## CS4004 GROUP PROJECT

**Made by:**

Eoin McDonough – 18241646

Jakub Pažej – 18260179

Aleksandr Jakusevs – 18257038

Shuo Pei – 19107657

## TABLE OF CONTENTS

[Cover Page 1](#_Toc24823938)

[Table of Contents 2](#_Toc24823939)

[Task #1: Bug Reporting 3](#_Toc24823940)

[Task #2: Test Design and Development using JUnit 17](#_Toc24823941)

## **Task #1: Bug Reporting**

**The Log:**

* Brainstorming about all the possible bugs 1.11.2019
* Noting down the bugs in a word document 1.11.2019
* Submitting 5 bugs on Friday 8.11.2019
* Submitting 14 bugs on Tuesday 12.11.2019

**Reflections:**

Jakub: I learned a few important parts of bug reporting like how to keep your bug reports short while simultaneously getting your point through efficiently.

Eoin: I improved my ability to write bugs and had to think about the severity of each possible case and whether the bugs were server side or client side.

Aleksandr: I learned how to think about bugs and how to report them while taking the developer and middle management into account.

Shuo: I learned how to sum up bugs better and find common things between different bugs.

**List of Headings:**

* 1. Bug Report 1 - Seat reservation doesn't work for some airplanes.
  2. Bug Report 2 - Layover duration in connecting flights not updating correctly.
  3. Bug Report 3 - Price of tickets changes after changing website language.
  4. Bug Report 4 - The power to change meeting location is not given appropriately.
  5. Bug Report 5 - Safety distance between trains not maintained.
  6. Bug Report 6 - Trains don't slow down before a zone with a lower speed limit.
  7. Bug Report 7 - Trains don't slow down before approaching a curve.
  8. Bug Report 8 - Train moves before doors close.
  9. Bug Report 9 - Train announcement system working incorrectly.
  10. Bug Report 10 - Non-Latin city names displayed incorrectly.
  11. Bug Report 11 - Printing of the e ticket is not supported in non-Latin alphabets.
  12. Bug Report 12 - When doing a multi-step flight, the system marks you as a frequent flyer.
  13. Bug Report 13 - When a flight is full it doesn’t stop customers from buying more tickets.
  14. Bug Report 14 - The currency exchange doesn’t update fast enough.
  15. Bug Report 15 - Cancelling a flight does not notify the system of a seat being freed.
  16. Bug Report 16 - Time zone is not adjusted for all the users.
  17. Bug Report 17 - The system doesn’t calculate the preferred time accurately.
  18. Bug Report 18 - Everyone is able to initiate a meeting.
  19. Bug Report 19 - System does not update after a book is returned.
  20. Bug Report 20 - Booking more rooms than the limit.
  21. **Bug Report 1 - Seat reservation doesn't work for some airplanes.**

Component: Web frontend

OS: All

Summary: Seat reservation doesn't work for some airplanes.

Overview: WOW Online Flight Reservation System bug

Steps to Reproduce:

|  |
| --- |
| 1. Go to the WOW flights website  2. Try to reserve a flight that’s on a RAN-D00m or B-01 type of airplane  3. Observe the graphical representation of the seats is not changed to that type of airplane, instead it displays the dflt-02 airplane |
|  | |

Actual Results: Seat reservation is working correctly for all available types of aircraft

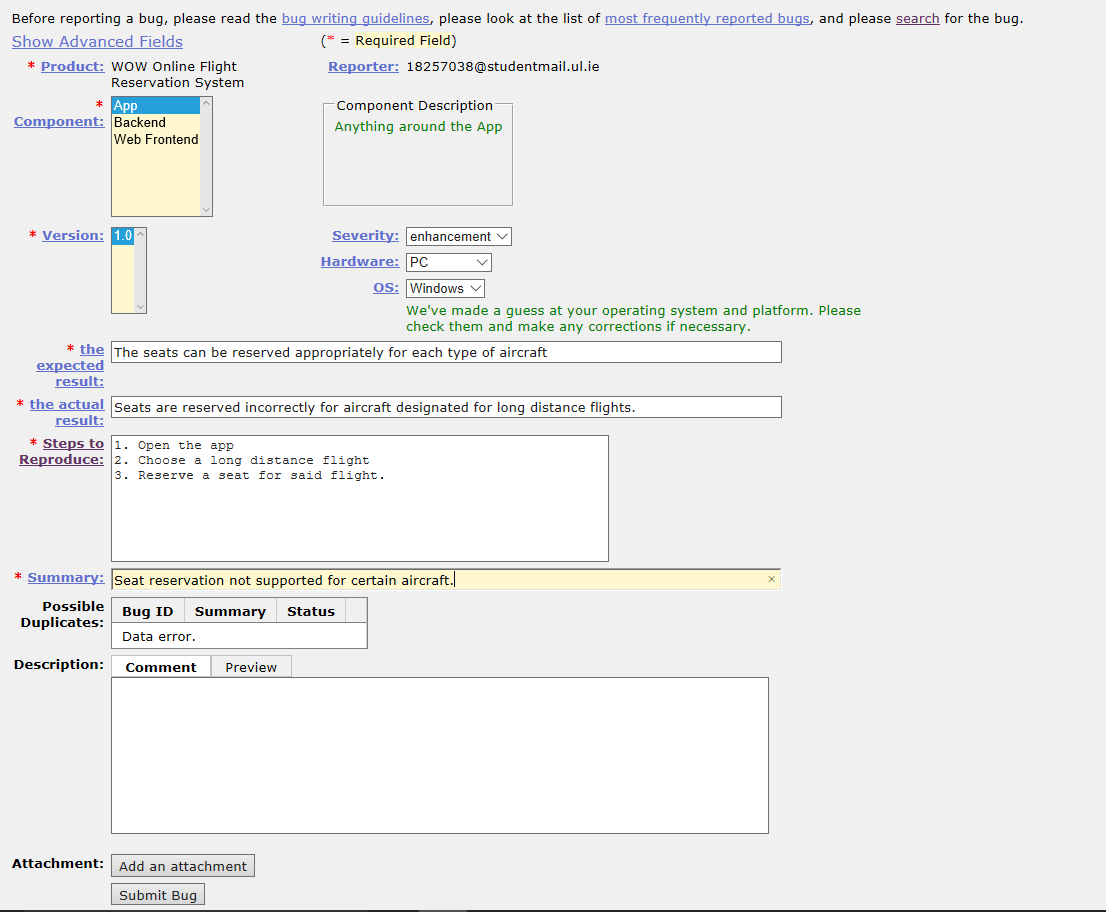
Expected Results: Seat reservation not supporting all types of aircraft

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: none

Additional Information: Importance major.

Screenshot of the submitted bug:



* 1. **- Bug Report 2 - Layover duration in connecting flights not updating correctly.**

Component: App

OS: Other

Summary: Layover duration in connecting flights not updating correctly.

Overview: WOW Online Flight Reservation System bug

Steps to Reproduce:

|  |
| --- |
| 1. Open the WOW flights app  2. Buy a flight that connects to another flight  3. Change the expected arrival or departure time on any of the bought flights  4. Observe that the expected layover time between them doesn't update |
|  | |

Actual Results: The layover duration between flights is not updated when there is a delay in either of them.

