|  |
| --- |
|  |
| Portfolio Question 5  Assessment Task 2 |
| |  |  |  | | --- | --- | --- | | Ben Royans [P205225] | 4/30/20 | Java III | |

|  |
| --- |
|  |
| Portfolio Question 5  Assessment Task 2 |
| |  |  |  | | --- | --- | --- | | Ben Royans [P205225] | 4/30/20 | Java III | |

TABLE OF CONTENTS

[UML 1](#_Toc39153036)

[Design Specifications 2](#_Toc39153037)

[Specification 2](#_Toc39153038)

[Review 2](#_Toc39153039)

[Version Control 3](#_Toc39153040)

[Selection 3](#_Toc39153041)

[Commit Type/Strategy 3](#_Toc39153042)

[Debugging 4](#_Toc39153043)

[Testing 5](#_Toc39153044)

[Test Table 5](#_Toc39153045)

[Screenshots 6](#_Toc39153046)

[Information Repositories 9](#_Toc39153047)

[BlackBoard 9](#_Toc39153048)

[GitHub 9](#_Toc39153049)

[Performance Screenshots 10](#_Toc39153050)

[Unit Testing 11](#_Toc39153051)

[References 12](#_Toc39153052)

# UML

Below are the class diagrams for the classes used in this project.



# Design Specifications

## Specification

An application is required to create a ‘To Do’ list from drag and drop elements. The application GUI should contain two separate lists, one containing a list of possible tasks and the other containing the selected tasks.

This application must have help files and include the use of 2D graphics on the GUI.

## Review

The GUI for this application will be developed in JavaFX. This will provide the functionality for the drag and drop elements as well as the 2D graphics.

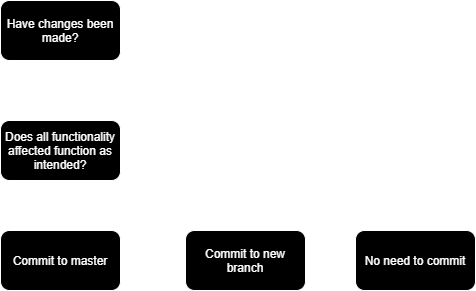
# Version Control

## Selection

GitHub will be used as the system to manage the version control for this project, with a commit being made after each development session.

## Commit Type/Strategy

The type of commit being made will follow this simple flow chart:

