Jyle Darling

M093437

Predesign specification document

AT3

Table of Contents

[UML Class Diagrams 1](#_Toc57296712)

[Server Application 1](#_Toc57296713)

[Client Application 1](#_Toc57296714)

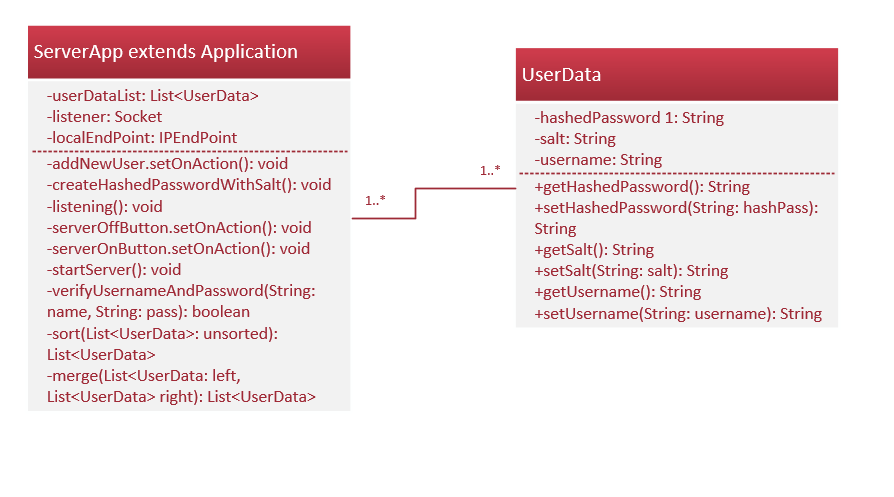
[Test Cases 2](#_Toc57296715)

[Junit Sample Tests 3](#_Toc57296716)

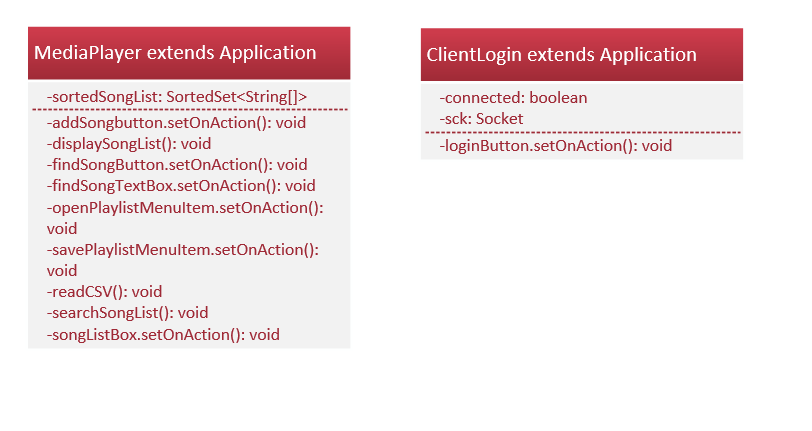
[Source Control 6](#_Toc57296717)

# UML Class Diagrams

## Server Application



## Client Application



# Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Description** | **Expected Result** | **Actual Result** | **Comments** |
| 01. | Server - Creating a new user in server application |  |  |  |
| 02. | Server - Starting server socket connection |  |  |  |
| 03. | Server - Stopping the server connection |  |  |  |
| 04. | Client – Login attempt when server not connected. |  |  |  |
| 05. | Client – Login attempt invalid credentials while server is connected. |  |  |  |
| 06. | Client – Login attempt valid credentials server connected. |  |  |  |
| 07. | Client – Opening CSV file containing song playlist via menu bar item. |  |  |  |
| 08. | Client – Adding new songs to the playlist |  |  |  |
| 09. | Client – playing a song by double clicking the item in the list box |  |  |  |
| 10. | Client – Removing a song from the list box |  |  |  |
| 11. | Client – Removing a song from the list box with nothing selected. |  |  |  |
| 12. | Client – Searching for a song in the list |  |  |  |
| 13. | Client – Searching for a song not in the list |  |  |  |
| 14. | Saving a playlist to a csv file |  |  |  |
| 15 | Loading recently saved file to ensure data was saved correctly |  |  |  |

# Junit Sample Tests

Below is a demonstration of Junit testing.

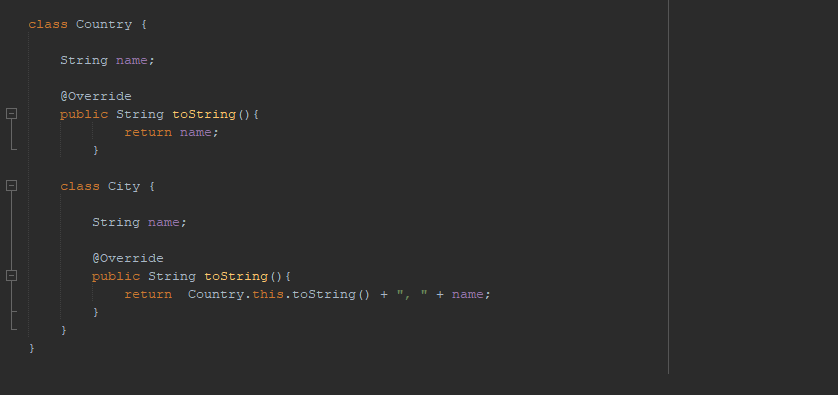


Figure 1 Country Class with nested City Class

The above figure is a simple class used in an earlier project. It demonstrated nested classes, and overwrites the toString() method to better illustrate this



Figure 2 JUnit Test setup

In the above figure we can see a Junit test being set up to test the output of the toString() method.

