|  |
| --- |
|  |
| Portfolio Question 7  Assessment Task 2 |
| |  |  |  | | --- | --- | --- | | Ben Royans [P205225] | 4/30/20 | Java III | |

|  |
| --- |
|  |
| Portfolio Question 7  Assessment Task 2 |
| |  |  |  | | --- | --- | --- | | Ben Royans [P205225] | 4/30/20 | Java III | |

TABLE OF CONTENTS

[UML 1](#_Toc40967264)

[Server 1](#_Toc40967265)

[Client 2](#_Toc40967266)

[Design Specifications 3](#_Toc40967267)

[Specification 3](#_Toc40967268)

[Review 3](#_Toc40967269)

[Version Control 4](#_Toc40967270)

[Selection 4](#_Toc40967271)

[Commit Type/Strategy 4](#_Toc40967272)

[Debugging 5](#_Toc40967273)

[Testing 6](#_Toc40967274)

[Screenshots 7](#_Toc40967275)

[Information Repositories 14](#_Toc40967276)

[BlackBoard 14](#_Toc40967277)

[GitHub 14](#_Toc40967278)

[Performance Screenshots 15](#_Toc40967279)

[Server 15](#_Toc40967280)

[Client 17](#_Toc40967281)

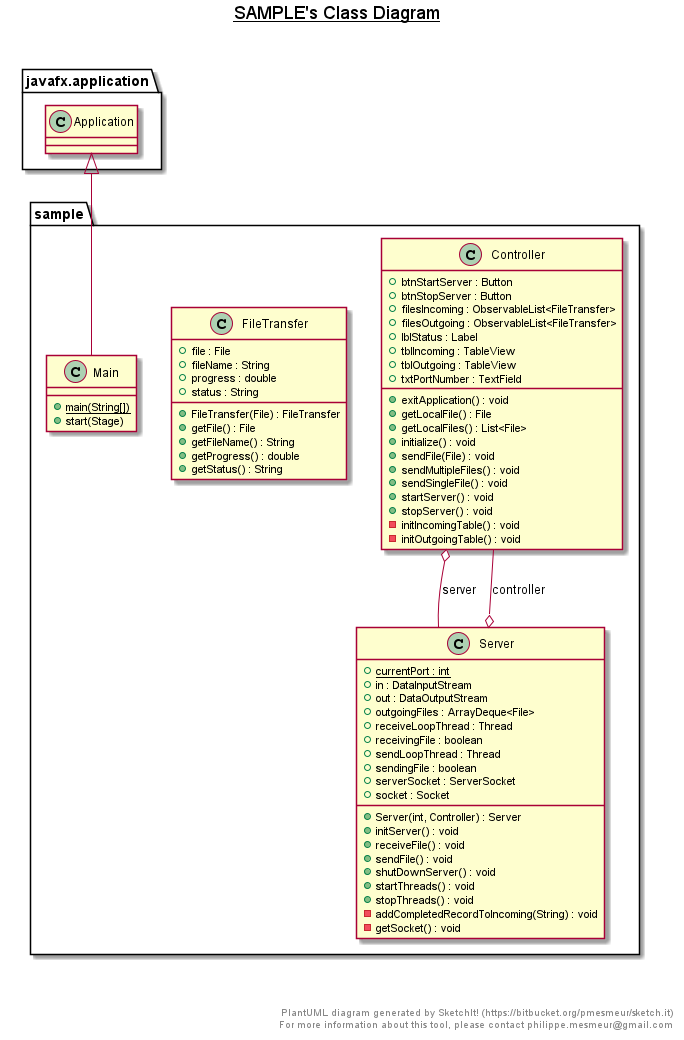
[Unit Testing 19](#_Toc40967282)

[References 20](#_Toc40967283)