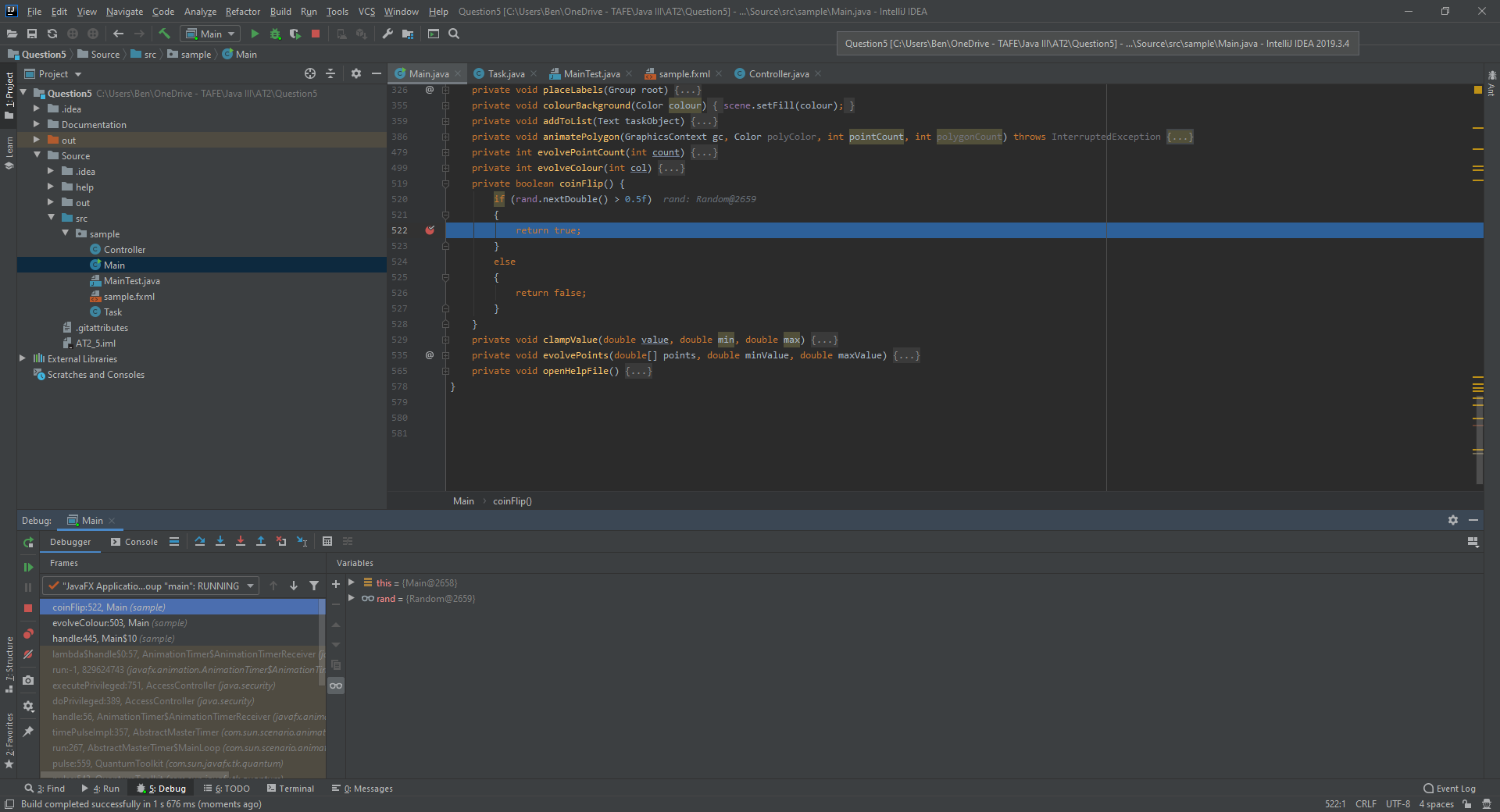


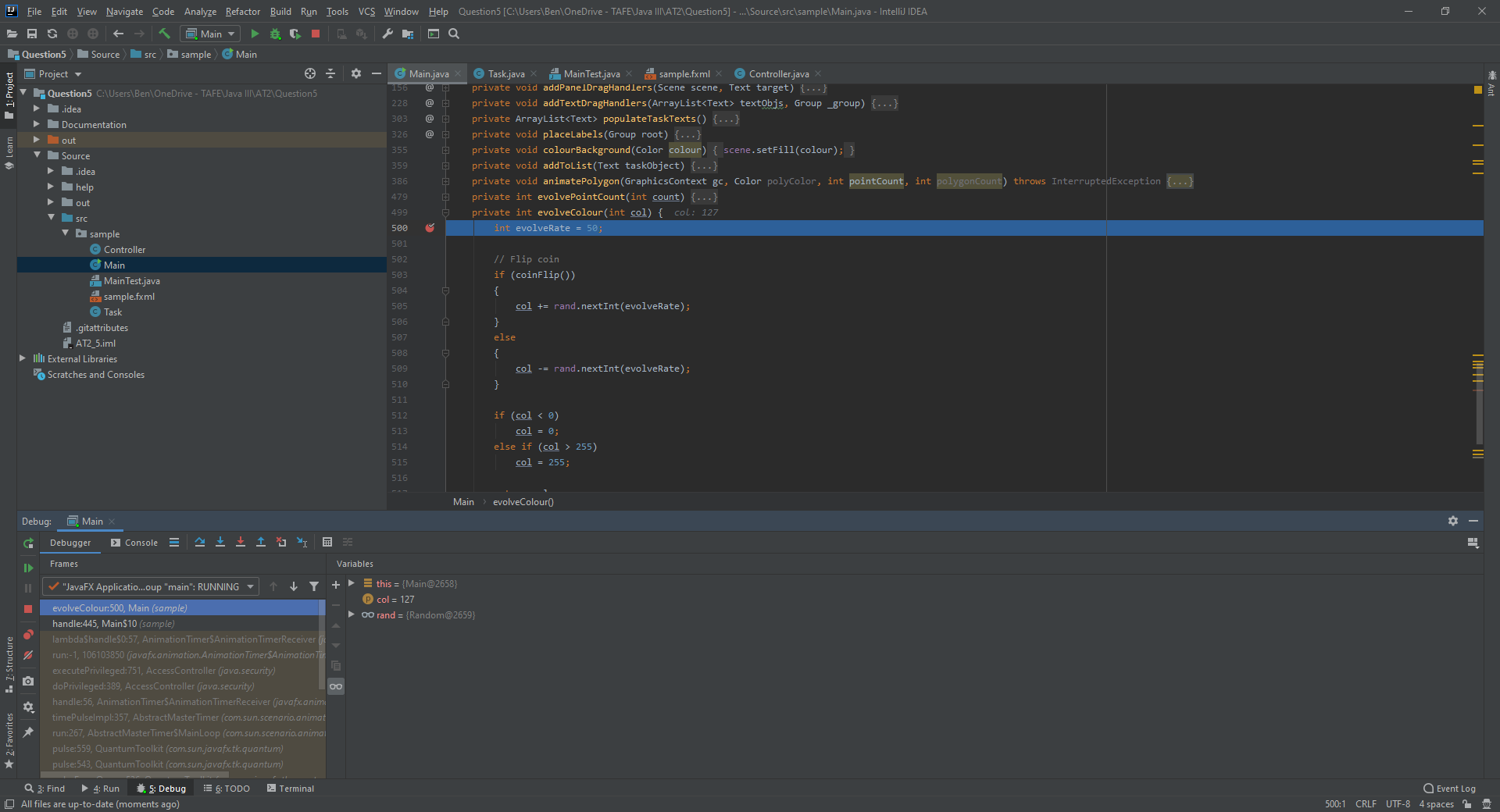
*Figure 1. Version control commit type flowchart.*

