

CS4004

SOFTWARE TESTING AND INSPECTION PROJECT

Written by

Laszlo Szlatki

Patrick James O'Neill

Michelle Hourihan Watanabe

Richard O'Brien

Submitted November 2019

Contents

1.	Bugzilla Reporting	3
1.1	Bug reporting log	3
1.2	Reflections	4
1.3	Evidence	5
1.3.1	Bug report	5
1.3.2	Bug report	6
1.3.3	Bug report	7
1.3.4	Bug report	7
1.3.5	Bug report	8
1.3.6	Bug report	9
1.3.7	Bug report	10
1.3.8	Bug report	11
1.3.9	Bug report	12
1.3.10	Bug Report	12
1.3.11	Bug report	13
1.3.12	Bug report	14
1.3.13	Bug Report	15
1.3.14	Bug Report	15
1.3.15	Bug Report	16
1.3.16	Bug Report	17
1.3.17	Bug Report	18
1.3.18	Bug Report	19
1.3.19	Bug Report	19
1.3.20	Bug Report	20
2	Test Design & Development using JUNIT.	26
2.1	TDD log.....	26
2.2	Reflections	30
2.3	Evidence.....	29
2.3.1	TDD for password generator	29
2.3.2	TDD for Paint Calculator	33
2.4	Reflections	39
3	Coverage Criteria Based Testing	37

3.1	Coverage Criteria Log.....	40
3.2	Reflections	37
3.3	Evidence.....	38
3.3.1	Coverage criteria.....	38
3.3.1.1	Example 1. Testing a single 'if' statement.	38
3.3.1.2	Example 2 – Testing nested 'if' statements.....	45
3.3.1.3	STATEMENT COVERAGE.....	49
3.3.1.4	BRANCH TESTING COVERAGE	52

1. Bugzilla Reporting

1.1 Bug reporting log

Subtask	Title	author	Date
Bug Report 1.3.1	All airports are listed in list of destinations, even if not served by chosen airline.	Laszlo Szlatki	10/18/2019
Bug Report 1.3.2	Department member cannot request journal subscriptions.	Laszlo Szlatki	10/18/2019
Bug Report 1.3.3	Some e-mail invitations are not being sent.	Laszlo Szlatki	10/18/2019
Bug Report 1.3.4	A train is allowed to enter an already occupied parking track.	Laszlo Szlatki	10/18/2019
Bug Report 1.3.5	Meeting initiation meant for authorized personnel is also being shared with unauthorized personnel.	Patrick O'Neill	10/18/2019
Bug Report 1.3.6	The door of one of the new autonomous trial trains has been not closing and opening mid trip.	Patrick O'Neill	10/18/2019
Bug Report 1.3.7	Library system is giving different dates for availability and return.	Patrick O'Neill	10/18/2019
Bug Report 1.3.8	App crashing once combined flights is ready to be purchased.	Patrick O'Neill	10/18/2019
Bug Report 1.3.9	No dropdown box appears to distinguish between adult and child passengers.	Michelle Hourihan Watanabe	10/25/2019
Bug Report 1.3.10	Members unable to search catalogue offsite as error page shows.	Michelle Hourihan Watanabe	10/25/2019
Bug Report	Doors not closing within specified time,	Michelle	10/25/2019

1.3.11	causing several attempts.	Hourihan Watanabe	
Bug Report 1.3.12	Meetings links being sent in undesired time zone.	Michelle Hourihan Watanabe	10/25/2019
Bug Report 1.3.13	Software Double Booking Rooms cause customer's to be not get meeting.	Richard O'Brien	10/17/2019
Bug Report 1.3.14	Self scan service of books not recording to account.	Richard O'Brien	10/17/2019
Bug Report 1.3.15	Latency of data causes flight reservation problems.	Richard O'Brien	10/17/2019
Bug Report 1.3.16	Book ticket button not working on all Browsers.	Richard O'Brien	10/17/2019
Bug Report 1.3.17	Child price given the wrong price for their age.	Richard O'Brien	11/1/2019
Bug Report 1.3.18	Trains directed to the wrong terminal, cause customer's to be late for flights.	Richard O'Brien	11/1/2019
Bug Report 1.3.19	Wrong Booking type cause second client not booked in Company ID instead of Customer ID	Richard O'Brien	11/1/2019
Bug Report 1.3.20	Wrong date recorded on book loan, causes strike on several Customer's.	Richard O'Brien	11/1/2019

1.2 Reflections

Michelle's Reflection

Overall, I felt pleased with the process we employed of discussing the case studies when creating a list of bugs, as this helped me to understand the types of bugs that can be reported. However, after writing up my reports independently, I realized that many of them had summaries that were far too long. Getting feedback from the group helped me to review my summaries. Reading each other's reports and seeing how we summarized our issues; how clear our descriptions were; and whether our steps to reproduction were 'reproducible', really helped me understand the importance of reporting in a clear way to different audiences.

Laszlo's Reflections

To create the bug reports in Bugzilla, I had to gain a thorough understanding of the bug reporting process and current industry best practices. At the beginning, I found it difficult to write the summary, as it was too long, compare to other examples.

Paddy's Reflections

After writing the bug reports and attending labs and lectures I came to understand that bug reports have to be clear. My first bug reports were not very concise. I realised by adding bullet-points into the reproduction steps they could be clearer. When reviewing my reports, I tried to think of the people reading it and ask myself if I was getting the main points across. Will the next person be able to understand what the problem is? When writing a bug report, you have to be specific but not by writing essays. The language you use has to be understandable and not used in an inappropriate tone. "The point of writing problem report (bug report) is to get bugs fixed" – By Cem Kaner

Richard's Reflections.

I found that after writing the bug report's that help me get the confidence that I needed. I had to be very direct and specific so another person could read them. The steps that I went into for the step to reproduce was aimed for someone that never used the product in their life. I felt that I learnt a lot from doing the bug reports on this assignment, as I aided and got help from other members of the team.

1.3 Evidence

1.3.1 Bug report

WOW Online Flight Reservation System

Status: CONFIRMED

Alias: None

Product: WOW Online Flight Reservation System

Component: Web Frontend

Version: 1.1.6

Severity: minor

Hardware: PC

OS: Linux

Assignee: Patrick O'Neill

URL: <https://wowflights.com>

Personal Tags: Incorrect destination list

Depends On: 6314: Germany as destination country inaccurate

Blocks: N/A

Reported: 2019-10-18 12:25 IST by Laszlo Szlatki

Modified: 2019-10-18 12:26 IST

Orig. Est.: 1.0

Current Est: 1.0

Hours Worked: 0.5

Hours Left: 0.5

% Complete: 50%

Deadline: 2019-11-17

The expected result: After selecting the departure airport and airline, only the destinations offered by said airline are available to choose from.

The actual result: After selecting departure airport and airline, still all destination airports are visible.

Steps to reproduce:

1) Navigate to <https://wowflights.com> using a PC running on Linux OS

- 2) Select departure country from list
- 3) Select departure airport from drop down menu
- 4) Select from available airlines from drop down menu
- 5) Select Germany as destination country from list
- 6) Select destination airport

Summary: All airports are listed in list of destinations even if not served by chosen airline

Description: After departure Ireland, Shannon and WizzAir are selected, in the destination country field, the options are filtered to only those served by the chosen airline. If Germany is selected as destination, all airports are listed as possible arrival locations, despite some not being served by the WizzAir. When I try to select such an airport, the system warns that "the selected airline does not service this destination".

1.3.2 Bug report

UWON Library System

Status: CONFIRMED

Alias: None

Product: UWON Library System

Component: Backend

Version: 1.2.2

Severity: normal

Hardware: PC

OS: Windows 10

Assignee: Richard O'Brien

URL: <https://UWON.library.com>

Personal Tags: New subscriptions

Depends On: 6426: Inconsistent tracking of staff ID

Blocks: 6118: New subscriptions to certain journals

Reported: 2019-10-18 13:25 IST by Laszlo Szlatki

Modified: 2019-10-18 14:19 IST

Orig. Est.: 5.0

Current Est: 5.0

Hours Worked: 2.0

Hours Left: 3.0

% Complete: 40%

Deadline: 2019-11-17

The expected result: Head of Department can request new journal subscriptions

The actual result: None of the Nursing Department members can request new journal subscriptions

Steps to reproduce:

- 1) Log in to UWON's library system using a Nursing department members credentials
- 2) Navigate to journals
- 3) Click on journal subscriptions
- 4) Click on request a new journal subscription

Summary: Department member cannot request journal subscriptions

Description: After logging in with Head of Nursing Department credentials, I navigated to the section where a new journal subscription can be requested. I filled out all necessary details to request a new subscription to the "Nurse Education Today" journal, but when I clicked on the Submit button, the system displayed an error message: "The request has been denied, you don't have rights to request new journal subscriptions."

1.3.3 Bug report

WSS Meetings

Status: CONFIRMED

Alias: None

Product: WSS Meetings

Component: General

Version: 1.0

Severity: major

Hardware: PC, Mac

OS: Windows, Mac OS, Linux

Assignee: Michelle Hourihan Watanabe

URL: <http://www.wssmeetings.com>

Personal Tags: Unsent invitations

Depends On: N/A

Blocks: 5780: Meeting initiation meant for authorized personnel is also being shared with unauthorized personnel

Reported: 2019-10-18 13:59 IST by Laszlo Szlatki

Modified: 2019-10-18 14:00 IST

Orig. Est.: 1.0

Current Est: 1.0

Hours Worked: 0.5

Hours Left: 0.5

% Complete: 50%

Deadline: 2019-11-17

The expected result: All invited parties receive meeting invitations sent by e-mail regardless of service provider.

The actual result: Account holders of free e-mail providers not receiving meeting invites

Steps to reproduce: 1) Log in to <http://www.wssmeetings.com>

2) Fill out necessary details about the meeting

3) Add invited parties

3/a) Invite people with professional e-mail addresses

3/b) Invite people with e-mail addresses from free providers

Summary: Some e-mail invitations are not being sent

Description: After inviting parties to attend the meeting, I realised some of the targeted addressees didn't receive an invitation. Upon closer examination it was obvious that participants with e-mail addresses from @gmail.com, @bigmailbox.com and @qq.com did not receive invitations. All these are free e-mail address provider domains.

1.3.4 Bug report

WAX Transportation

Status: CONFIRMED

Alias: None

Product: WAX Transportation

Component: Parking track

Version: 1.0

Severity: major

Hardware: PTS2K39

OS: Linux

Assignee: Laszlo Szlatki

URL: N/A

Personal Tags: Parking track

Depends On: 5782: train door opens while moving

Blocks: 6649: Order of trains leaving parking track

Reported: 2019-10-18 14:34 IST by Laszlo Szlatki

Modified: 2019-10-22 13:53 IST

Orig. Est.: 3.0

Current Est: 2.0

Hours Worked: 1.5

Hours Left: 0.5

% Complete: 75%

Deadline: 2019-11-17

The expected result: Train is being directed to an empty parking track

The actual result: Train was directed to a parking track, where a train was already parking

Steps to reproduce: 1) Divert trains both on track PT01, PT02 and PT03 and park them

2) Remove the train from PT02 or PT03

3) Direct a train to an unspecified parking track

Summary: A train is allowed to enter an already occupied parking track

Description: After all parking tracks were in use, and one becomes available, the system directs the next train to be parked in PT01, a possibly still occupied track, instead of directing it to the next available empty track. It only appears once all parking tracks were filled, and PT01 is not the first one, that becomes empty.

1.3.5 Bug report

WSS Meetings

Status: CONFIRMED

Alias: None

Product: WSS Meetings

Component: Meeting initialization

Version: 1.0

Severity: Major

Hardware: PC

OS: Windows 10

Assignee: Laszlo Szlatki

URL: <http://www.wssmeetings.com>

Personal Tags: Initialization

Depends On: N/A

Blocks: N/A

Reported: 2019-10-18 14:21 IST by Patrick O'Neill

Modified: 2019-10-18 14:30 IST

Orig. Est.: 8.0

Current Est: 5.0

Hours Worked: 4.0

Hours Left: 1.0

% Complete: 80%

Deadline: 2019-11-17

The expected result: Meeting initiation should be restricted to authorized personnel.

The actual result: Meeting initiation being sent to unauthorized personnel.

Steps to Reproduce:

- Open WSS meetings.
- Compile meeting initiation.
- Add authorized personnel to recipients.
- Send meeting announcement.
- Look at the recipients after the meeting announcement has been sent.
- You can now see unauthorized people have been included and they can now respond to the announcement.

Summary: Meeting initiation meant for authorized personnel is also being shared with unauthorized personnel.

Description: A new meeting initiation system which was only supposed to include authorized personnel that had been selected for the meeting has been sent to unauthorized personnel that had not been included in the recipients. Unauthorized personnel can respond to the meeting announcement and can also see the private company information.

1.3.6 Bug report

Wax Transportations

Status: CONFIRMED

Alias: train door opens while moving

Product: WAX Transportations

Component: TDS56

Version: V3

Severity: major

Hardware: MacBook

OS: Mac OS

Assignee: Michelle Hourihan Watanabe

URL: <http://www.waxtransport.com>

Personal Tags: door opens and closes unexpectedly

Depends On: N/A

Blocks: 5781: A train is allowed to enter an already occupied parking track

Reported: 2019-10-18 15:20 IST by Patrick O'Neill

Modified: 2019-10-22 13:53 IST

Orig. Est.: 25.0

Current Est: 40.0

Hours Worked: 32.0

Hours Left: 8.0

% Complete: 80%

Deadline: 2019-11-17

The expected result: Opening and closing of train doors automatically while entering and exiting train station.

The actual result: Door not closing successfully and opening mid trip in between train stations.

Steps to Reproduce:

- Train enters station.
- Opens doors.
- Train closes doors.
- One door remains open.
- The doors are all opened and closed again successfully.
- While on route to the next station the same door that had a problem closing now opens mid-trip.

Summary: The door of one of the new autonomous trial trains has been not closing and opening mid trip

Description: While testing the new autonomous trains. The train enters the station. It passes sensor to allow opening of doors once it has stopped. Train closes doors after allowing people to enter and exit the train. One door has remained open. The doors are all opened and closed again successfully. While on route to the next station the same door that had a problem closing now opens mid trip.

1.3.7 Bug report

UWON Library systems

Status: CONFIRMED

Alias: None

Product: UWON Library System

Component: Backend

Version: 1.3.1

Severity: minor

Hardware: PC

OS: Windows 7

Assignee: Richard O'Brien

URL: <https://UWON.library.com>

Personal Tags: Different available and due dates on reservation

Depends On: N/A

Blocks: N/A

Reported: 2019-10-18 16:41 IST by Patrick O'Neill

Modified: 2019-10-18 16:45 IST

Orig. Est.: 2.5

Current Est: 2.5

Hours Worked: 0.0

Hours Left: 2.5

% Complete: 0%

Deadline: 2019-11-17

The expected result: Once a book has been reserved the system will let you know what date the book will be available.

