

User-Documentation (Admin)

PSEC CA1 Assignment Documentation

# Welcome!

* Welcome to the administrative side of the Quiz Application! If you’re here, you are probably one of the new admins of the application who is still trying to figure out to utilize your new position as admin in the application!
* Fear not, for I will guide you through the vast features that will make your experience as a quiz admin very fun!
* Good luck and have fun!

# Scripts for Admin Users

Input-Output Files:

* question\_pool.txt (Stores questions, answers, marks, and correct answers)
* quiz\_results.txt (Stores results, questions attempted, answers and correct answers)
* quiz\_settings.txt (Stores quiz settings and configurations)
* userid\_pswd.txt (Stores usernames, encrypted passwords, and user privileges)

Main Script for Admin:

* admin.py

External Files

* quiz\_report.csv (Excel CSV to generate quiz results for every attempt recorded)

IMPORTANT NOTE: Open the entire 1B05-ErvinLee folder in VS Code for the file locations to work

# Input-Output Files

## question\_pool.txt

This is a data file where all questions with the properties (Marks, Choices, Correct Answer) are stored for each question. They are all stored in key-value pairs in dictionaries for easy access to the values via the keys.

Data Fields (Example)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Keys | Question | A | B | C | D | Marks | Correct Answer |
| Values | What is 1+1? | 1 | 2 | 3 | 4 | 2 | B |
| Data Types | String | String | String | String | String | String | String |
| Dictionary | | | | | | | |

## userid\_pswd.txt

This data file stores all user accounts that are register in the quiz application. Each user account stored consists of role (Default User or Administrator), username and password. For security reasons, the password stored is encrypted using a simple Caesar cipher algorithm.

Data Fields (Example)

|  |  |  |
| --- | --- | --- |
| Role | Username | Password |
| U | Ervin | oljkwSdlqwlqj5$C |
| String | String | string |

## quiz\_settings.txt

This data file stores all the quiz settings and configurations which are set by the admins. Examples include the option to randomize the order of questions, set number of attempts a user can take for the quiz and set a time limit for the quiz.

Data Fields (Example)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Setting | RandomizeQuestion | TimeLimit | Duration | Questions Tested | Attempts |
| Values | On | On | 5 | 5 | 3 |
| Data Types | String | String | Integer | Integer | Integer |

## quiz\_results.txt

This data file stores all the information about a user’s quiz attempt. Each line is a dictionary with keys corresponding to the information sent by the user during their attempt in a quiz. This data file will be read to export the results to a CSV file in Excel.

Data Fields (Example)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UserID | Questions | Answers | Correct Answers | Score | Dated Attempted | Module | Topic |
| Ervin | [“What is 1+1”, “What is the capital city of USA?”, …] | [“A”, “C”, “D”…] | [“A”, “B”, “C”, …] | 90.0 | 1/1/2021 | “Geography” | “America” |
| String | List | List | List | Float | String | String | String |

## quiz\_report.csv

This file will be exported from the admin.py script by reading the data values in the quiz\_results.txt file. It will represent all the attempts done by every registered user with the past randomized questions, answers, correct answers, dates, marks, and date in an Excel sheet.

Sample Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UserID | Date Attempted | Number of Questions | Score (%) | Question 1 | Entered Answer for Question 1 | Correct Answer for Question 1 |
| ervin2 | 23/11/2021 | 5 | 83.333 | Which of these cities is not a national capital? | D | D |
| String | String | Integer | Float | String | String | String |

## quizzes.txt

This file stores all the quizzes created by the administrators in the form of a dictionary. Each quiz dictionary stores the list of modules tested; list of topics selected, and the list of question dictionaries based on the tested module and topic

Sample Table

|  |  |  |  |
| --- | --- | --- | --- |
| QuizName | Module | Topics | Questions |
| “Math 101” | “Mathematics” | [(3, “Addition”), (2, “Subtraction”)] | [{Q1: “What is 1+1? … } ….] |
| String | String | List of tuples | List of dictionaries |

## modules.txt

This is where all modules and topics created by the administrators are stored. Each line in the file is represented by a dictionary, with module name and topic names acting as dictionary keys.