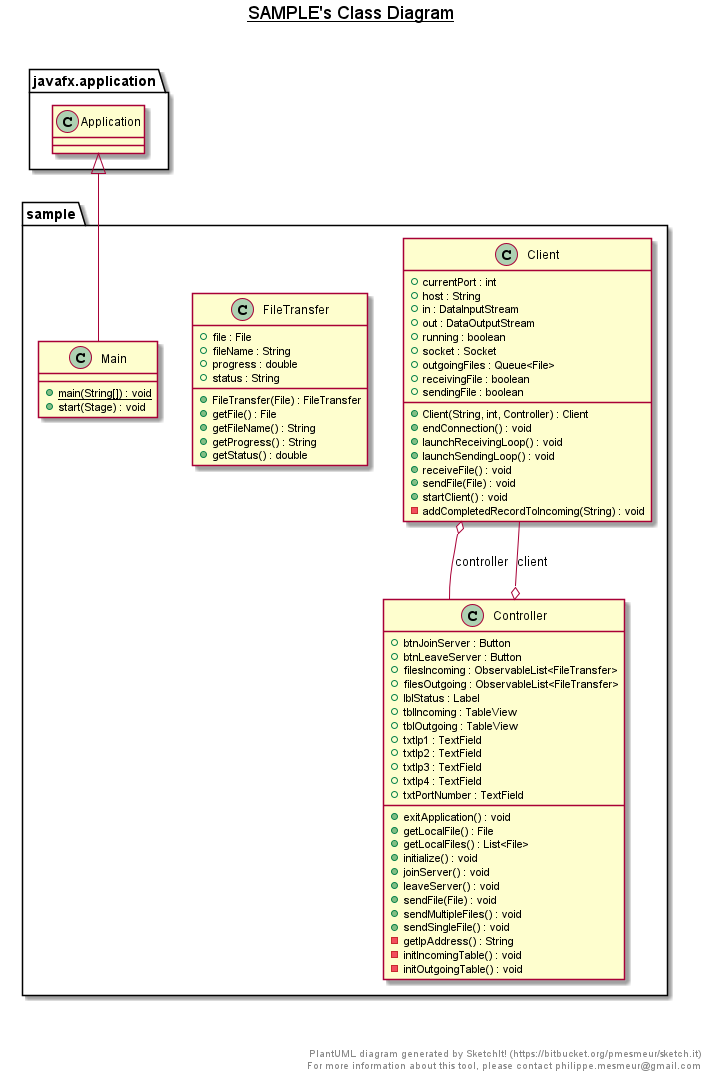
# UML

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

## Server



## Client



# Design Specifications

## Specification

An application is required to transfer data files between remote locations via a sockets protocol.

## Review

The application will be developed in JavaFX to provide the user with a UI and ease of file selection when transferring files. The Server and the Client applications will be separate coding projects in order to separate their logic, however some functions will be shared between the two.