The actual result: The system is giving the wrong date for availability.

Steps to Reproduce:

- Log onto the UWON library system.
- Reserve a book that has already been taken out.
- Check dates of availability.
- Cross reference with dates where the book has to be returned.

Summary:	Library system is giving different dates for availability and return
Description:	When a book is taken out it produces a return date automatically. When someone reserves the same book it should be given a date to pick up the book after it has been returned but the system is giving the wrong availability dates. It says that the book should be available for pick up even though the book is not due to be returned.

1.3.8 Bug report

WOW Online Flight Reservation System

Status: CONFIRMED

Alias: None

Product: WOW Online Flight Reservation System

Component: Android app

Version: 1.3.1

Severity: major

Hardware: Samsung Galaxy S7

OS: Android Marshmallow (6.0)

Assignee: Laszlo Szlatki

URL: <https://wowflights.com>

Personal Tags: Need to fix app crashing ASAP

Depends On: N/A

Blocks: 6127: Layout modification of online boarding passes

Reported: 2019-10-18 17:33 IST by Patrick O'Neill

Modified: 2019-11-08 10:54 GMT

Orig. Est.: 15.0

Current Est: 16.0

Hours Worked: 12.5

Hours Left: 3.5

% Complete: 70%

Deadline: 2019-11-17

The expected result: The App is used to buy flights and combine flights with other operators.

The actual result: The App is crashing when trying to purchase tickets after combining flights.

Steps to Reproduce:

- Open App.
- Open 'Select flights'
- Select Dublin to London Heathrow (Ryanair)
- Combine
- London Heathrow to New York(British airways)
- Select 'Purchase tickets'
- App will now crash

Summary: App crashing once combined flights is ready to be purchased.

Description: App is crashing once flights have been combined and are ready for purchase. There has been a sale on tickets and there has been a much higher volume of users on the app than normal. Single operator flights have not been affected.

1.3.9 Bug report

WOW Online Flight Reservation System

Status: CONFIRMED
Alias: None
Product: WOW Online Flight Reservation System
Component: Web Frontend
Version: 1.1.6
Severity: normal
Hardware: PC
OS: Windows
Assignee: Michelle Watanabe
URL: <https://wowflights.com>
Personal Tags: No dropdown box
Depends On: N/A
Blocks: N/A
Reported: 2019-10-25 11:47 IST by Michelle Watanabe
Modified: 2019-10-25 12:54 IST
Orig. Est.: 8.0
Current Est: 6.0
Hours Worked: 3.0
Hours Left: 3.0
% Complete: 50%
Deadline: 2019-11-17

The expected result: When booking a flight with both adult and child passengers travelling together, the drop-down box should allow the passenger to select passenger type e.g. adult or child.

The actual result: No dropdown box is being displayed and so adult passengers are later being registered as child passengers and vice a versa after payment.

Steps to reproduce:

- 1) Navigate to <https://wowflights.com> using a PC running on Windows
- 2) Go to flight search, select destination, dates and passengers.
- 3) Choose one adult passenger and one child.
- 3) Choose flight prices.
- 4) Go to checkout.
- 5) Input passenger details. System asks for first name, last name, date of birth and passport details without giving a child/adult dropbox.

Summary: No dropdown box to distinguish between adult and child passengers.

Description: When making the booking and inputting passenger details there is no way of choosing between an adult and child passenger. Then when advancing and checking details after payment, the child is registered as an adult or vice a versa and it is not possible to amend the details online without calling the help desk. This also means issuing the correct tickets at the airport rather than online.

1.3.10 Bug Report

UWON Library System

Status: CONFIRMED
Alias: None
Product: UWON Library System

Component: Web Frontend

Version: 1.1.6

Severity: major

Hardware: PC

OS: Windows 10

Assignee: Laszlo Szlatski

URL: <https://UWON.library.com>

Personal Tags: Login offsite

Depends On: Browser used. Only fault in Chrome.

Blocks: Waiting to schedule maintenance window from Chrome to continue testing websites issues.

Reported: 2019-10-24 17:18 IST by Laszlo Szlatki

Modified: 2019-10-26 09:54 IST

Orig. Est.: 4.0

Current Est: 4.0

Hours Worked: 1.0

Hours Left: 3.0

% Complete: 25%

Deadline: 2019-11-17

OS: Windows

The expected result: Library members using should be able to search the library catalogue when offsite, once registered, using their membership and password.

The actual result: Library members using google chrome can login but there is an error message and members are unable to make a search for content when accessing the library catalogue via the web login.

Steps to reproduce:

1) Using Google Chrome Web Browser, go to library website

<https://UWON.library.com/login>

2) Log in to UWON's library system using the library website.

3) Go to 'search catalogue'.

4) Enter keyword, book title, and author or reference number and click search.

Summary: Members unable to search catalogue using Google Chrome.

Description: Members using Google Chrome who have signed up to use the library catalogue online remotely are able to log in to the library catalogue successfully using the website, but when trying to search the library catalogue an error page is showing, stating content is unavailable and for restricted access only.

1.3.11 Bug report

WAX Transportation

Status: CONFIRMED

Alias: N/A

Product: WAX Transportations

Component: ZX1190 Sensors

Version: V5

Severity: major

Hardware: Other

OS: Linux

Assignee: Michelle Hourihan Watanabe

URL: <http://www.waxtransport.com>

Personal Tags: door not closing on time
 Depends On: N/A
 Blocks: N/A
 Reported: 2019-10-25 16:13IST by Patrick O'Neill
 Modified: 2019-10-28 15:53 IST
 Orig. Est.: 20.0
 Current Est: 15.0
 Hours Worked: 5.0
 Hours Left: 10.0
 % Complete: 30%
 Deadline: 2019-11-17

The expected result: Train doors should be fully closed within 5 seconds of the warning signal.

The actual result: Train doors are taking longer to close than the allotted time slot, several attempts are then being made, causing the doors to open and close several times before departure.

Steps to reproduce:

- 1) Put train into drive mode
- 2) Turn on the departure signal to start the warning sound
- 3) Doors should then close automatically before departure.

Summary: Doors not closing within specified time, causing several attempts.

Description: Some doors are not able to close within the allocated time slot even if there are no mechanical or environmental issues. When they do not close in the allocated time slot, this is causing repeat attempts and significant delays.

1.3.12 Bug report

WSS Meetings

Status: CONFIRMED
 Alias: None
 Product: WSS Meetings
 Component: Frontend
 Version: 1.0
 Severity: Normal
 Hardware: PC
 OS: All
 Assignee: Laszlo Szlatki
 URL: <http://www.wssmeetings.com>
 Personal Tags: Time Zone
 Depends On: N/A
 Blocks: N/A
 Reported: 2019-10-24 14:21 IST by Patrick O'Neill
 Modified: 2019-10-25 14:30 IST
 Orig. Est.: 10.0
 Current Est: 12.0
 Hours Worked: 4.0
 Hours Left: 8.0
 % Complete: 30%
 Deadline: 2019-11-17

The expected result: When a web conference is scheduled in a specified time zone, all members should receive an email with the correct time zone chosen and a link that will allow them to join the

meetings for that time.

The actual result: Despite a non-default time zone being specified, the links sent to all meeting participants only work at the default time zone. For example, when someone schedules a web conference logged in as a user with a default of Central European Time and schedules a meeting in Greenwich Mean Time, the links sent out actually only work at the default CET time. Steps to reproduce:

- 1) Log in to <http://www.wssmeetings.com>
- 2) Fill out necessary details about the meeting.
- 3) Add invited parties
- 4) Invite people with professional e-mail addresses
- 5) Choose a time slot specifying a time zone other than the default time.
- 6) Open participant email and check the time zone of the meeting link sent.

Summary: Meetings links being sent in undesired time zone.

Description: Users are unable to schedule meetings in time zones other than their default time zone, even if a different time zone is selected. For example, when scheduling a web conference in another time zone, if the default setting for the user is Central European Time, the links sent to all invited participants is set to the default time and not the selected time zone. This means the meeting link is only available at default time and not the desired time in the correct time zone.

1.3.13 Bug Report WSS Meetings

Summary: Software Double Booking Rooms cause customers to not get their meeting.

Status: Confirmed.

Alias: Double Book

Product: WSS Meetings.

Component: Backend.

Version: 1.0

Hardware: PC – Windows

Importance: High – High

Assignee: Patrick O'Neill

URL: B

Personal Tags: Double Book

Depends On: Meetings.

Blocks: Meetings from happening.

Original Est Time: 2.0

Current Est Time: 2.0

Hours Worked: 0.0

Hours Left: 2.0

% Complete: 0%

Gain: 0.0

Deadline: 2019-11-18

The expected Result: When the room is booked, it should not allow the user to book the same room.

The actual Result: The When the user goes to search for a room, it says that it has not been booked out and it has been.

Description: This is a problem as businesspeople don't have time on their side. Having rooms double booked is a problem as just say we had to turn a customer away due to been booked out due to our negligence, they could warn off future customers.

1.3.14 Bug Report

UWON Library System

Summary: Self scan service of books not recording to account.

Status: Confirmed.

Alias: Record1

Product: UWON Library System.

Component: Backend.

Version: 1.0

Hardware: PC – Windows

Importance: Normal – Normal

Assignee: Patrick O'Neill

URL: D

Personal Tags: Record.

Depends On: Record to account.

Blocks: Quick access for customers.

Original Est Time: 2.0

Crurent Est Time: 2.0

Hours Worked: 0.0

Hours Left: 2.0

% Complete: 0%

Gain: 0.0

Deadline: 2019-11-18

The expected Result: If you scan the bar code from the book, it should add the book to the account that you logged in on, as part of the self-scan service.

The actual Result: The scanner does not write the bar code to the file of the account, as part of the self-scan service.

Steps to Reproduce:

- 1): Scan the library card with scanner, to log you into your account.
- 2): Do a search for a particular book.
- 3): That will show if it's in stock or not.
- 4): It will show you the aisle and shelf the book location is at.

Description: This is a problem as the Library users like the convenient way of checking out the books. Note this problem happened after an update was applied to the system, roll back to the previous update as this might be the cause.

1.3.15 Bug Report

WOW Online Flight Reservation System.

Summary: Latency of data causes flight reservation problems.

Status: Confirmed.

Alias: Price Latency.

Product: WOW Online Flight Reservation System.

Component: Backend

Version: 1.0

Hardware: PC – Windows

Importance: High – Major

Assignee: Michele Hourihan Watanabe

URL: E

Personal Tags: Latency 1.

Depends On: Faster Speeds for transactions.

Blocks: Instant transactions for customers.

Original Est Time: 9.0

Crurent Est Time: 0.0

Hours Worked: 0.0

Hours Left: 9.0

% Complete: 0%

Gain: 0.0

Deadline: 2019-11-18

The expected Result: When you add a flight to basket and go to check out it should reserve your flight instantly.

The actual Result: When you click reserve flight to basket and by the time you go to reservation screen it tells you the flight has been booked out.

Steps to Reproduce:

- 1): Login or Create Account.
- 2): Search for Destination.
- 3): Select Arrival and Departure airport.
- 4): Select date and time for Arrival and Departure flights.
- 5): Press place flight reservation.
- 6): Add info like Name, Address, DOB, and Credit Card Details.
- 7): Press Confirm Flight to finalize the booking of the flight.

Description: This is a problem as a customer may get a good deal on a flight and by the time they get to pay for it the deal could be gone. This could affect our reputation as a company. Note this may be external issue as the hosting company that hosts our site has many the same problems with different site, like mandmdirect.ie etc.

1.3.16 Bug Report

WAX Transportation.

Summary: Book ticket button not working on all Browsers.

Status: Confirmed.

Alias: Button Faulty.

Product: WAX Transportation.

Component: Backend

Version: 1.0

Hardware: PC – Windows

Importance: High – Major

Assignee: Michele Hourihan Watanabe

URL: F

Personal Tags: Button trouble.

Depends On: Having the right implemented to work with Chrome.

Blocks: Customers fail to book their train ticket reservation.

Original Est Time: 2.0

Curent Est Time: 0.0

Hours Worked: 0.0

Hours Left: 2.0

% Complete: 0%

Gain: 0.0

Deadline: 2019-11-18

The expected Result: When you use Chrome Browser to select the journey you want to book and press book ticket button it takes you to reservation page.

The actual Result: The Book ticket button on web page doesn't work on Chrome Browser, but it

works on other browsers.

Steps to Reproduce:

- 1): Login or continue as guest.
- 2): Search for Destination.
- 3): Select Arrival Station and Departure Station.
- 4): Select date and time for Arrival and Departure stations.
- 5): Press place train trip reservation.
- 6): Add info like Name, Address, DOB and Credit Card details.
- 7) Press Confirm trip to finalize the booking of the train trip.

Description: The book ticket button, is in working Microsoft Edge, Internet Explorer and Firefox. This means it wasn't tested properly, this is a problem as Google Chrome .

1.3.17 Bug Report

WOW Online Flight Reservation System.

Summary: Child price given the wrong price for their age.

Status: Confirmed.

Alias: Price Child wrong.

Component: General

Version: 1.0

Product: WOW Online Flight Reservation System.

Component: Backend.

Version: 1.0

Hardware: PC – Windows

The expected Result: Children under the age of 11 should be recorded as a child at a discounted price.

The actual Result: The customer's ten-year-old child is been recoded as an adult.

Steps to Reproduce:

- 1): Create account or Login.
- 2): Search for flight Departure Location and Destination Location.
- 3): Select the Date, Time and choose first class or economy.
- 4): Choose the amount of passengers and their age class.
- 5): Press place flight reservation.
- 6): Add info like Name, Address, DOB, and Credit Card Details.
- 7): Press Confirm Flight to finalize the booking of the flight.

Importance: High – Enhancement

Assignee: Laszlo Szlatki

URL: G

Personal Tags: Wrong Age.

Depends On: Right age groups.

Blocks: Customers pay more for their children under eleven.

Original Est Time: 1.0

Current Est Time: 0.0

Hours Worked: 1.0

Hours Left: 1.0

% Complete: 0%

Gain: 0.0

Deadline: 2019-11-18

Description: The flights site that give customers a search through different sites for the cheapest

deal. The customer's ten-year-old child is recorded as an adult. Where on the main site of the airline, children are aged between 2-11.

1.3.18 Bug Report WAX Transportation.

Summary: Trains directed to the wrong terminal, causes customers to be late for flights.

Status: Resolved Fixed.

Alias: Outofsync.

Product: WAX Transportation.

Component: Backend

Version: 1.0

Hardware: PC – Windows

Importance: High – Major

Assignee: Michele Hourihan Watanabe

URL: A

Personal Tags: Train wrong direction.

