**CS 1632 - DELIVERABLE 6: Developing a website using BDD and testing it with Selenium**

github.com/marcelo-dalmeida/CS-1632-DELIVERABLE-6

Philip Ni

Marcelo d’Almeida

1. **Introduction**

In this project, we created a car buying webapp using behavior-driven development (BDD). The webapp currently allows users to see cars that are available and calculates the total cost after taxes, down-payments, and interest. The user is given information about total price and the price per month they would have to pay. It also allows managers to register for an account which will allow them to create or modify car postings, and generate new registered users.

As future work, Cart and Payment functionality will be added, so the webapp can, in fact, sell cars, as well as support for different location (State) buyers. The database will store payment history, information about the clients, and information about the car distribution locations.

BDD seemed to be the best way to develop this app because it allowed us to focus on getting the website and its basic features working as priority. Being feature focused, we decided to divide our user stories into core functionalities: 1) database functionalities (e.g. insert, search, and deleting car entries), 2) login/user registration, and 3) payment calculation. Splitting the user stories by functionality helped up develop more modularly. We found integrating components to be easier this way.

One of the problems of developing large applications is that it is difficult to know when to stop adding features to a user story. This is the advantage we found in using BDD. By defining what functionality a particular user story should have in advance, we could focus more on just getting those specifications met and then coming back to the user story if there is more to add.

Another issue we had to consider was leaving room to add more scenarios, in order to cover as many edge and corner cases as possible. Because we started testing for basic use cases, we were able to get all core functionalities developed first and then worry about further testing. The advantage of developing this way, is that we had a functional webapp throughout the development process. On the other hand though, things like style and user experience took a major hit, as we focused more on getting the features working. Most of our tests were used to determine if the database backend was operating correctly and enabling other functionalities. For example, logging in relies on the database to only allow registered users. That way only valid users can insert and delete car entries, and create new user accounts.

Because feature verification was our primary goal, we could not focus on security or performance. Right now, the database is relatively small. So, I would like to see how the site behaves when the database has hundreds of thousands of entries rather than just one or ten. Also, we have proven that SQL injection is a big issue in many of our text entry areas. So, I would like to perform some security testing to see where exactly the problem areas are, and find out a way to eliminate security risks.

*github.com/marcelo-dalmeida/CS-1632-DELIVERABLE-6*

1. **Quality Assessment**
2. CSS and Style [YELLOW]:

Many of the objects are misaligned, making some things difficult to read. This is very apparent, and may make the use of the website unbearable to some. It does not affect the major functionality of the site’s features. This is a more difficult issue to fix as we need to consider frameworks and designs to use.

1. SQL Injection [RED]:

This is a severe threat that has the potential to break the backend or potentially corrupt the database entries.

1. Back Button throws “Document Expired” when a form is submitted [GREEN]:

Currently, this is solved by refreshing (Ctrl+R or F5) the page. This is an annoying problem, but does not break any functionality, surprisingly. It can be easily resolved, given some more time.

1. Manager can enter empty entry to the database [YELLOW]:

This is a severe threat, but can be easily resolved in time. It has the potential to break security features of logging in, but it mainly affects car entries.

1. User Experience [GREEN]:

Users may have difficulty understanding how to use the site because there is very little instruction. This is an easy issue to fix.

1. Dynamically Generated Links Messing with our Unit Tests [GREEN]:

Some of our database tests are failing because links are dynamically generated after looking up information from the database. This does not mean that any features are broken; it just means that we have to rethink how we are writing our tests. A quick fix would be to add a thread wait