Changes to the code that have not been fully tested or are not completely functional as intended will be committed to a new branch. The new branch will continue to be developed until it meets the criteria to be committed to the master branch.

# Debugging

Below are some screenshots of the debugging process.





# Testing

## Test Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case #** | **Functionality Tested** | **Data/Procedure**  **Used** | **Expected**  **Outcome** | **Pass / Fail** |
| 1 | Populate list. | Empty ToDo list. | Task is added to list. | PASS |
| 2 | Populate list. | ToDo list with under 18 tasks. | Task is added to list. | PASS |
| 3 | Populate list. | ToDo list with 18 tasks. | Error message. | PASS |
| 4 | Reset list. | Populated list. | List clears to 0 items. | PASS |
| 5 | Reset list. | Un-populated list. | List clears to 0 items. | PASS |
| 6 | Reset background. | Populated list. | Background 2D art is reset, list is retained. | PASS |
| 7 | Reset background. | Un-populated list. | Background 2D art is reset, list remains un populated. | PASS |

## Screenshots

|  |  |
| --- | --- |
| **Case #** | **Screenshot** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |

# Information Repositories

## BlackBoard

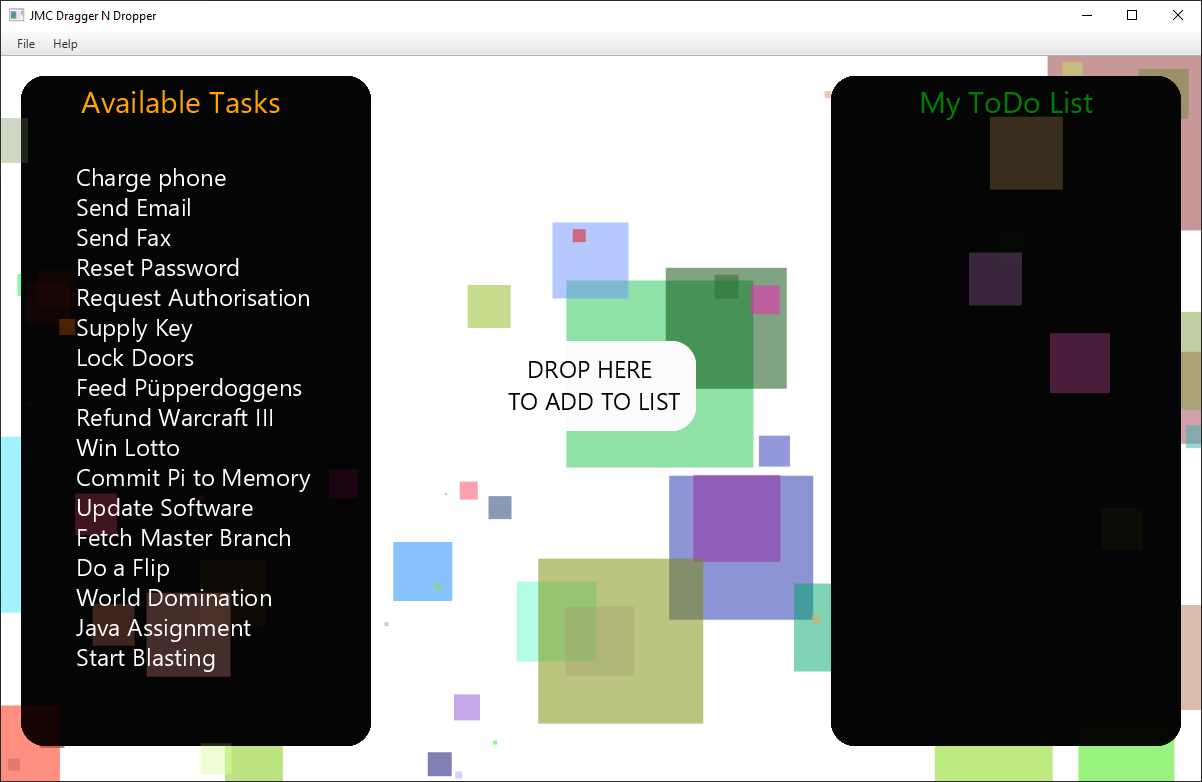
The assessment specifications and requirements have been supplied through BlackBoard. BlackBoard also provides the learning content used for this assessment task and will provide the methods for deployment (handing in the assessment).