Depends On: Time been kept up to date.

Blocks: Customers fail to make their flight on time.

Original Est Time: 2.0

Current Est Time: 2.0

Hours Worked: 2.0

Hours Left: 2.0

% Complete: 100%

Gain: 0.0

Deadline: 2019-11-18

The expected Result: For example, Train A should go to Terminal A etc.

The actual Result: Display is showing the wrong train for the required terminal.

Steps to Reproduce:

- 1): Create Account or Login.
- 2): Pick Departure and Arrival Location.
- 3): Select date and time for Arrival and Departure stations.
- 4): Enter Payment details.
- 5): Confirm Payment.

Description: The Train A is going to Terminal B when it should be going to Terminal A. Train B is going to Terminal C when it should be going to Terminal B. Then finally Train C is going to Terminal A, when it should be going to Terminal C.

Note Passengers for Terminal A missed their flight to London.

1.3.19 Bug Report WSS Meetings

Summary: Wrong booking type cause second client not booked in Company Id instead of Customer ID.

Status: Confirmed.

Alias: Wrong Type

Product: WSS Meetings.

Component: Backend.

Version: 1.0

Hardware: PC – Windows

Importance: Normal – Major

Assignee: Patrick O’Neill

URL: G

Personal Tags: Wrong Data Booking.

Depends On: Meetings.

Blocks: Two customers from having meetings from happening from the same company at the same time.

Original Est Time: 3.0

Current Est Time: 3.0

Hours Worked: 0.0

Hours Left: 3.0

% Complete: 0%

Gain: 0.0

Deadline: 2019-11-18

The expected Result: Booking should be done under Customer ID instead of Company ID.

The actual Result: The booking reservation was booked under Company ID instead of Customer ID.

Steps to Reproduce:

1. Create Account or Login.
2. Check Date and Time Available.
3. Pick Date and Time.
4. Confirm Booking.

Description: The booking reservation was booked under Company ID instead of Customer ID. The problem that caused was by this was two different clients from the same company, booked two different meetings at the same time, so the meeting was only booked once, since it was booked by Company ID.

1.3.20 Bug Report UWON Library System

Summary: Wrong date recorded on book loan, causes strike on several customer’s account.

Status: Confirmed.

Alias: Bstrike

Product: UWON Library System.

Component: Backend.

Version: 1.0

Hardware: PC – Windows

Importance: Normal – Major

Assignee: Patrick O’Neill

URL: H

Personal Tags: Book Strike.

Depends On: Record to account.

Blocks: The customer to be fairly treated, on getting an underserved strike.

Original Est Time: 4.0

Current Est Time: 4.0

Hours Worked: 0.0

Hours Left: 4.0

% Complete: 0%

Gain: 0.0

Deadline: 2019-11-18

The expected Result: When the Librarian checks up on the account it shouldn’t say a book is overdue when it’s not.

The actual Result: The customer of the library is told by that the book is late on the account by a day when it's not.

Steps to Reproduce:

- 1): Enroll for account in Library.
- 2): Pick a book from shelve.
- 3): Give the book to librarian and give your account no.

Description: The customer took out a loan of a book on 25-10-19 for a week. When the customer brought back the book on the 1-11-19 it said it was a day late. Each customer can borrow only two books per week. When a book is late the account gets a strike, when the account gets ten strikes, that account then can't take out a book for a month.

Bugzilla – Enter Bug: WAX Transportation

Home | New | Browse | Search | Search [?] | Reports | Preferences | Administration | Help
Log out 18169902@studentmail.ul.ie

Before reporting a bug, please read the [bug writing guidelines](#), please look at the list of [most frequently reported bugs](#), and please [search](#) for the bug.

[Show Advanced Fields](#) (* = Required Field)

* **Product:** WAX Transportation **Reporter:** 18169902@studentmail.ul.ie

* **Component:** General Component Description
The 'Catch All' Component

* **Version:** 1.0 **Severity:** major
Hardware: Other
OS: Other
We've made a guess at your operating system and platform. Please check them and make any corrections if necessary.

* **the expected result:** Train is being directed to an empty parking track

* **the actual result:** Train was directed to a parking track, where a train was already parking

* **Steps to Reproduce:** 1) Divert trains both on track [PT01](#), [PT02](#) and [PT03](#) and park them
2) Remove the train from [PT02](#) or [PT03](#)
3) Direct a train to an unspecified parking track

* **Summary:** A train is allowed to enter an already occupied parking track

Possible Duplicates:

Bug ID	Summary	Status
	Data error.	

Description: Comment Preview

After all parking tracks were in use, and one becomes available, the system directs the next train to be parked in [PT01](#), a possibly still occupied track, instead of directing it to the next available empty track. It only appears once all parking tracks were filled, and [PT01](#) is not the first one, what becomes empty.

Attachment: Add an attachment

Submit Bug

Home | New | Browse | Search | Search [?] | Reports | Preferences | Administration | Help
Log out 18169902@studentmail.ul.ie
My Bugs

BUG 1.3.4

Bug 5783 - Library system is giving different dates for availability and return [\(edit\)](#)
Save Changes

[Status:](#) **CONFIRMED** [\(edit\)](#)
[Alias:](#) None [\(edit\)](#)
[Product:](#) UWON Library System
[Component:](#) General [\(show other bugs\)](#)
[Version:](#) unspecified
[Hardware:](#) PC [Windows](#)
[Importance:](#) Normal [minor](#)
[Assignee:](#) MICHELLE.HOURIHAN WATANABE [\(edit\)](#) [\(take\)](#)
[URL:](#)
[Personal Tags:](#)
[Depends on:](#)
[Blocks:](#)

[Reported:](#) 2019-10-18 16:41 IST by [Patrick O'Neill](#)
[Modified:](#) 2019-10-18 16:45 IST [\(History\)](#)
[CC List:](#) ☐ Add me to CC list
0 users [\(edit\)](#)
[Ignore Bug Mail:](#) ☐ (never email me about this bug)

[the expected result:](#) Once a book has been reserved the system will let you know what d
[the actual result:](#) The system is giving the wrong date for availability.
[Steps to Reproduce:](#) [\(edit\)](#)

Log onto the UWON library system.
Reserve a book that has already been taken out.
Check dates of availability.
Cross reference with dates where the book has to be returned.

Orig. Est.:	Current Est.:	Hours Worked:	Hours Left:	%Complete:	Gain:	Deadline:
0.0	0.0	0.0 + 0	0.0	0	0.0	<input type="text"/>

[Summarize time \(including time for bugs blocking this bug\)](#)

Attachments
[Add an attachment](#) (proposed patch, testcase, etc.)

Additional Comments:

Comment
Preview

[Status:](#) **CONFIRMED** [Mark as Duplicate](#)
Save Changes

[Patrick O'Neill](#)
2019-10-18 16:41:36 IST
Description
[tag] [reply] [-]

When a book is taken out it produces a return date automatically. When someone reserves the same book it should be given a date to pick up the book after it has been returned but the system is giving wrong availability dates. It says that the book should be available for pick up even though the book is not due to be returned.

[Add Comment](#)

[Collapse All Comments](#)
[Expand All Comments](#)

[Format For Printing](#) - [XML](#) - [Clone This Bug](#) - [Top of page](#)

Bugzilla - Bug 3940 Summary: No dropdown box appears to distinguish between adult and child passengers. Last modified: 2019-11-07 09:55 GMT

Home New Bugs Search Search [Reports](#) [Preferences](#) [Administration](#) [Help](#) [Logout](#) [C/C++/C#](#) [C#](#) [C++](#) [C#](#)

Bug List (1 of 1) (First/Last/Prev/Next) [Show list search results](#)

Bug 3940 - Summary: No dropdown box appears to distinguish between adult and child passengers. [View](#) [Save Changes](#)

Status: [CONFIRMED](#) [\[edit\]](#)

Alias: [None](#) [\[edit\]](#)

Product: [KON Online Flight Reservation System](#) [\[edit\]](#)

Component: [Web Frontend](#) [\[edit\]](#) [\[show other bugs\]](#)

Version: [1.0](#)

Hardware: [PC](#) [\[edit\]](#) [\[show other bugs\]](#)

Importance: [Normal](#) [\[edit\]](#) [\[show other bugs\]](#)

Assignee: [Alexander Schwan](#) [\[edit\]](#) [\[show other bugs\]](#)

URL:

Personal Tags:

Depends on:

Blocks:

Reported: 2019-11-07 09:55 GMT by [MICHELLE MOURSHAN BATHAM](#)

Modified: 2019-11-07 09:55 GMT [\[edit\]](#)

CC List: [Add me to CC list](#)

0 users [\[edit\]](#)

Ignore Bug Mail: [\[edit\]](#) (never email me about this bug)

[See expected result](#) The expected result: When booking a flight with both adult and child passengers traveling together the drop down box should allow the passenger to

[See actual result](#) The actual result: No dropdown box is being displayed and as adult passengers are later being registered as child passengers and vice versa after pa

[Steps to Reproduce](#) [\[edit\]](#)

- 1) Navigate to <https://konflights.com> using a PC running on Windows
- 2) Go to Flight search, select destination, dates and passengers.

[\[1\] Please open and the passenger and vice versa. Add it.](#)

Orig. Est.	Current Est.	Hours Worked	Hours Left	%Complete	Gain	Deadline
0.0	0.0	0.0 + 0	0.0	0	0.0	

[Summerize time including time for bugs blocking this bug](#)

Attachments

[Add an attachment](#) (proposed patch, test case, etc.)

Additional Comments

[Comment](#) [Preview](#)

Status: [CONFIRMED](#) [\[edit\]](#)

[Mark as Duplicate](#) [Save Changes](#)

[MICHELLE MOURSHAN BATHAM](#) 2019-11-07 09:55:00 GMT [Description](#) [\[edit\]](#) [\[show\]](#) [\[hide\]](#) [Collapse All Comments](#) [Expand All Comments](#)

Description: When making the booking and inputting passenger details there is no way of choosing between an adult and child passenger. Then when advancing and checking details after payment, the child is registered as an adult or vice versa and it is not possible to amend the details online without calling the help desk. This also means loading the correct tickets at the airport rather than online.

[Add Comment](#)

Bug List (1 of 1) (First/Last/Prev/Next) [Show list search results](#) [Open in Firefox](#) [Open in Chrome](#) [Open in Safari](#)

Home New Bugs Search Search [Reports](#) [Preferences](#) [Administration](#) [Help](#) [Logout](#) [C/C++/C#](#) [C#](#) [C++](#) [C#](#)

0.0

Bug List: (8 of 10) [First](#) [Last](#) [Prev](#) [Next](#) [Show last search results](#)

Bug 5772 - Self scan service of books not recording to account. [\(edit\)](#)

Save Changes

Status: [CONFIRMED](#) [\(edit\)](#)

Alias: [None](#) [\(edit\)](#)

Product: [UWON Library System](#)

Component: [General](#) [\(show other bugs\)](#)

Version: [1.0](#)

Hardware: [Other](#) [Windows](#)

Importance: [Normal](#) [normal](#)

Assignee: [Patrick O'Neill](#) [\(edit\)](#) [\(take\)](#)

URL:

Personal Tags:

Depends on:

Blocks:

Reported: 2019-10-17 10:50 IST by [Richard O'Brien](#)

Modified: 2019-11-07 10:15 GMT [\(History\)](#)

CC List: ☐ Add me to CC list
0 users [\(edit\)](#)

Ignore Bug Mail: ☐ (never email me about this bug)

[the expected result:](#)

If you scan the bar code from the book, it should add the book to the account that your logged in on, as part

[the actual result:](#)

The scanner does not write the bar code to the file of the account, as part of the self scan service.

[Steps to Reproduce:](#) [\(edit\)](#)

1) Scan the Library card with scanner, to log you into your account.
2) Do a search for a particular book.
3) That will show if it's in stock or not.
4) It will show you the aisle and shelf the book location is at.

Orig. Est.:	Current Est.:	Hours Worked:	Hours Left:	%Complete:	Gain:	Deadline:
0.0	0.0	0.0 + 0	0.0	0	0.0	

[Summarize time \(including time for bugs blocking this bug\)](#)

Attachments

[Add an attachment](#) (proposed patch, testcase, etc.)

Status: [CONFIRMED](#)

Save Changes

[Mark as Duplicate](#)

[Richard O'Brien](#) 2019-10-17 10:50:29 IST

[Description](#) [\[tag\]](#) [\[reply\]](#) [\[-\]](#)

[Collapse All Comments](#)

[Expand All Comments](#)

[Richard O'Brien](#) 2019-10-25 13:33:37 IST

[Comment 1](#) [\[tag\]](#) [\[reply\]](#) [\[-\]](#)

This is a problem as Library users like the convenient way of checking out the books. Note this problem happened after an update was applied to system roll back to previous update as this might be the cause.

Bug 1.3.14

2 Test Design & Development using JUNIT.

2.1 TDD log

Subtask	Title	Author	Date
TDD 4.1	TDD for password generator.	Laszlo Szlatki	10/23/2019
TDD 4.2	First test fails Password generator as class does not exist yet.	Laszlo Szlatki and Richard O'Brien	10/23/2019
TDD 4.3	Second test passes after code written for password generator class.	Laszlo Szlatki and Richard O'Brien	10/23/2019
TDD 4.4	Third test fails range of the random number generator method is not yet implemented.	Laszlo Szlatki and Richard O'Brien	10/24/2019
TDD 4.5	Fourth test passes after writing the method in the source file.	Laszlo Szlatki and Richard O'Brien	10/24/2019
TDD 4.6	Fifth test fails as method to change random number to character does not exist yet.	Laszlo Szlatki and Richard O'Brien	10/25/2019
TDD 4.7	sixth test passes after implementing the necessary method The number 0 is hard coded in the method, because this is the minimum requirement to make the test pass.	Laszlo Szlatki and Richard O'Brien	10/25/2019
TDD 4.8	Seventh test fails because the value 0 is hardcoded into the source file.	Laszlo Szlatki and Richard O'Brien	10/26/2019
TDD 4.9	eight test passes after passing in a value, with the value of the random generated number.	Laszlo Szlatki and Richard O'Brien	10/26/2019
TDD 4.10	Ninth test fails as the length of the password has not yet	Laszlo Szlatki and Richard O'Brien	10/27/2019

	been implemented in the source file.		
TDD 4.11	Tenth test passes after implementing the method for what is accepted in the password length.	Laszlo Szlatki and Richard O'Brien	10/27/2019
TDD 4.12	TDD for Paint Calculator.	Patrick O'Neill and Michelle Hourihan	11/7/2019
TDD 4.13	First test fails for Paint Calculator as class does not exist.	Patrick O'Neill and Michelle Hourihan	11/7/2019
TDD 4.14	Second test passes after class is created.	Patrick O'Neill and Michelle Hourihan	11/7/2019
TDD 4.15	Designing of the third test to check the method will calculate the wall area.	Patrick O'Neill and Michelle Hourihan	11/7/2019
TDD 4.16	Failure of the third test as method WallAreaCalc does not exist.	Patrick O'Neill and Michelle Hourihan	11/7/2019
TDD 4.17	Development of the method WallAreaCalc	Patrick O'Neill and Michelle Hourihan	11/7/2019
TDD 4.18	Fourth test passes once method WallAreaCalc exist and return expected value for wall area.	Patrick O'Neill and Michelle Hourihan	11/7/2019
TDD 4.19	Test designed to check the method to calculate the total paint needed returns the correct result expected.	Patrick O'Neill and Michelle Hourihan	11/7/2019
TDD 4.20	Fifth test fails as method is not yet created.	Patrick O'Neill and Michelle Hourihan	11/7/2019
TDD 4.21	Design of totalPaintCalc method.	Patrick O'Neill and Michelle Hourihan	11/7/2019
TDD 4.22	Sixth test passes after implementing the method and testing to see if it returns the expected value.	Patrick O'Neill and Michelle Hourihan	11/7/2019

2.2 Reflections

Michelle's Reflections

Having limited experience with Junit and programming, I personally found the task extremely challenging, but it gave me a valuable opportunity to learn hands-on about Test-Driven Development. The aspect that was the most difficult was learning to use Junit for both testing and developing code concurrently. I became more comfortable by practicing individually and designing tests for very simple programs. At first, I kept developing too much code before testing but then I realized that I had to not create or continue any method before starting each test case. Overall, I really felt designing each test before developing the next step of the program helped me to appreciate how TDD can make my programs more reliable.