Sample Table

|  |  |
| --- | --- |
| Module | TopicList |
| Math | [“Addition”, “Subtraction”, “Multiplication”] |
| String | List |

# Instructions to run administrator script

## Script Overview

* Login Page
* Manage Question Pool
  + Add Question
  + Remove Question
  + Edit Question
  + Quiz Settings
* Manage User Accounts
  + Add User
  + Remove User
  + Edit User
  + List Users
* View Quiz Results
  + View Past Attempts
  + View Quiz Performance
  + Export CSV File
  + Generate Management Report

## External Modules to Import

* For the management report in quiz results, the external module PrettyTables is used to generate a visually appealing string-based table.
* For the table generation to work, the module needs to be installed.
* This can be done using the following command:

python -m pip install -U prettytable

## User Accounts

* Users are not allowed to login to the administration panel. They will be denied from logging if they attempt to do so.
* Only admins can create new administration privileges, while users are only allowed to create accounts with default user privileges.
* Only admins have the power to edit account details such as username and passwords and remove user accounts

## Server

1. The admin program connects to the server as a client via socket programming to send data packets for the server to read, update, write data files in its database.
2. The server must be running before running any client program as the server listens for client connections
3. Open two terminals on VSCode
4. On one terminal, type in the following command to run the server:

python ./Code/server.py

1. Similarly on another terminal, type in the following command to run the admin program:

python ./Code/admin.py

1. The server should be listening for any connections:

Text

Description automatically generated

1. Only the server can access all the data files. If the admin program wishes to modify any data file, the request needs to be sent to the server for the requested action to occur. Here is an example of the client requesting the server to update changes to the module file.

Text

Description automatically generated

## Running the Admin Script

On admin.py

1. The login screen will be shown below. You are given two options:
   * Login to script
   * Quit and stop the script

Note: Inputs are validated (Range and Type Check)

A screenshot of a video game

Description automatically generated with medium confidence

1. There is a default admin account in the userid\_pswd.txt which you will use to login into the admin script. (WARNING: CASE SENSITIVE)
   * Username: admin1
   * Password: 1Qwer#@!

A picture containing text, clock

Description automatically generated

1. Once logged in, you will have access to your administrative settings of the quiz application

## Navigating the Script

1. Quiz Settings
   1. In this section, you have the power to make your quiz!
   2. You can add, edit, and delete questions in the quiz
   3. You can make changes to the marks, choices, question description and the correct answer for each question

Text

Description automatically generated with low confidence

* Adding a question:

Text

Description automatically generated

* Removing a question:

Text

Description automatically generated

* Editing a question:

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

* Quiz settings:

Text

Description automatically generated

1. Module Settings
   1. In this section, you can create, edit, and remove modules that you wish to test in your quizzes in this application!
   2. Within each module, you can also add, edit, and delete topics!
   3. You can add as many topics and modules as you like, but each module should have at least two topics!

* Adding a module

Text

Description automatically generated

* Removing a module

Text

Description automatically generated

* Editing a module and topic:

Text

Description automatically generated

Text

Description automatically generated

* Editing a module name

Text

Description automatically generated

* Adding a new topic

Text

Description automatically generated

* Removing a module

Text

Description automatically generated

* Editing a topic

Text

Description automatically generated

1. Manage User Accounts
   1. In this section, you have the power to create, remove and edit users in the application!
   2. Only admins can create more admin accounts and only you have the power to change the passwords of any existing user
   3. You can view a list of all users register to the quiz application

Graphical user interface, text, application

Description automatically generated

* List Users

Text

Description automatically generated

* Add a new user:

