Parking Garage System Classes for

Master of Science in Information Technology

Software Design and Programming

James McKenna

University of Denver College of Professional Studies

10/05/2025

Faculty: Nirav Shah, M.S.

Director: Cathie Wilson, M.S.

Dean: Michael J. McGuire, MLS

Abstract

This portion of the class is building off of the use case diagram and classes we made for our initial individual project earlier. Our assignment was now to build off of the parking garage system that was planned around initially. Our classes were an address class, car class, car type enum class, customer class and a parking lot class. Each of these classes needed to have private member variables as well as getters that could be called upon if other classes were using composition or inheritance so as to not alter the variable itself directly. After all the classes and their toString functions were created, JUnit Tests needed to be created in order to ensure the correct functionality of all the classes worked the way they were constructed and designed.

CONTENTS

What did you find difficult or easy?

The basic coding structure that was provided at the beginning of this assignment was super helpful in terms of what variables and methods we needed to have established. Personally, what I found to be the hardest was trying to implement every class correctly, and have some classes be implemented into other classes for easier use. One case of this would be the class for our Car.java where I needed to implement LocalDate and didn't realize this was a pre-installed library within Java to use. Then I had to figure out how to implement this into the code win terms of getters, setters, and in the toString function I made. Another use case with this is the Customer.java class that uses the Address.java class we made. This has been the first major assignment where I've had to heavily use private variables as well as getters and setters because of how java encapsulates its coding structure. I'm not used to this because of being more heavy on the Python side with my knowledge. This also popped up once again with ParkingLot.java where I used a list of Car classes to help with capacity and making sure our parking lots weren't full.

What Helped You?

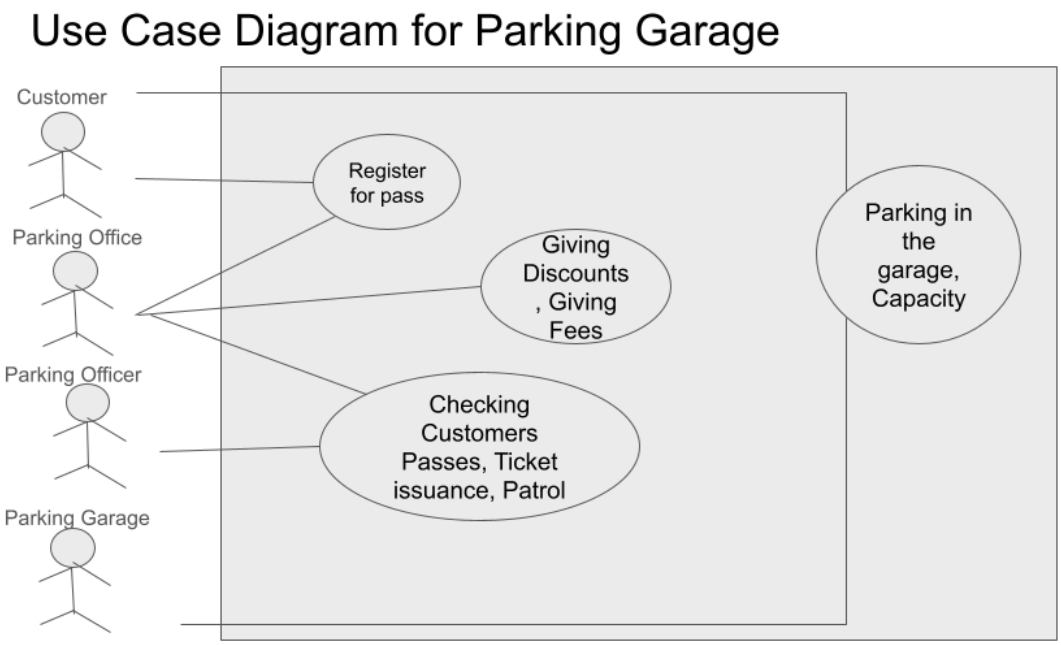
What helped me was slowly realizing that a lot of the base classes were very similar to one another in terms of setters, getters, and how they're all constructed. What varied a bit was trying to implement certain functions that could use these variables and how to use them appropriately. I was getting confused early on and spent a lot of time trying to figure out good use cases for our methods and why they needed to be constructed they way they were.

