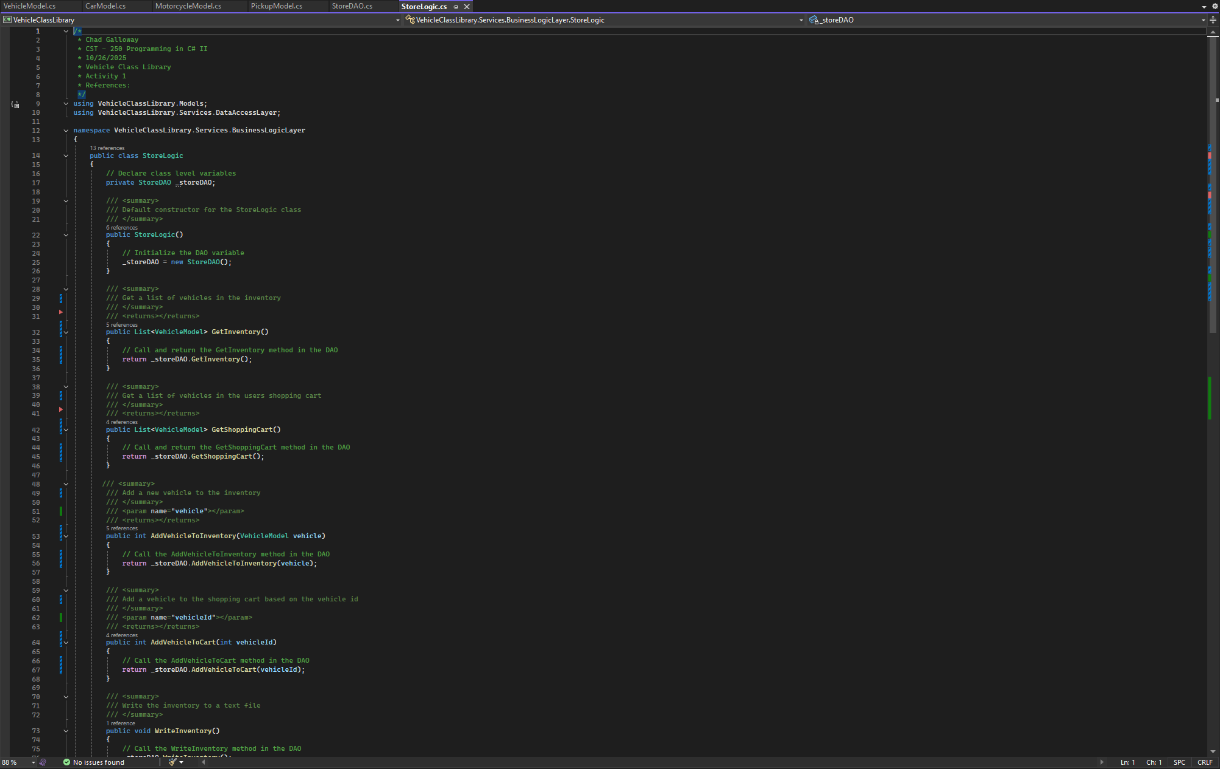


Figure 14: StoreLogic class citations and constructor

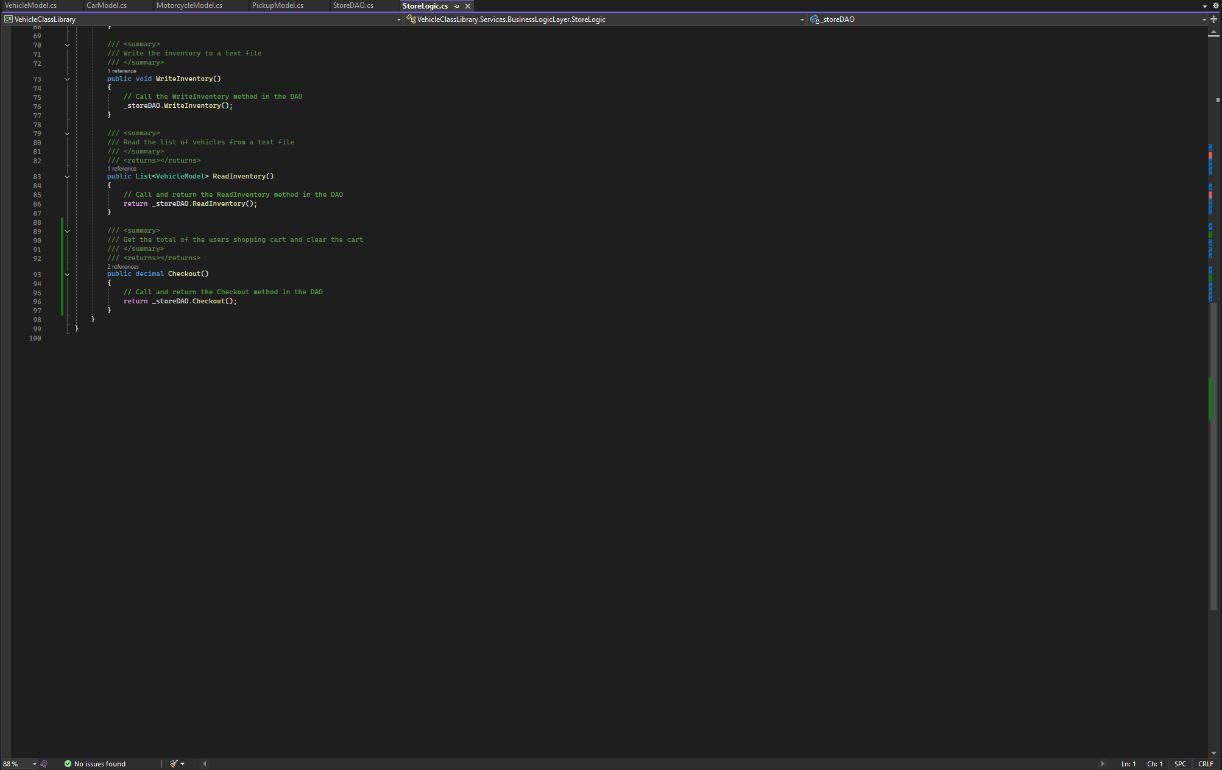


Figure 15: StoreLogic methods

Figure 14 and 15 are screenshots of the StoreLogic class.

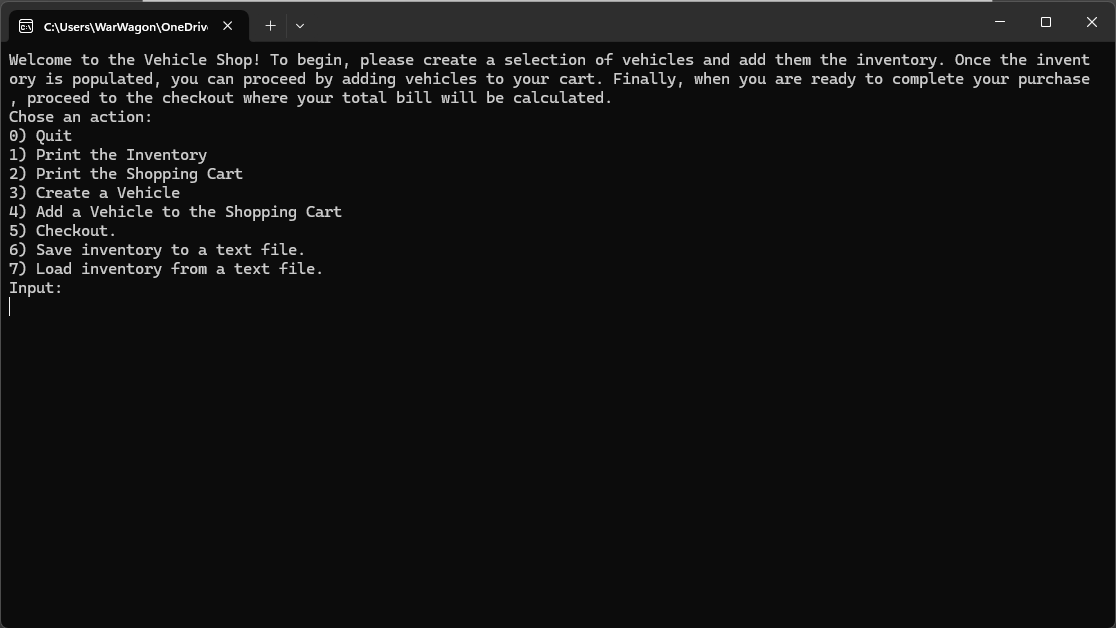
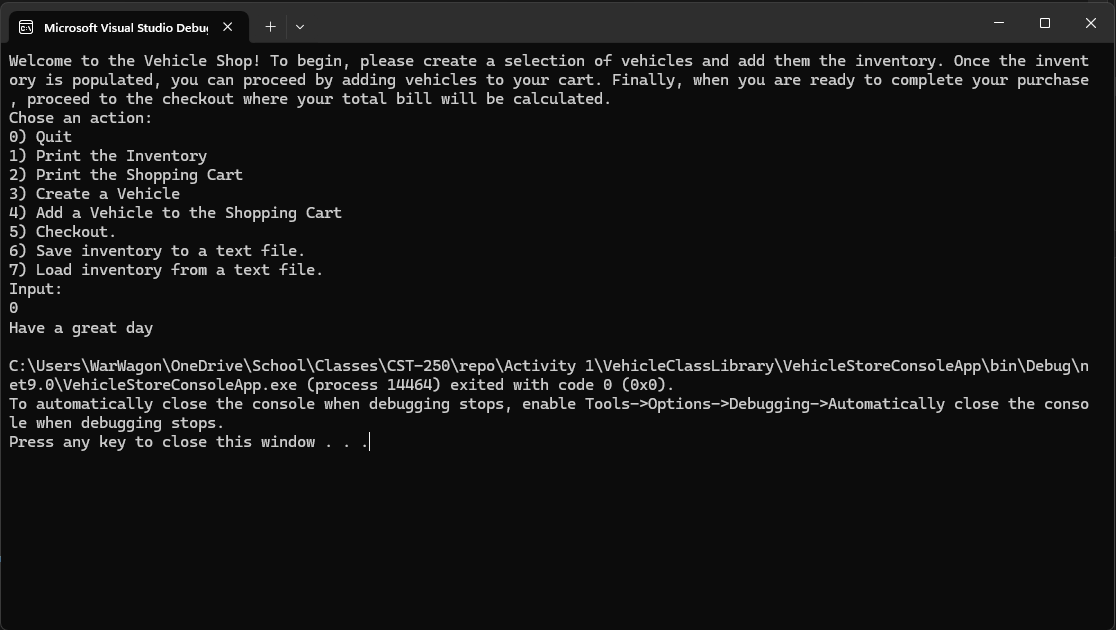