Text

Description automatically generated

* Remove a User

Text

Description automatically generated

* Edit User Details

Text

Description automatically generated

Text

Description automatically generated

1. Viewing Results
   1. You can see behind the scenes and check out the general performance of the users that attempted the quiz
   2. Check out the overall statistics of the quiz (e.g., mean, median, mode, min, max, standard deviation)
   3. You can export your results to CSV to view them in a table format on Excel
   4. Every attempt with all the past questions, answers and correct answers are recorded, even the date! You can view the list of past attempts and view each one of them in detail.
   5. You can generate a management report that allows admins to see the detailed performance of users for each quiz e.g., how many users got a question in the quiz correct.
   6. You can clear all attempts to save data as you please by entering option 4. USE WITH CAUTION!!!

Text

Description automatically generated

* View Past Attempts

Text

Description automatically generated

Text

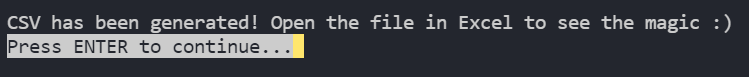
Description automatically generated

* Generate a Report

Text

Description automatically generated

* Export Results to CSV
  + Go to your file explorer and open the CSV file in the PSEC Code folder in Excel to check out the results projected in table format
  + Note: Once the table is open, the table will not be formatted. You are required to auto-format the columns for a better and neater table.



A screenshot of a computer

Description automatically generated

Graphical user interface, text, application

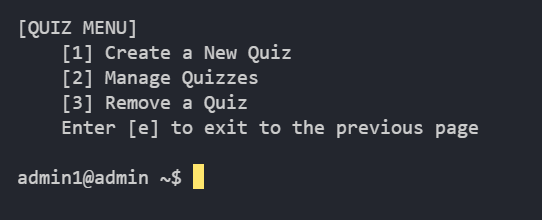
Description automatically generated

* Generate a management report

A picture containing table

Description automatically generated

1. Manage Quizzes
   1. This is where admins can create various quizzes testing different modules and different topics to put to the test a variety of knowledge for our users.
   2. Create, edit, and remove quizzes as many as you like!



* Creating a new quiz (Specify the number of questions you wish to test for each topic, 0 to test all questions)

Shape

Description automatically generated with medium confidence

Text

Description automatically generated

* Manage quizzes

Graphical user interface, text, application

Description automatically generatedText

Description automatically generated

* Editing module tested for quiz

A picture containing graphical user interface

Description automatically generated Graphical user interface, text

Description automatically generated Text

Description automatically generated

Text

Description automatically generated

That is all the features! Have fun exploring and making fun and educational quizzes!

**Post-Implementation Support:** If there are any issues or queries, feel free to contact Ervin at [ervinleequanjun@outlook.com](mailto:ervinleequanjun@outlook.com).

END OF USER-ADMIN DOCUMENTATION



User-Documentation (User)

PSEC CA1 Assignment Documentation

# Welcome user!

* Thank you for using our application! If you have chanced upon our application for the very first time, then you have come to the right place!
* You will be taught about the features our quiz application has to offer and how to navigate around our app with ease.
* Hopefully by the time you have finished reading the instructions, you will be ready for to journey into the great test of knowledge, trivia, and of course, lots of fun!

# Getting Started

* As a normal user, you only need to run the user.py script (our server program will be running in the background beforehand!)
* Before we start, make sure you have Visual Studio Code installed! If you don’t have it, do follow this [useful guide here](https://code.visualstudio.com/docs/setup/windows) before we get started! (If you’re using Mac, follow [this one](https://code.visualstudio.com/docs/setup/mac) instead!)
* You should download the application zip file and extract it before continuing the following steps

1. Once you have downloaded Visual Studio Code, launch the application.
2. You should be greeted by this launch page when you start the application.