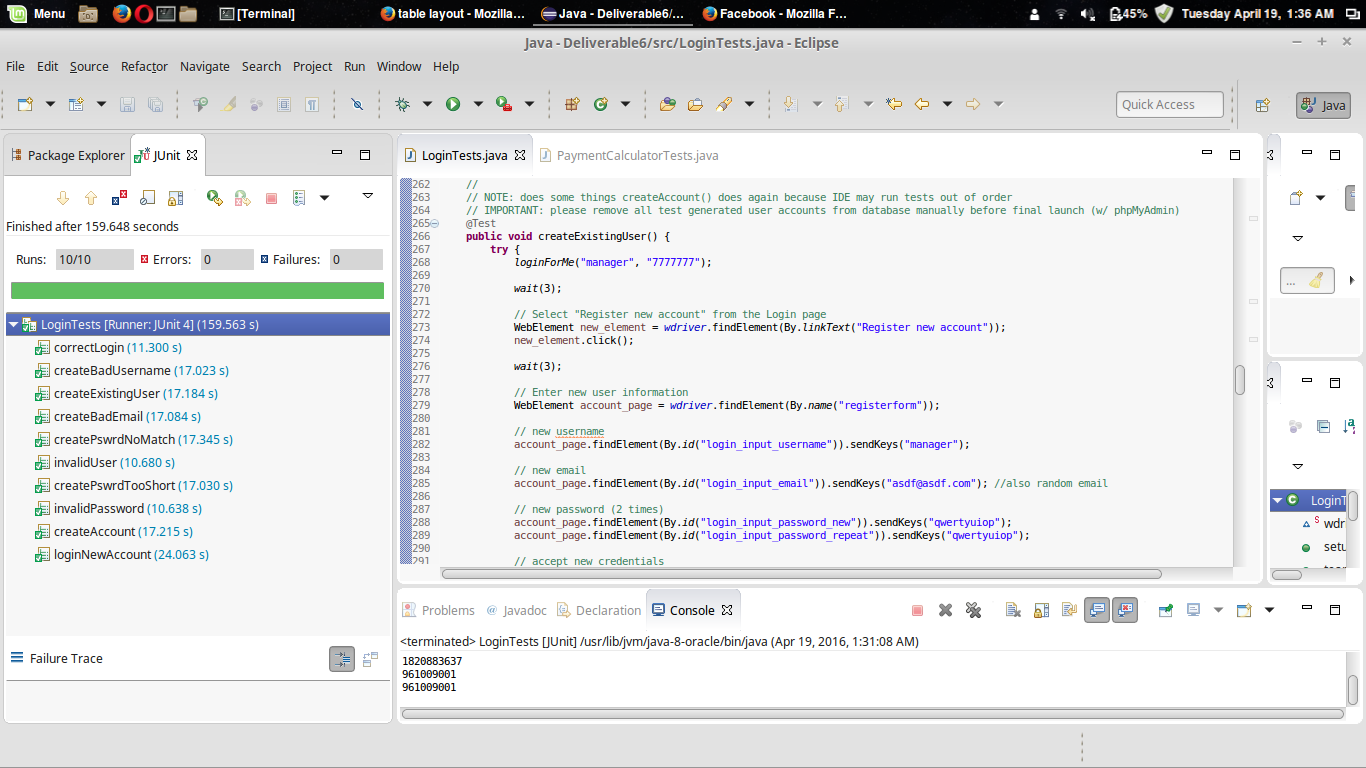
Conclusion:

Overall, we have met all of the goals and milestones we have set out for ourselves when we started. We also have a high percentage of passed tests. We may need to think of a better way to test certain scenarios. However, most of the features seem to be working as we specified.