Figure 16: Initial load state of the console app

Figure 17: user input 0

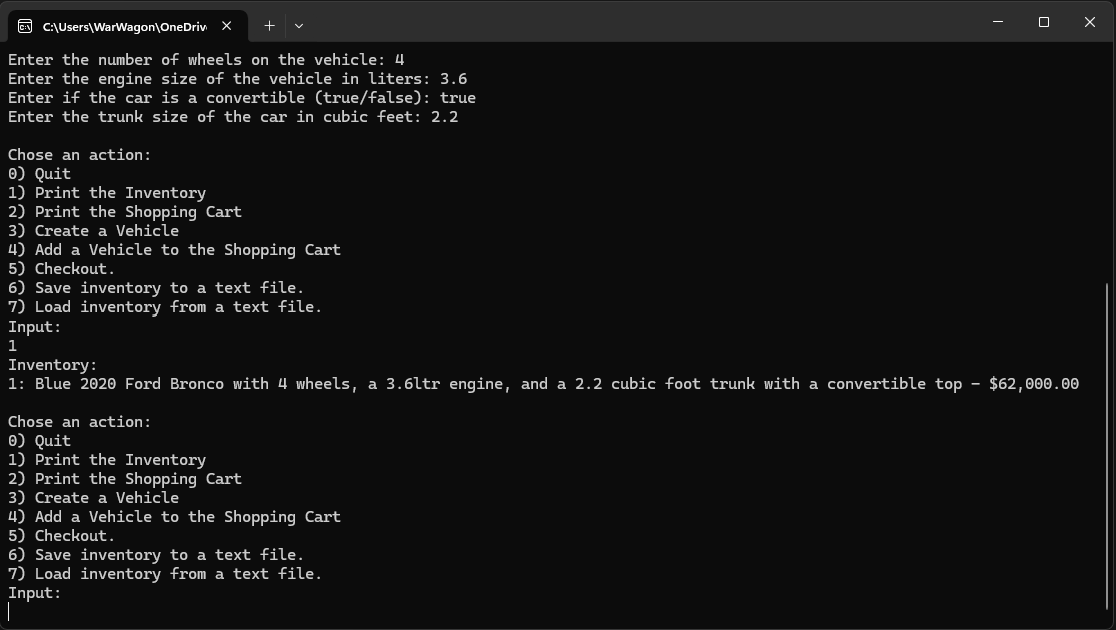


Figure 18: user input 1

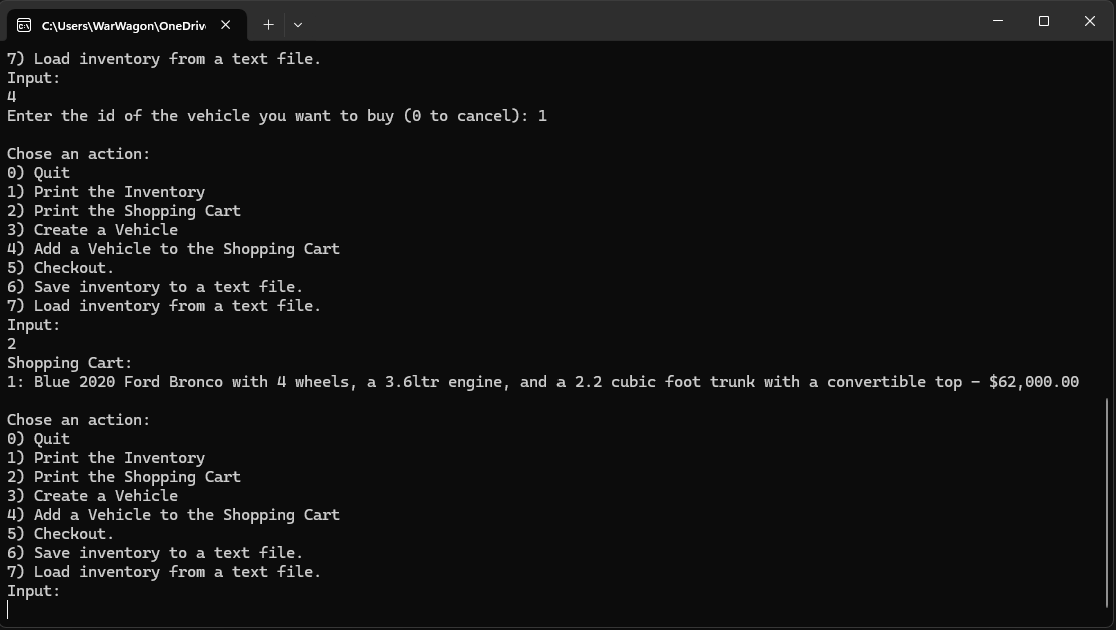


Figure 19: user input 2

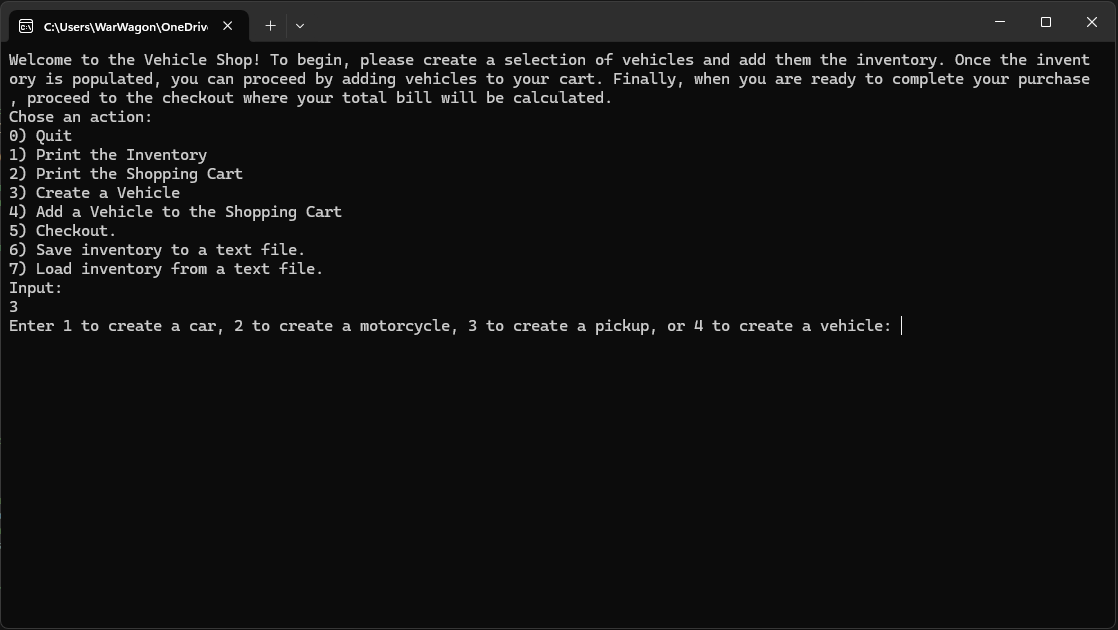


Figure 20: user input 3

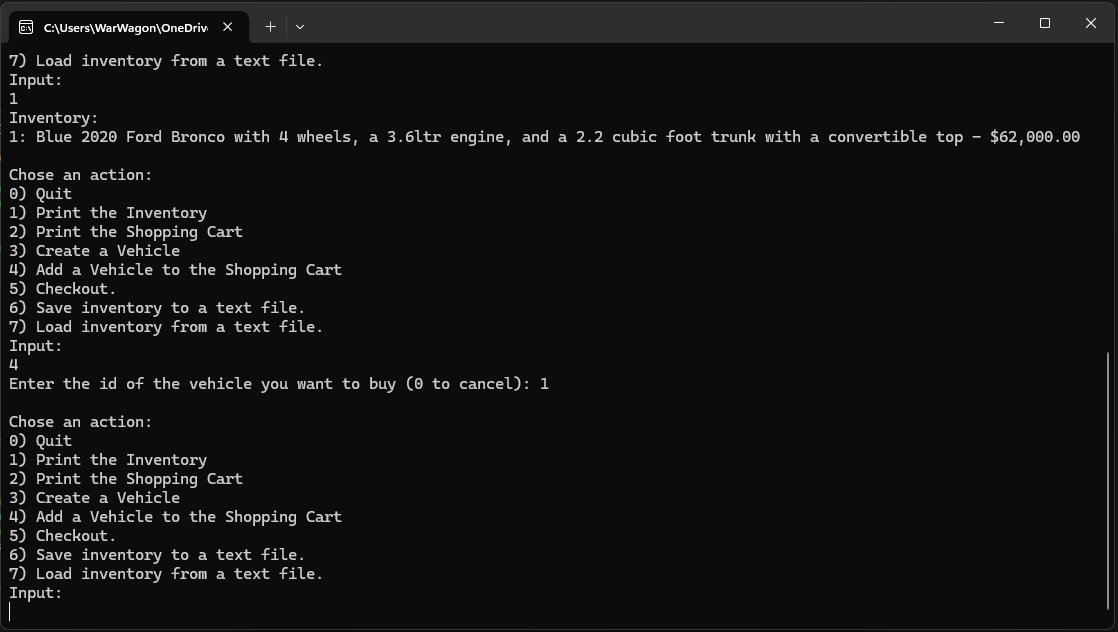


Figure 21: user input 4

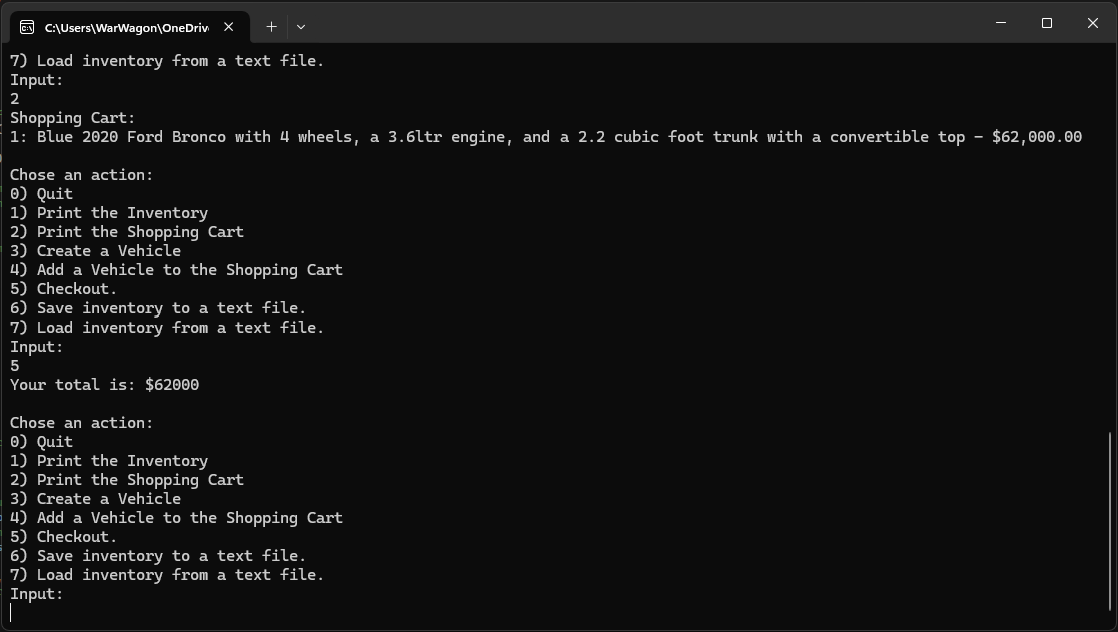