A screenshot of a computer

Description automatically generated

1. Click on “Open Folder,,,” and navigate to the quiz application extracted folder. Double click on it to select and open it on VS Code.
2. Once the folder is opened, drop down on the “Code” folder and select the user.py file.
3. To run the quiz application, create a new terminal (Go to Terminal > New Terminal) or use the shortcut (Ctrl + Shift + `)
4. On the terminal, simply type the command:

python ./Code/user.py

# Running The Script

1. With the script running, you will be directed to the main login page with three options: (We will be going through all the respective options below, feel free to skip to the main quiz section if it does not apply to you)
   1. Login (if you have an existing account)
   2. Register (register a new account if you don’t have an existing account)
   3. Forgot Password (If you have an existing account in the application but you forgot and wish to change your password)

Icon

Description automatically generated with medium confidence

* Logging In (Option 1)
  + If you have an account, enter your username and password. The inputs will be validated to see if your account exists and if you entered the correct credentials. (WARNING: CASE SENSITIVE)

Text

Description automatically generated

* Registering a New Account (Option 2)
  + If you’re new, you can freely register a new account that you can log into our applications at any time.
  + Just enter a new unique username that does not exist in our database and a secure password (the program will show you the requirements) and you’re good to go!

Text

Description automatically generated

* Changing Your Password
  + Forgot your password? Fear not, we will help you recover your account by simply entering your username
  + You will need to input your username so that we can change your password for that account.
  + There you have it! You are ready to go without worrying about being locked out because of a forgotten password!

Text

Description automatically generated

# Taking the Quiz

Congratulations! You have successfully logged into the quiz application. Now it’s time for some learning fun, putting your knowledge and trivia to the test! You will be taking a quiz with a limited number of attempts, random questions with a time limit, all set up by our administrators working to provide you the best quiz experience!]

Here is what you will see when you have logged in:

Text

Description automatically generated

On the server side, your login credentials will be received from the client user program to validate your details based on the userid\_pswd data file and will send back the verification status. The server will then extract the quiz questions the admins have set for you to take!

Text

Description automatically generated

Here is a sample question you can expect to see from the quiz:

Text

Description automatically generated

When the time limit runs out…

Text

Description automatically generated

When attempts run out…

Text

Description automatically generated

* You are required to choose from 4 options (a, b, c, d). Inputs will be validated so don’t make any typos or mistakes 😊
* Case does not matter!
* Your time limit will be displayed and will be updated when you skip, go back, or answer a question
* You can freely skip (press Q) or move back (press P) between questions
* Once you have answer or skipped the last question of the quiz, a summary page of your questions and responses will be shown. If you have any unanswered questions, they will also be shown
* You can jump to any question as and when you like and change your answers
* Once you are done or time is up, you will be shown your results to tell you how you did.
* If you have more than one attempt, you have the option to use another attempt, or you could stop the quiz and submit your results. (The quiz result will submit the best of the attempts you take)
* Your results will be saved in a database where admins can view your attempt in detail and track performance

Summary:

Text

Description automatically generated

Quiz Results:

Text

Description automatically generated

* Once you have done the quiz, you will be redirected back to the main login page and the program will send your results over to the server to store for the admins to review!

That is all for the quiz. Hopefully you are now able to freely navigate around the quiz application with ease and have lots of fun learning new things from the quiz!

Post Implementation Support:

* If you have issues, queries or require assistance, feel free to contact Ervin Lee via email at [ervinleequanjun@outlook.com](mailto:ervinleequanjun@outlook.com)

Happy Learning! 😊

# Settings Icon | Android Lollipop Iconset | dtafalonso

# Documentation (System)

PSEC CA1 Assignment Documentation

# Application Overview

Application Structure

* Shows all files and what they do

’

Basic Application Flow

* Brief flowchart describing the flow of the quiz application
* More detailed flowchart will be shown later as part of program design documentation

