# Program Specifications

|  |  |
| --- | --- |
| **Project Name:** | **SimpleLogin** |
| Programmer: | Ming Chi Hsiao |
| Supervisor: | Jawdat Moussa, Software Development Team Leader |
| Purpose of program: | Create a simple way for user to login with their password and account name |
| Programming language: | python |
| Software to be used: | Visual studio code |
| Project brief:  *Simple English description of the program to be created (maximum 50 words)* | **Login** – for users who have previously registered. Username and password to be checked for validity.  **Register** – create a new user account.  **Passwords** – new users given the option to enter their own password or generate one.  **Generated passwords** – user given the option to choose the password character types – numbers, symbols or letters. The default password length should be applied, but users should also be allowed to choose how many characters.  **Save file**– usernames and passwords should be saved to a text file **accounts.txt.**  **Exit** – delay for 2 seconds before exit.  **View accounts** – to display user account information from the accounts.txt file (assuming that only admin have access to this program). |

### Table of Contents

[Program Specifications 1](#_Toc81985354)

[Coding Rules 3](#_Toc81985355)

[Data Library Variables 3](#_Toc81985356)

[Variable Scope 4](#_Toc81985357)

[Program Test Data 5](#_Toc81985358)

[Program Algorithm 6](#_Toc81985359)

[Program Code 7](#_Toc81985360)

[Software Test Report 8](#_Toc81985361)

[Overview 8](#_Toc81985362)

[Changes Required from Feedback 9](#_Toc81985363)

[Test Cases 10](#_Toc81985364)

[Software Evaluation 12](#_Toc81985365)

[Programmer’s Checklist 13](#_Toc81985366)

[Software Final Approval 14](#_Toc81985367)

[When you have completed your report, select the table (above) and press F9 to update the entire table. Once done, delete this text]

Coding Rules

|  |  |
| --- | --- |
| **Rule** | **Brief explanation of rule** |
| 1. safe | The coding should be able to be used and doe not cause harm or damage to others or company. |
| 1. Secure | The coding should be secure, others without authority should not be able to access the information and document involved. |
| 1. Reliable | The coding should be functional all the time and every time the function should be able to be the same. |
| 1. Testable | The coding should be testable and debugged, to make sure all the designed function are working condition. |
| 1. Maintainable | The code should be able to be maintained, so the coding can always to make sure has the best performance as designed. |

Data Library Variables

|  |  |
| --- | --- |
| **Variable name** | **How variable will be used** |
| **Login:** |  |
| Username | Use to storage the user input login username |
| Password | Use to storage the user input login password |
| Accounts\_username | Use to storage the account\_username which we have in our data. |
| Accounts\_password | Use to storage the account\_password which we have in our data |
| ErrorMessage | Use to storage the error message send to user when the information the user provide is wrong. |
| **Register:** |  |
| Username | Use to storage the user input login username |
| Password | Use to storage the user input login password |
| Accounts\_username | Use to storage the account\_username which we have in our data. |
| Accounts\_password | Use to storage the account\_password which we have in our data |
| ErrorMessage | Use to storage the error message send to user when the information the user provide is wrong. |
| **Account View:** |  |
| Username | Use to storage the user input login username |
| Password | Use to storage the user input login password |
| Accounts\_username | Use to storage the account\_username which we have in our data. |
| Accounts\_password | Use to storage the account\_password which we have in our data |
| ErrorMessage | Use to storage the error message send to user when the information the user operate is wrong. |

Variable Scope

|  |  |
| --- | --- |
| **Scope** | **Variable Name** |
| Local | Login variable, all local scopes to the function login  Username, password, Accounts\_username, Accounts\_password, Error message |
| Local | Register variable, all local scopes to the function register  Username, password, Accounts\_username, Accounts\_password, Error message |
| Local | AccountsView variable, all local scopes to the function accountsView  Username, password, Accounts\_username, Accounts\_password, Error message |
| Choose an item. |  |
| Choose an item. |  |
| Choose an item. |  |
| Choose an item. |  |
| Choose an item. |  |
| Choose an item. |  |
| Choose an item. |  |
| Choose an item. |  |
| Choose an item. |  |
| Choose an item. |  |
| Choose an item. |  |
| Choose an item. |  |
| Choose an item. |  |

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

[paste a copy of the pseudocode or flowchart here and delete this text]

Program Code

[paste a copy of program code or blueprint here and delete this text]

# Software Test Report

Overview

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

|  |  |
| --- | --- |
| Name: |  |
| Date: |  |
| Software being tested: |  |

Changes Required from Feedback

|  |  |  |
| --- | --- | --- |
| **Object to be changed** | **Changes required** | **Completed** |
| *Sample*  *Spelling error in main menu “optoins” should be “options”* | *Correct the spelling error* |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Test Cases

| **Test case ID** | **Test case name** | **Summary** | **Expected results** | **Screenshot filename** | **Actual result** |
| --- | --- | --- | --- | --- | --- |
| *Sample:*  *Var\_Menu1* | *Test the login options in main menu* | *Data = 1*  *Test that the 1 on main menu works* | *Login screen is displayed* | *Var\_Menu1.jpg* | *Login screen is displayed* |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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) |
| ***Sample***  *Menu displays when program starts* | *Menu displays when program started with no spelling or grammatical errors* |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 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: |  |