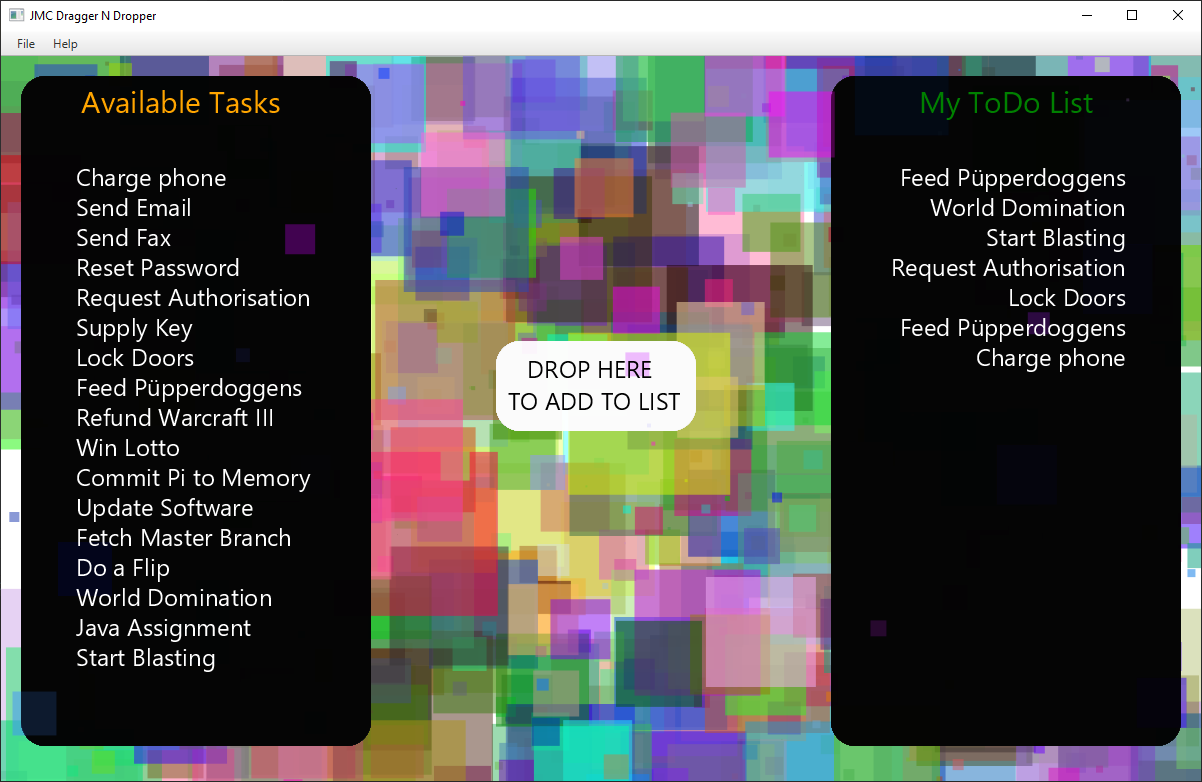
## GitHub

GitHub will be used as version control software for the project. Regular and systematic commits to either the master or branches of the code will provide a safety net for unexpected errors on the working application.