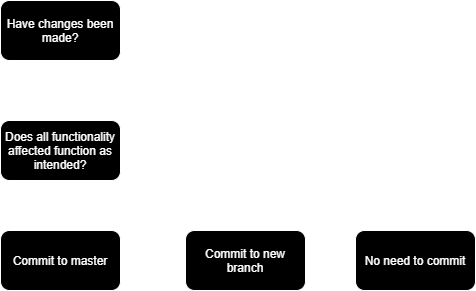
# Version Control

## 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.

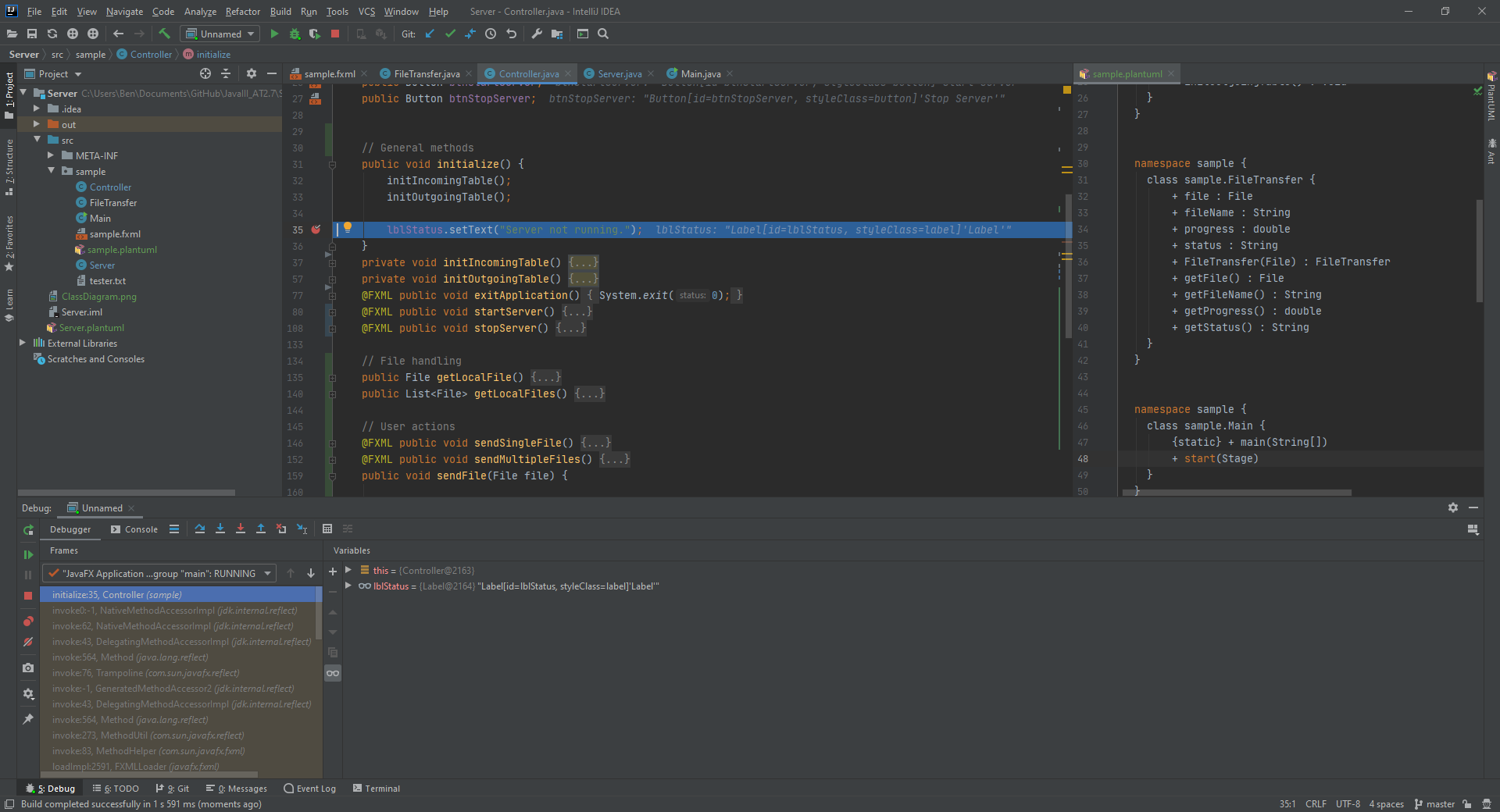


Figure . Debugging Server.java.