What did you wish you knew before?

I personally wish I knew more about how to set up these classes appropriately and how to unit test them more thoroughly. As I stated earlier, my experience and knowledge with code is more heavy with Python than any other coding language. Making the switch to Java and then having to write all these unit tests, compile the code and then make sure all the unit tests executed accurately was a very steep learning curve. However I managed to figure out the code and how to do appropriate unit testing on my end because of it. I also got better experience with Java and how to execute this code I wrote because of the unit tests helping me test the code I developed.Another add on is I wish I knew more about certain Java libraries to enhance my coding and unit testing.

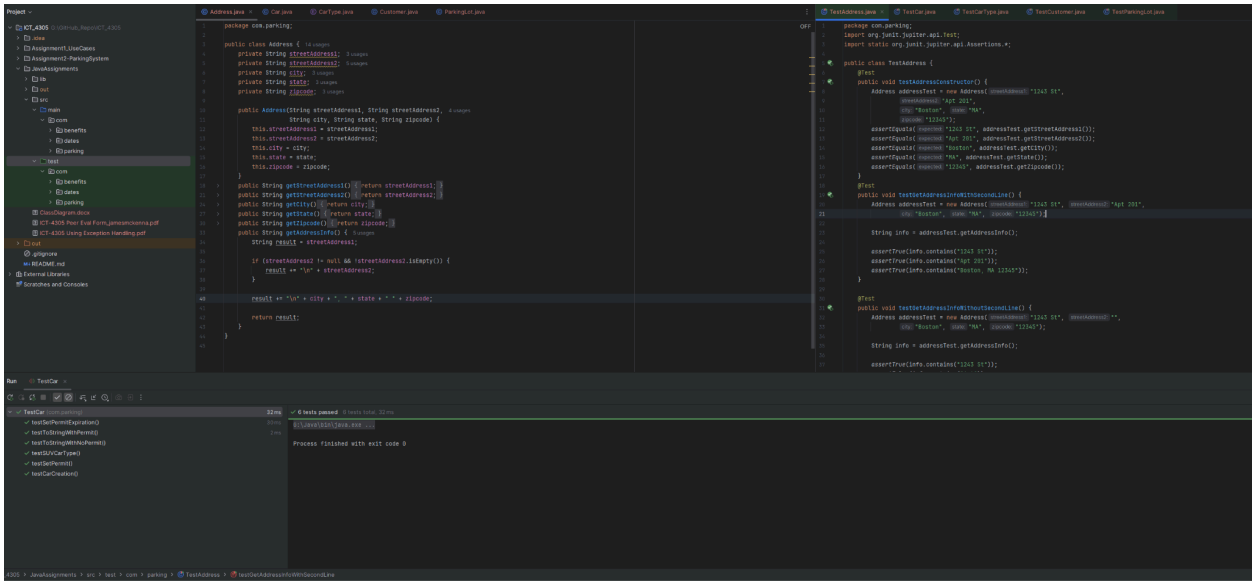
Implementations and Design Decisions

For the overall design and implementation of the code I developed. I decided to go through and look at my original use case diagram. This was updated a little after going through my code. But I realized I couldn't implement all the features such as fees, parking officers going and submitting tickets, and then the customer doing a check to see if they were within the permitted time or had the correct permit. I realized I needed to make this more simple and go forward based off that with future assignments. Currently my design decisions were to make all the appropriate classes such as the Address, Car, Car Type, Customer, and Parking Lot. I incorporated some into one another such as the Customer having a Car class a part of it, the Parking Lot having the Address and the Car class into it, and the Car also having the CarType enum class apart of it. Otherwise it was mostly creating getters and setters, as well as doing the appropriate unit testing to see if all the classes were constructed and made appropriately.