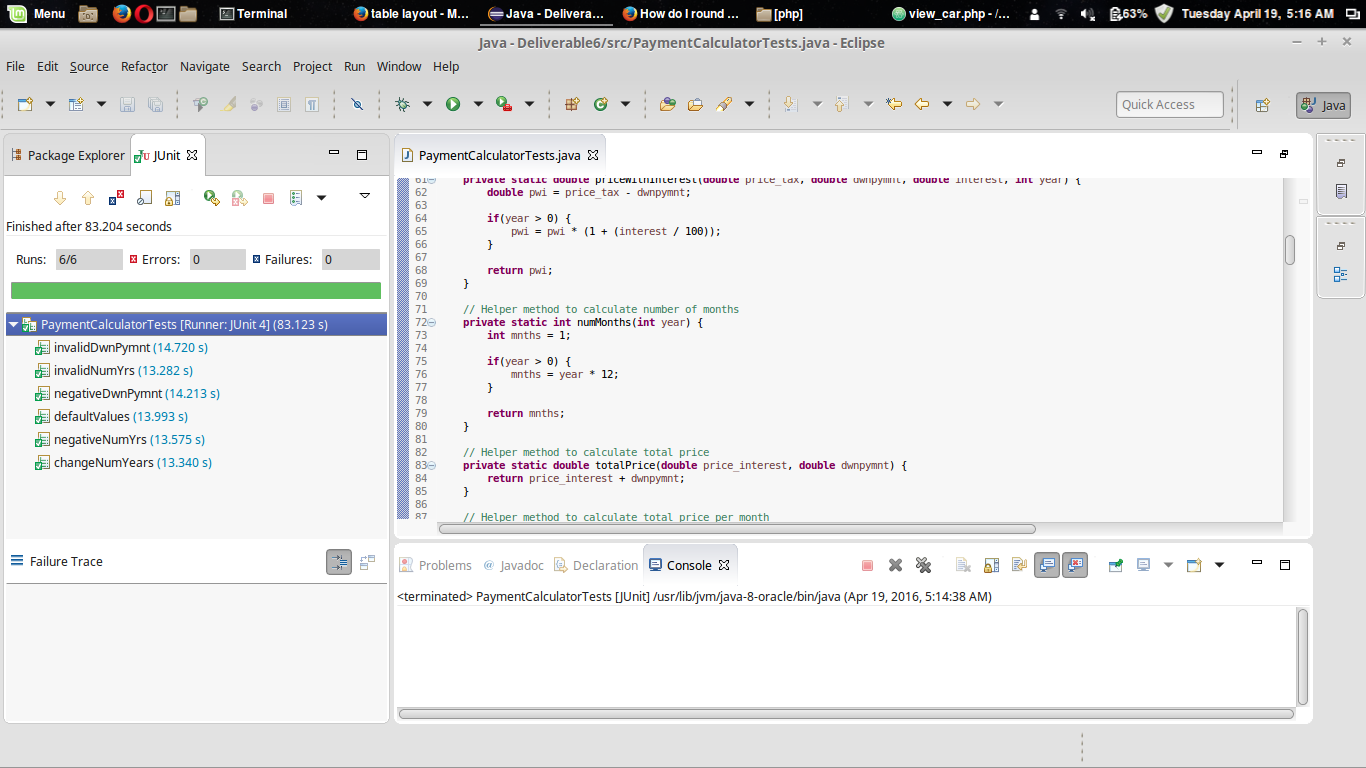
On the other hand, this webapp needs a lot more work. The main purpose of creating this webapp was to explore PHP and web programming. Now, that we know how this all works, we can spend more time focusing on making the site look and feel more professional. As it is now, the site is usable, but unbearable.

Given more time, we could add or modify certain features to make the webapp more secure and responsive. We could also use PHPUnit and some profiling tools to test the backend code directly, in the future, to see where the performance and security issues are.