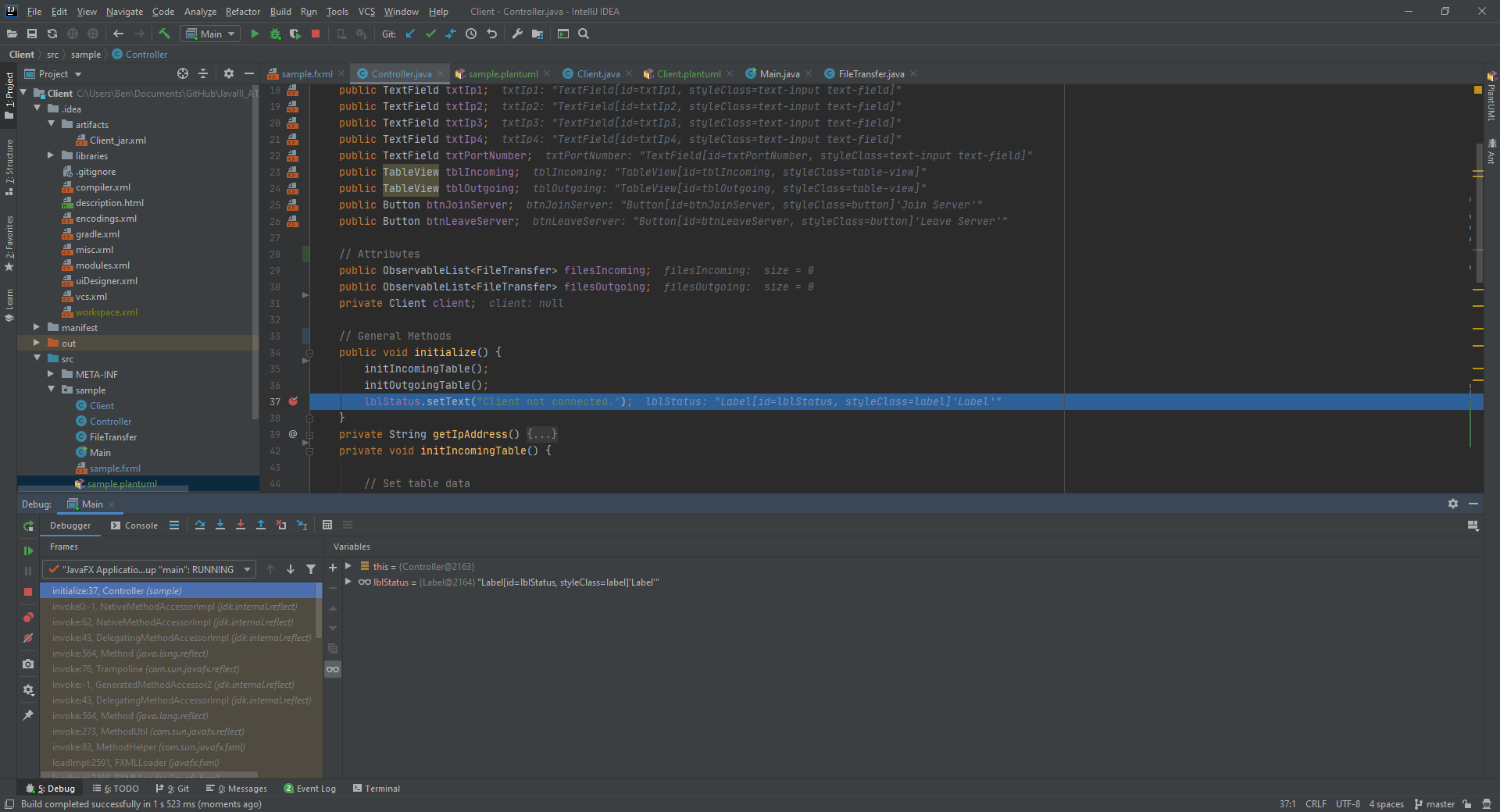


Figure . Debugging Client.java

# Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case #** | **Functionality Tested** | **Data/Procedure**  **Used** | **Expected**  **Outcome** | **Pass / Fail** |
| 1 | Start Server | * - | Server starts on specified port. | PASS |
| 2 | Client Join Server | * Invalid IP Address used. | Client fails to connect, and continues to retry on loop. | PASS |
| 3 | Client Join Server | * Valid IP Address, * Wrong port. | Client fails to connect, and continues to retry on loop. | PASS |
| 4 | Client Join Server | * Valid IP Address, * Correct port, * Server not running. | Client fails to connect, and continues to retry on loop. | PASS |
| 5 | Client Join Server | * Valid IP Address, * Correct port, * Server running. | Client connects to server. | PASS |
| 6 | Server – Send File (Single) | * A single .csv file is sent. | File is sent successfully by the Server, and received by the Client. | PASS |
| 7 | Server – Send File (Multiple) | * Multiple files of various formats used. | All files sent successfully by the Server, and received by the Client. | PASS |