Figure 22: user input 5

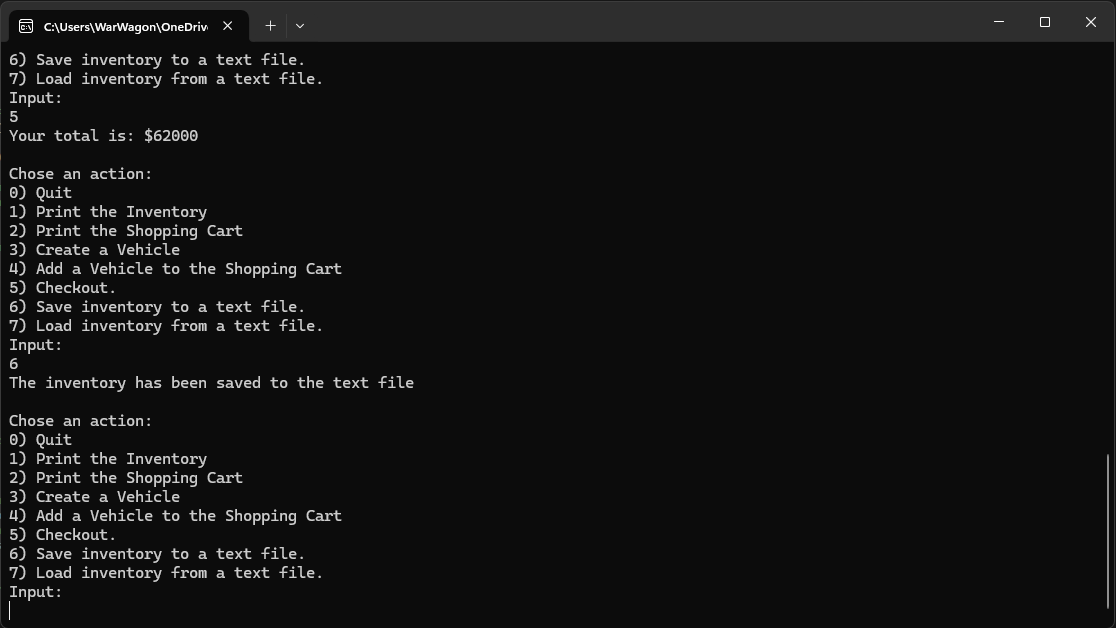


Figure 23: user input 6

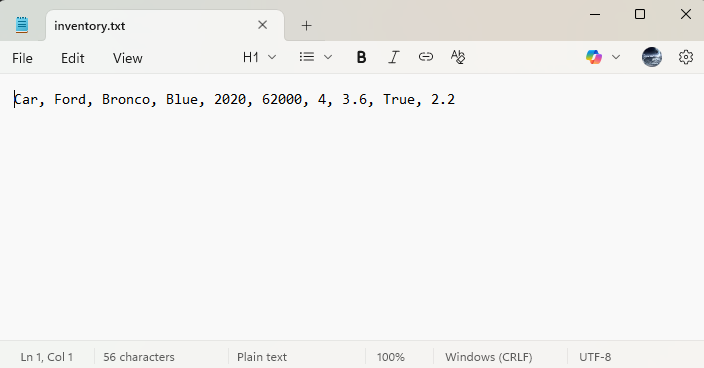


Figure 24: File contents after save (number 6)

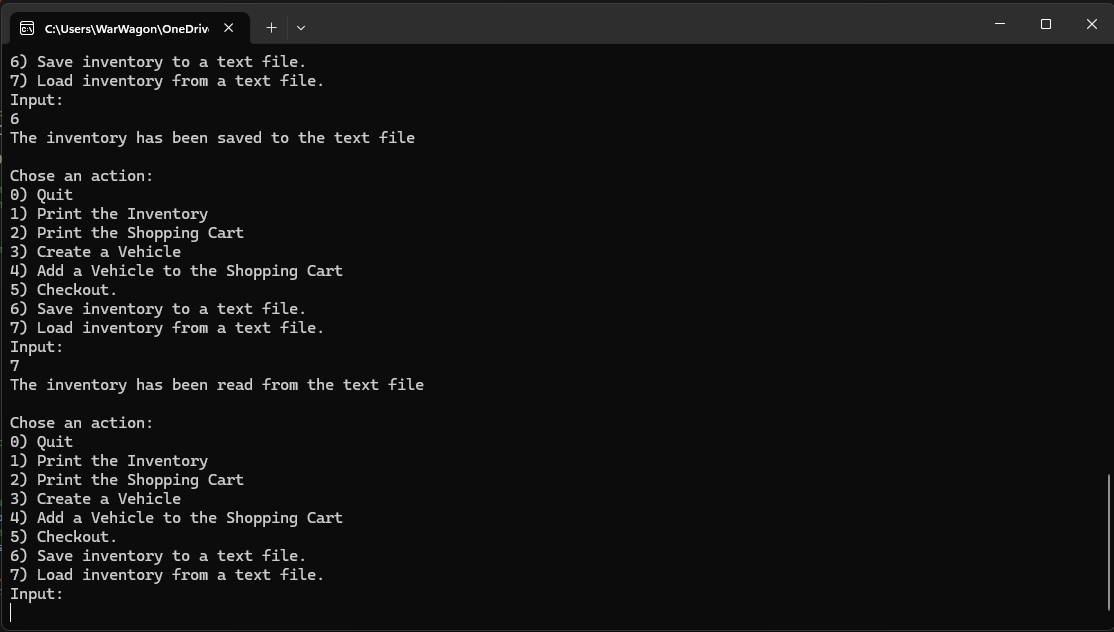
Figure 25: user input 7

Figure 16- 25 are screenshots of the application running and demonstrating the different menu option selections made by the user