Public link to repository: <https://github.com/RuggedRadius/JavaIII_AT2.5/>

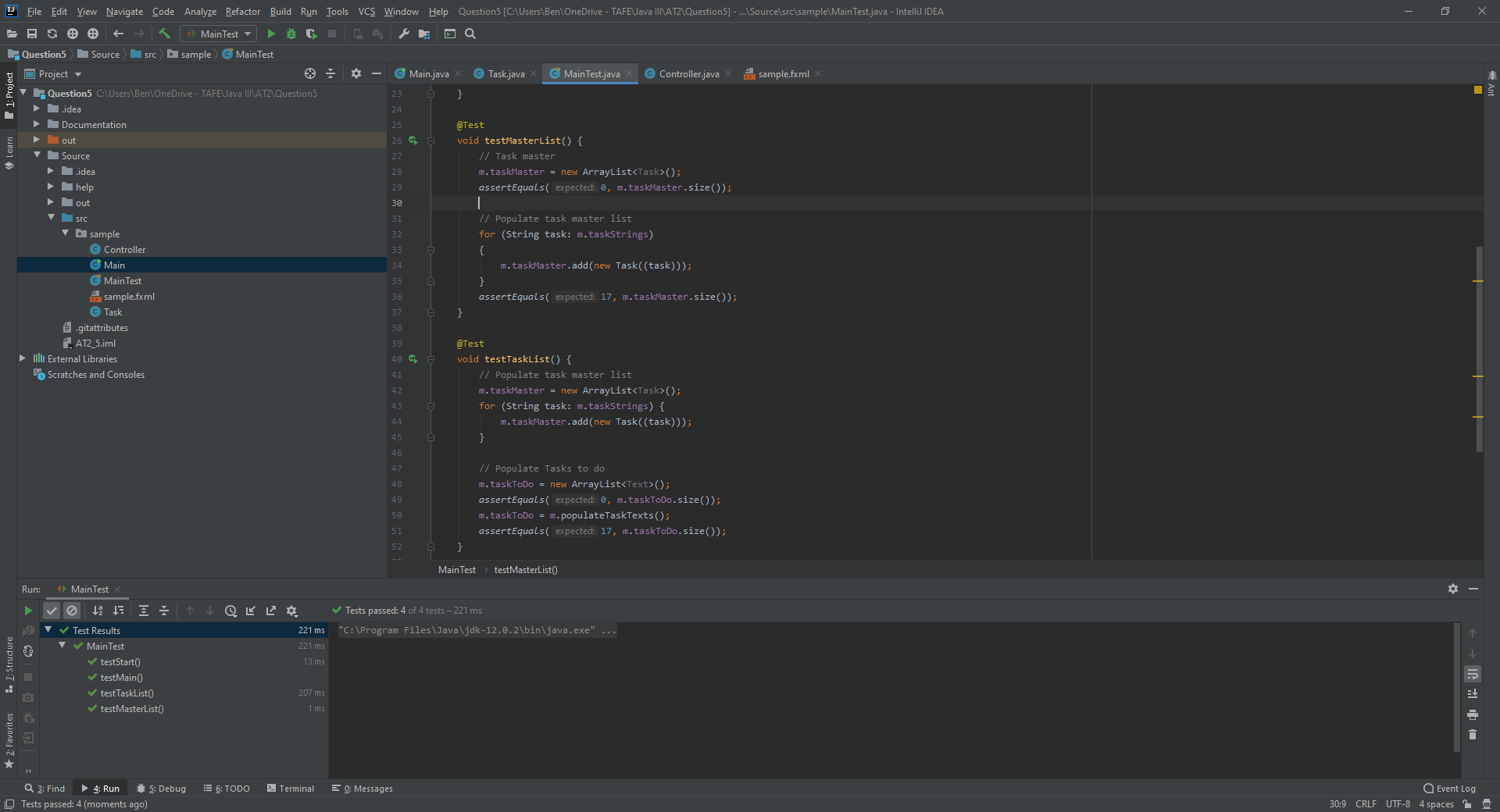
# Performance Screenshots





# Unit Testing

Junit 5 was used to complete the unit testing for this project. The Junit test class is included in the project files. Below is a screenshot of the Junit tests completing successfully.



# References

Bechtold, S. (2020). *JUnit 5 User Guide*. Retrieved from Junit: https://junit.org/junit5/docs/5.3.0-RC1/user-guide/index.html#installation

Lacar, J. (2014). *How to test method without no return value by using JUnit.* Retrieved from Code Ranch: https://coderanch.com/t/611976/engineering/test-method-return-JUnit

South Metropolitan TAFE. (2020). *Session 10 - Testing*. Retrieved from BlackBoard: https://blackboard.southmetrotafe.wa.edu.au/webapps/blackboard/content/listContent.jsp?course\_id=\_12140\_1&content\_id=\_1170302\_1