| 8 | Client – Send File (Single) | * A single .csv file is sent. | File is sent successfully by the Client, and received by the Server. | PASS |
| 9 | Client – Send File (Multiple) | * Multiple files of various formats used. | All files sent successfully by the Client, and received by the Server. | PASS |
| 10 | Server – Send File (Single) | * Server stopped. * No client connected | Error message is displayed. | PASS |
| 11 | Server – Send File (Multiple) | * Server stopped. * No client connected | Error message is displayed. | PASS |
| 12 | Client – Send File (Single) | * Server stopped. * No client connected | Error message is displayed. | PASS |
| 13 | Client – Send File (Multiple) | * Server stopped. * No client connected | Error message is displayed. | PASS |

## Screenshots

|  |  |
| --- | --- |
| **Case #** | **Screenshot** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |

# 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_AT2.7/>

# Performance Screenshots

## Server

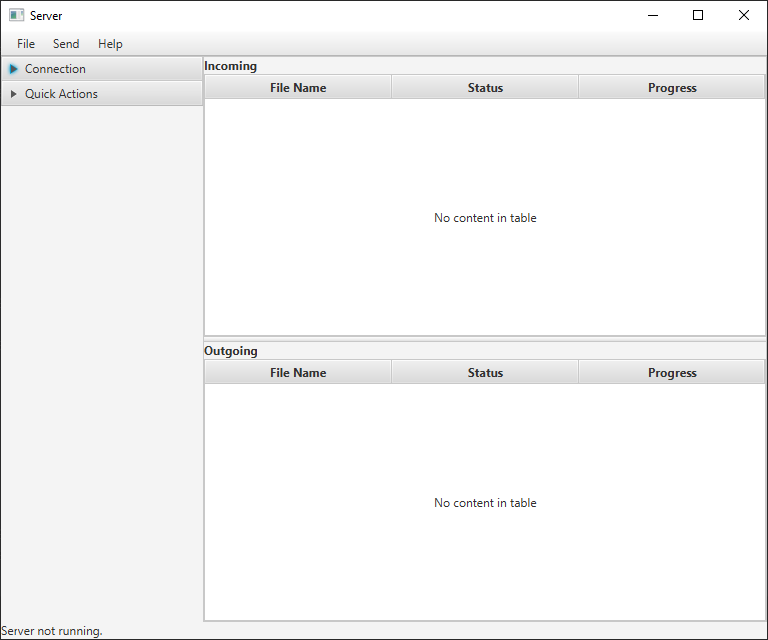


Figure . On initialisation.

