|  |
| --- |
|  |
| Portfolio Question 6  Assessment Task 2 |
| |  |  |  | | --- | --- | --- | | Ben Royans [P205225] | 7/5/20 | Java III |   7/ |

|  |
| --- |
|  |
| Portfolio Question 6  Assessment Task 2 |
| |  |  |  | | --- | --- | --- | | Ben Royans [P205225] | 7/5/20 | Java III | |

TABLE OF CONTENTS

[UML 1](#_Toc40260297)

[Design Specifications 2](#_Toc40260298)

[Specification 2](#_Toc40260299)

[Review 2](#_Toc40260300)

[Version Control 3](#_Toc40260301)

[Selection 3](#_Toc40260302)

[Commit Type/Strategy 3](#_Toc40260303)

[Debugging 4](#_Toc40260304)

[Testing 5](#_Toc40260305)

[Test Table 5](#_Toc40260306)

[Screenshots 6](#_Toc40260307)

[Information Repositories 9](#_Toc40260308)

[BlackBoard 9](#_Toc40260309)

[GitHub 9](#_Toc40260310)

[Performance Screenshots 10](#_Toc40260311)

[Unit Testing 12](#_Toc40260312)

[References 13](#_Toc40260313)

# UML

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



# Design Specifications

## Specification

The client (JMC) requires an application to read and write .csv (Comma Separated Values) files using a Graphical User Interface (GUI). This application must use a 3rd party library.

## Review

This application will be developed in JavaFX to provide a modern feel for the UI/UX and appropriate controls for displaying the data.

Apache Commons CSV will be used at the 3rd party library for this application as it is a popular and proven choice for dealing with .csv files. It will be used to read .csv files into a data structures, and write the data from the structures back to .csv files.

JavaFX’s TableView control requires the implementation of a class object as the ‘model’ for each row. A Row class will be constructed to hold the values of each cell in a given row.