Laszlo' Reflections

I really enjoyed the TDD approach of writing a new program. Attending the guest lecture helped me to understand the concept more. Before I couldn't imagine, how I can test something, what doesn't even exist at the time? Because it was my first time trying the Test driven development method, it took a lot longer to write the program, than it would without the tests, but going through the process, I can understand why the TDD is the preferred method for software development. The resulted code is more modular, reliable and less clumsy as a result. I aim to use it in the future too, even in college in future assignments.

Paddy's Reflections

I found something unusual with TDD at first it seemed strange writing a test knowing it was going to fail before you could move on and write the minimum amount of functional code to correct the issue. At the same time, it was very satisfying when you got the required result at the end. It allows you to code in small steps rather than large steps, for example writing 3 or 4 lines of code which will probably have something small to fix, rather than having 3 or 4 hundred lines of code with a lot of mistakes that will take a more time to resolve.

Richard's Reflection's

I really did find the TDD the most difficult to do, as I found TDD up to the assignment very hard. I think the breakthrough was when I went to Anila's tutorial's where that all made a lot of sense to me. I can see where TDD would be used in testing as you can test the program and make changes to it at a quicker pace than some of the other forms of testing.

2.3 Evidence

2.3.1 TDD for password generator

The group agreed to create 2 programs, the first was an amount of paint calculator and the second was a password generator program, using the Test-Driven Development (TDD) method. We decided to do two test driven-development methods so we could work more effectively in pairs and fully understand the TDD process.

In the program specification, we determined the characters to be used in the generated passwords (lower and upper-case letters and numbers 0 to 9), and that the length of the password can be determined by the program user. This program is appropriate to generate passwords for the most common use cases.

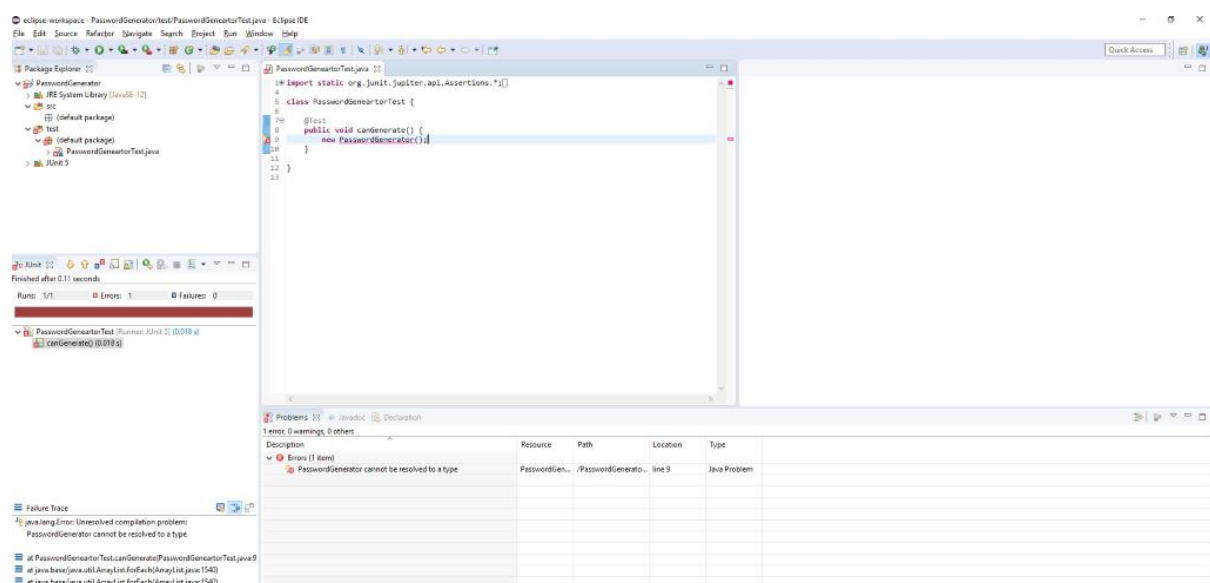
TDD is a software development process, which relies on a short development cycle repeated over and over. The main idea behind TDD is to code the tests first and then run the tests. This causes the tests to fail, and only then do we write just the minimum amount of code to make the tests pass in turn. Then, finally, we need to refactor the new code to adhere to acceptable standards and best practices.

The first step was to set up Eclipse and to create the new PasswordGenerator project. In this project, we also created new source folder called test to store all test related files.

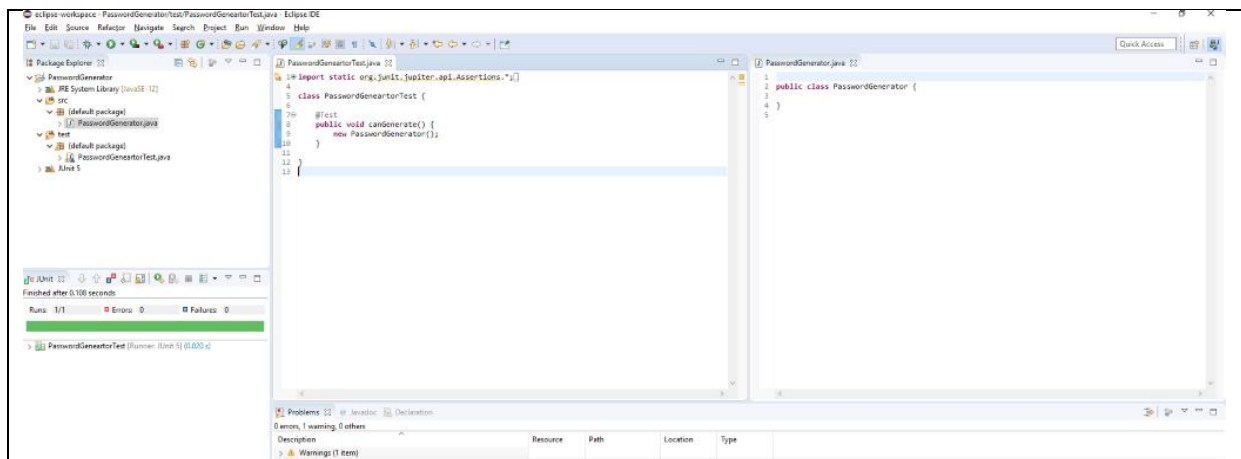
In the test folder, we created PasswordGeneratorTest.java file to assist with the TDD of the password generator application.

When everything was set up for easy use and clear navigation, in the test environment, we tested the PasswordGenerator program.

We tried to create a new instance of the PasswordGenerator object, but it failed because the PasswordGenerator class did not exist yet.



After writing just the necessary amount of code - creating the PasswordGenerator class- the test passed.

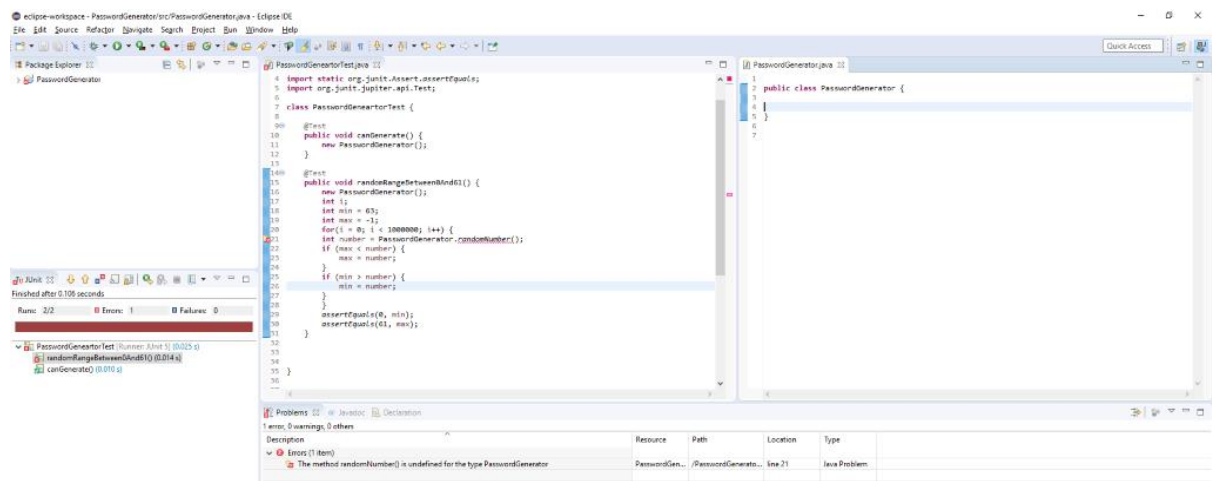


Next, we tested the range of the random number generator, by saving the min and max generated numbers. We tried to call the method 1 million times, to ensure that both borders of the range were generated.

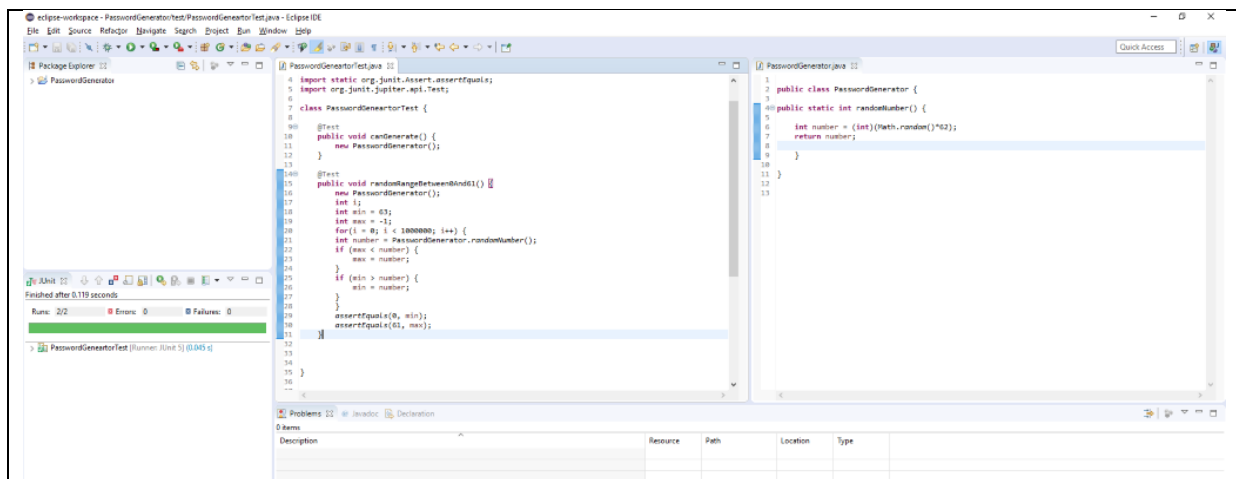
We tested if the minimum value equals 0 and the maximum value equals to 61. Between 0 and 61, we can store 62 numbers. We need 62 numbers, because in the agreed list of characters, there are 62 different variations.

To save the generated min and max values, we initialized min to be more than the upper end of the range and max to be less than the lower end of the range. This way we can ensure that after the first iteration, both values will be over-written, and an appropriate number will be stored in both. The rest of the iterations only extend this stored range to the edges of our allowed range of 0 and 61.

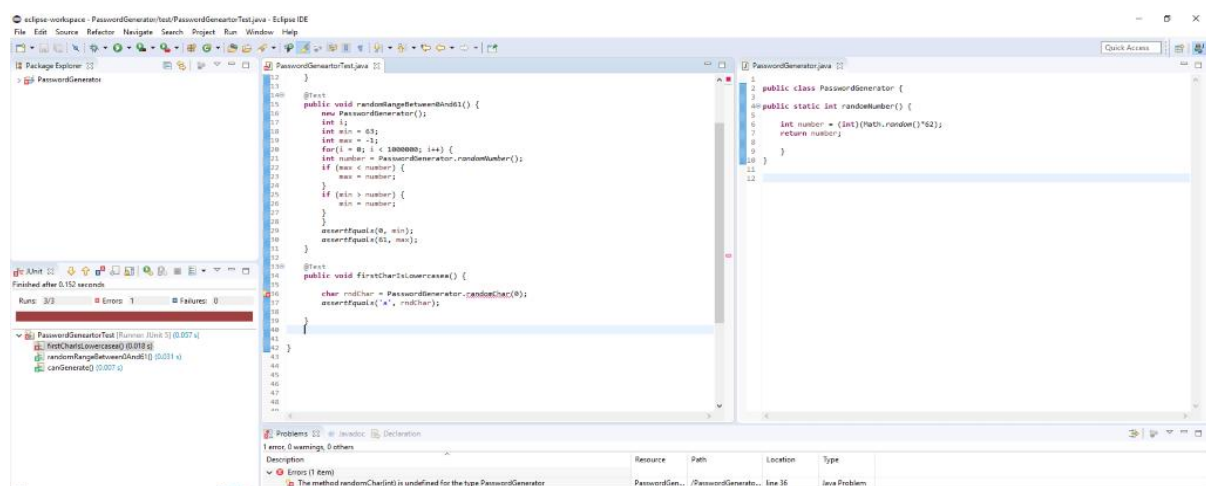
It fails, because the method was not implemented yet.