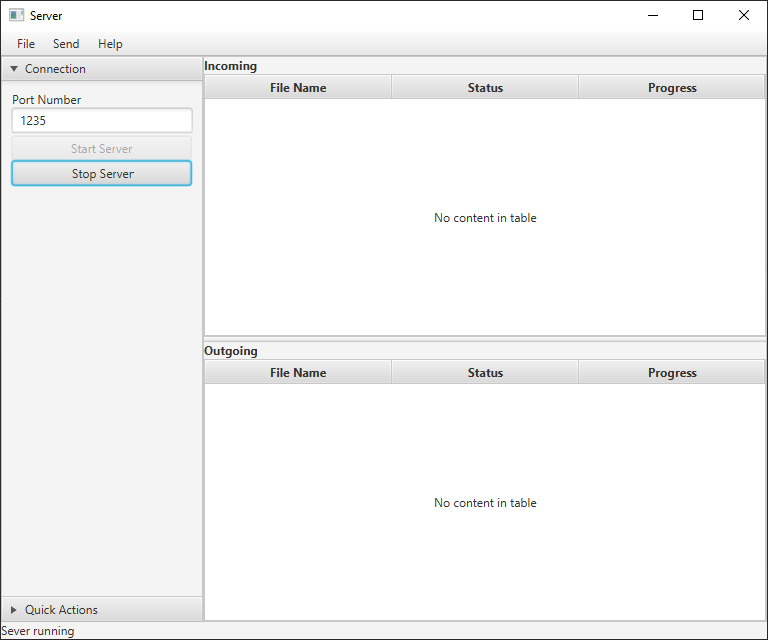


Figure . Starting a server.

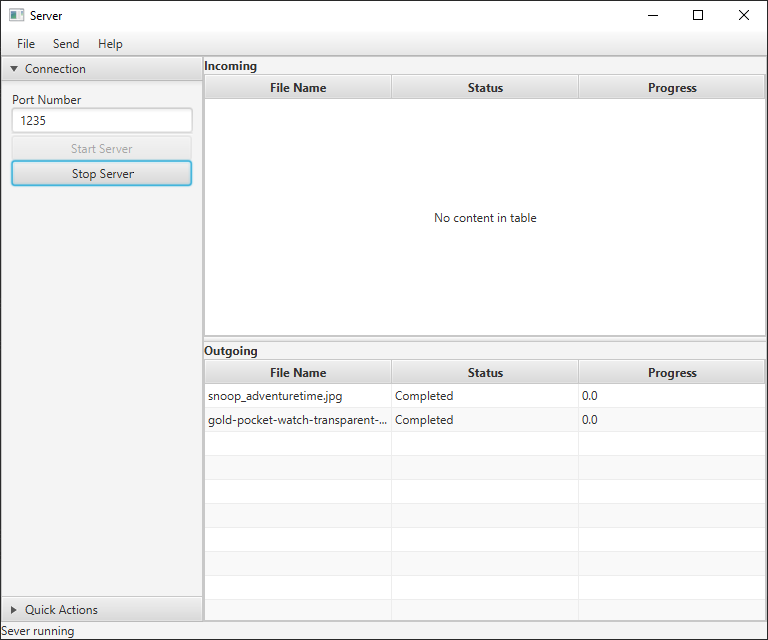


Figure . After transferring files.

## Client

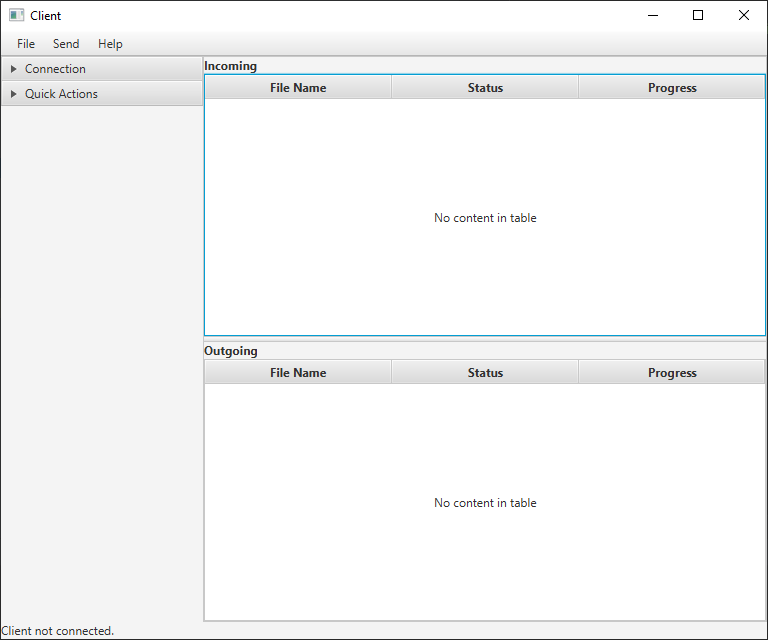


Figure . On initialisation.