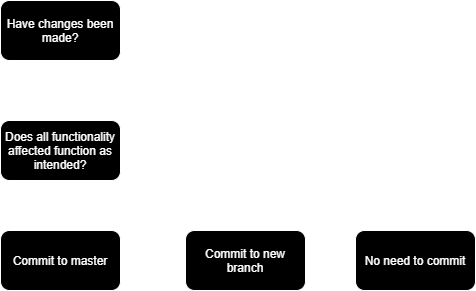
# Version Control

## 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.

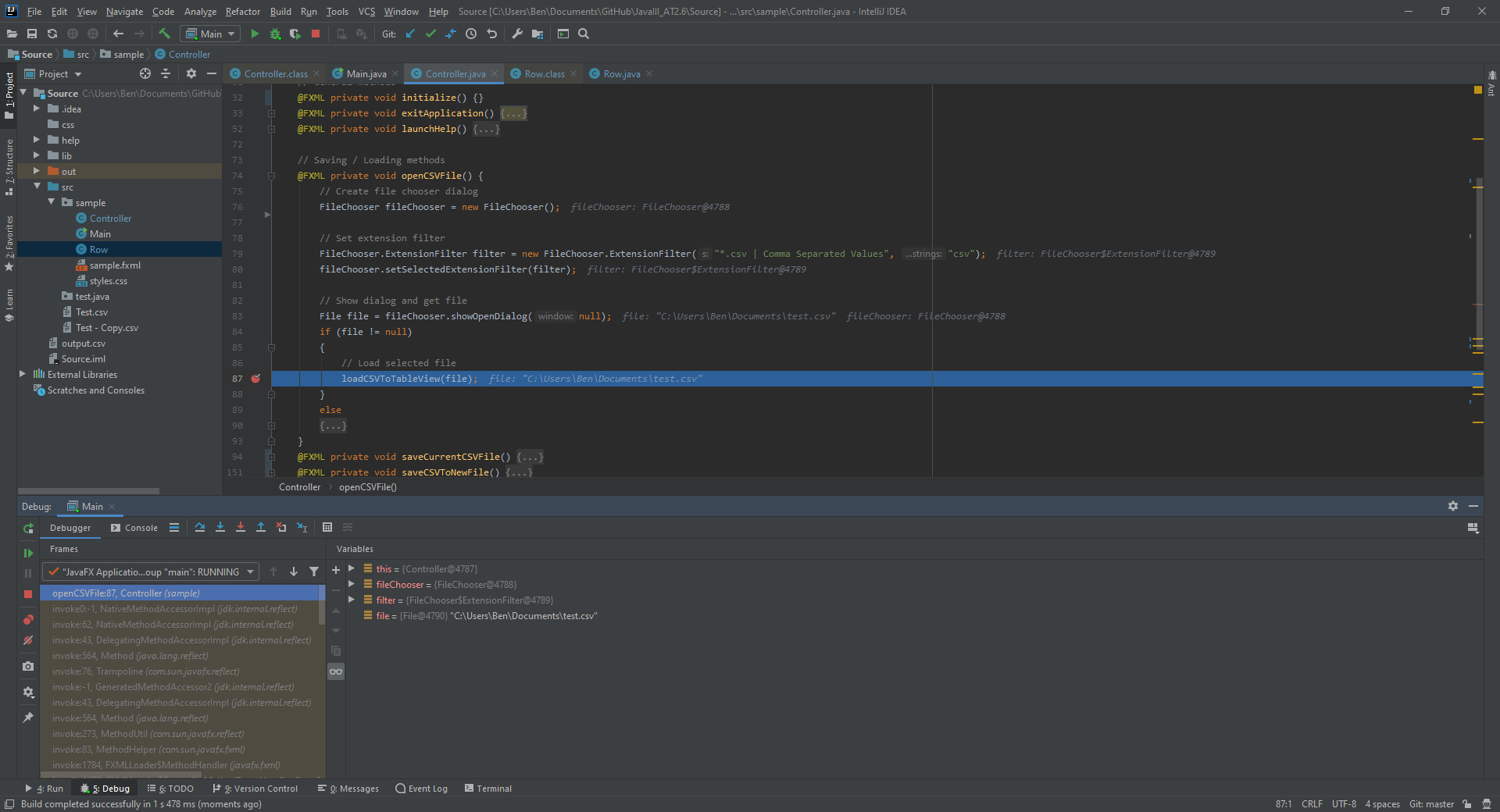


Figure . Debugging the loading process.