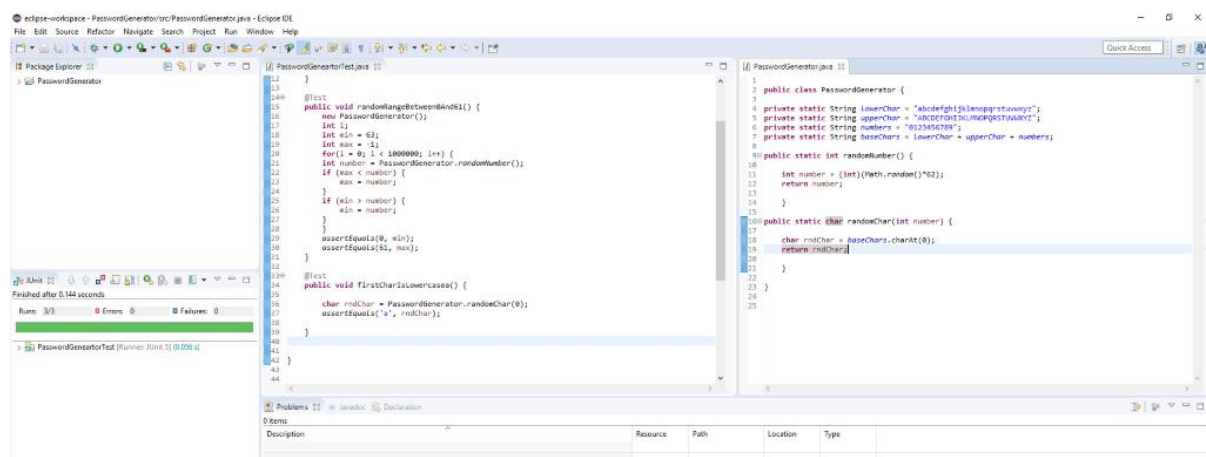
After writing the method in the source file, the issue was resolved. The test passed.



The next task was to test the character conversion. As the program generates a random number between 0 and 61, we needed to find a solution to test the indeterminate. To achieve this, the random generated number needed to be replaced with a hard-coded number, to allow the testing of an indeterminate output. Firstly, we tested for the first character to be 'a' using the input value 0. The test clearly failed, as the method to change the random number to a character did not exist yet.

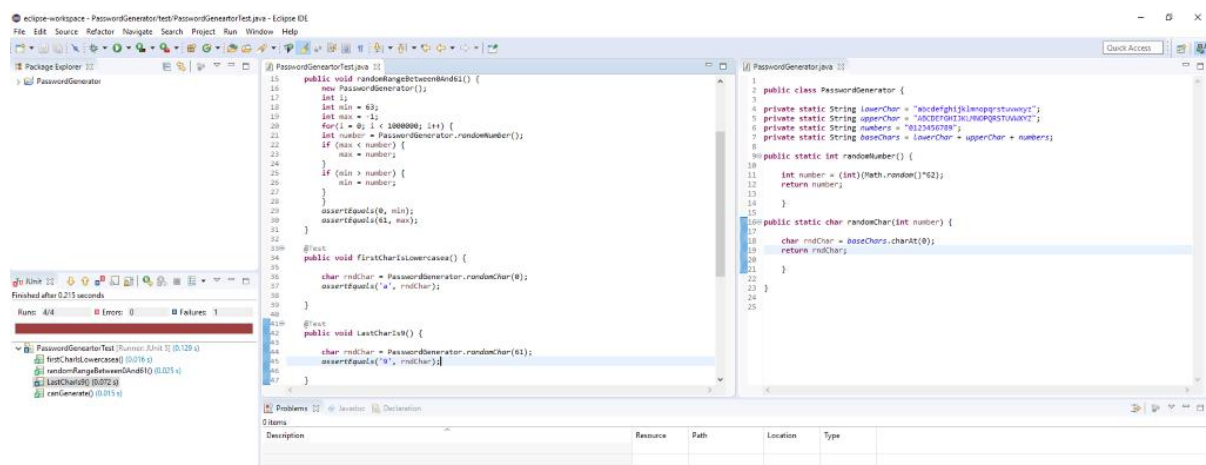


After implementing the necessary method, the test passed. The number 0 is hard coded in the method, because this is the minimum requirement to make the test pass.



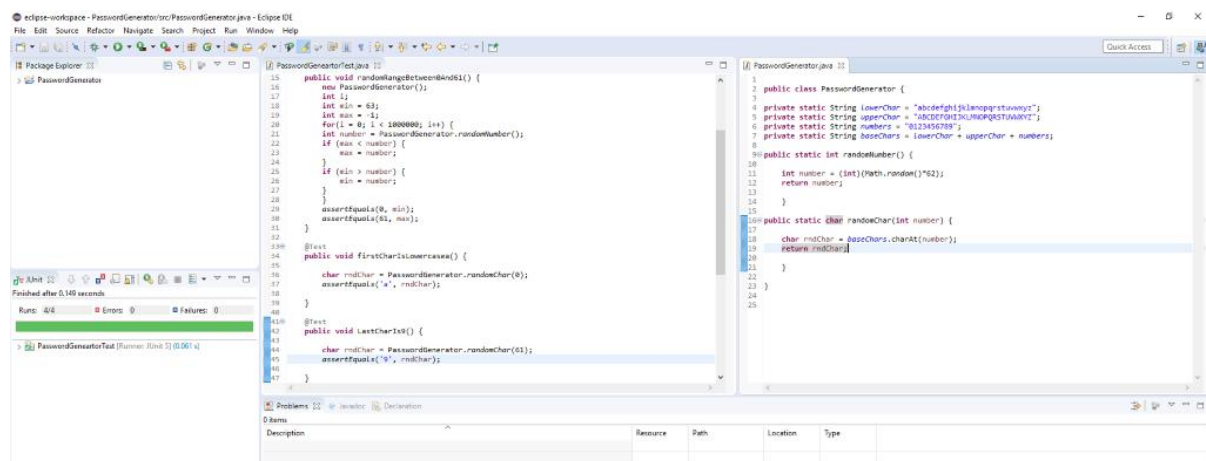
Now we needed to check for the last possible character to be '9'. For this, the method needed the input to be 61. The test failed, because the value 0 is hardcoded into the source file. As a result,

regardless of the input value, the method always returns the character 'a'.



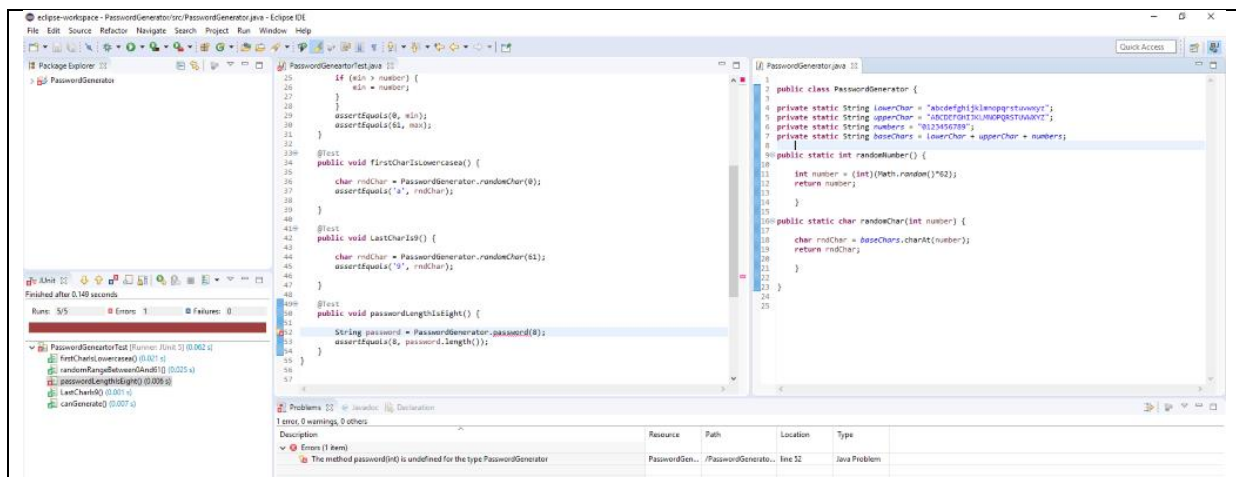
As expected, the test failed, due to the value 0 being hardcoded in the method. To rectify the issue, we need to pass in a variable, with the value of the random generated number.

Now the test passes. After checking the range for the generated random value and checking the two ends of the range, we can assume, the char conversion works satisfactorily and not check for each of the remaining 60 possibilities.

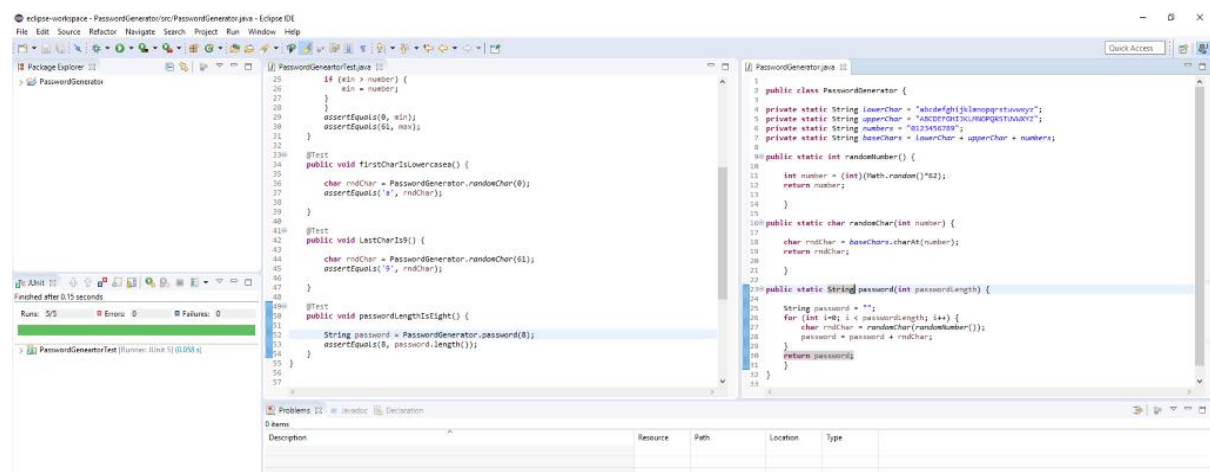


Lastly, we needed to test the length of the password. In our test case, we requested the password to be 8 characters long.

This test also failed because it is not yet implemented in the source file. At this point, the program only generates one random character.



After implementing the method, that accepts user input for the password length, this test also passes.

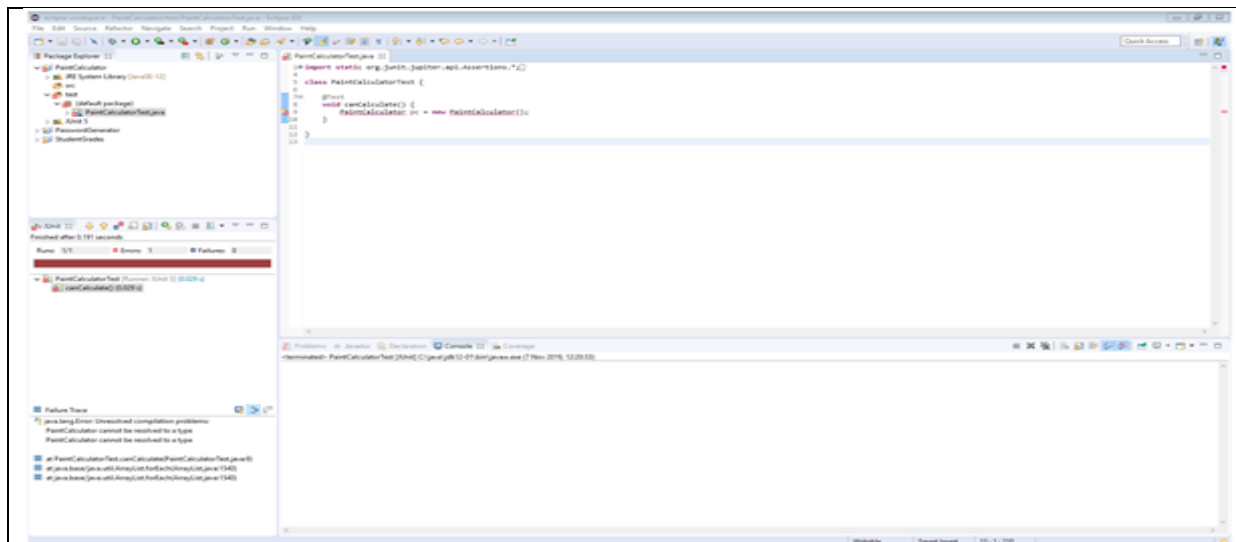


In the above example, we created a password generator program using the Test-Driven Development method. First, we wrote the test code, and after failing, we implemented the necessary code in the source file, to satisfy the specific test. Thanks to the TDD method, we can be confident the finished program works as expected.

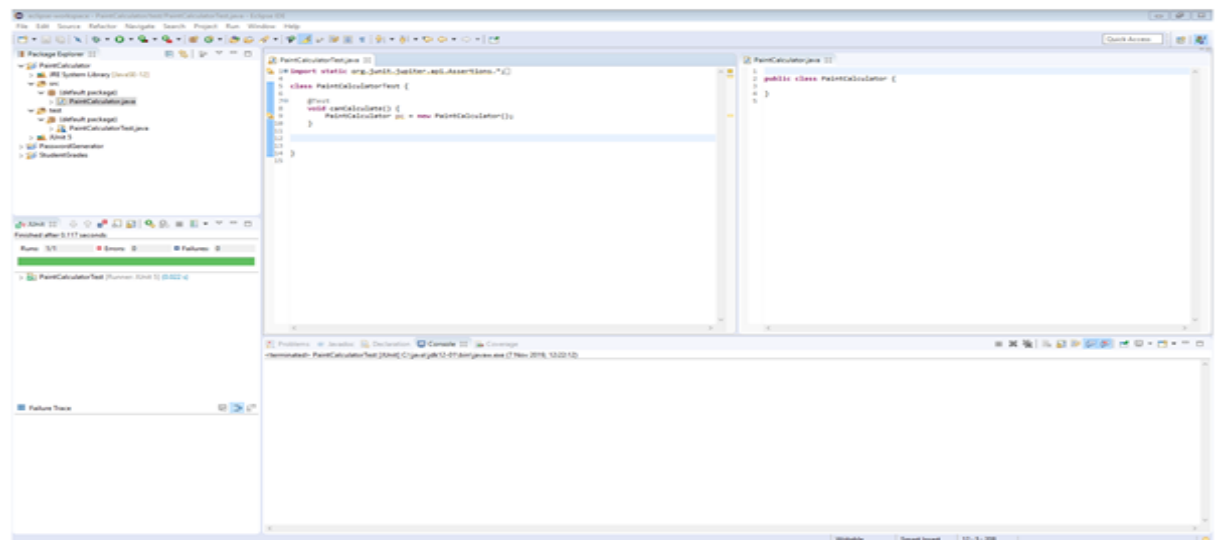
2.3.2 TDD for Paint Calculator

The first step was to start Eclipse and create a paint calculator project. We also had to create a new source folder called test to store all the test related files. When everything was set up for easy use and clear navigation, in the test environment, we test the paint calculator program.