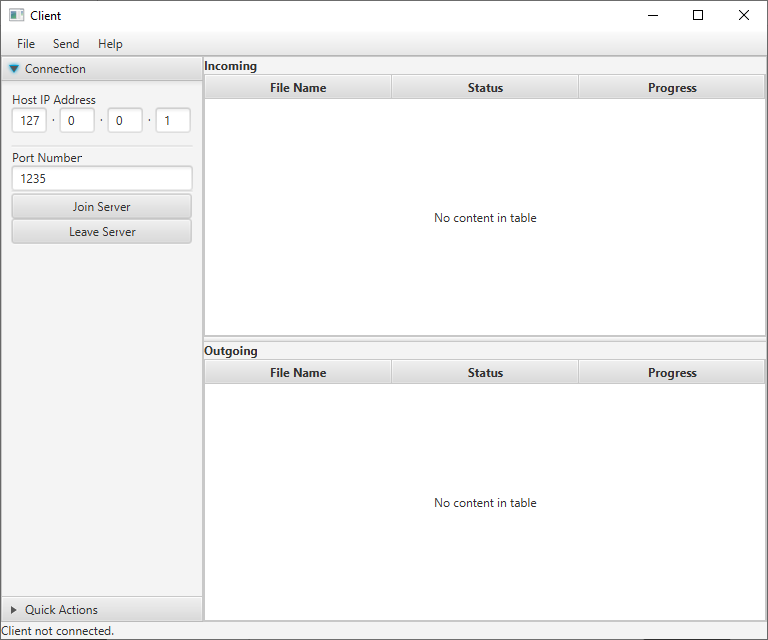


Figure . Connecting to a server.

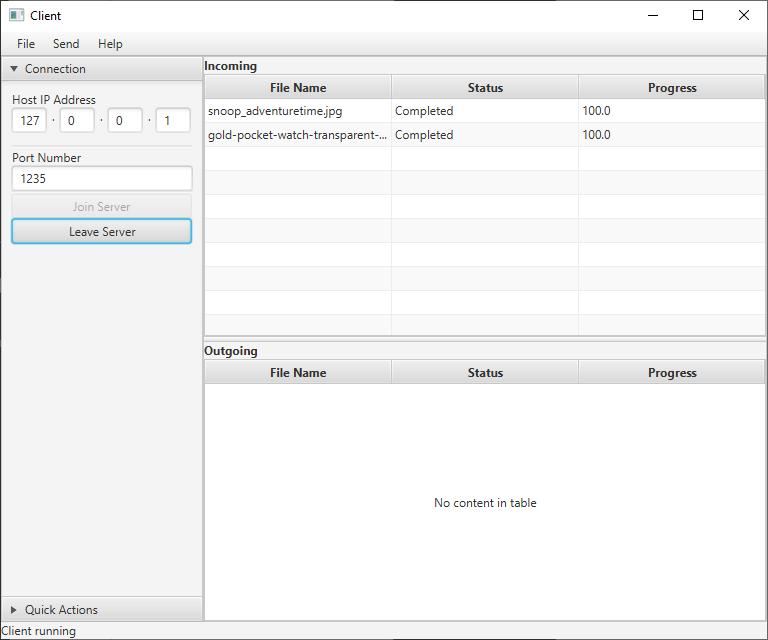


Figure . Receiving files from the server.