Expected Results: The layover duration is updated whenever there is any delay or early arrival between the two flights.

Build Date & Hardware: Build 1 for other hardware.

Additional Builds and Platforms: none

Additional Information: importance major

* 1. **Bug Report 3 - Price of tickets changes after changing website language.**

Component: Web Frontend

OS: All

Summary: Price of tickets changes after changing website language.

Overview: WOW Online Flight Reservation System bug

Steps to Reproduce:

|  |
| --- |
| 1. Go to the WOW flights website  2. Observe the prices on some of the tickets  3. Change the language of the website in the top right corner  4. Find the tickets previously checked and notice the price changed |
|  | |

Actual Results: prices of tickets change to the selected countries local price whenever you change the language on the website

Expected Results: The prices only changes to different countries local prices after you check the users IP address and it's from the different country.

Build Date & Hardware: build 1 for all hardware

Additional Builds and Platforms: none

Additional Information: Importance critical.

* 1. **Bug Report 4 - The power to change meeting location is not given appropriately.**

Component: General

OS: All

Summary: The power to change meeting location is not given appropriately.

Overview: WSS Meetings

Steps to Reproduce:

|  |
| --- |
| 1. Create a meeting with a few accounts in different positions in it  2. Set a meeting location from the most powerful account(effectively removing the option for all the other accounts to vote on it)  3. Go to any other of the accounts and observe they are all able to change the meeting location |
|  | |

Actual Results: Everyone can change the meeting location

Expected Results: Only the people on the highest positions and the people that have permission from them are able to change the meeting location

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: none

Additional Information: importance critical

* 1. **Bug Report 5 - Safety distance between trains not maintained.**

Component: General

OS: All

Summary: Safety distance between trains not maintained.

Overview: WAX Transportation bug

Steps to Reproduce:

|  |
| --- |
| 1. Create a delay in the usual train schedule  2. The train behind the delayed train doesn't slow down when approaching from behind  3. The distance between the trains gradually decreases over time  4. Possibility of collision occurs if the first train (the one in front) decides to suddenly brake. |
|  | |

Actual Results: The train does not manage to meet the specified safety distance between trains accordingly.

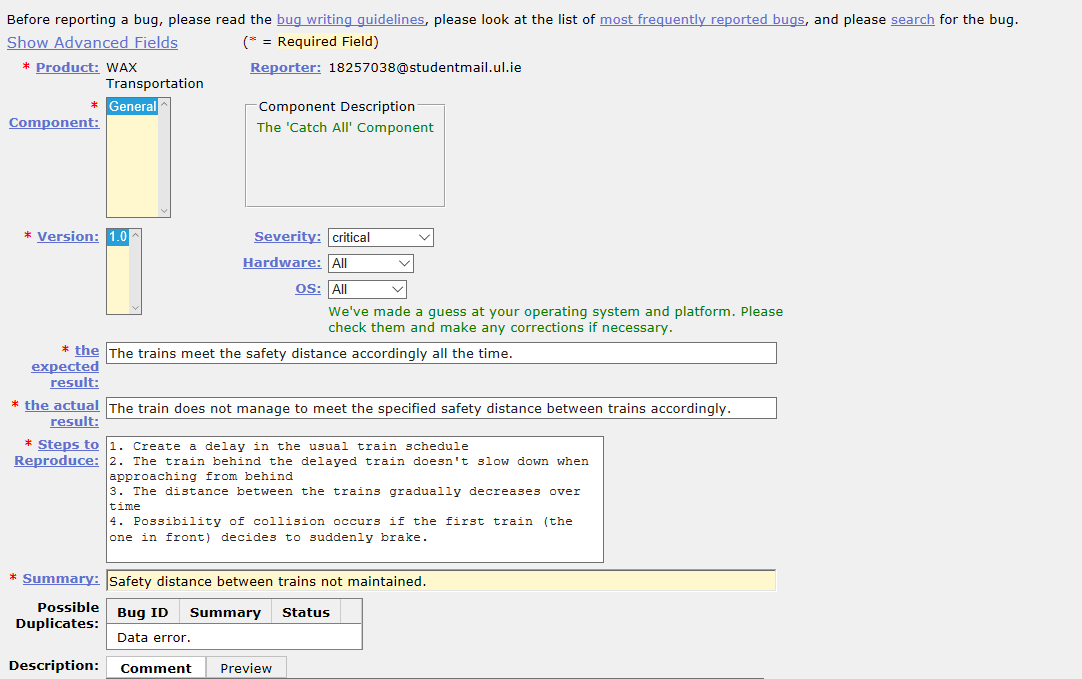
Expected Results: The trains meet the safety distance accordingly all the time.

Build Date & Hardware: build 1 for all hardware

Additional Builds and Platforms: none

Additional Information: importance critical

Screenshot of the submitted bug:



* 1. **Bug Report 6 - Trains don't slow down before a zone with a lower speed limit.**

Component: General

OS: All

Summary: Trains don't slow down before a zone with a lower speed limit.

Overview: WAX Transportation

Steps to Reproduce:

|  |
| --- |
| 1. Make a train accelerate and subsequently maintain the speed of a certain block.  2. Have the train enter a block with a lower speed limit.  3. The train decelerates only after entering the block with a lower speed limit. |
|  | |

Actual Results: Trains exceed the specified speed limit when entering a block with a lower speed limit.

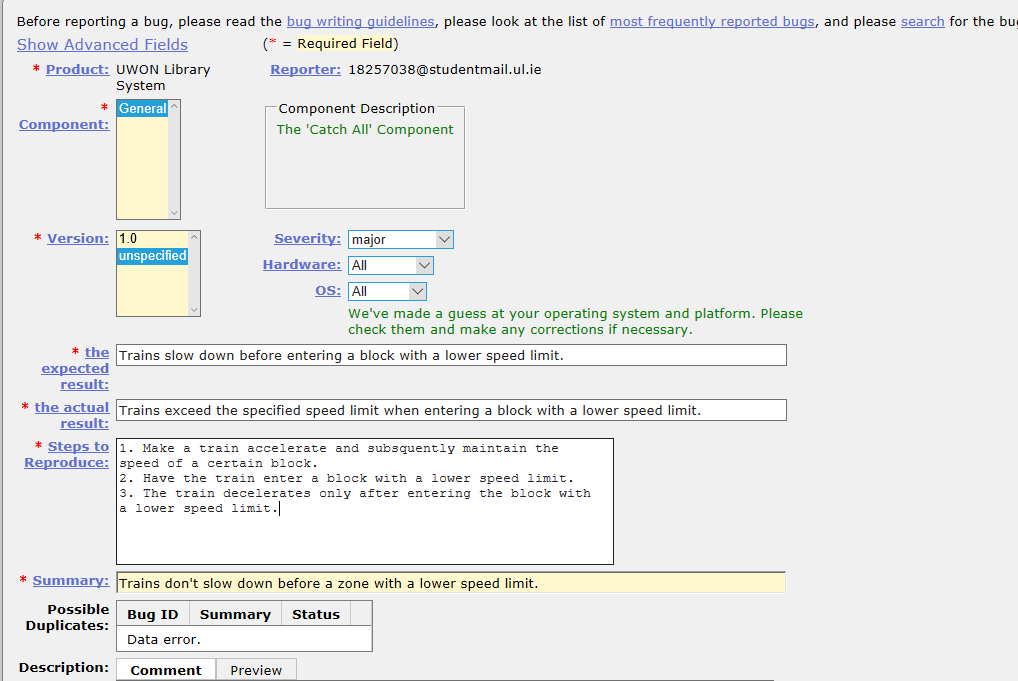
Expected Results: Trains slow down before entering a block with a lower speed limit.