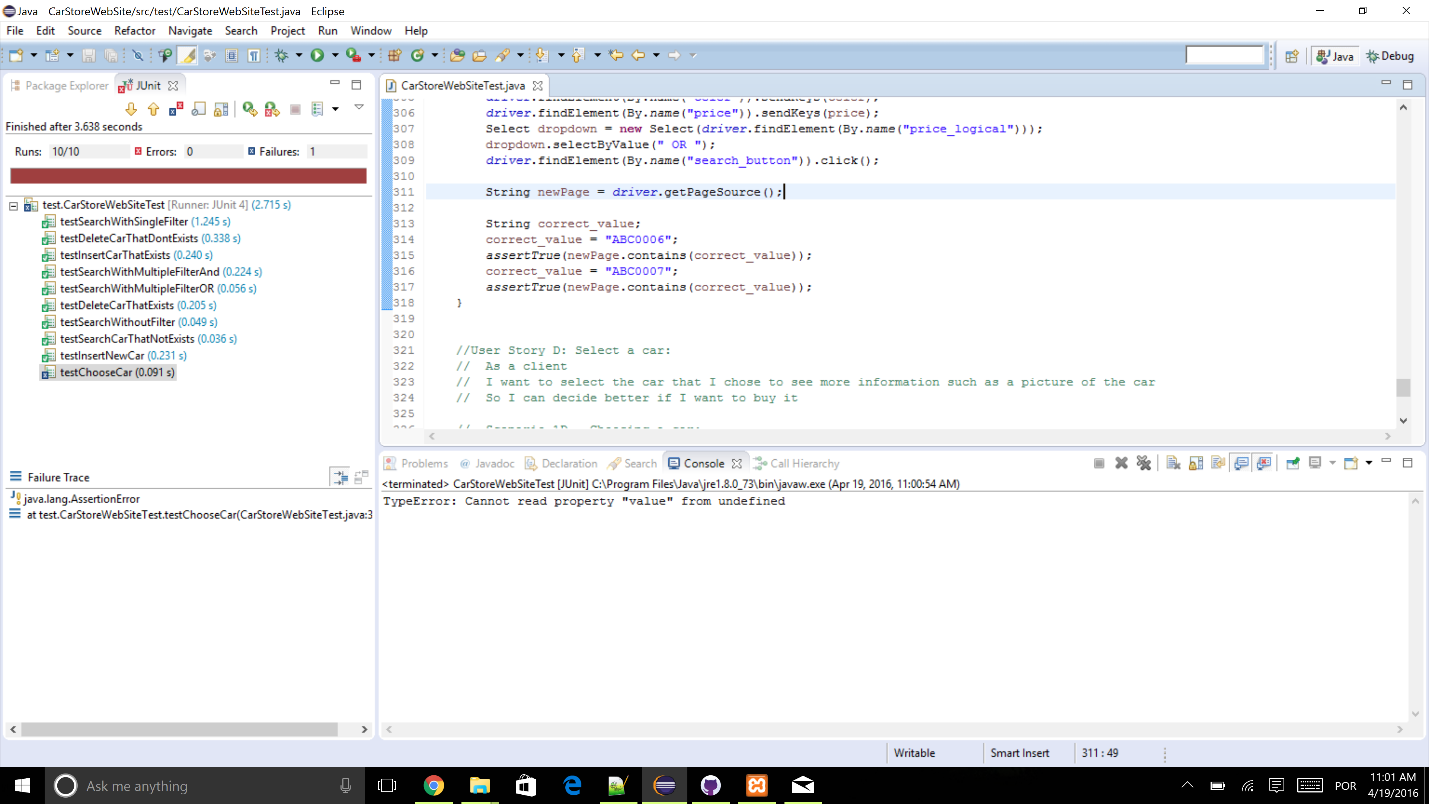
1. **Test Run**



Login Tests



Payment Calculator Tests



Database Functionality Tests

1. **User Stories and Scenarios**

**User Story A: Insert on database:**

As a manager

I want to be able to insert new cars to the database

So I can let new cars be available for the client

**Scenario 1A – Insert a new car correctly**

Given a unique license plate

And all other car related information

When the manager tries to insert the car information to the database

Then the car is inserted in the database and it’s available for both manager and client to search

**Scenario 2A – Insert a car with existing license plate**

Given a license plate

And all other car related information

When the manager tries to insert the car information

Then a message appears saying that this license plate is already in the database

**User Story B: Delete on database:**

As a manager

I want to be able to delete cars from the database

So I can let cars be unavailable for the client

**Scenario 1B – Delete a car correctly**

Given a unique license plate

When the manager tries to delete the car information from the database

Then the car is deleted from the database and it’s unavailable for search

**Scenario 2B – Delete a car that is not in the database**

Given a unique license plate

When the manager tries to delete the car information from the database

Then a message appears saying that there is not any car with that license plate

**User Story C: Search on database:**

As a client

I want to be able to filter for cars through a defined parameter such as color

So I can choose the right car for me

**Scenario 1C – Search for a car that is not in the database**

Given a not registered license plate

When the client tries to search

Then a message appears saying that no results were found

**Scenario 2C – Search for a car without filtering**

Given an empty list of filters

When the client tries to search

Then a list containing all the cars from database appears

**Scenario 3C – Search for a car through a single field filtering**

Given a year

When the client tries to search

Then a list of cars corresponding the filtering appears

**Scenario 4C – Search for a car through multiple field filtering (AND)**

Given a car model

And a color

When the client tries to search

Then a list of cars corresponding the filtering appears

**Scenario 5C – Search for a car through multiple field filtering (OR)**

Given a color

Or a price

When the client search

Then list of cars corresponding the filtering appears

**User Story D: Select a car:**

As a client

I want to select the car that I chose to see more information such as a picture of the car