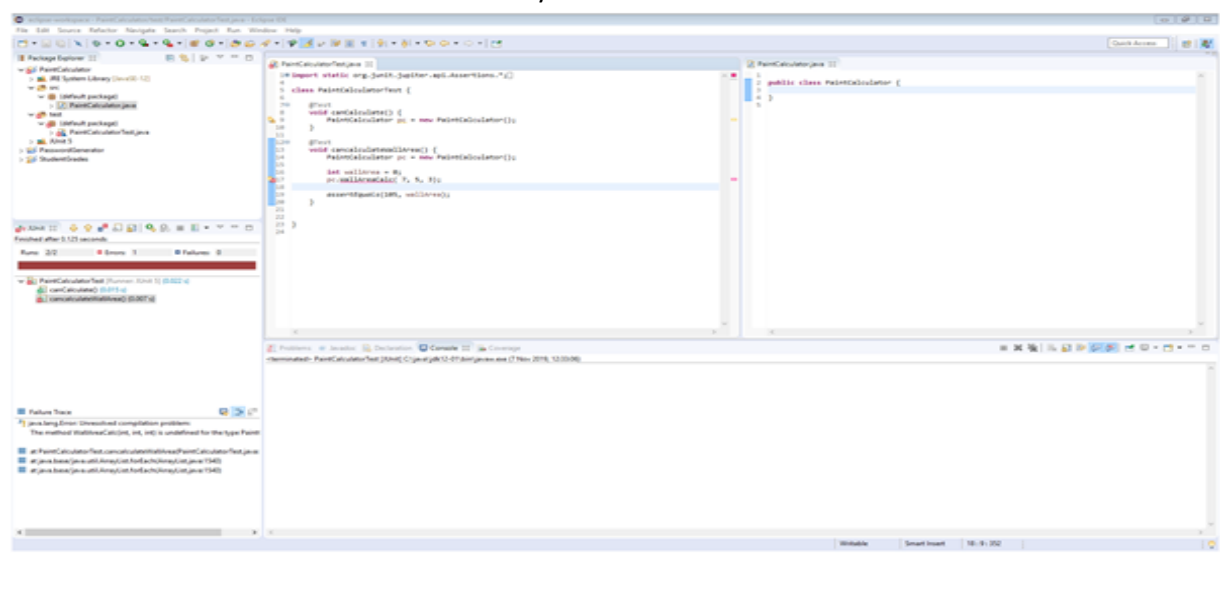
We tried to create a paint calculator object, but it failed because we have not created a paint calculator class yet.



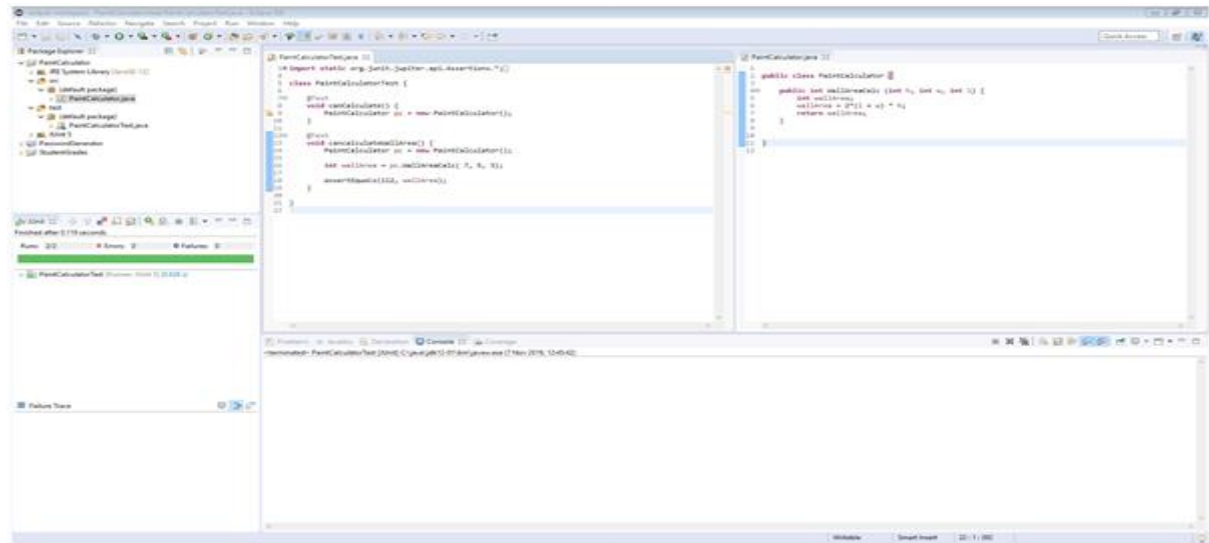
Once we created a class in the source folder “PaintCalculator.java” and ran the program again the test now passes.



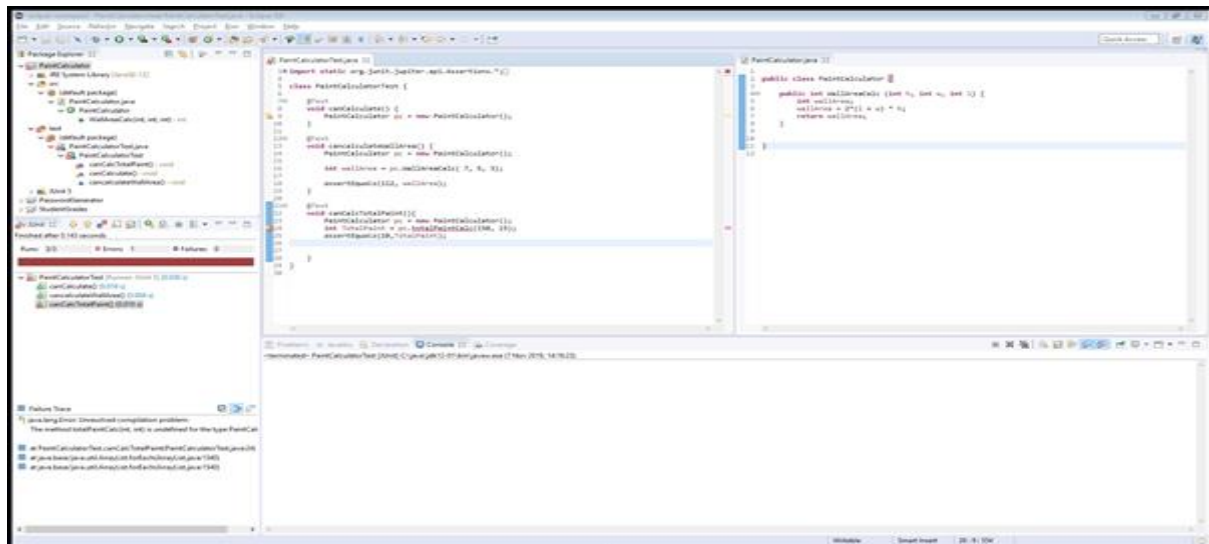
We designed the third test to check the method will calculate the wall area. This test failed as the method WallAreaCalc had not been created yet.



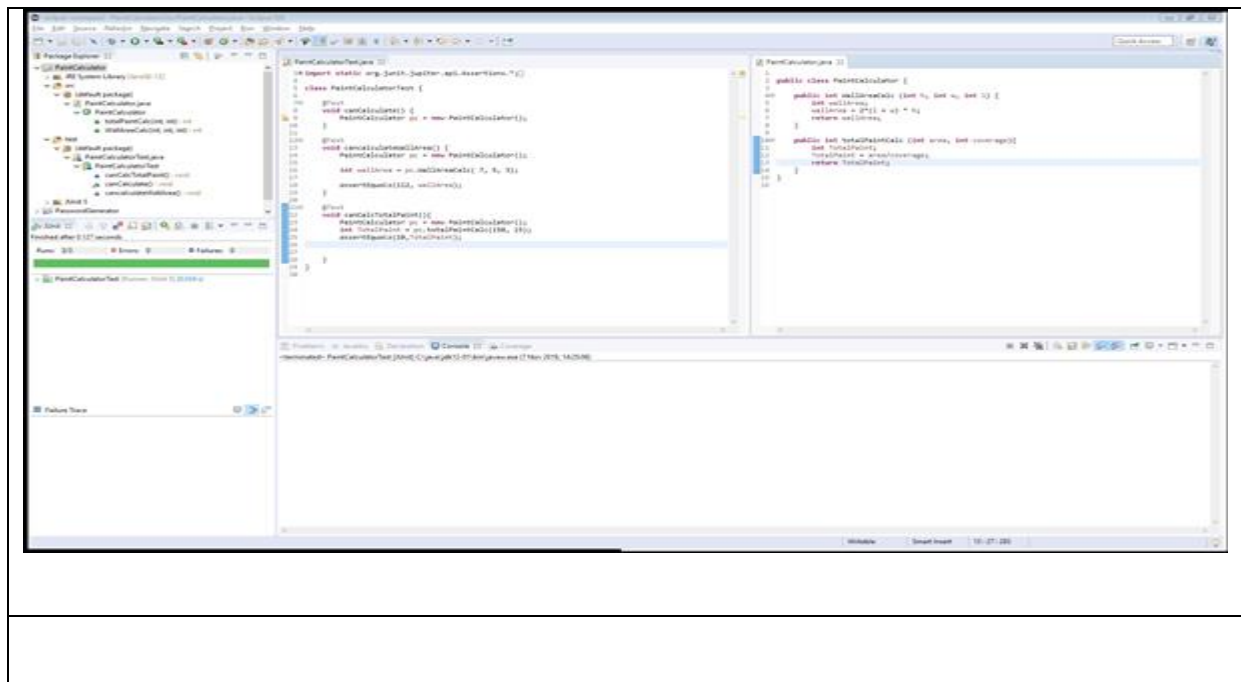
Once we found the reason for the previous test failure our next challenge was to create the method needed for the WallAreaCalc. We created the formula needed to return the wallArea. Our formula was $2 * (l + w) * h$. The test now passed after returning the expected value for the wall area.



We designed a test to check the method returns the correct result expected regarding the amount of paint needed, but because we have not created the method the test failed.



The sixth test passes after implementing the method and testing to see if it returns the expected value. It does by dividing the area by the coverage. Thanks to the above design the program is working as expected.



3 Coverage Criteria Based Testing

3.1 Coverage Criteria Log

Subtask	Title	Author	Date
3.3.1.1	Analysis of test cases on graph coverage for isEven () function	Michelle Hourihan Watanabe	11/10/2019
3.3.1.2	Input Domain Modelling and Input Partitioning test cases for isEven () function	Patrick O'Neill	11/10/2019
3.3.1.3	Tests were ran in Junit to check for full branch coverage	Michelle Hourihan Watanabe & Patrick O'Neill	11/11/2019
3.3.1.4	Creation of program GradeCalculator()	Laszlo Szlatki	11/11/2019
3.3.1.5	Establishing of equivalence classes for the test	Laszlo Szlatki & Richard O'Brien	11/11/2019
3.3.1.6	Statement coverage	Patrick O'Neill	3/11/2019
3.3.1.7	Branch coverage	Laszlo Szlatki	4/11/2019
3.3.1.8	Condition coverage	Laszlo Szlatki & Richard O'Brien	6/11/2019
3.3.1.9	Statement Coverage	Richard O'Brien	14/11/2019

3.2 Reflections

Michelle

I found criteria-based testing difficult to implement. On the one hand it seemed quite simple in theory, as in some cases it can be obvious that certain values should be tested to see if a function works. However, while trying to produce graphs and analyse input domain and graph coverage, it became apparent that as a program gets long or if function have many conditions or loops it can be easy to miss paths or not find a simple way to ensure complete coverage. I was satisfied with how we determined coverage criteria for our methods but feel I still have a lot to learn about how to use these methods for more 'complex' programs.

Laszlo

It might be because of the appropriate choice of a program to be tested, and because I kept the TDD principals in mind while writing the program to be tested, but I found the coverage criteria-based testing straight forward. I understand the difference between the various coverage tests and know their shortcomings. I also found eclipse's built in tools to test coverage very helpful.

Paddy

Coverage criteria-based testing was great because it assured the quality of the testing. It showed what parts of the code were actually touched. It also shows you what areas in application that are not touched upon. Great at finding faults and easy to use, as per our lectures criteria-based testing maximises our bang for buck. It also shows you how many tests are needed.

Richard

I really enjoyed this the most and I learnt so much from it. I did my statement coverage and branch coverage on a game, as that made me relate to how it works. This assignment on these types of coverage gave a better understanding of how they work. I can see why they are so popular in the industry, as statement is more popular than branch coverage due to price, as branch coverage is more rigorous.

3.3 Evidence

3.3.1 Coverage criteria

3.3.1.1 *Example 1. Testing a single 'if' statement.*

Testing a function that determines and prints if an integer is even or odd.

The function tested was function `isEven()`. The specifications for this function are that for a positive or negative integer, the function will determine and return if the number is odd or even.

Since the possibilities of testing any/all integers are very large or even infinite, it was necessary to find a coverage-criteria that we felt would satisfactorily test the methods functionality, which is to distinguish between odd and even. Firstly, we drew a graph to represent this function to better visualize the possible paths of control flow.

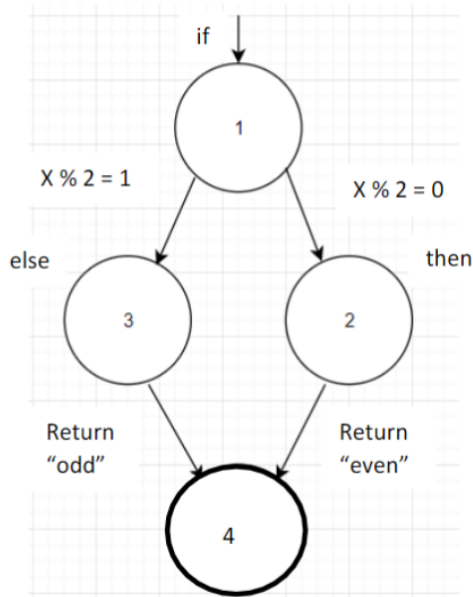


Diagram 1.1

Pseudo code

Read number

If number % 2 = 0 then

Print "Number is even"

Else

Print "Number is odd"

End if

Exit

Entry

Node 1

Node 2

Go to Node 4

Node 3

Node 4

Node 4

$N = \{1,2,3,4\}$, $N_i = \{1\}$, $N_f = 4$, $E = (1,2), (1,3), (2,4), (3,4)$

If we look at our control flow graph and analyse the graph coverage (see diagram 1.2), we can establish which test cases will cover our graph as fully as possible. We can see we have 4 nodes and 4 edges and two valid test paths (Diagram 1.1). In order to cover test path 1 (t1), we need a test case with an even number. In order to cover test path 2(t2) we need to choose an odd number. In this way we can in fact get complete path coverage by covering both paths which cover all nodes and all edges. Paths between [1,4] and [3,4] are invalid and not possible to test.