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: none

Additional Information: importance major

Screenshot of the submitted bug:



* 1. **Bug Report 7 - Trains don't slow down before approaching a curve.**

Component: General

OS: All

Summary: Trains don't slow down before approaching a curve.

Overview: WAX Transportation bug

Steps to Reproduce:

|  |
| --- |
| 1. Have a train move at high speed  2. Have the train approach the curve on the tracks.  3. Have the train pass over the curve. |
|  | |

Actual Results: Trains do not slow down before curves on the train tracks.

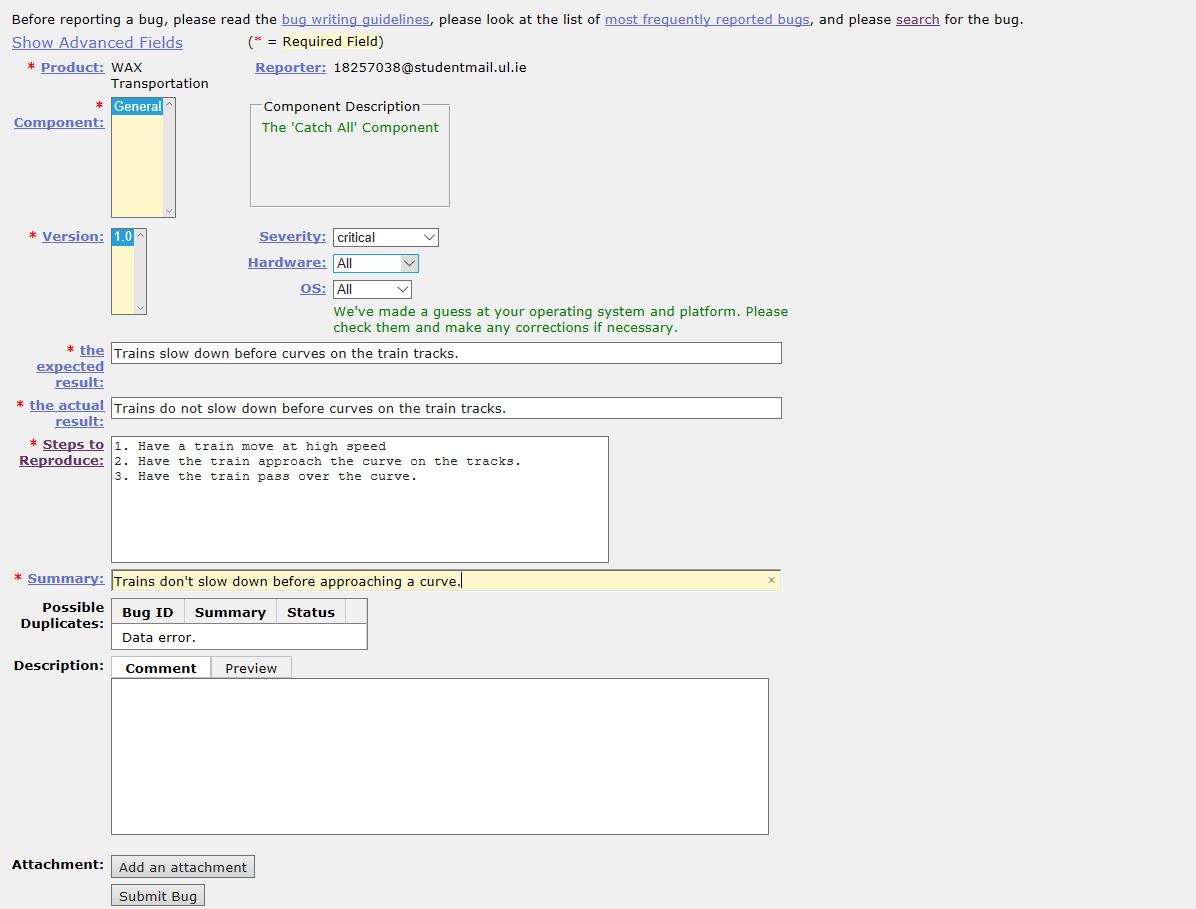
Expected Results: Trains slow down before curves on the train tracks.

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: none

Additional Information: Importance critical

Screenshot of the submitted bug:



* 1. **Bug Report 8 - Train moves before doors close.**

Component: General

OS: All

Summary: Train moves before doors close.

Overview: WAX Transportation bug

Steps to Reproduce:

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | 1. Train arrives in a station.  2. Train stops.  3. Train opens the doors.  4. Passengers board the train.  5. The boarding time finishes.  6. Train tries to close the doors.  7. Block the train doors from closing properly.  8. The train will still depart even without waiting for the obstacle to be removed and the doors still open. | |  | | |
|  | |

Actual Results: The train starts moving before the doors close properly.

Expected Results: The train does not move as long as the doors are closed.

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: None

Additional Information: Importance critical

* 1. **Bug Report 9 - Train announcement system working incorrectly.**

Component: General

OS: All

Summary: Train announcement system working incorrectly.

Overview: WAX Transportation bug

Steps to Reproduce:

|  |
| --- |
| 1. Have the train prepare for departure  2. Block the train doors.  3. Have the train get delayed enough for the announcement to play. |
|  | |

Actual Results: The speakers announce departure even if the train has not departed.

Expected Results: The speakers announce departure upon the actual departure

Build Date & Hardware: build 1 for all hardware

Additional Builds and Platforms: None

Additional Information: Importance minor

* 1. **Bug Report 10 - Non-Latin city names displayed incorrectly.**

Component: App

OS: All

Summary: Non-Latin city names displayed incorrectly.

Overview: WOW Online Flight Reservation System bug

Steps to Reproduce:

1. Open the app

2. Search for a non-Latin city name.

Actual Results: City names in non-Latin alphabet displayed with black boxes in the app.

Expected Results: All city names displayed properly in the app

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: None

Additional Information: importance major

* 1. **Bug Report 11 - Printing of the e ticket is not supported in non-Latin alphabets.**

Component: Backend

OS: All

Summary: Printing of the e ticket is not supported in non-Latin alphabets.

Overview: WOW Online Flight Reservation System bug

Steps to Reproduce:

1. Order an e ticket.

2. Choose to print the ticket.

3. Select the language to print it in. (in this case a non-Latin language such as Russian.)

4. Print the ticket.

Actual Results: The letters that are meant to be non-Latin characters are displayed as weird black boxes.

Expected Results: The e ticket when printed in a non-Latin alphabet is printed correctly.

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: None

Additional Information: Importance major

* 1. **Bug Report 12 - When doing a multi-step flight, the system marks you as a frequent flyer.**

Component: App

OS: All

Summary: When doing a multi-step flight, the system marks you as a frequent flyer.