# 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.

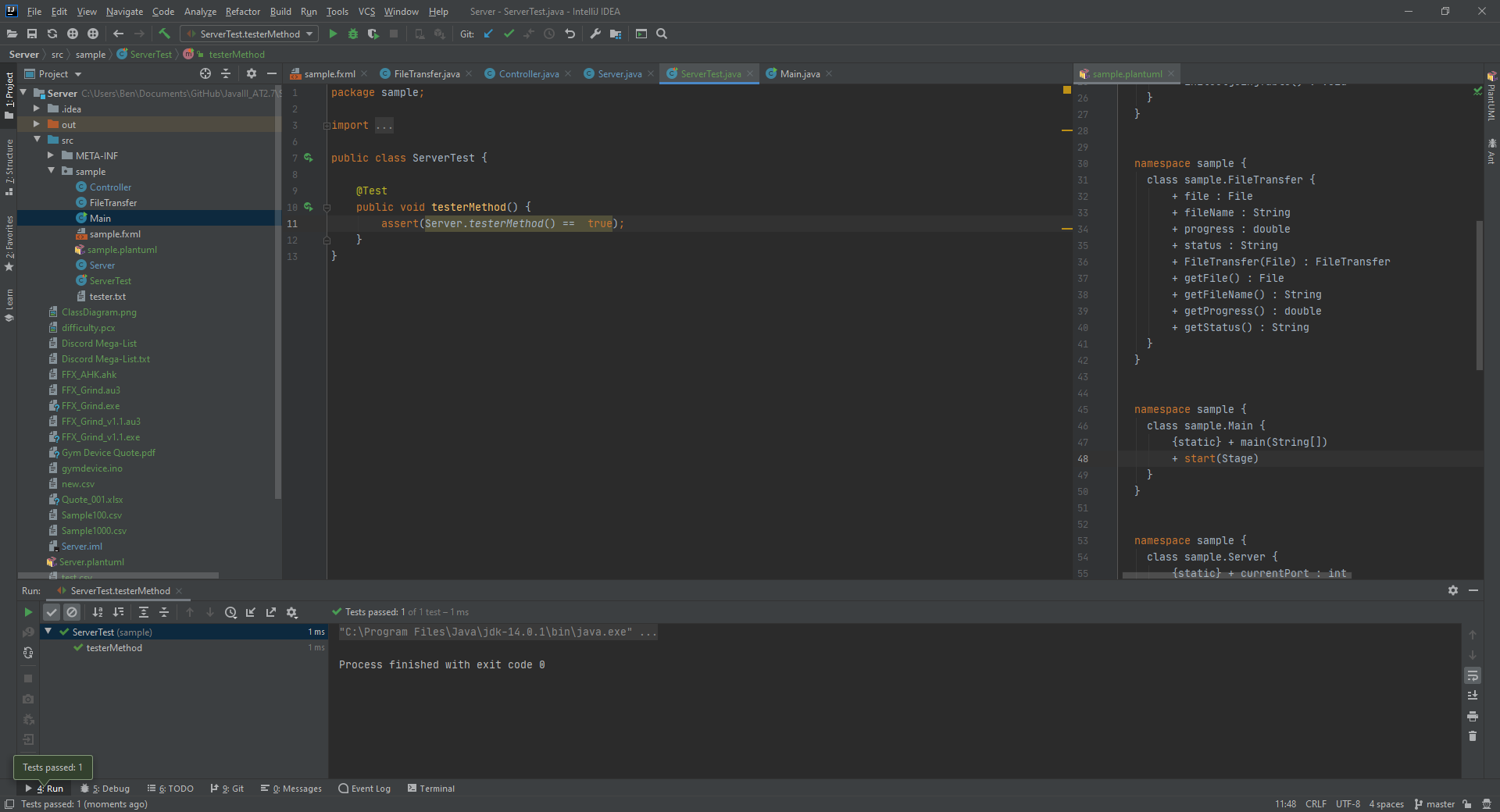


Figure . JUnit test completed successfully.

# 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