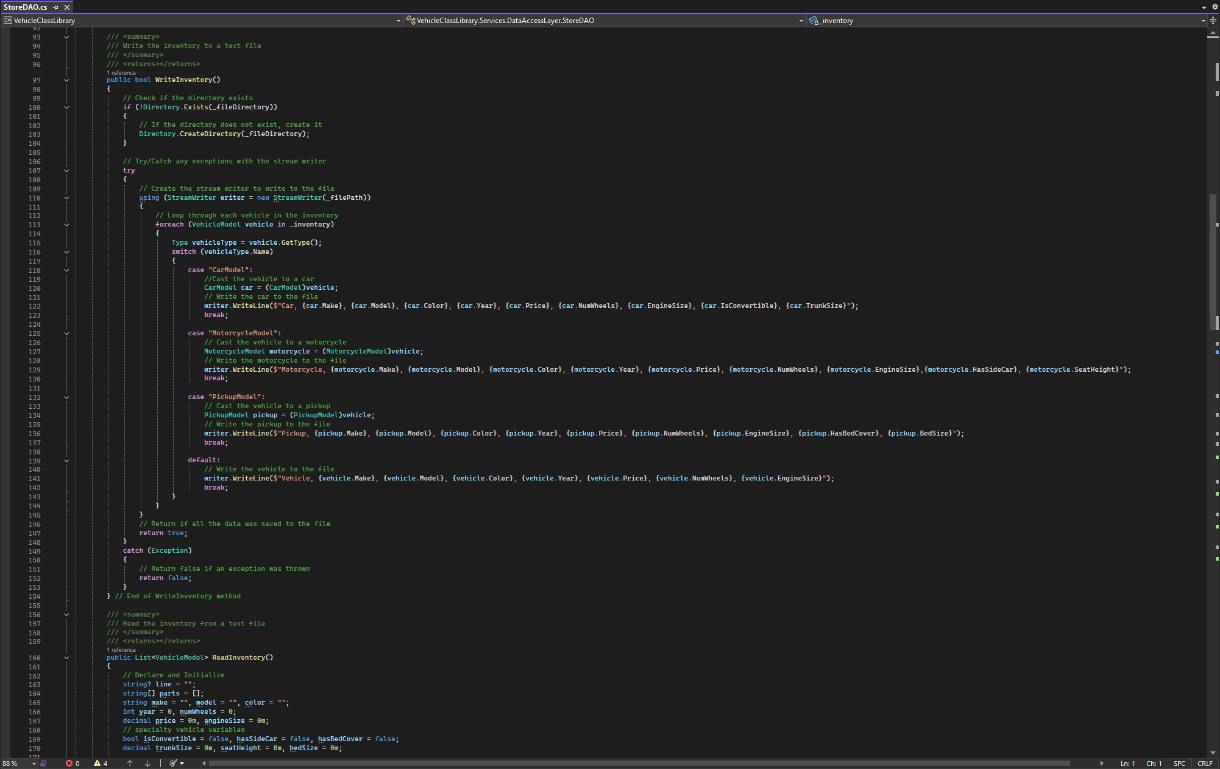


Figure 26: Challenge Changes

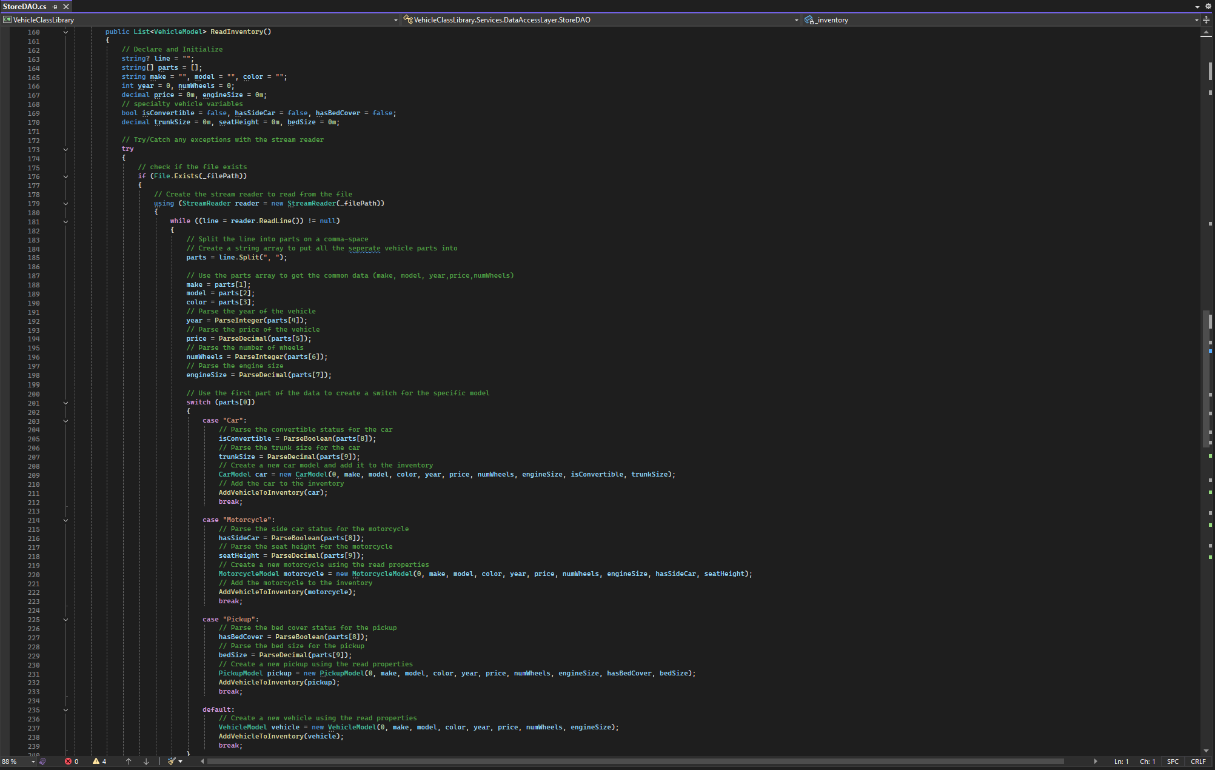


Figure 27 Challenge changes

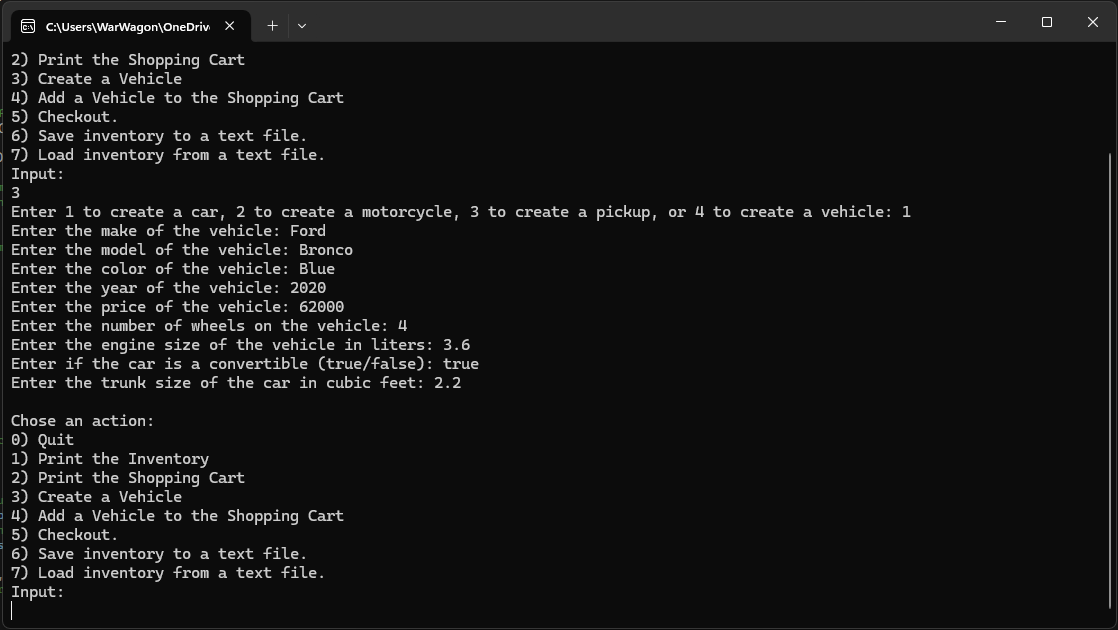


Figure 28: Changes running

Figures 26-28 are the changes to add the new properties and the application running with the new properties added to the create vehicle screens