Overview: WOW Online Flight Reservation System bug

Steps to Reproduce:

1. Book a multi-step flight.

2. Take the flight.

3. Look up the frequent flyer info to see if you are on it.

Actual Results: The system marks you as a frequent flyer after just one multi step flight.

Expected Results: When doing a multi-step flight the system does not mark you as a frequent flyer.

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: None

Additional Information: importance minor

* 1. **Bug Report 13 - When a flight is full it doesn’t stop customers from buying more tickets.**

Component: App

OS: All

Summary: When a flight is full it doesn’t stop customers from buying more tickets.

Overview: WOW Online Flight Reservation System bug

Steps to Reproduce:

1. Book a plane ticket.

2. Keep booking the plane tickets until the flight is full

3. Book again.

4. Check if your booking went through.

Actual Results: The customer/s is/are able to buy tickets for a completely full flight.

Expected Results: The customer cannot purchase more tickets for a flight that has no more empty seats.

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: None

Additional Information: Importance critical

* 1. **Bug Report 14 - The currency exchange doesn’t update fast enough.**

Component: Backend

OS: All

Summary: The currency exchange doesn’t update fast enough.

Overview: WOW Online Flight Reservation System bug

Steps to Reproduce:

1. Look up current currency exchange rates.

2. Exchange currency with the currency exchange system.

3. Compare the results.

Actual Results: The currency exchange updates too slowly resulting in mismatched payments.

Expected Results: The currency exchange updates at sufficient speed.

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: None

Additional Information: Importance major

* 1. **Bug Report 15 - Cancelling a flight does not notify the system of a seat being freed.**

Component: Backend

OS: All

Summary: Cancelling a flight does not notify the system of a seat being freed.

Overview: WOW Online Flight Reservation System bug

Steps to Reproduce:

1. Order a seat online.

2. Cancel said order.

3. Check the system to see if that seat is still taken.

Actual Results: The system is not notified of a seat being freed.

Expected Results: The system is notified of a seat being freed.

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: None

Additional Information: Importance major

* 1. **Bug Report 16 - Time zone is not adjusted for all the users.**

Component: General

OS: All

Summary: Time zone is not adjusted for all the users.

Overview: WSS Meetings bug

Steps to Reproduce:

1. Create a meeting.

2. Change the timezone in your system and log in from a different account

3. Look at the timezone set for the meeting.

Actual Results: The timezone is set to the timezone of the user who organised the meeting.

Expected Results: The timezone is adjusted according to the timezones of the users for each user.

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: None

Additional Information: Importance critical

* 1. **Bug Report 17 - The system doesn’t calculate the preferred time accurately.**

Component: General

OS: All

Summary: The system doesn’t calculate the preferred time accurately.

Overview: WSS Meetings bug

Steps to Reproduce:

|  |
| --- |
| 1. Create a meeting  2. setup your schedule.  3. Have another person join or join with another account.  4. repeat steps 2 and 3 until the preferred meeting time becomes strange such as 26:150 on the 50th of January. |
|  | |

Actual Results: The system does not calculate the preferred time for all participants.

Expected Results: The system accurately calculates the preferred time for all participants.

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: None

Additional Information: Importance major

* 1. **Bug Report 18 - Everyone is able to initiate a meeting.**

Component: General

OS: All

Summary: Everyone is able to initiate a meeting.

Overview: WSS Meetings bug

Steps to Reproduce:

|  |
| --- |
| 1. Create a meeting.  2. Have people join or join with separate accounts.  3. Give some accounts the authorization to start a meeting and some not.  4. Test each account to see if they are able to start a meeting.  5. The unauthorized accounts are able to initiate the meeting. |
|  | |

Actual Results: Meeting initiation can be performed by every participant of the meeting.

Expected Results: Meeting initiation should be reserved to authorized people

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: None

Additional Information: Importance major

* 1. **Bug Report 19 - System does not update after a book is returned.**

Component: General

OS: All

Summary: System does not update after a book is returned.

Overview: UWON Library System bug

Steps to Reproduce:

|  |
| --- |
| 1. Borrow a book from the library.  2. System shows that the book is borrowed.  3. Return the book and update information in the system.  4. Observe the book is still "borrowed" in the system. |
|  | |

Actual Results: The book is still borrowed in the system after being returned

Expected Results: The book is available to borrow when it is returned

Build Date & Hardware: Build unspecified for all hardware

Additional Builds and Platforms: None

Additional Information: Importance enhancement

* 1. **Bug Report 20 - Booking more rooms than the limit.**

Component: General

OS: All

Summary: Booking more rooms than the limit.

Overview: UWON library System bug

Steps to Reproduce:

1. Book a study room online.

2. Book another room in quick succession. (E.g. another tab)

3. Both rooms will be booked.

Actual Results: Sometimes you can book more than one room at a time.

Expected Results: Should only be able to book one study room at a time

Build Date & Hardware: Build 1 for all hardware

Additional Builds and Platforms: none

Additional Information: importance normal

## **Task #2: Test Design and Development using JUnit**

**The Log:**

* Started and completed 8 JUnit tests on Tuesday 12.11.2019
* Finished another 12 JUnit tests on Friday 15.11.2019

**Reflections:**

Jakub: I think the most important thing I got from this part is that no matter how difficult the code you’re testing is, you will always follow the same steps.

Aleksandr: I learned how to think about code from the perspective of the developer and the perspective of the bug tester.

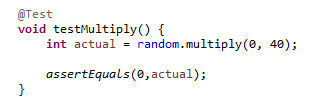
Eoin**:** I learned more about JUnit testing and using Eclipse IDE. I also had to think about the strategies for testing different tests.

Shuo: I have developed a better understanding of White-box testing and Black-box testing. Also I learned how to design a test according to the logic of the code and tried to get the test a 100% coverage.

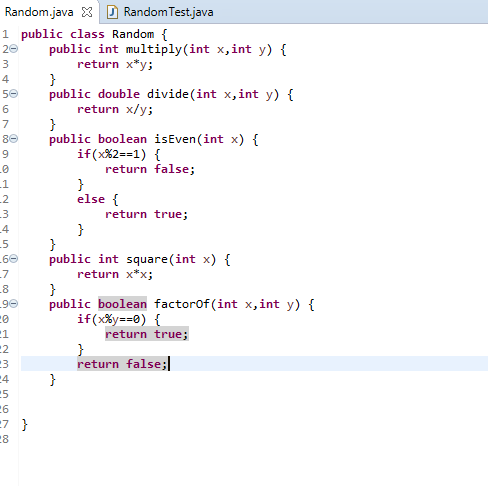
**4.1 TEST #1 – Multiplying**

We tested the maximum and minimum integer sizes, negative numbers and 0 in different test cases.