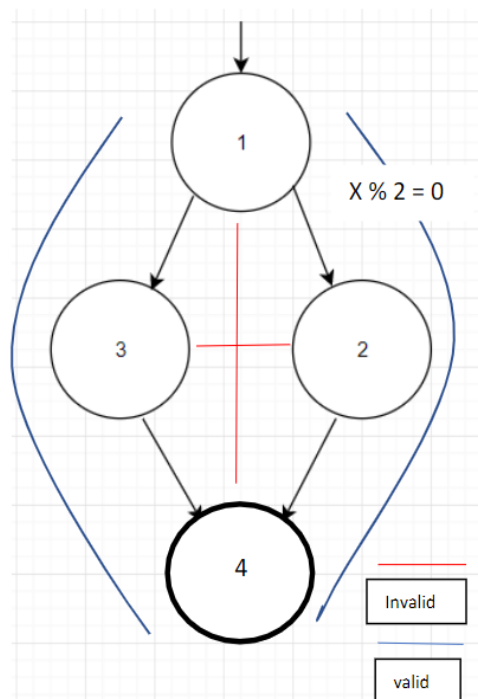


Diagram 1.2

$N = \{1, 2, 3, 4\}$, $N = \{1\}$, $E = (1, 2), (1, 3), (2, 4), (3, 4)$

$TR = \{1, 2, 3, 4\}$

Minimal Test path to meet TR		
t1	[1, 2, 4] (can be tested with $x = 2$)	Covers nodes 1, 2, 4 and edges [1, 2][2, 4]
t2	[1, 3, 4] (can be tested with $x = 1$)	Cover nodes 1, 3, 4 and edges [1, 3][3, 4]

Our test requirements need to be covered by both paths. These two paths visit all edges and all nodes. All nodes are reachable.

Invalid Test Paths	
1	1, 4
2	2, 3

3.3.1.2 Input Domain Modelling and Input Partitioning

Since we have such a large input domain, we were not satisfied that simply testing an odd or an even number would suffice, so an input domain modelling approach was taken to further define the test requirements. An input domain model helps to partition a very large domain of possible test values. Given that `isEven()` is a testable function, the characteristics of the functions input were determined and the input domain was partitioned into blocks and then further sub-partitioned. Again, we can separate the domain between odd and even, or in this case if the number is even (true/false). However, we also sub-partitioned our input and checked the values with reference to zero.

Partitioning and sub-partitioning for `isEven()`'s inputs:

Characteristic	Block 1			Block 2		
q1 = number is even	True			False		
	Less than 0	Equal to zero	Greater than 0	Less than 0	Equal to zero	Greater than 0

Based on this information we can then add possible values for the blocks:

Parameter	b1			b2		
X	4			5		
X	-4	0	2	-5	0	1

Here we can see some redundant tests as testing 0 is repeated. Therefore, I refined the test cases, eliminated the duplication of testing of case where $x = 0$.

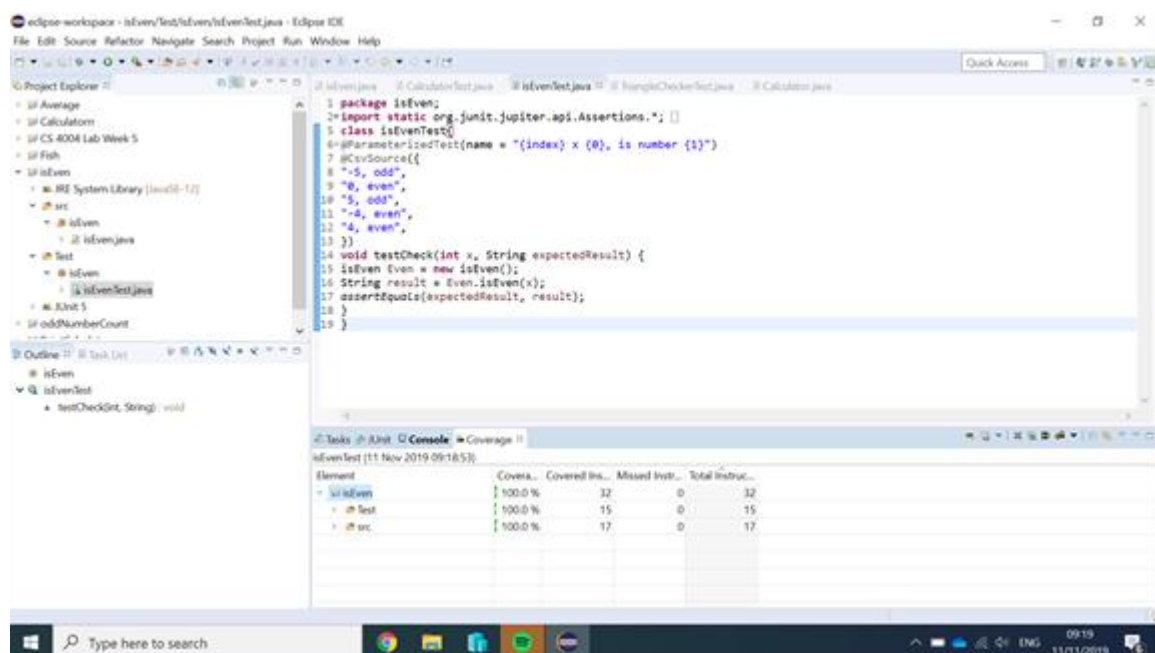
3.3.1.3 Boundaries

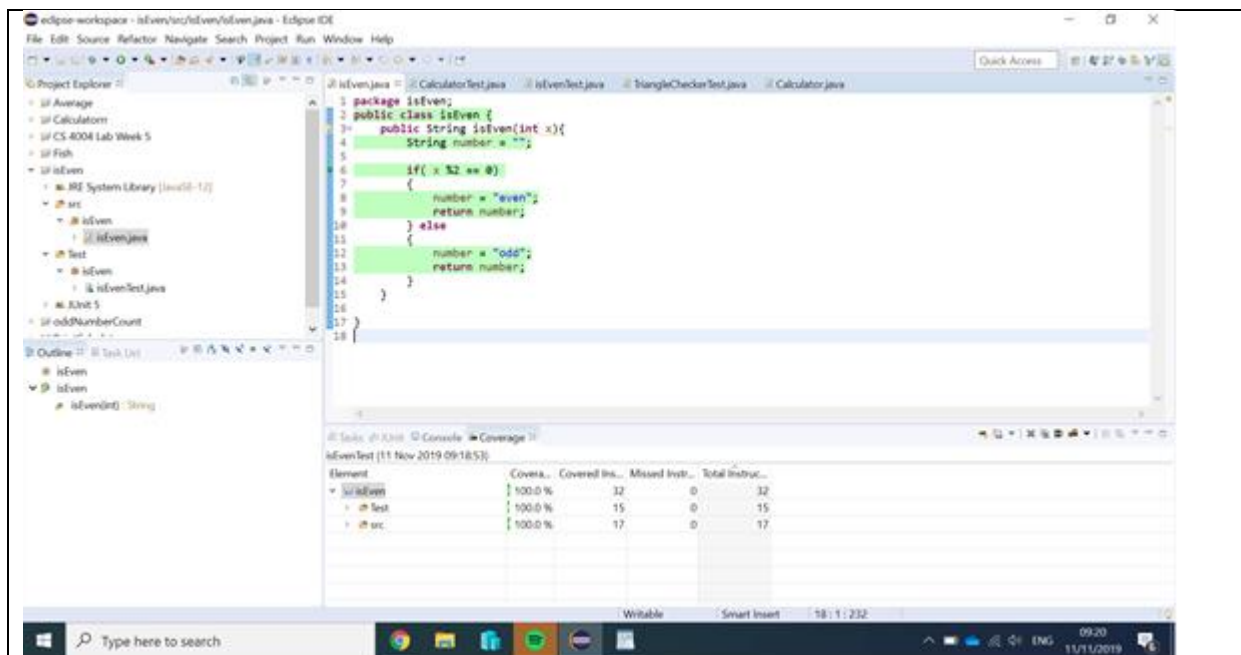
Finally, we tested limits. Therefore, we chose a very large integer to test of 1,000,000. This was a somewhat arbitrary choice. However, so that testing would not become excessive the final test cases were then chosen.

The final test cases and specifications were:

EC #	EC Description	Valid/Invalid	Specific example. Test case	Expected Outcome	Test paths
EC1	X = even	Valid	t1 = 2	even	[Test path p1: 1,2,4]
EC2	X = odd	Valid	t2 = 1	odd	[Test path: p2 1, 3,4]
EC3	X is even AND < 0	Valid	t3 = -4	even	[Test path: p1: 1, 2,4]
EC4	X = 0	Valid	t4 = 0	even	[Test path p1: 1,2,4]
EC 5	X is odd AND < 0	Valid	t5 = -5	odd	[Test path p2: 1,3,4]
EC 6	X is very large to test limits.	Valid	t6 = 1,000,000	even	[Test path p1: 1,2,4]

The tests were then ran on the code in Junit to confirm that all test cases provided full coverage of the two branches of the code. Indeed, there was full coverage of both paths/branches of code. We felt satisfied that the method worked correctly and had been tested sufficiently based on the coverage criteria and test specifications selected.





Program code:

```
package isEven;
public class isEven {
public String isEven(int x){
String number = "";
if(x % 2 == 0)
{
number = "even";
return number;
} else
{
number = "odd";
return number;
}
}
}
```

Test code:

```
package isEven;
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvSource;
class isEvenTest2{
@ParameterizedTest(name = "{index} x {0}, is number {1}")
@CsvSource({
"2, even",
"1, odd",
"-4, even",
"0, even",
"-5, odd",
"1000000, even"
})
void testCheck(int x, String expectedResult) {
isEven Even = new isEven();
String result = Even.isEven(x);
assertEquals(expectedResult, result);
}
```

3.3.1.4 *Example 2 – Testing nested 'if' statements*

In this exercise, we created a program to calculate the student's grade based on the score achieved. In the specifications, we established that the program had to calculate a student's letter grade given the following scores:

90-100 = A

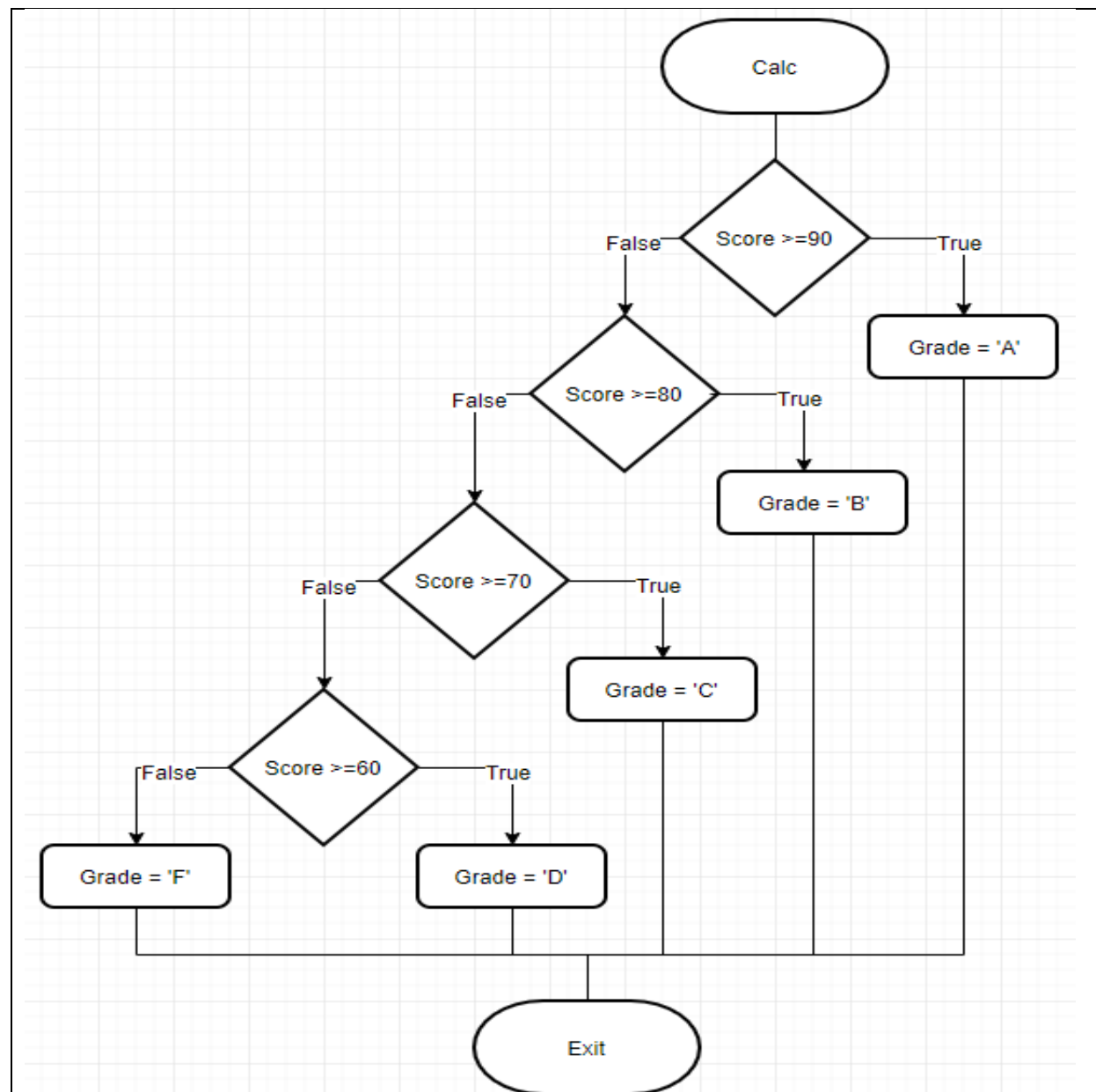
80-89 = B

70-79 = C

60-69 = D

Below 60 = F

To better illustrate, here is a flowchart for the module to calculate the grade.