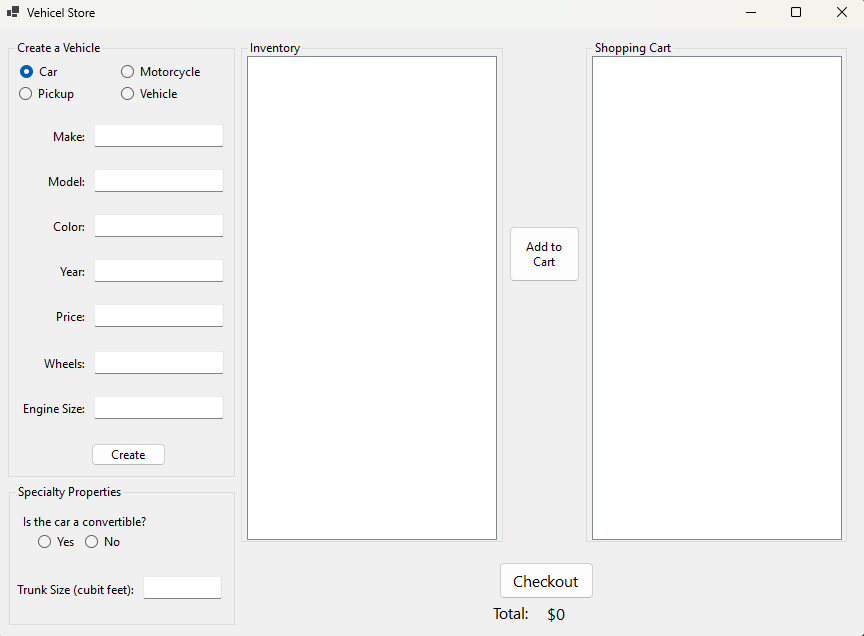
****

Figure 29: Initial state of UI

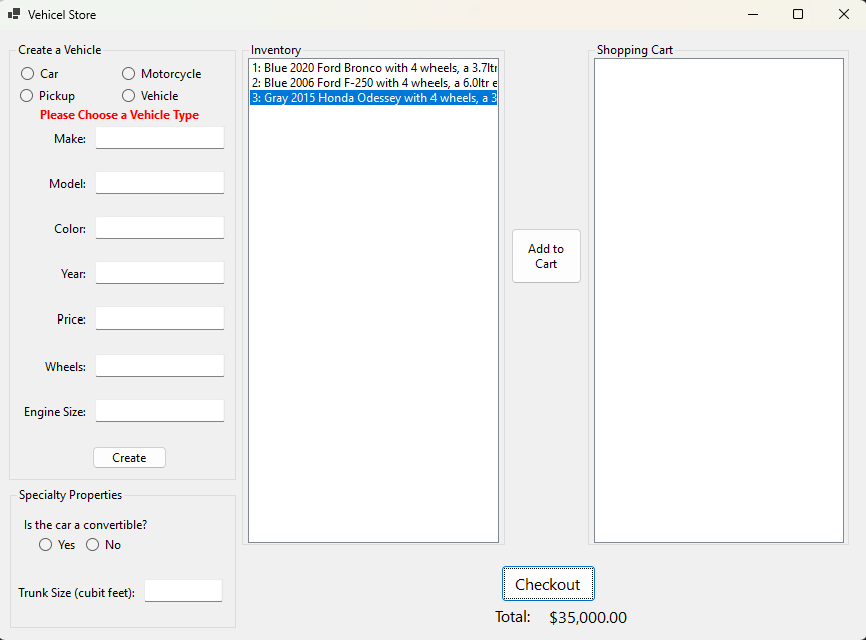


Figure 30: Ui after creating some vehicles and checking one out

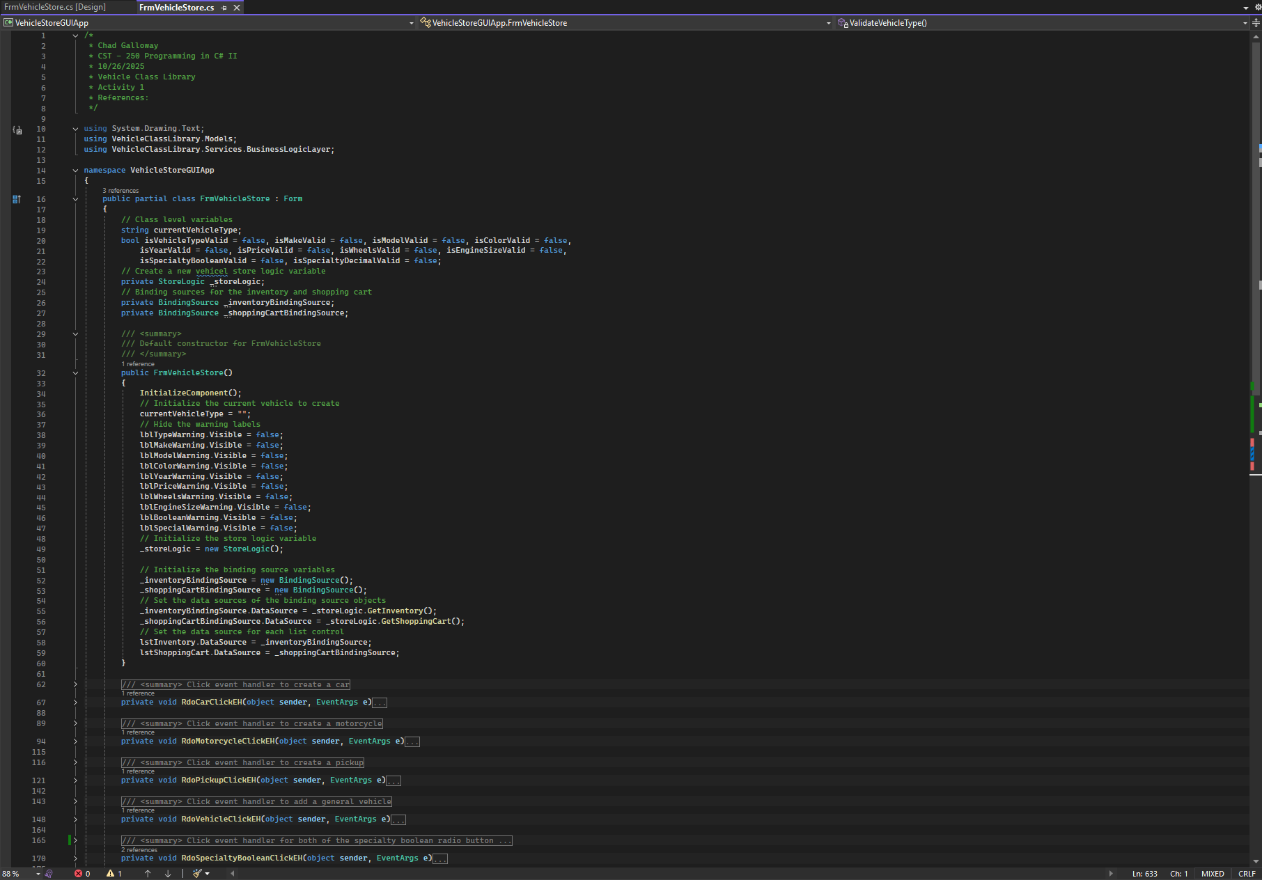


Figure 31: The FrmVehicleStore class declaration, citations, and constructors

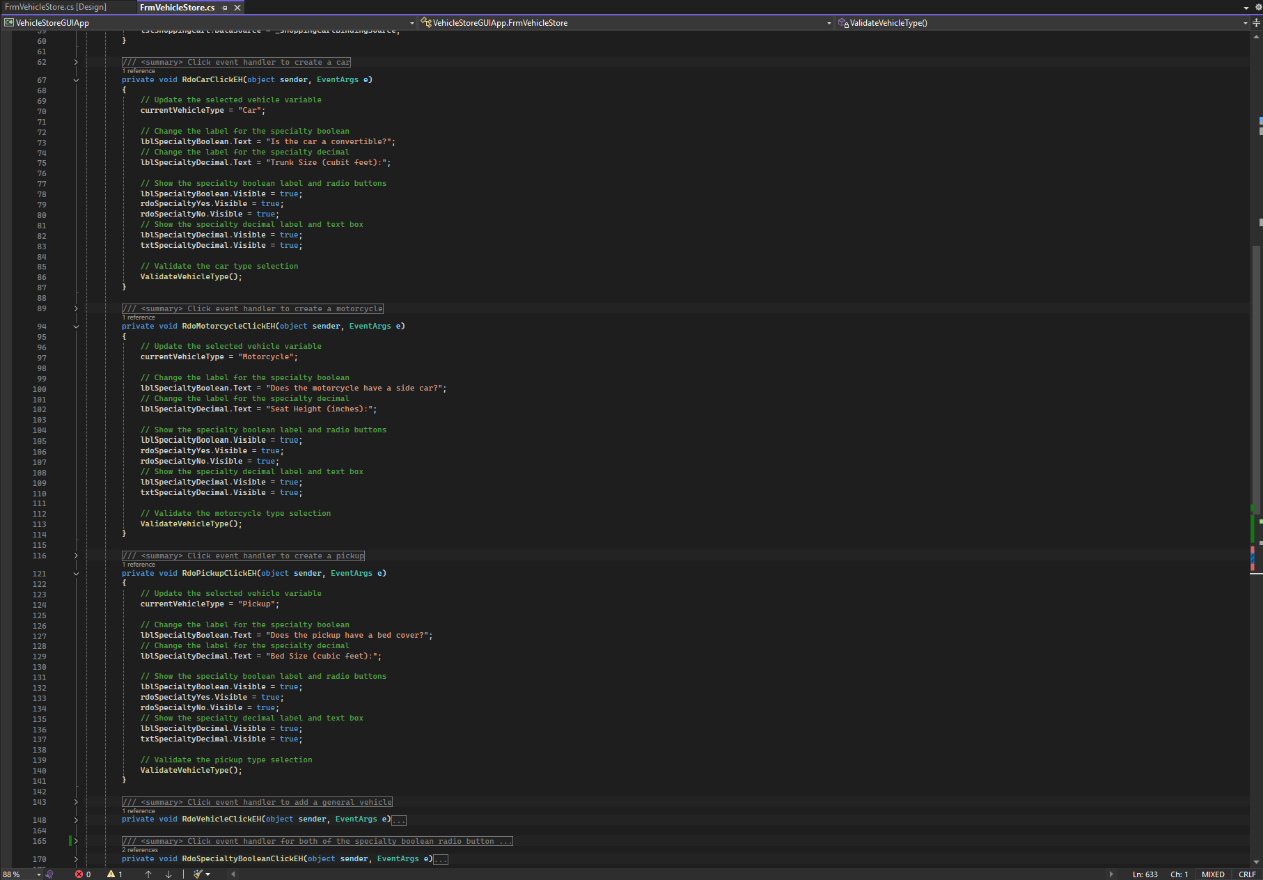


Figure 31: event handlers for Car, Motorcycle, and Pickup radio buttons



Figure 32: Event handlers for the remaining two radio buttons and the add to cart and checkout buttons.