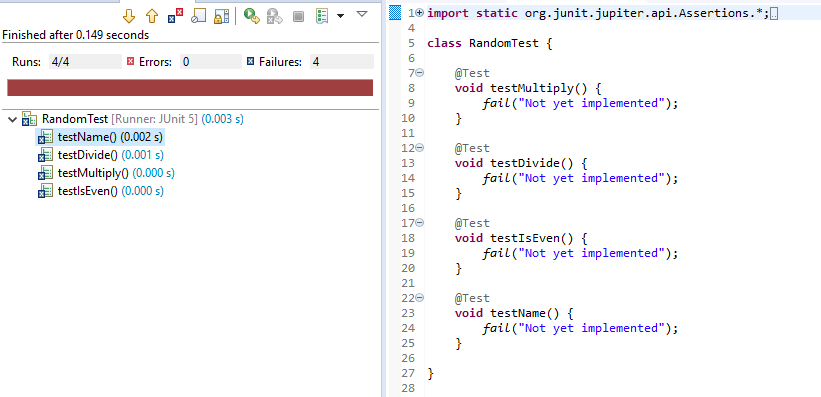
**4.1.1 Code developed for the test**

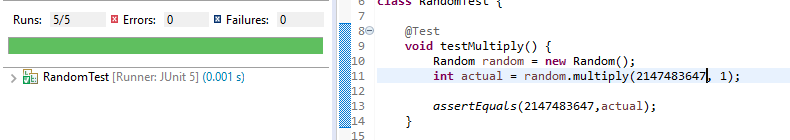
. 

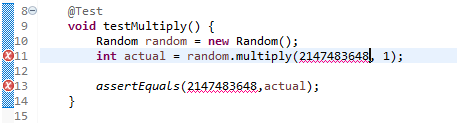
**4.1.2 Code developed for the program**

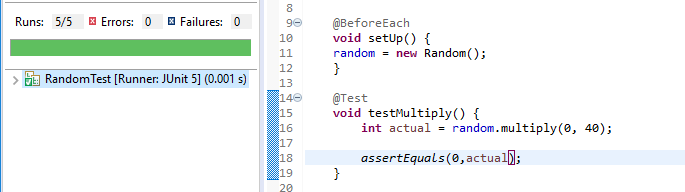


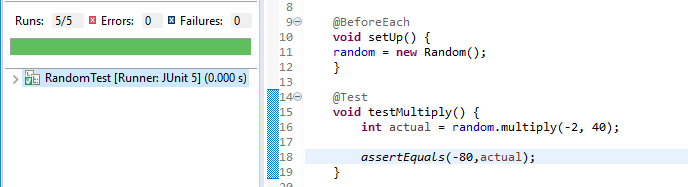
**4.1.3 Screenshots of failure and success**







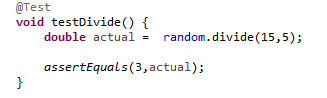




**4.2 TEST #2 – Dividing**

We checked for negative divisions, dividing by 0, the maximum values and for integer overflow.  
We also checked different scopes of numbers.

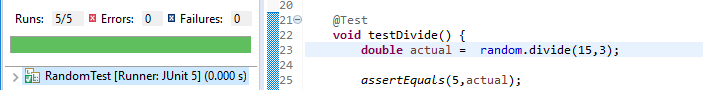
**4.2.1 Code developed for the test**

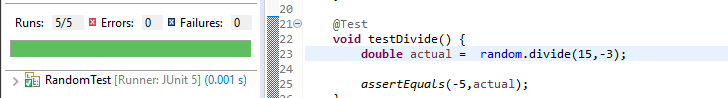


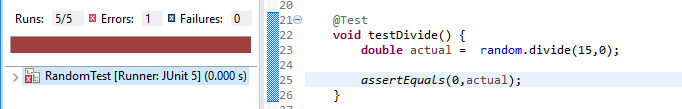
**4.2.2 Code developed for the program**

C:\Users\c18241646\AppData\Local\Microsoft\Windows\INetCache\Content.Word\originalcode.png

**4.2.3 Screenshots of failure and success**



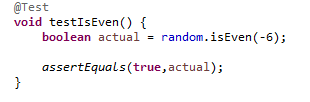




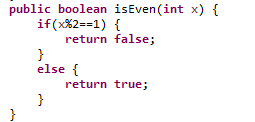
**4.3 TEST #3 – Checking if a number is Even.**

We tested both even and odd numbers, as well as positive and negative. We also checked 0 to see how that affected the code.

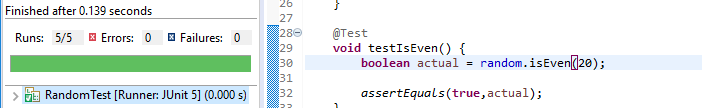
**4.3.1 Code developed for the test**

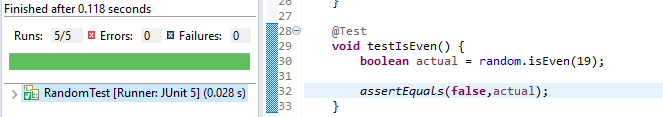
****

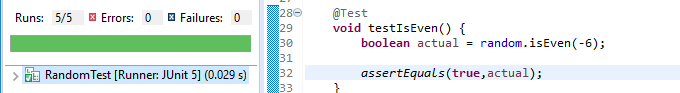
**4.3.2 Code developed for the program**

****

**4.3.3 Screenshots of failure and success**

****

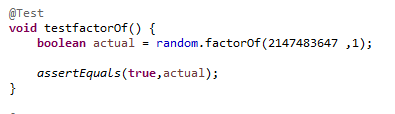
****

****

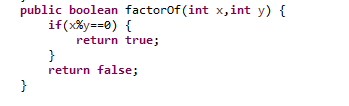
**4.4 TEST #4 – Is y factor of x**

We tested different scopes of numbers and if negative numbers properly counted as factors.  
We also made sure that the number itself counted as a factor of itself. As well as this , we made sure the maximum integer worked in the code.

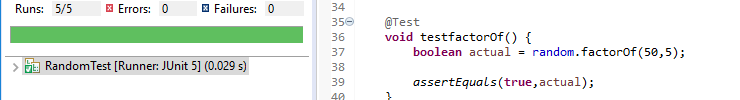
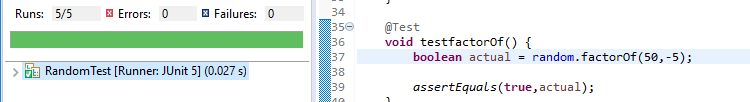
**4.4.1 Code developed for the test**

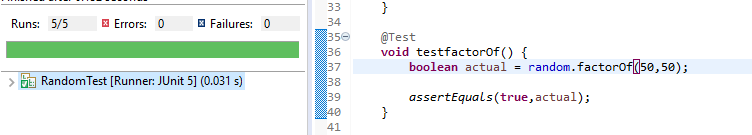
****

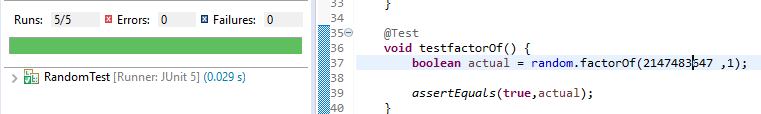
**4.4.2 Code developed for the program**

****

**4.4.3 Screenshots of failure and success**

****

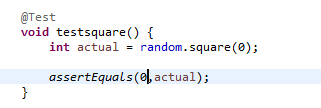
****

****

**4.5 TEST #5 – Squaring**

We made sure that the code worked as intended, and that negative numbers squared properly.  
We also made sure that 0 worked in the code.