To do a coverage criteria-based testing, we had to implement the appropriate source code.

```

public class GradeCalculator {
    public String ScoreToGrade(int score)
    {
        String grade = "";
        if (score > 100 || score < 0)
        { grade = "INVALID SCORE";
        }
        else if (score >= 90) {
            grade = "A";
        }
        else if (score >= 80) {
            grade = "B";
        }
        else if (score >= 70) {
            grade = "C";
        }
        else if (score >= 60) {
            grade = "D";
        }
        else {
            grade = "F";
        }
    }
}
  
```

```

    } else {
        grade = "F"; }
return grade;
}
}

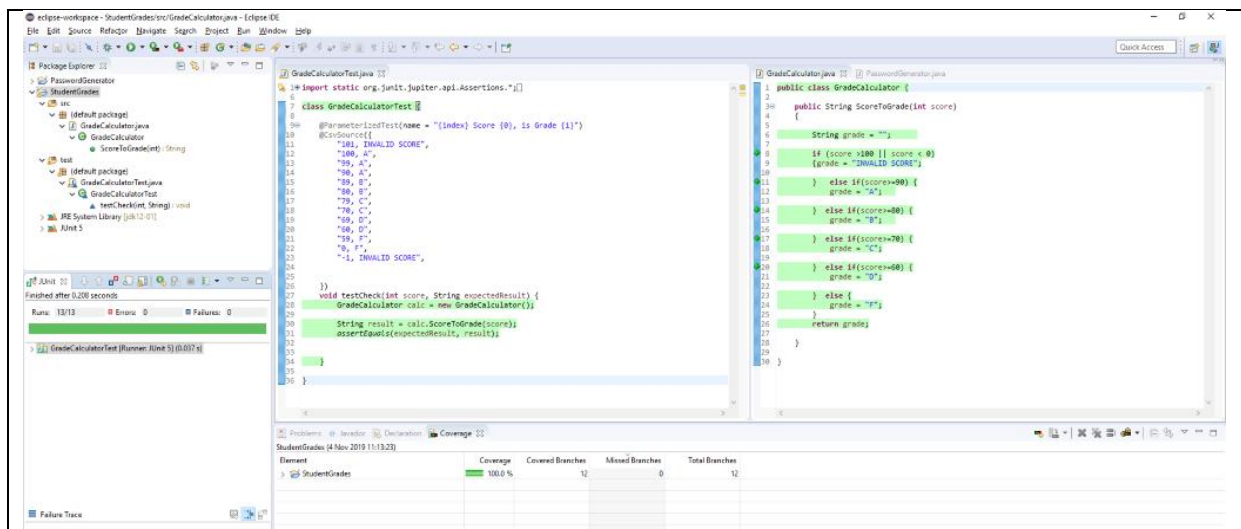
```

To test a code, we had to establish the test cases. For this, we used the partition borders: 0, 60, 70, 80, 90, 100. As per best industry practices, we checked for invalid values from outside of the allowed intervals (-1 and 101), the border cases (0, 60, 70, 80, 90, 100) and the score left to each border (59, 69, 79, 89, 99).

3.3.1.5 Equivalence classes for the test

EC#	EC Description	Valid/Invalid	Specific Example	Expected outcome
1	Score < 0	Invalid	-1	INVALID SCORE
2	Score = 0	Valid	0	F
3	Score < 60	Valid	59	F
4	Score = 60	Valid	60	D
5	60 <= Score < 70	Valid	69	D
6	Score = 70	Valid	70	C
7	70 <= Score < 80	Valid	79	C
8	Score = 80	Valid	80	B
9	80 <= Score < 90	Valid	89	B
10	Score = 90	Valid	90	A
11	90 <= Score <= 100	Valid	99	A
12	Score = 100	Valid	100	A
13	100 < Score	Invalid	101	INVALID SCORE

Using these test cases, we achieved 100% statement coverage, as per JUnit screen capture below:

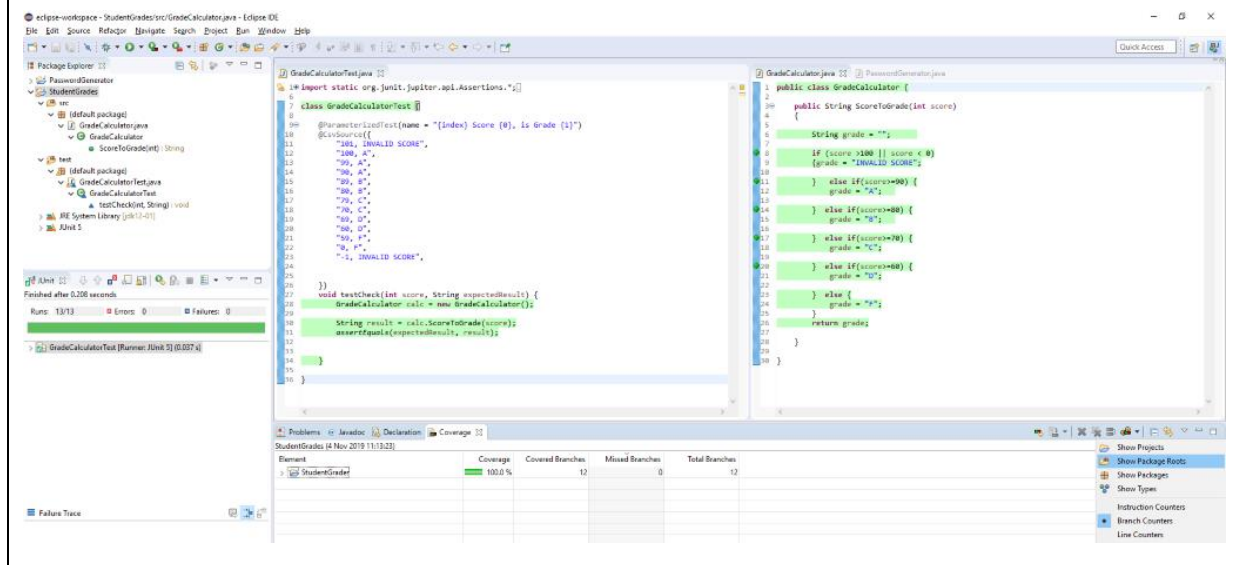


3.3.1.6 Statement Coverage

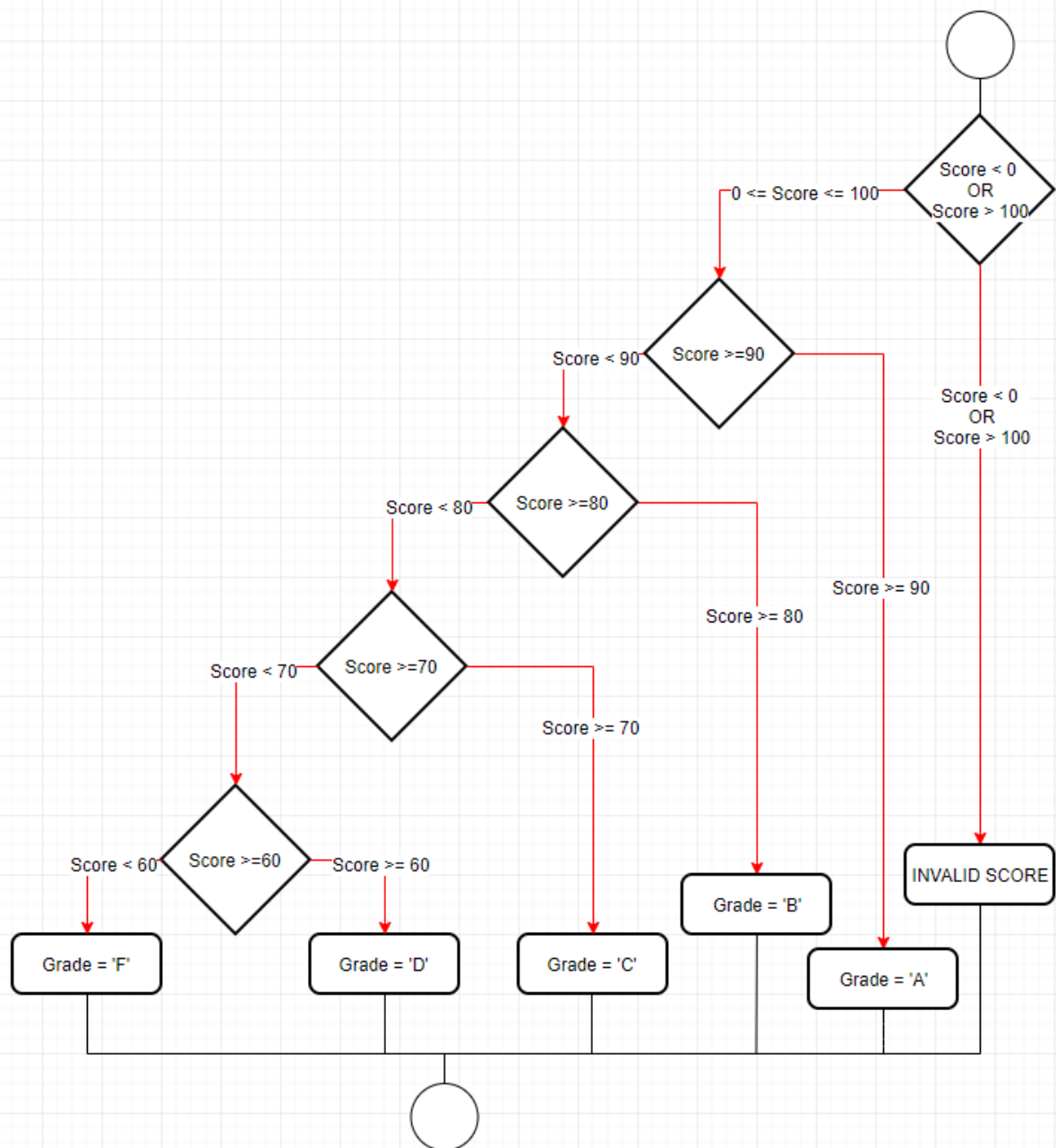
Statement coverage is the most widely used test in industry. It has its downsides, however. If there is an error in the programmer's logic, the statement coverage might not show it, because it only tests for each written statement. Equally, if a crucial statement is missing, the statement coverage test will not highlight it. Using JUnit, Coverage as JUnit Test function, it is easily automated.

3.3.1.7 Branch Coverage

As our program contains a significant amount of decision statements, we chose to do a branch coverage test in addition to the statement coverage. Branches are the outgoing edges of the decision points. The goal of the branch coverage test is to execute all the branches in the program. JUnit has a nice feature, where just with one click of a button; it can check the branch coverage too. The coverage is measured by taking the number of executed branches and divide it by the total number of branches. As the below screen capture shows, the code achieved 100% on this test too, meaning the test cases chosen cover all possible branches of the complex nested if structure in our code.



Another way to map the branch coverage is by using a Control Flow Graph (CFG)



It does not matter which kind of coverage we utilize, and how much coverage we achieve, there is always a chance that we might miss something.

3.3.1.8 Condition Coverage

In the first 'if' statement, we check for two conditions simultaneously. The score cannot be less than 0 or greater than 100. Any one of these cases tested would be enough to achieve 100% branch coverage, but it would not be enough i.e. if we check for 101, the statement will return false, and as a result, this branch is considered to be covered. However, we need to ensure the program can handle negative numbers also. To do this, we had to also consider the condition coverage while designing the test cases. In the condition testing, we test for each condition in a decision box, not

just each direction of the outcome.

The code used in the test:

```
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvSource;

class GradeCalculatorTest {

    @ParameterizedTest(name = "{index} Score {0}, is Grade {1}")
    @CsvSource({
        "-1, INVALID SCORE",
        "0, F",
        "59, F",
        "60, D",
        "69, D",
        "70, C",
        "79, C",
        "80, B",
        "89, B",
        "90, A",
        "99, A",
        "100, A",
        "101, INVALID SCORE",
    })
    void testCheck(int score, String expectedResult) {
        GradeCalculator calc = new GradeCalculator();

        String result = calc.ScoreToGrade(score);
        assertEquals(expectedResult, result);
    }
}
```

The code used in the program:

```
public class GradeCalculator {
    public String ScoreToGrade(int score)
    {
        String grade = "";
        if (score >100 || score < 0)
        {grade = "INVALID SCORE";
        }else if(score>=90) {
            grade = "A";
        } else if(score>=80) {
            grade = "B";
        } else if(score>=70) {
            grade = "C";
        } else if(score>=60) {
            grade = "D";
        } else {
            grade = "F";
        }
        return grade;
    }
}
```


3.3.1.9 Using STATEMENT COVERAGE to test an “if else” statement.

This is a statement coverage-based testing the method `verify()` which is used to check if a user's age is over 18 before they can buy a game. We looked at statement coverage-based testing because it is a frequently used method of white box testing. The first test case chosen was to check the age with the value of 19 (t1, age = 19). This test covers 70% of the code. The reason for this is it checks the user versus the game age rating and as the age is greater than 18 it will print out “Congratulations on game purchase” and then finish the program. Therefore, it only went through the if statement but never went through the ‘else if’. Therefore, we had to create another test case. The second test case was that the user is under the age of 18 and the game rating stays the same. The if statement asks if the user is over 18 which the user is not then it goes onto the else if. When the else if is executed it prints out “you're not old enough for the game order”. This then covers the last three lines, so that's 100% of the statement is covered by our two tests. We felt satisfied that both statements were covered and stopped testing at this point for this method.

The highlighted blue in the diagram below demonstrates test case 1 where the code passes through, skipping the else if.

Pseudo code	Test case#1
Naming variables inside verify function verify. User = 19; gage=18 If(user>gage) then Print “Congratulations on game purchase” Else if user<gage then Print “You're not old enough for the game order” End if Exit	<pre> 3 public void verify(int user,int gage); 4 user= 19; // user age 5 gage= 18; // age of game 6 if (user>gage) 7 { 8 System.out.print("Congratulations on game purchase"); 9 } 10 11 else if(user < gage) { 12 System.out.print("you're not old enough for the game order"); 13 } 14 } </pre>

The highlighted code in the next diagram shows that all code is executed with test cases 1 and 2.

Pseudo code	Test case#2
-------------	-------------

Naming variables
inside verify
function verify.
User = 17; gage=18
If(user>gage)
Print
"Congratulations on
game purchase"
Else if(user<gage)
Print "You're not
old enough for the
game order"
End if
Exit

```
4 public void verify(int user, int gage) {  
5     user = 17; // user age  
6     gage = 18; // age of game  
7     if(user>gage){  
8  
9         System.out.print("Congratulations on game purchase");  
10    }  
11    else if(user < gage) {  
12        System.out.print("you're not old enough for the game order");  
13    }  
14 }  
15 }
```