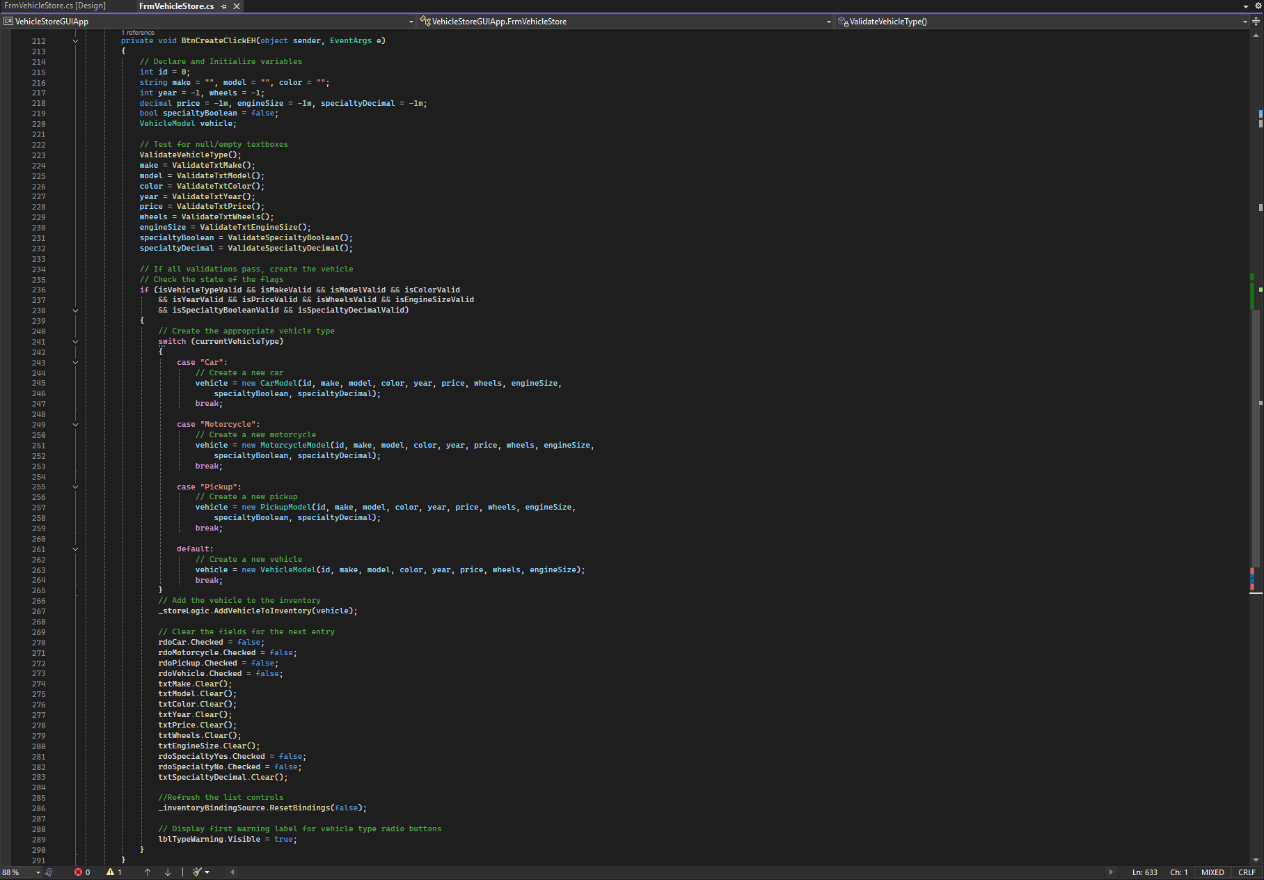


Figure 33: Create Button click event handler.