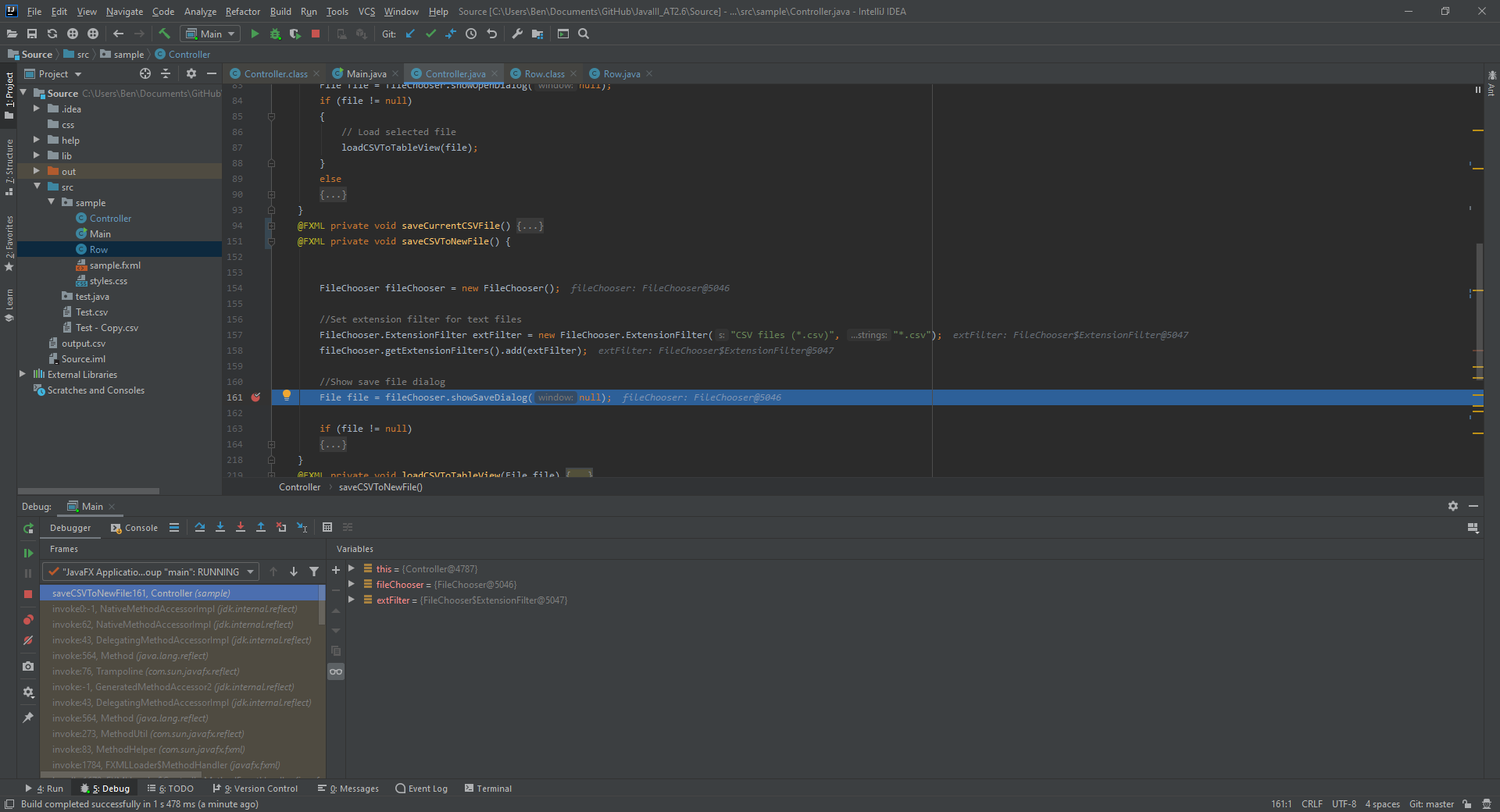


Figure . Debugging the saving to new file process.

# Testing

## Test Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case #** | **Functionality Tested** | **Data/Procedure**  **Used** | **Expected**  **Outcome** | **Pass / Fail** |
| 1 | Open File | Open file type that is not .csv. | Error Message | PASS |
| 2 | Open File | Open file type that is .csv. | .csv file is loaded into TableView in GUI. | PASS |
| 3 | Save File | Click File -> Save. | Current .csv file is saved with any modifications to file. | PASS |
| 4 | Save File As | Click File -> Save As.. (Using a unique file name). | File is saved as a new .csv as named. | PASS |
| 5 | Save File As | Click File -> Save As.. (Using a duplicate file name). | A confirmation dialog is shown. | PASS |
| 6 | Edit .csv data | Double click any cell and edit it’s value. | Value in cell is modified. | PASS |

## Screenshots

|  |  |  |
| --- | --- | --- |
| **Case #** | **Screenshot** | |
| 1 |  | |
| 2 |  | |
| 3 |  | |
| 4 |  | |
| 5 |  | |
| 6 | Figure . Before | Figure . After |

# 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_AT2.6/>

# Performance Screenshots

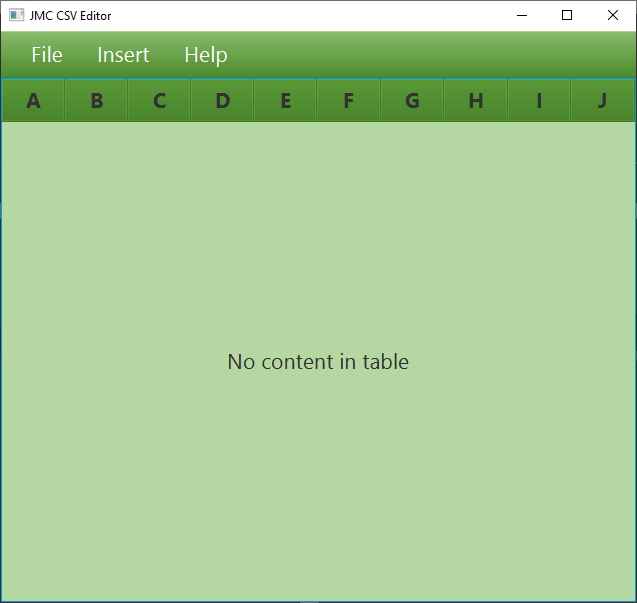


Figure . After initialisation.

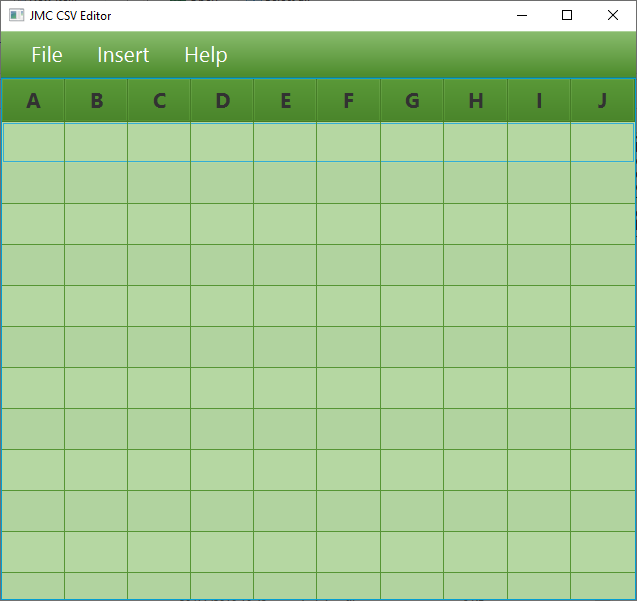


Figure . Empty CSV displayed.

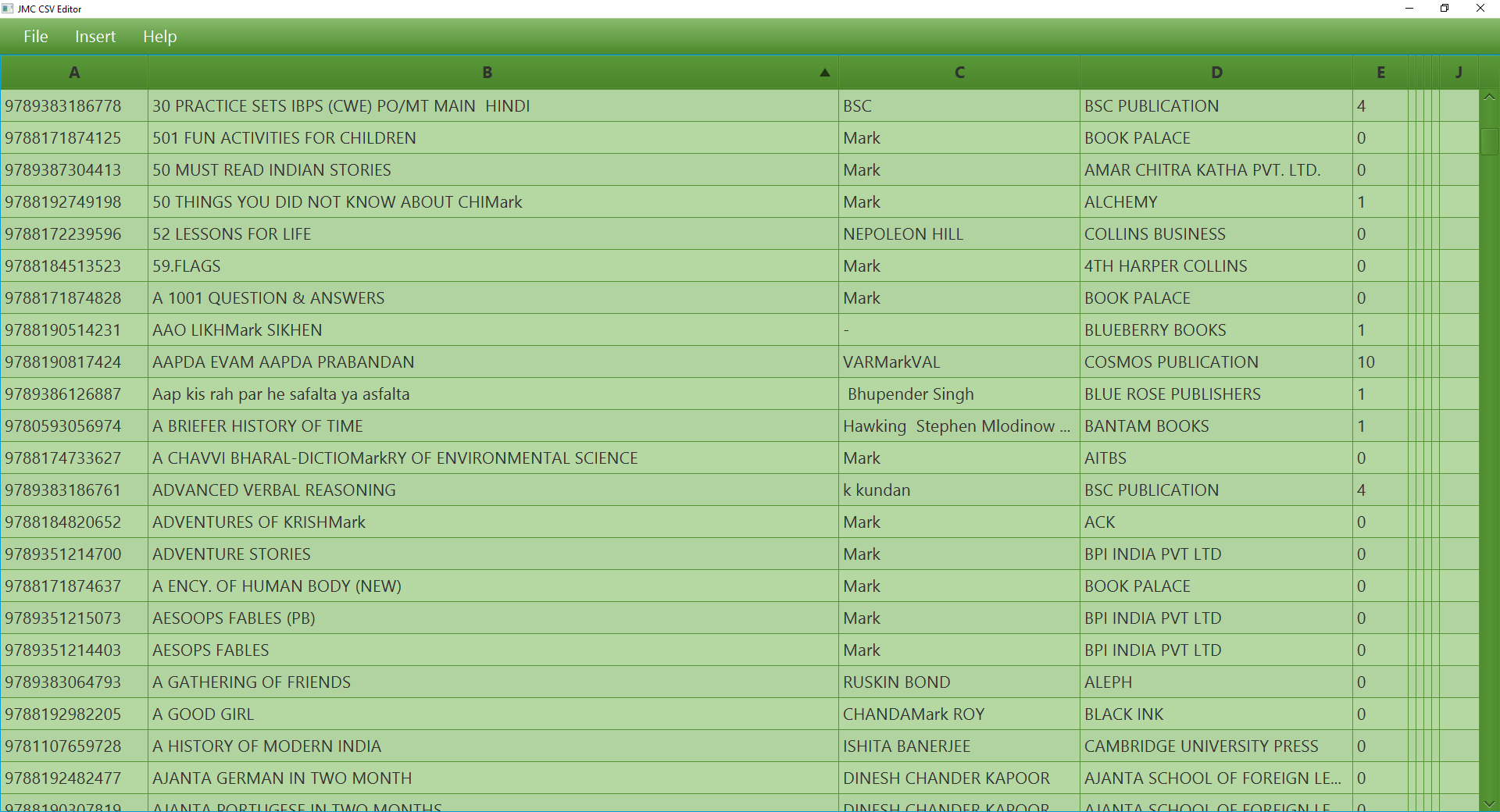


Figure . Fields populated with data.

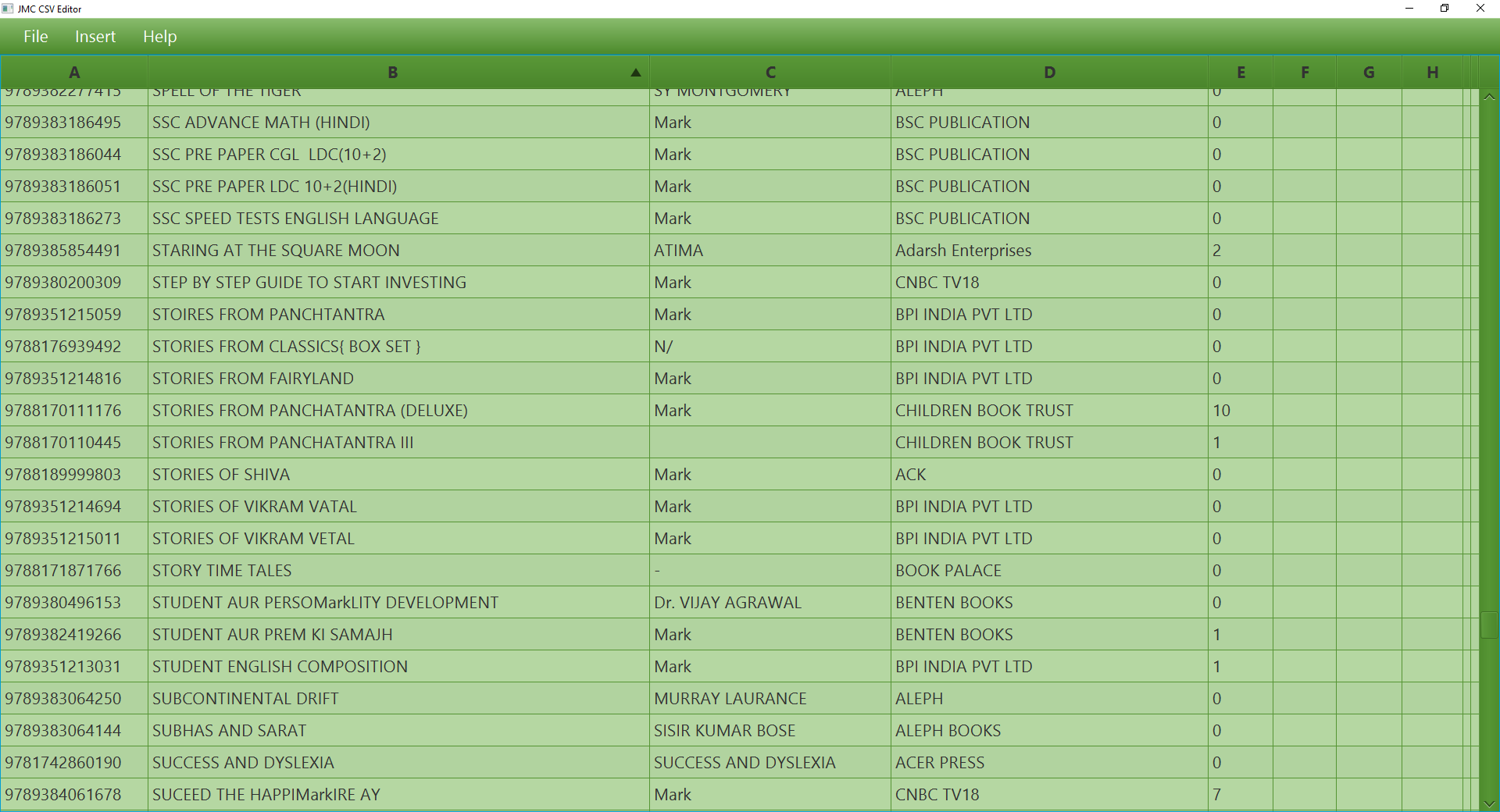


Figure . Performance screenshot.

# 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