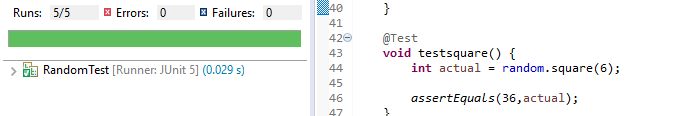
**4.5.1 Code developed for the test**

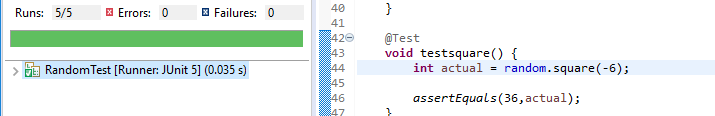
****

**4.5.2 Code developed for the program**

**C:\Users\c18241646\AppData\Local\Microsoft\Windows\INetCache\Content.Word\originalcode.png**

**4.5.3 Screenshots of failure and success**

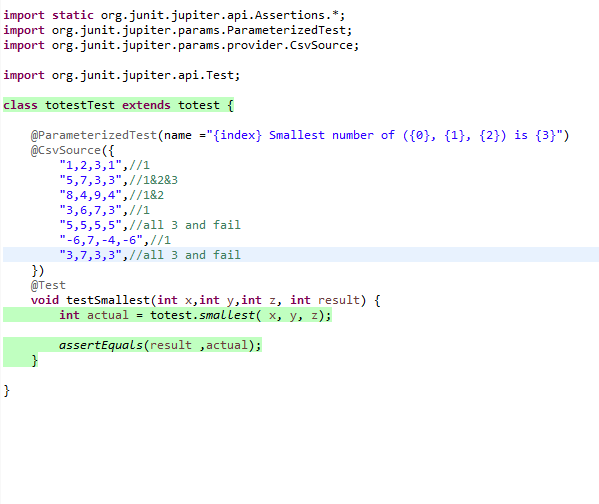
****

****

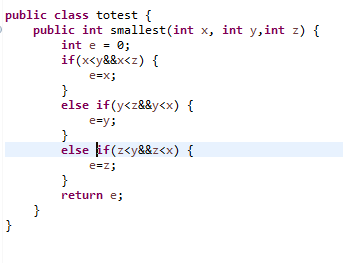
**4.6 TEST #6 – Smallest Number**

We made sure each of the variables worked as intended. We tested what happens when two or more of the numbers are the same value. We inputted the smallest integer into different slots to make sure we thoroughly tested every line of code. The tests covered 100% of the code.

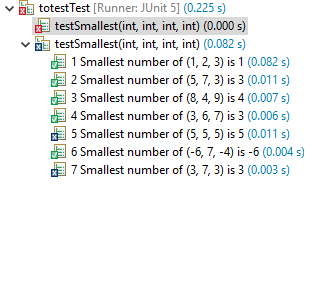
**4.6.1 Code developed for the test**



**4.6.2 Code developed for the program**



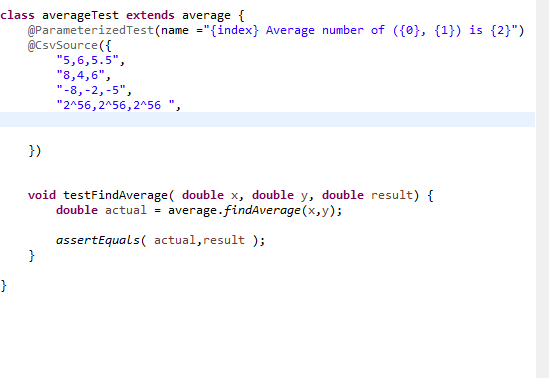
**4.6.3 Screenshots of failure and success**



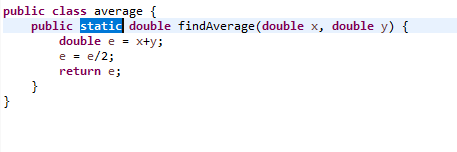
**4.7 TEST #7 –Average Number**

We thought about decimal and negative numbers. We also had to think about double overflow.

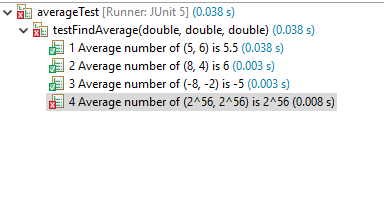
**4.7.1 Code developed for the test**



**4.7.2 Code developed for the program**



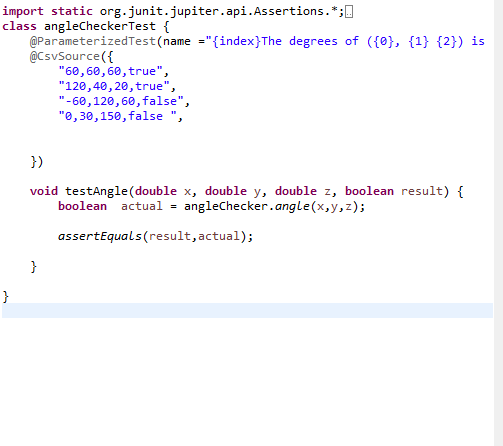
**4.7.3 Screenshots of failure and success**



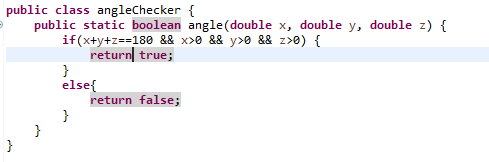
**4.8 TEST #8 – Angle Checker in a Triangle**

We used white box testing for this test case .We tested for negative values as the angle sizes because angles can’t be a negative number. They also can’t be angled at 0 degrees so we tested this also.

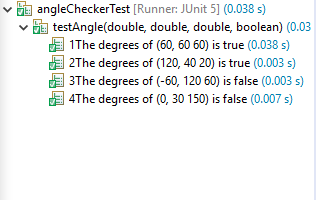
**4.8.1 Code developed for the test**



**4.8.2 Code developed for the program**



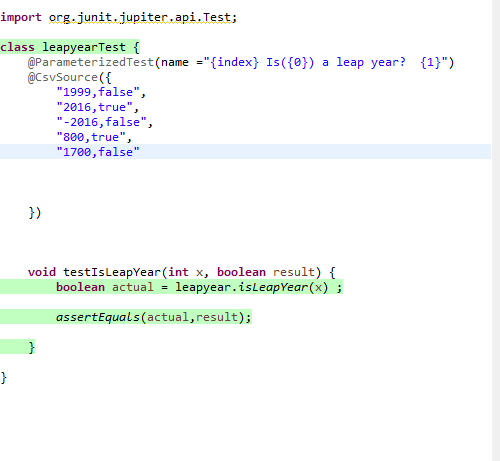
**4.8.3 Screenshots of failure and success**



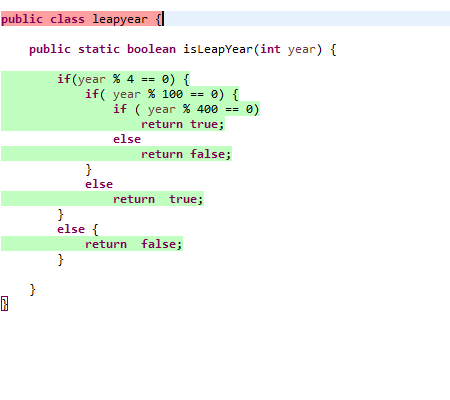
**4.9 TEST #9 –Leap Year Checker**

We covered 87% of the program code in our test. We tested negative numbers, and years that are divisible by 4 yet are not leap years. We tested both leap years and non leap years.

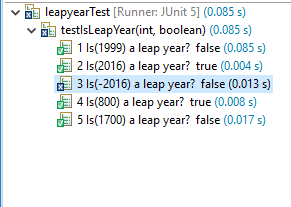
**4.9.1 Code developed for the test**



**4.9.2 Code developed for the program**



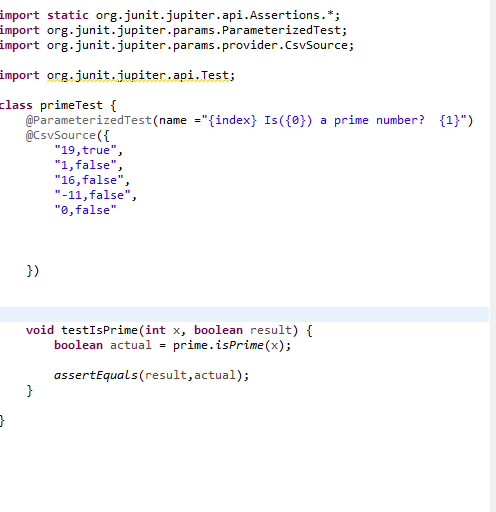
**4.9.3 Screenshots of failure and success**



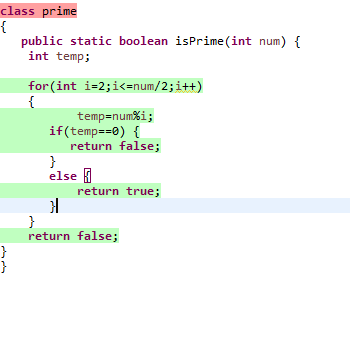
**4.10 TEST #10 – If a number is prime.**

For this test we used black box testing. We checked primed and nonprime numbers, as well as 1, 0, and unnatural numbers. We covered 86% of the program code.

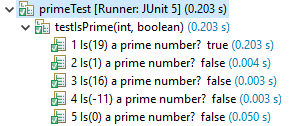
**4.10.1 Code developed for the test**



**4.10.2 Code developed for the program**



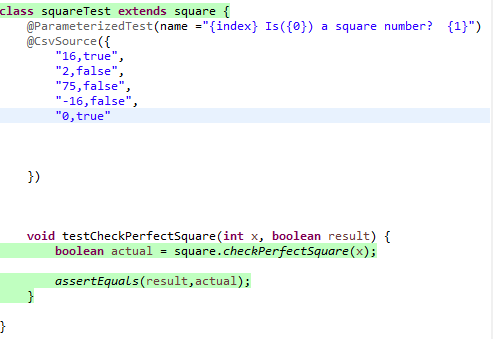
**4.10.3 Screenshots of failure and success**



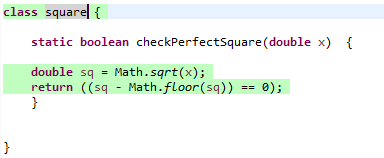
**4.11 TEST #11 – If a number is a Square number**

We used black box testing for this test case. We checked square and non square values , such as negative numbers and 0. We covered 100% of the original code in our tests.

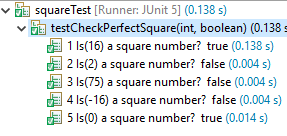
**4.11.1 Code developed for the test**



**4.11.2 Code developed for the program**

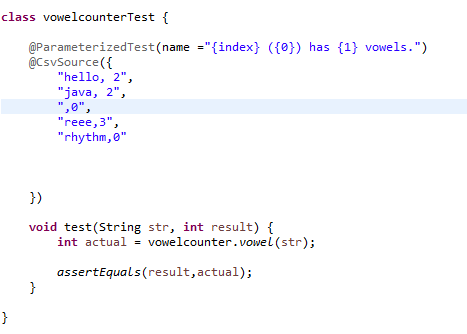


**4.11.3 Screenshots of failure and success**

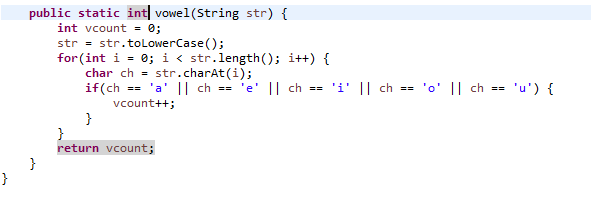


**4.12 TEST #12 –Average Number**

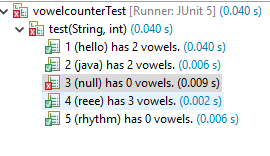
**4.12.1 Code developed for the test**



**4.12.2 Code developed for the program**



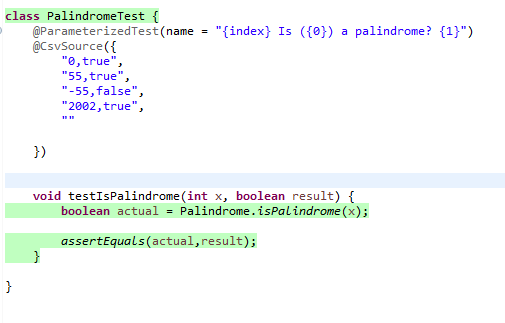
**4.12.3 Screenshots of failure and success**



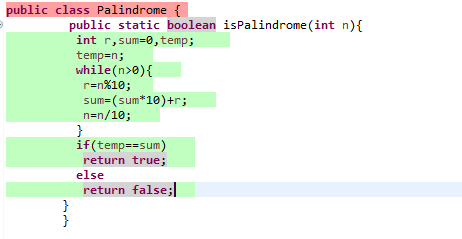
**4.13 TEST #13 –Palindrome Number**

In this test we covered 90.3% of the code. We checked both palindrome numbers and non-palindrome numbers, such as 0 and negative numbers.

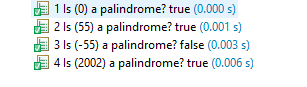
**4. 13.1 Code developed for the test**



**4. 13.2 Code developed for the program**



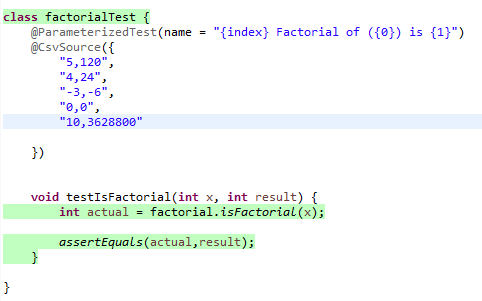
**4. 13.3 Screenshots of failure and success**



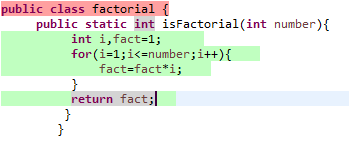
**4.14 TEST #14 –Factorials**

We tested 83% of the code in this test. We tested negative numbers, aswell as 0 and different positive numbers.

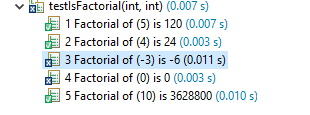
**4. 14.1 Code developed for the test**



**4. 14.2 Code developed for the program**



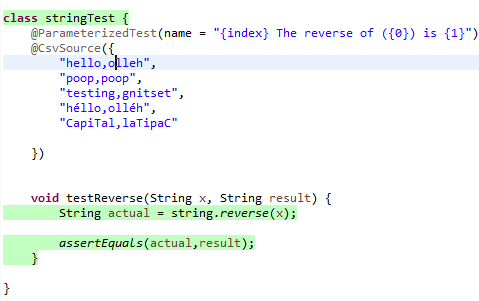
**4. 14.3 Screenshots of failure and success**



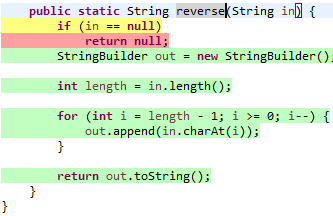
**4.15 TEST #15 –Reversing a String**

In this test we covered 84% of the code. We tested words with accents, as well as words with capital letters.

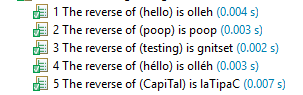
**4.15.1 Code developed for the test**



**4.15.2 Code developed for the program**



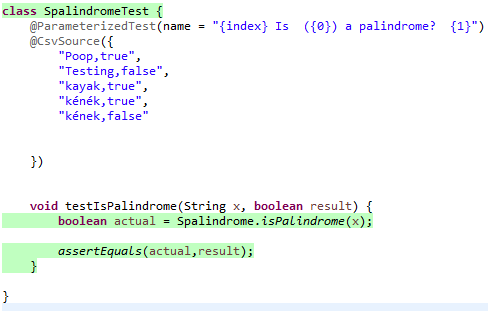
**4.15.3 Screenshots of failure and success**



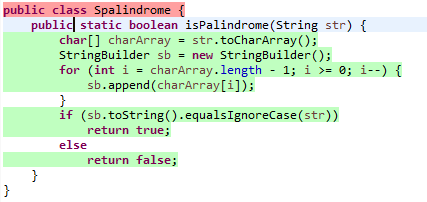
**4.16 TEST #16 –Average Number**

In this test we checked to see if characters with accents were equal to the same character without accents. We checked for this also with capital letters. We covered 91% of the code in this test.

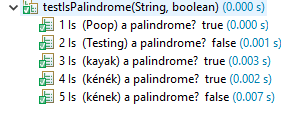
**4.16.1 Code developed for the test**



**4.16.2 Code developed for the program**



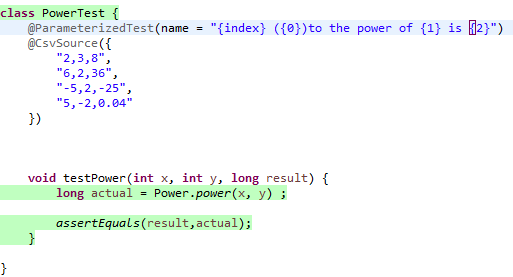
**4.16.3 Screenshots of failure and success**



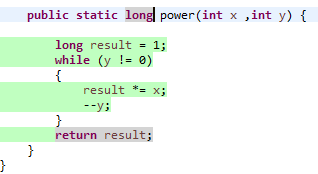
**4.17 TEST #7 –A number to the power of another number**

In this test we checked to see if negative powers gave the correct values, as well as a negative number to the power of an even number. We covered 81% of the code.

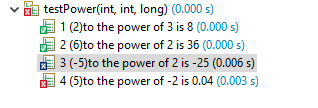
**4.17.1 Code developed for the test**



**4.17.2 Code developed for the program**



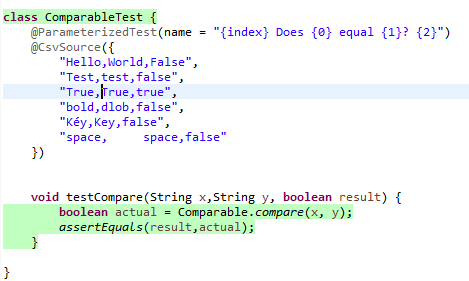
**4.17.3 Screenshots of failure and success**



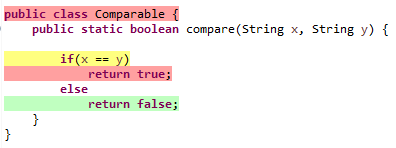
**4.18 TEST #18 –Comparing two Strings**

In this test we checked if capitals were seen as different to lowercase, as well as accents on letters. We also checked if spaces would make one string differ to another. We coved 71% of the code in this test

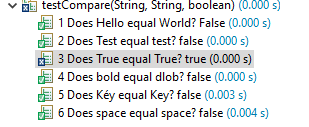
**4.18.1 Code developed for the test**



**4.18.2 Code developed for the program**



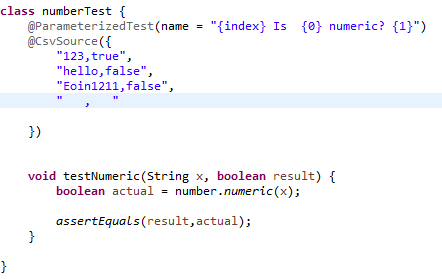
**4.18.3 Screenshots of failure and success**



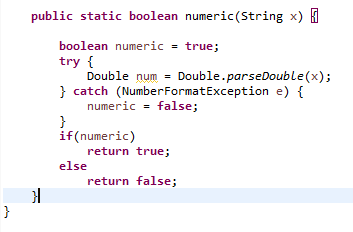
**4.19 TEST #19 –Numeric String**

In this test we checked numeric and non-numeric Strings. We also tested Strings that had both letters and numbers in it, and NULL Strings. We had a coverage of 84.2% in this test

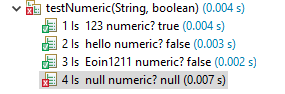
**4.19.1 Code developed for the test**



**4.19.2 Code developed for the program**



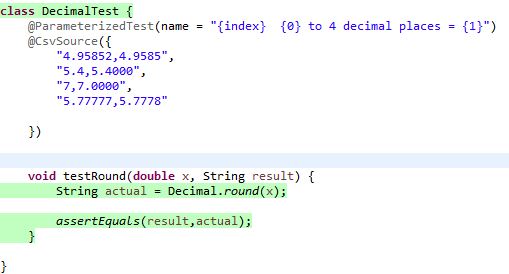
**4.19.3 Screenshots of failure and success**



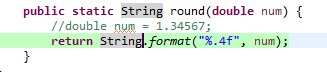
**4.20 TEST #20 –Rounding to 4 decimal places**

In this test we tested numbers with more and with less than 4 decimal places, as well as numbers with none. 76.9% of the code was covered in this test.

**4.20.1 Code developed for the test**



**4.20.2 Code developed for the program**



**4.20.3 Screenshots of failure and success**