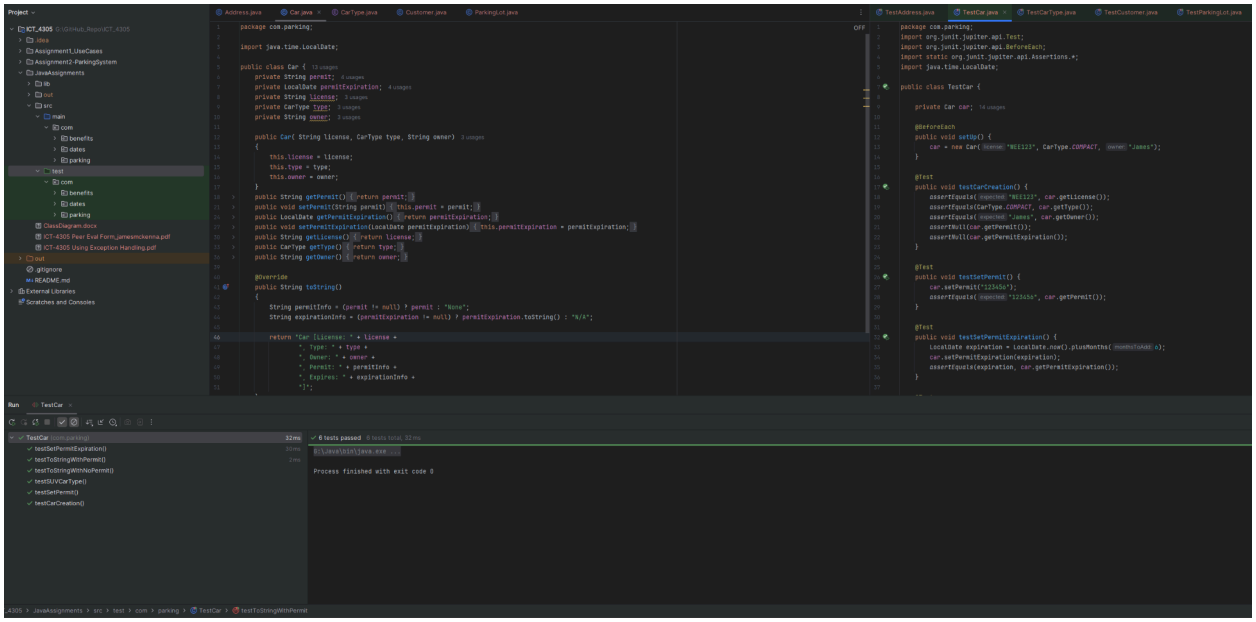


Successful Unit Test Screenshots

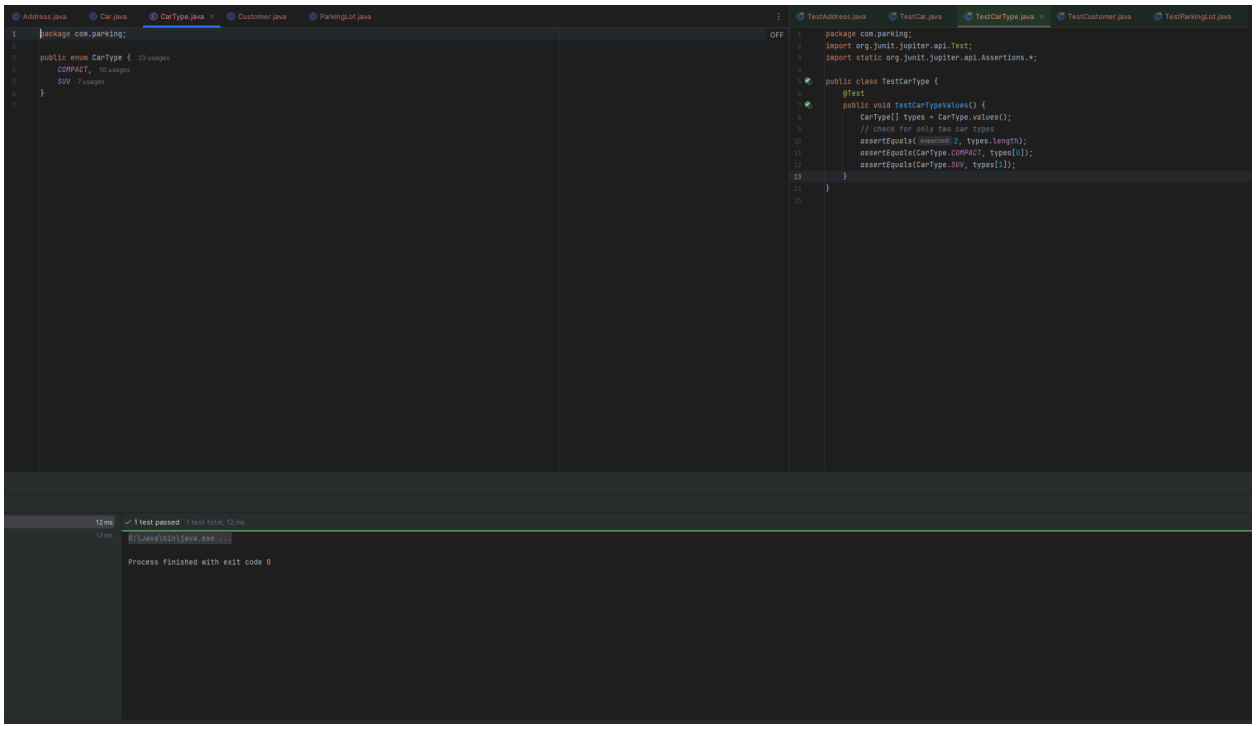