We first create an test instance of a new Country object. We then set the name of the Country instance to “CountryNameTest”.

After this we specify to the Test what the expected output should be from the test, in this case its “CountryNameTest”.

We then assign the result of the test to the country instance toString() method, and then determine if the result matches the expected result.

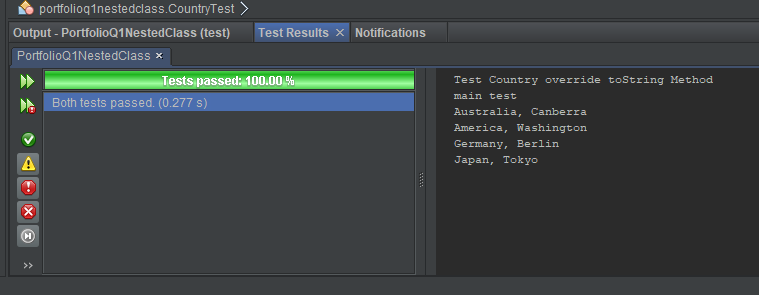


Figure 3 Test results

In the above image we can see the results of the test, another test was also run to test the main method. As you can see both tests passed, which means that our expected results matched out actual results.

# Source Control

For this project I have opted to use GitHub as the repository for this applications code and documentation.

The publically accessible Repository can be found at: <https://github.com/Jely101/Java>

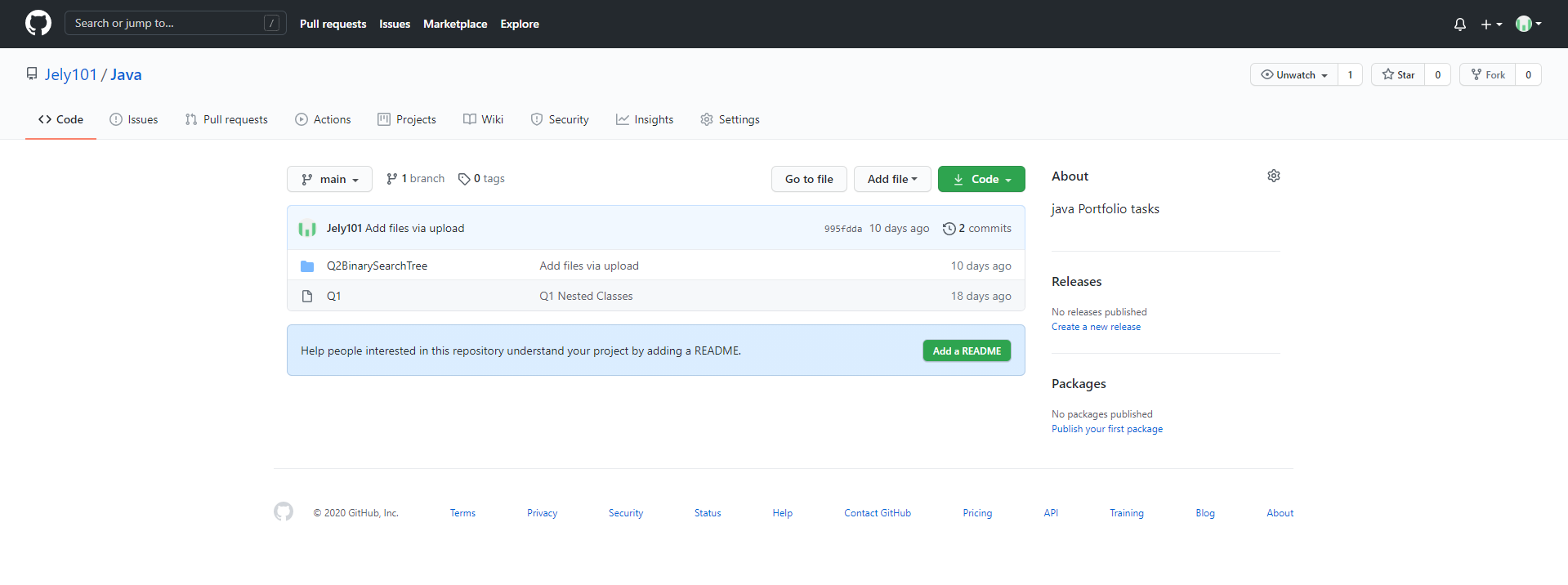


Figure 4 Image of the Repository at the time of producing this document.