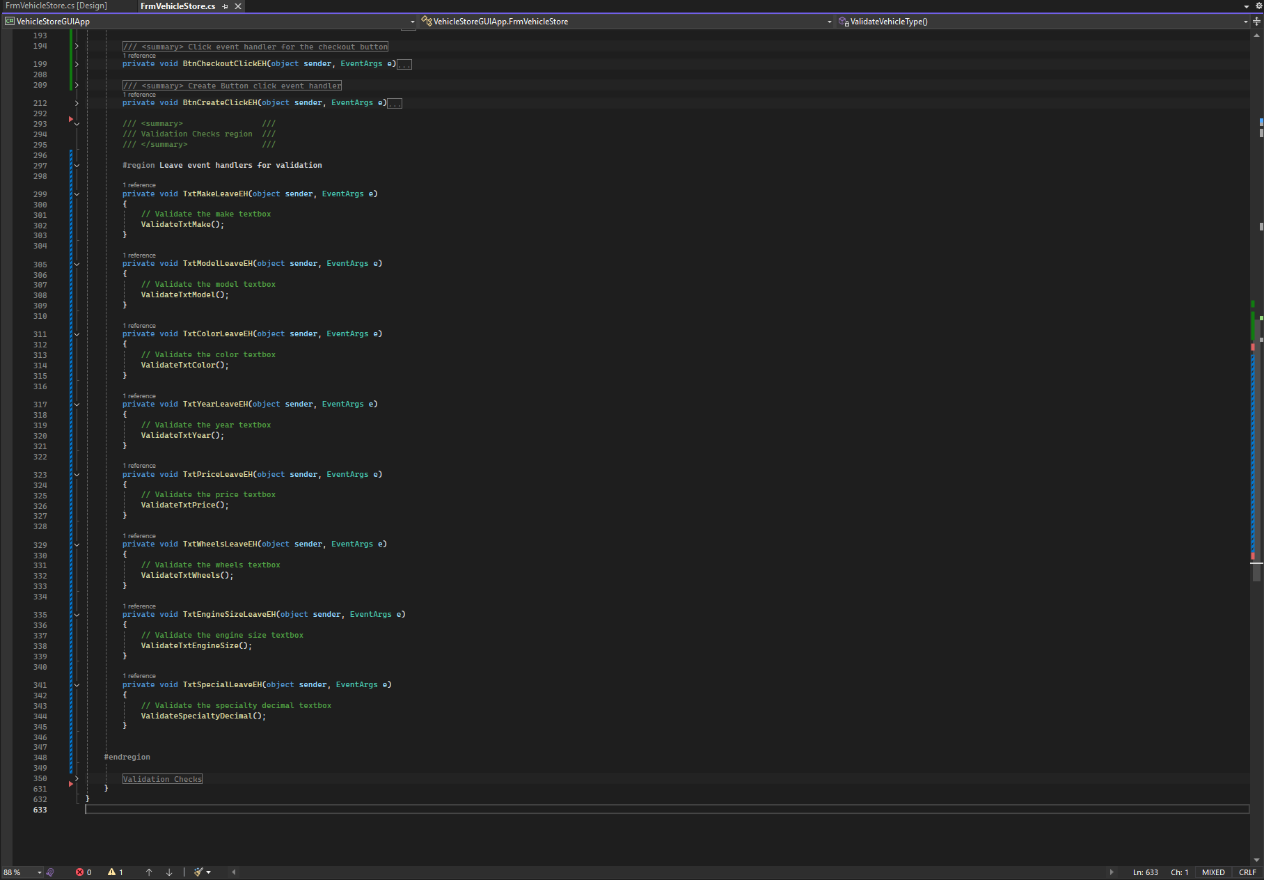


Figure 34: All the leave event handlers

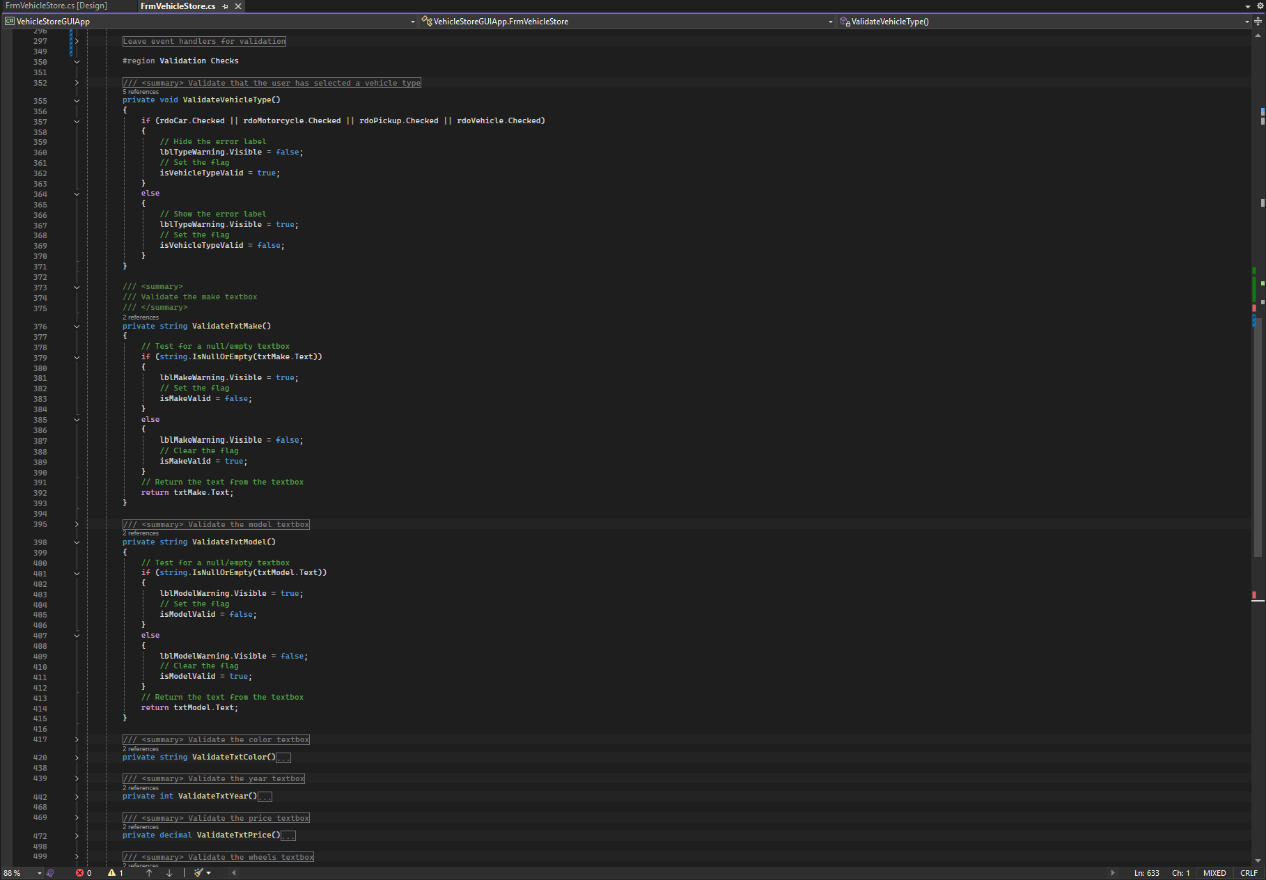
****

Figure 35: Textbox and radio button validation checks

**Part 2**

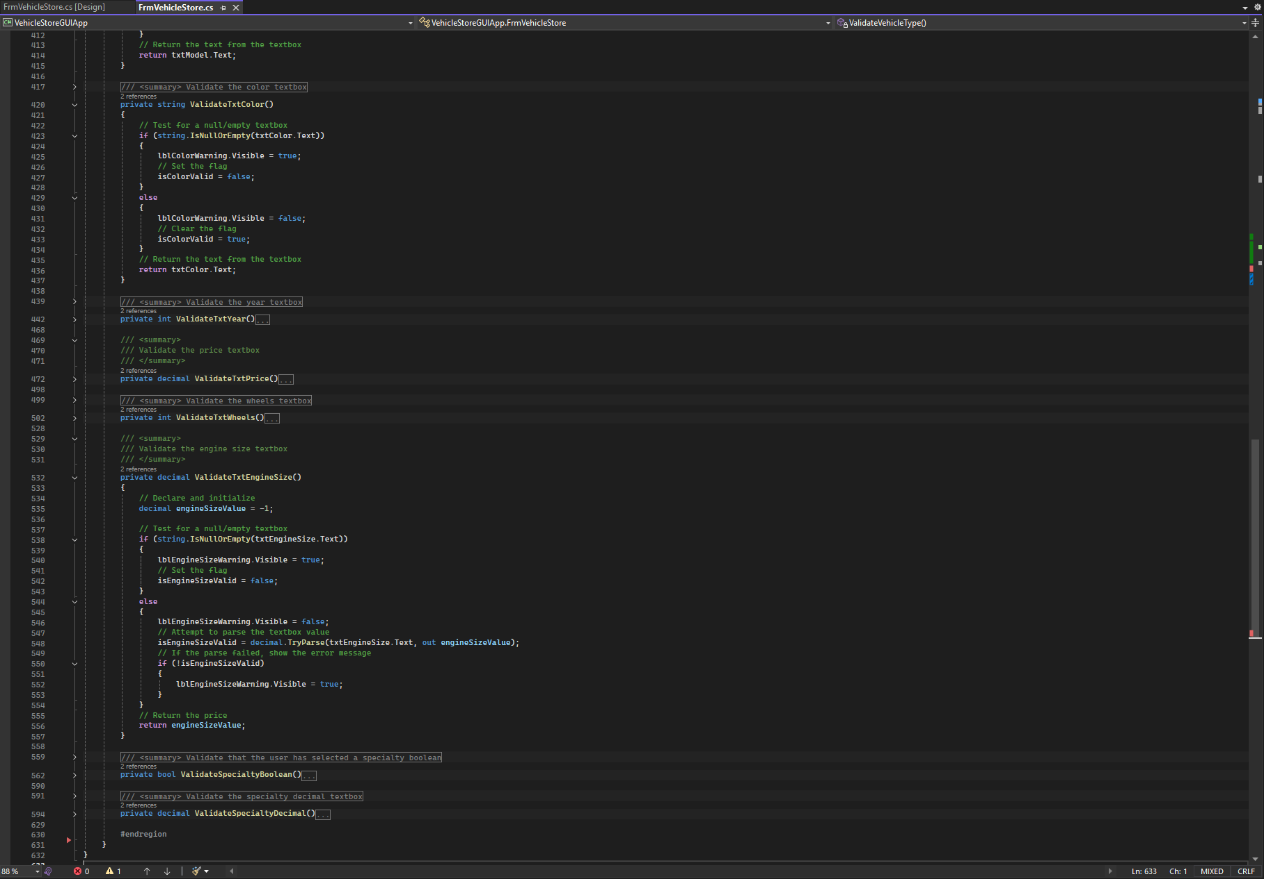


Figure 36: New property validation checks

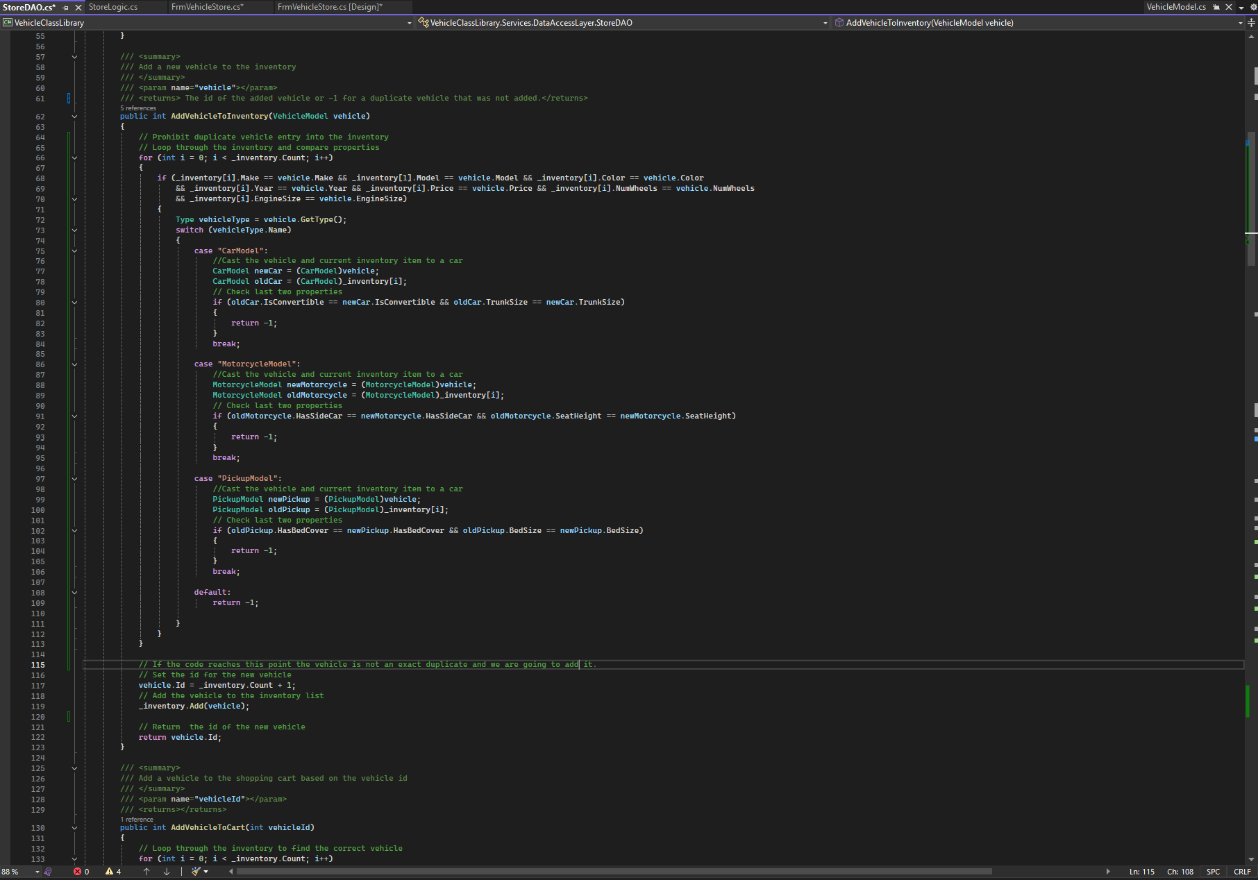


Figure 37: new code added to StoreDAO to prevent duplicates

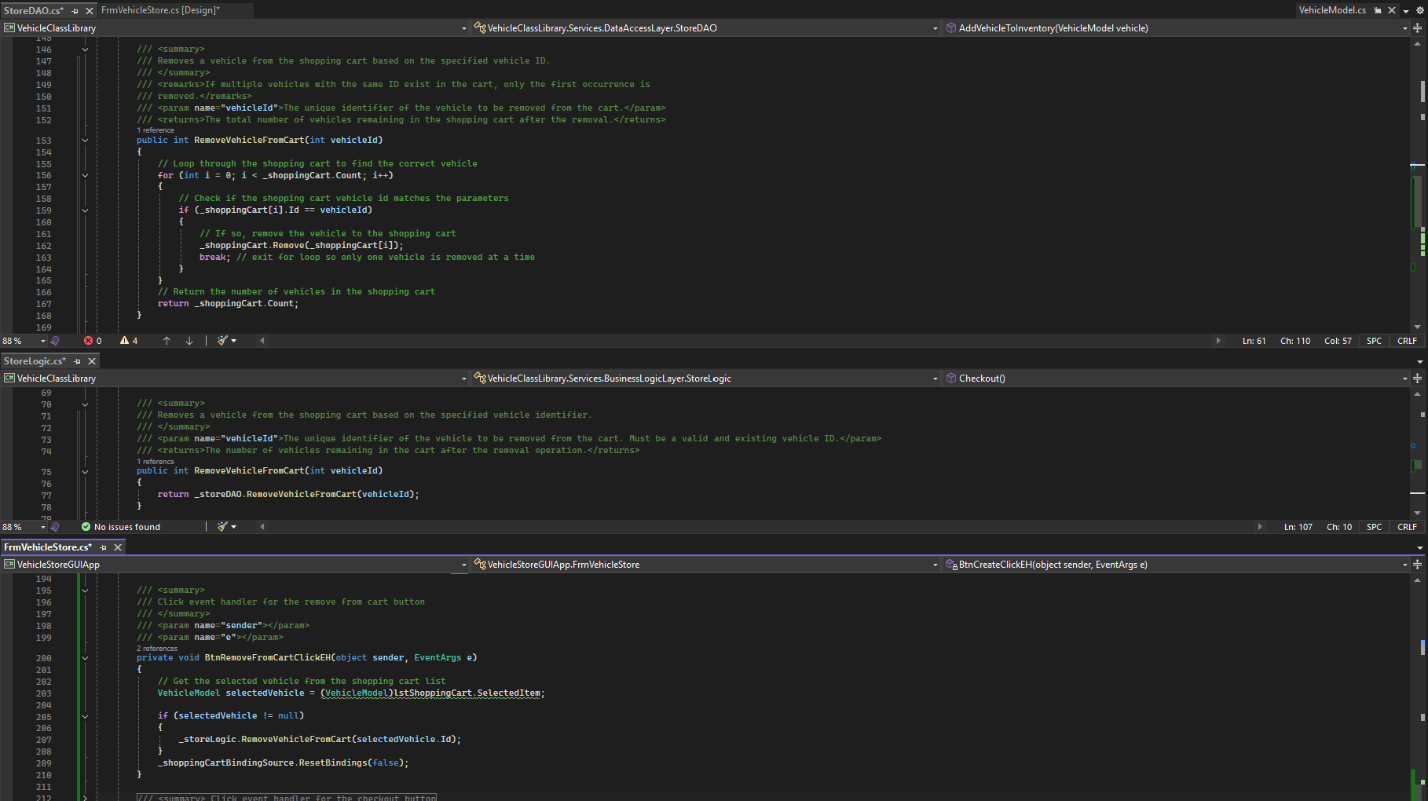


Figure 38: The three methods responsible for removing a vehicle from the cart in all three layers, the StoreDAO, the StoreLogic and the FrmVehicleStore

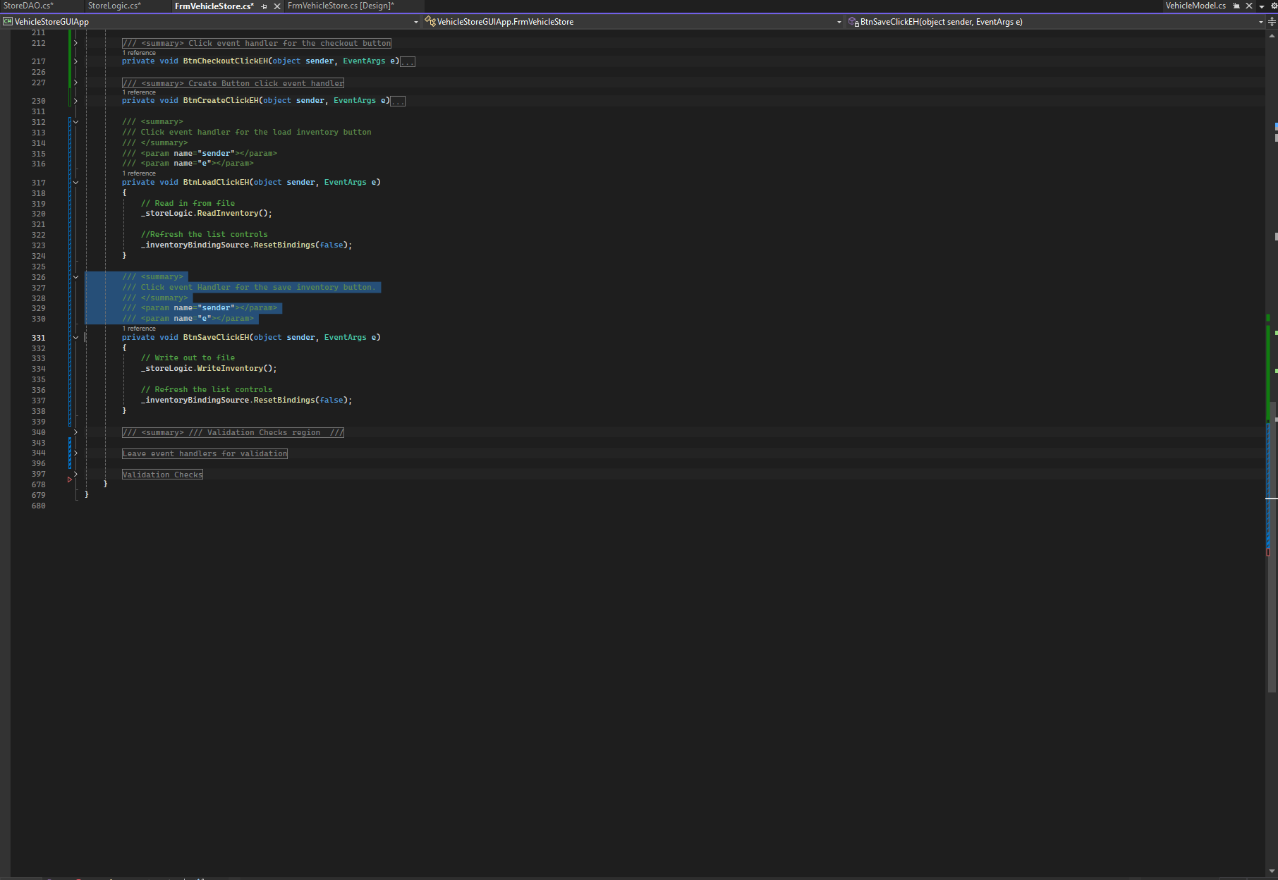


Figure 39: Implementation of the save and load functionality for the GUI

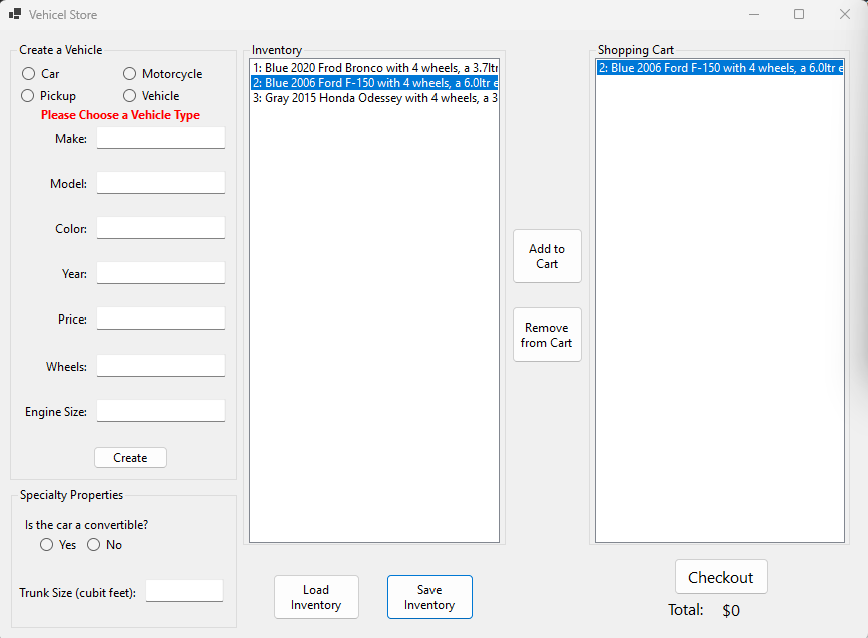
****

Figure 40: the GUI with all the new features implemented

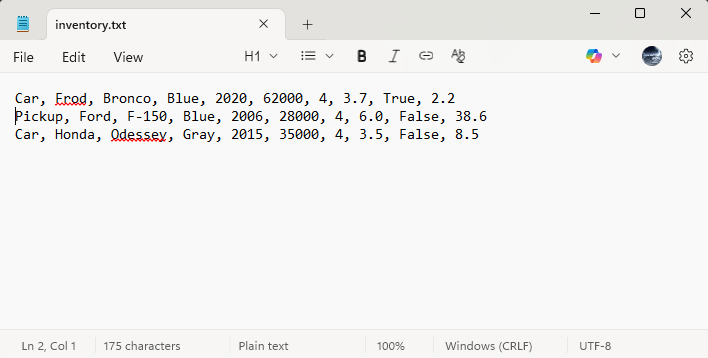


Figure 41: The contents of the save file after hitting save inventory in the above figure 40 screenshot

**Part 3**

Flowchart

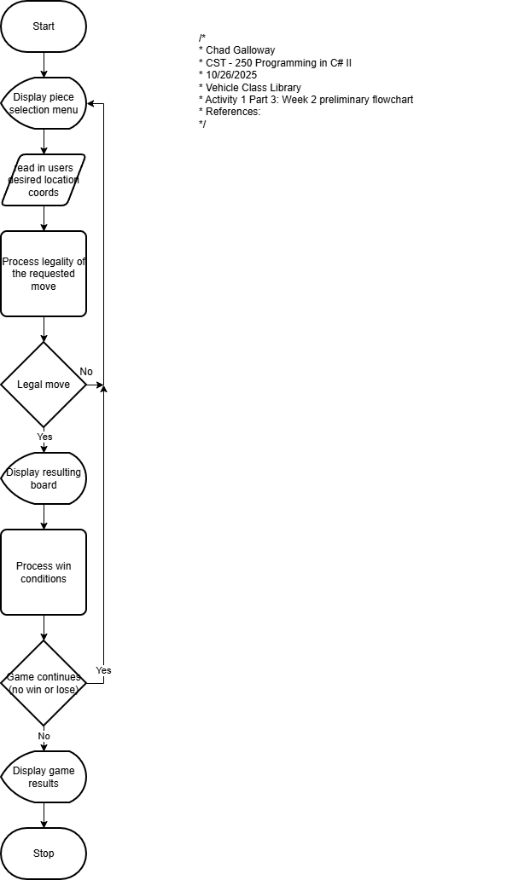


Figure 42: Screenshot Flowchart for week 2

UML



Figure 43: Screenshot of week 2 UML