TestAddress.java



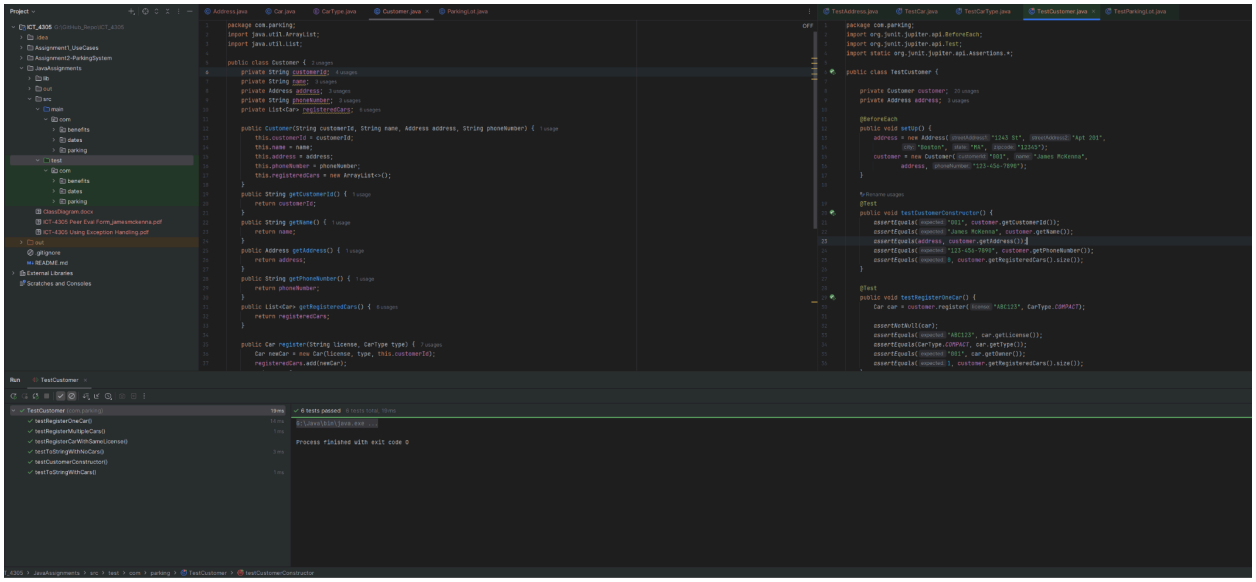
TestCar.java



TestCarType.java



TestCustomer.java



TestParkingLot.java: