# Program Specifications

|  |  |
| --- | --- |
| **Project Name:** | **SimpleLogin** |
| Programmer: | Samuel Batey |
| Supervisor: | Christina Kaiser, Software Development Team Leader |
| Purpose of program: | To add a login and password generation functionality |
| Programming language: | C# .Net |
| Software to be used: | Visual Studio |
| Project brief:  *Simple English description of the program to be created (maximum 50 words)* | Simple login program with the ability to generate passwords |

### Table of Contents

[Program Specifications 1](#_Toc101025024)

[Coding Rules 3](#_Toc101025025)

[Data Library Variables 3](#_Toc101025026)

[Variable Scope 6](#_Toc101025027)

[Program Test Data 8](#_Toc101025028)

[Program Algorithm 9](#_Toc101025029)

[Program Code 10](#_Toc101025030)

[Software Test Report 16](#_Toc101025031)

[Overview 16](#_Toc101025032)

[Changes Required from Feedback 17](#_Toc101025033)

[Test Cases 18](#_Toc101025034)

[Software Evaluation 22](#_Toc101025035)

[Programmer’s Checklist 24](#_Toc101025036)

[Software Final Approval 25](#_Toc101025037)

Coding Rules

|  |  |
| --- | --- |
| **Rule** | **Brief explanation of rule** |
| 1. Give variables appropriate names | Makes the code easier to understand |
| 1. Use comments | Makes the code easier to understand and maintain |
| 1. Use appropriate data types | Avoids errors and using excess resources |
| 1. End lines with ; | In C#, lines of code must be ended with ; |
| 1. Give methods appropriate names | Makes the code easier to understand |

Data Library Variables

|  |  |
| --- | --- |
| **Variable name** | **How variable will be used** |
| userIn | Used in Main to store the user’s input |
| running | Used in Main to keep track of if the program should be running |
| accounts | Used in Mian to store each username and password in the accounts file |
| usernameIn | Used in Login to store the user’s input for their username |
| passwordIn | Used in Login to store the user’s input for their password |
| account | Used in Login to store a single username password pair when looping through accounts |
| loginSuccess | Used in Login to keep track of if the login succeeded |
| pass | Used in ValidatePass to store the password being validated |
| hasSymbol | Used in ValidatePass to keep track of if the password has a symbol in it |
| hasLetter | Used in ValidatePass to keep track of if the password has a letter in it |
| hasNumber | Used in ValidatePass to keep track of if the password has a number in it |
| correctLength | Used in ValidatePass to keep track of if the password is the correct length |
| symbols | Used in ValidatePass to store all the valid symbols |
| symbol | Used in ValidatePass to store a symbol when looping through symbols |
| letters | Used in ValidatePass to store all the valid letters |
| letter | Used in ValidatePass to store a letter when looping through letters |
| numbers | Used in ValidatePass to store all the valid numbers |
| number | Used in ValidatePass to store a number when looping through numbers |
| fileName | Used in Register to store the name of the accounts file |
| accountsFile | Used in Register to store the StreamWriter |
| usrSlct | Used in Register to store the user’s selection |
| usrName | Used in Register to store the user’s input for their username |
| usrPass | Used in Register to store the user’s input for their password |
| validOption | Used in Register to keep track of if the user’s selection was valid |
| validPass | Used in Register to keep track of if the user’s password is valid |
| validChars | Used in Register to store all the valid characters for the password generation |
| r | Used in Register to store an instance of the Random class |
| pass | Used in Register to store the generated password |
| accounts | Used in ViewAcc to store each username and password in the accounts file |
| account | Used in ViewAcc to store a single username password pair when looping through accounts |
| fileName | Used in UpdateAcc to store the name of the accounts file |
| fileLines | Used in UpdateAcc to store the lines of the accounts file |
| accounts | Used in UpdateAcc to store each username and password in the accounts file |
| line | Used in UpdateAcc to store a line when looping through lines |

Variable Scope

|  |  |
| --- | --- |
| **Scope** | **Variable Name** |
| Local | userIn |
| Local | running |
| Local | accounts |
| Local | usernameIn |
| Local | passwordIn |
| Local | account |
| Local | loginSuccess |
| Local | pass |
| Local | hasSymbol |
| Local | hasLetter |
| Local | hasNumber |
| Local | correctLength |
| Local | symbols |
| Local | symbol |
| Local | letters |
| Local | letter |
| Local | numbers |
| Local | number |
| Local | filename |
| Local | accountsFile |
| Local | usrSlct |
| Local | usrName |
| Local | usrPass |
| Local | validOption |
| Local | validPass |
| Local | validChars |
| Local | r |
| Local | pass |
| Local | account |
| Local | accounts |
| Local | line |

Program Test Data

|  |  |
| --- | --- |
| **Username** | **Password** |
| fredsmart1 | 12345678 |
| jrobertson4 | r@=%8(\_W=1 |
| bob101 | 1234598 |
| marcusw | 3#tr@9dw%4 |
| popeyedd | 1989eidjce |
| junkman00 | p3\*(kd8&ld |
| sbj2021 | $d5e(ep2(d |
| robotman | 7777Spy007 |

Program Algorithm

BEGIN Main()

WHILE the program is running

UpdateAcc()

IF accounts file doesn’t exist THEN

end program

ENDIF

DISPLAY menu options

GET user’s INPUT

IF user selected login option THEN

Login()

ELSE IF user selected register option THEN

Register()

ELSE IF user selected view accounts option THEN

ViewAcc()

ELSE IF user selected exit option THEN

end program

ELSE

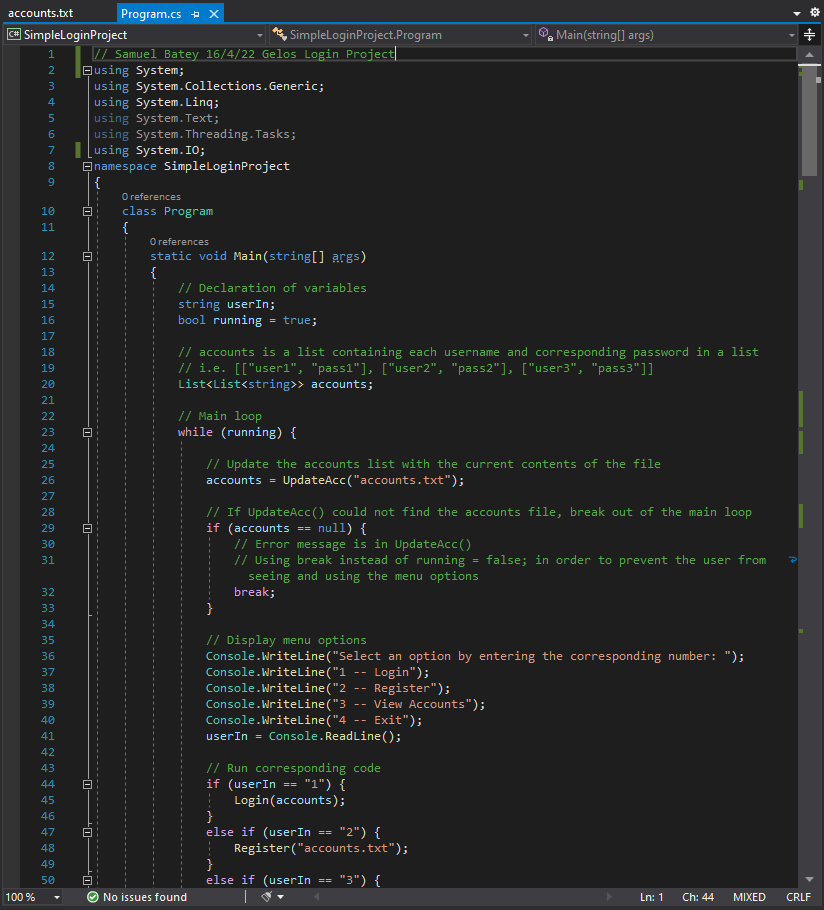
DISPLAY invalid option

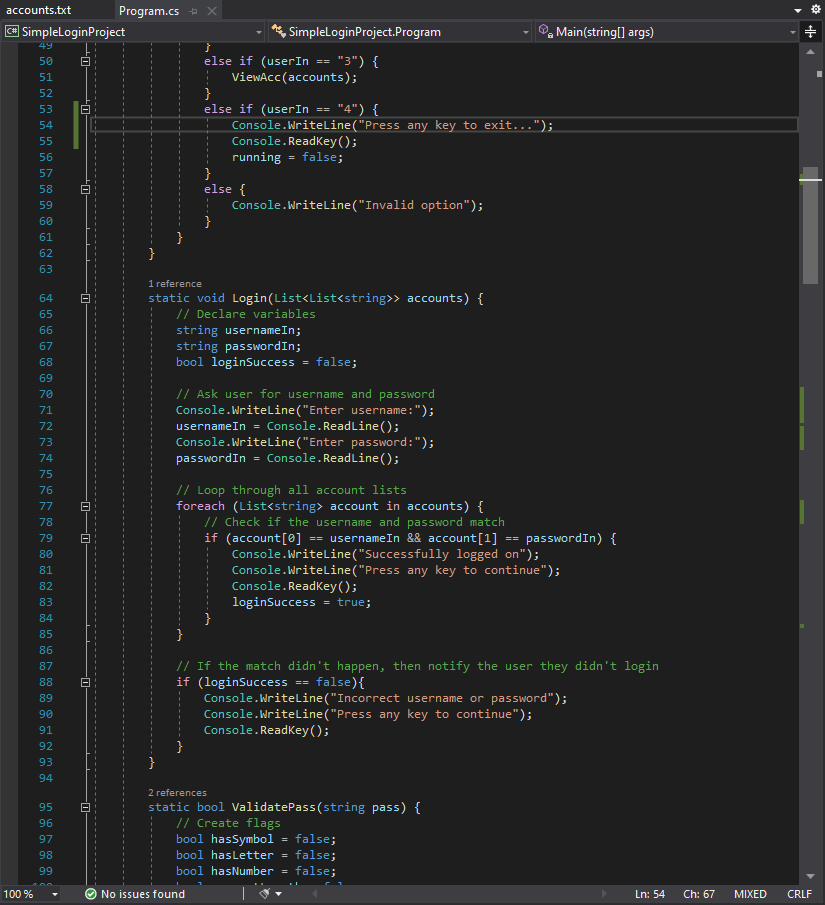
ENDIF

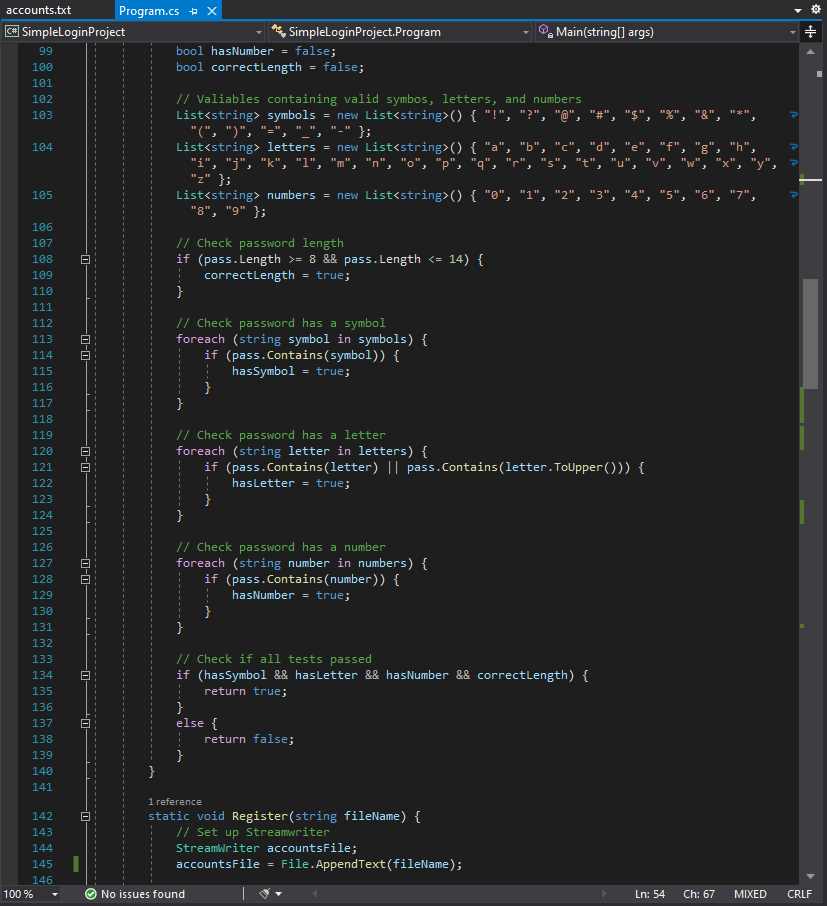
ENDWHILE

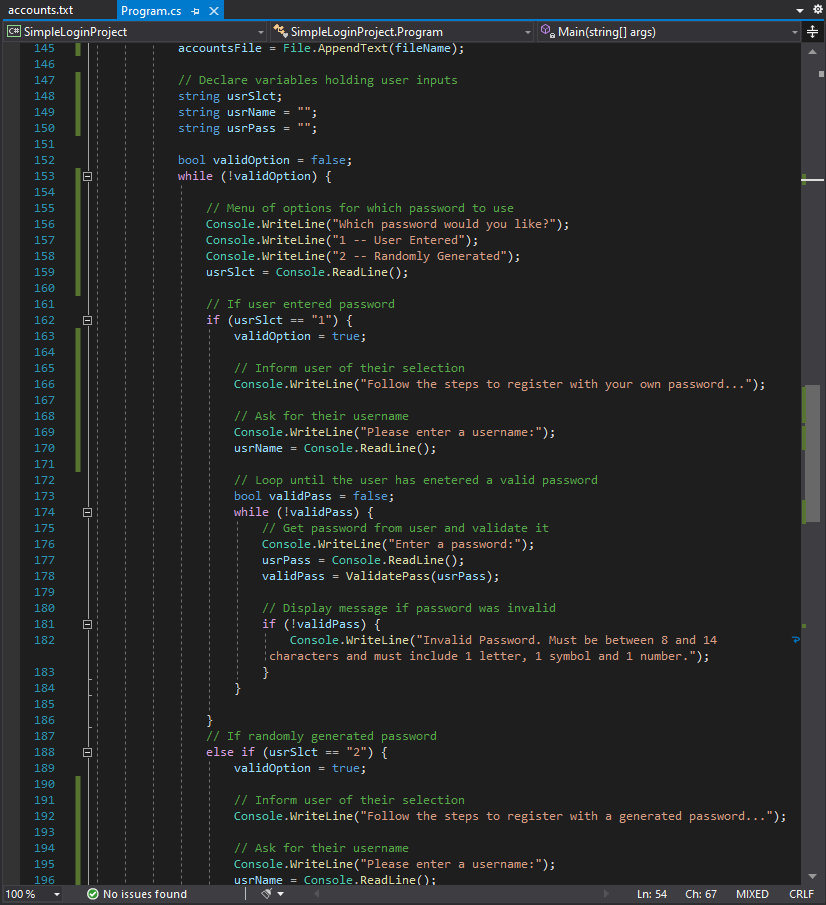
END Main()

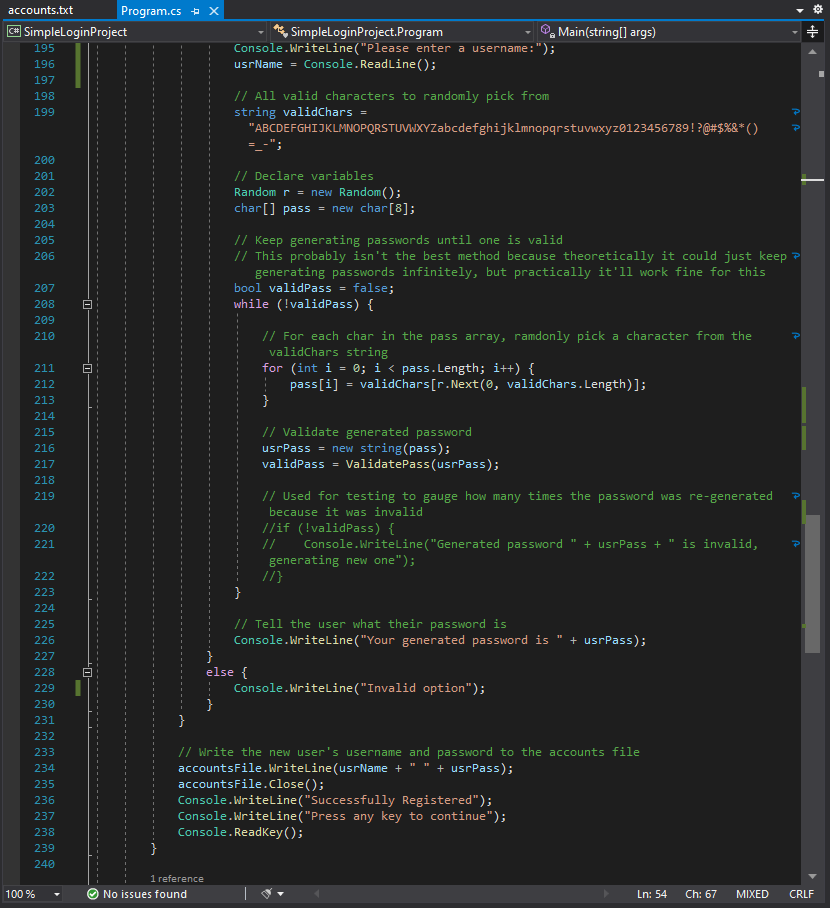
Program Code

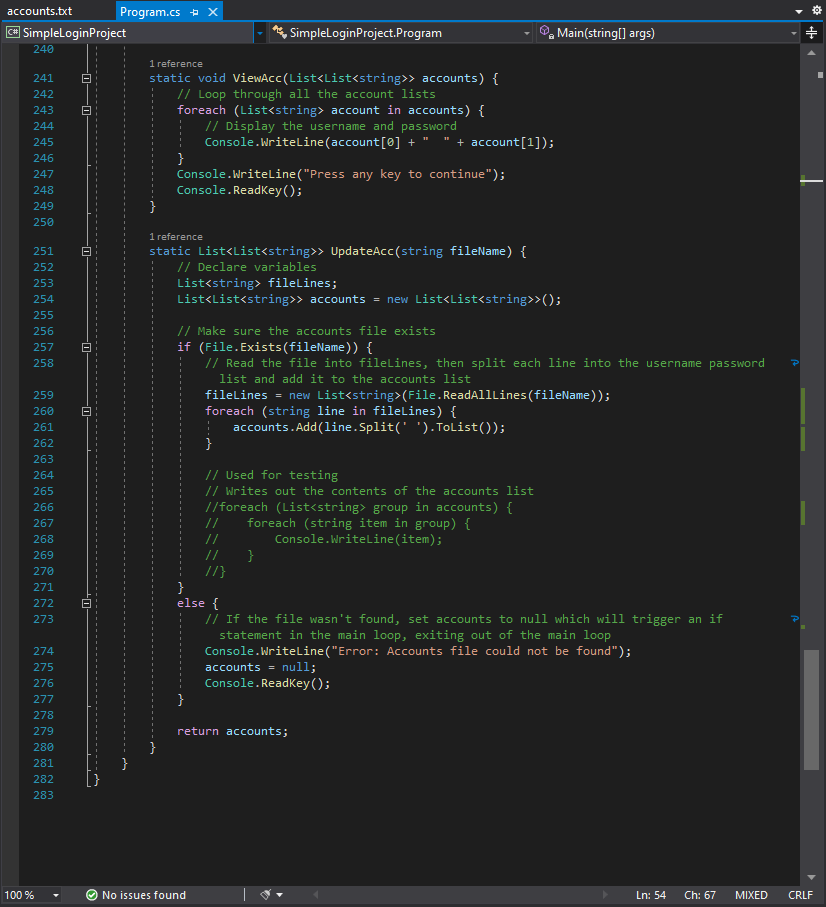












# Software Test Report

Overview

This document is the software test report of the testing phase of the SimpleLoginProject software development project. It contains the results of tests, which were executed during the testing phase.

|  |  |
| --- | --- |
| Name: | Samuel Batey |
| Date: | 15/4/22 |
| Software being tested: | SimpleLoginProject |

Changes Required from Feedback

|  |  |  |
| --- | --- | --- |
| **Object to be changed** | **Changes required** | **Completed** |
| N/A | N/A |  |

Test Cases

| **Test case ID** | **Test case name** | **Summary** | **Expected results** | **Screenshot filename** | **Actual result** |
| --- | --- | --- | --- | --- | --- |
| MenuLoginOption | Test the login option in main menu | User enters “1” | User is prompted to enter username | MenuLoginOption.PNG | User is prompted to enter username |
| MenuRegisterOption | Test the register option in main menu | User enters “2” | User is prompted with register menu | MenuRegisterOption.PNG | User is prompted with register menu |
| MenuViewAccountsOption | Test the view accounts option in main menu | User enters “3” | User is displayed the username and password of the accounts in accounts.txt | MenuViewAccountsOption.PNG | User is displayed the username and password of the accounts in accounts.txt |
| MenuExitOption | Tests the exit option in main menu | User enters “4” | User is told they are exiting the program | MenuExitOption.PNG | User is told they are exiting the program |
| MenuInvalidOption | Tests when an invalid option is selected in the main menu | User enters and invalid option, in this case “6” | User is displayed the invalid option text and is shown the main menu again | MenuInvalidOption.PNG | User is displayed the invalid option text and is shown the main menu again |
| RegisterUserPassOption | Tests the user entered password option in the register menu | User enters “1” | User is notified of the option they selected then prompted for their username | RegisterUserPassOption.PNG | User is notified of the option they selected then prompted for their username |
| RegisterGeneratedPassOption | Tests the generated password option in the register menu | User enters “2” | User is notified of the option they selected then prompted for their username | RegisterGeneratedPassOption.PNG | User is notified of the option they selected then prompted for their username |
| RegisterInvalidOption | Tests when an invalid option is selected in the register menu | User enters and invalid option, in this case “6” | User is displayed the invalid option text and is shown the register menu again | RegisterInvalidOption.PNG | User is displayed the invalid option text and is shown the register menu again |

Software Evaluation

|  |  |
| --- | --- |
| **Software Specification**  (As listed in the software specifications)  (No more than 40 words per response) | **Evaluation**  (How well does your software meet this specification?)  (No more than 20 words per response) |
| Login for users who have previously registered. Username and password to be checked for validity | User can login with a username and password that is check for validity in the accounts.txt file |
| Register for users to create a new account | User can register to create a new username and password that is added to the accounts.txt file |
| New users given the option to enter their own password or generate one | In the register menu, the user can select to enter their own password or generate a random one |
| User given the option to generate a random password and should be allowed to choose the length | User can select to register with a randomly generated password that is the minimum allowed length (8 characters) however, cannot select the length |
| Usernames and passwords should be saved to a text file accounts.txt | Usernames and passwords are saved to the accounts.txt text file |
| Exit the program after a delay of 2 seconds | User can select the option to exit the program and is then prompted to press any key to exit, however there is no time delay |
| View accounts to display user account information from the accounts.txt file | User can select the view accounts option that will display the usernames and passwords in the accounts.txt file |

# Programmer’s Checklist

Check that each of these items have been completed and **have been recorded in the Gelos Software Design Document** in the appropriate place.

|  |  |
| --- | --- |
| **Item to be checked** | **Completed** |
| Gelos Software Design Document fully completed with all details |  |
| Coding rules for chosen language included |  |
| Data library variables defined |  |
| Variable scope defined |  |
| Library of functions defined |  |
| Program test data file created |  |
| Program algorithm designed |  |
| Program code created |  |
| Program code uses good design principles (white space, indents, etc) |  |
| Program code is commented throughout |  |
| Test cases completed |  |
| Screenshots correctly named and saved and zipped |  |
| Software evaluation completed |  |
| Changes required from feedback completed |  |
| Self-Evaluation completed |  |

# Software Final Approval

***Supervisors use only***

|  |  |
| --- | --- |
| **Software Approval** |  |
| Name: |  |
| Role: |  |
| Program meets design specifications: |  |
| Signature: |  |
| Date: |  |