So I can decide better if I want to buy it

**Scenario 1D – Choosing a car**

Given a list of cars

When the client chooses a car

Then all information about the selected car and picture(s) appears

**User Story E: Login:**

As a car sales manager

I want to be able to securely login to the system

So that only account holders can add new user accounts

**Scenario 1E – Invalid user**

Given that, I am on the login page

When I enter a username that does not exist

Then I should not be able to proceed from the login page

And I should be prompted with a message stating what was wrong

**Scenario 2E – Invalid password**

Given that I am on the login page

When I enter a correct username

And an incorrect password

Then I should not be able to proceed from the login page

And I should be prompted with a message stating what is wrong

**Scenario 3E – Correct login**

Given that I am on the login page

When I enter valid login credentials

Then I should be able to access more features

And I should see a message indicating a successful

**Scenario 4E – Create Account**

Given that I am successfully logged in

When I select the "Login" option

Then I can create a new account

**Scenario 5E – Login new account**

Given that I have created a new account

When I log out

Then I can login and use the newly generated account

**Scenario 6E – Create existing user**

Given that I am successfully logged in

When I create a new manager account with the same username/email as an existing account

Then the original account will not be overwritten

And the new account will not be created

**Scenario 7E – Create bad username**

Given that I am successfully logged in

When I try to register a new account

And use a username that does not meet the pattern specified

Then I cannot proceed to a new page

**Scenario 8E – Create bad email**

Given that I am successfully logged in

When I try to register a new account

And use an email address that does not meet the pattern specified

Then I cannot proceed to a new page

**Scenario 9E – Create password that doesn’t match**

Given that I am successfully logged in

When I try to register a new account

And the password does not match the repeated password

Then I will be informed that the registration has failed

**Scenario 10E – Create password too short**

Given that I am successfully logged in

When I try to register a new account

And use a username that does not meet the pattern specified

Then I cannot proceed to a new page

**User Story F: Payment Calculator:**

As a general user (either a manager or non-account user)

I would like to be able to calculate the estimated price of a car

So that I know how much it would cost me to buy that car with taxes, interest, and other fees included

**Scenario 1F – Default Calculation Behavior**

Given that I have selected a car

And am currently viewing its information page

When I don't change any values in the price calculator

Then the correct total price will be calculated for the car for 1 month (0 yrs) with no down-payment

**Scenario 2F – Modifying Down-Payment Field**

Given that I have selected a car

And am currently viewing its information page

When I change only the down-payment in the price calculator

Then the correct total price will be calculated for the car for 1 month (0 yrs) with the specified down-payment

**Scenario 3F – Modifying Number of Years Field**

Given that I have selected a car

And am currently viewing its information page

When I change only the number of years in the price calculator

Then the correct total price will be calculated for the car for n \* 12 months (n yrs) with no down-payment

**Scenario 4F – Modifying Down-Payment Field with Non-Numeric Entry**

Given that I have selected a car

And am currently viewing its information page

When I enter non-numeric characters into the down-payment field

Then the results fields should be empty/not change (except for "Number of months" which will revert to its default value of 1)

**Scenario 5F – Modifying Number of Years Field with Non-Numeric Entry**

Given that I have selected a car

And am currently viewing its information page

When I enter non-numeric characters into the number of years field

Then the results fields should be the same as the default behavior

**Scenario 6F – Modifying Down-Payment Field with Negative Entry**

Given that I have selected a car

And am currently viewing its information page

When I enter a negative value into the down-payment field

Then the results fields should behave as normal using negative values

**Scenario 7F – Modifying Number of Years Field with Negative Entry**

Given that I have selected a car

And am currently viewing its information page

When I enter a negative value into the number of years field

Then the results fields should be the same as the default behavior