# Features and Status of Completion

Basic Features

|  |  |  |
| --- | --- | --- |
| Feature | Status Completed (%) | Remarks |
| Login System | 100 | * Able to validate username and password by reading from userid text file * Validate that user without admin privileges cannot access admin script even if they attempt to do so * Registering user was simple (add username, password, write to useid file) * Able to send to server to verify |
| Compliance of Password | 90 | * Password requirements are well enforced with re-entering for verification * Requirements was stricter than required in brief to reinforce security * However, password function has security flaws e.g., no way to verify authenticity of user changing password. * Regex saved lots of lines instead of string slicing to validate the password. |
| Validation of Input Values | 90 | * Feel that input validation is well implemented (range checks, type check, length check etc) * Could have made code shorter and more efficient if simple validation function written, but input requirements vary too much, difficult to implement function for every input |
| Color Coding of Output texts | 95 | * Made program look pleasing but was very tedious to implement as strings needed to be appended individually * Could have more efficient functions to color code text without adding manually with print function |
| Overall text UI | 99 | * Was very organized with the use of os.system(“cls”), got rid of stacking outputs * Only worked with navigating from section to section, input validations still stacking but overall, not very important |
| Quiz Question Management | 99 | * Add Question function was well implemented and can write to question pool without any issues (although wished could be more dynamic as only supports 4 options) * Edit Question also manages to edit all aspects of a quiz question (e.g., marks, correct answer), same as Add Question: could be more dynamic. |
| User Management | 100 | * Able to perform all possible actions an admin can do e.g., list users register, add, remove, and edit usernames and passwords. |
| View Results | 99 | * Feature that views list all attempts and can review each one in detail using a saved dictionary with keys to access * Can review basic statistics that track overall performance of all quiz attempts (e.g., mean, median, mode, standard deviation) |
| Quiz Settings | 100 | * Using clearing function, was very neat seeing the settings toggle when selected * Very user friendly with detailed validations for every setting made |
| Quiz Navigation | 100 | * Done lots of validation for choices and able to check if question answered or unanswered * Easy to navigate to previous or next question * Summary page well printed and organized with inputted answers * Able to jump to specific question and navigate from there * Repeats quiz when there are multiple attempts |
| Result Computation | 100 | * Able to tally marks and display for user * Able to take the best attempt out of multiple ones * Prints the required messages based on how much marks (by percentage) * Could have a grading system and implemented grade statistics for more detailed report. |
| Password Encryption | 100 | * Used a simple Caesar cipher to shift ASCII values of each password character up by 3 and stored in userid file.’ * Decrypts when reading password for verification |
| Modules and Topics | 100 | * Modules and topics are easily changeable and dictionaries are very dynamic. |
| Socket Programming | 100 | * Server can receive “calls” from clients to perform a function that deals with reading or writing data files instead of directly modifying the file in the admin/user program. |

Advanced Features

|  |  |  |
| --- | --- | --- |
| Feature | Status Completed (%) | Remarks |
| Export to CSV Function | 90 | * Export to CSV well organized, could have added more details such as actual questions and marks. |
| Time Limit | 99 | * Done using time.time(), when time is up, the user can enter last question before finishing quiz * Originally wanted an immediate time stop for quiz but found it difficult to implement without killing code |
| Multiple Attempts | 90 | * Able to specify number of attempts |
| Management Report in PrettyTables | 100 | * PrettyTables report well projected |
| Multithreading for Socket programming | 100 | * Successfully implement client threads on the server and server receives calls from clients very well and can handle multiple client programs at the same time |

# Program Design (Flowchart)

View the flowchart for the admin and user scripts separately in jpeg format in the Documentation Folder.

# Input Validations

## Admin.py Script

Title Screen (Type and Range Check)

A screenshot of a computer

Description automatically generated with low confidence

Text

Description automatically generated with medium confidence

Text

Description automatically generated

Logging In (Invalid username or password) – Clears and prints invalid output message

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Admin Menu (Type and Range Check)

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Quiz Settings Menu

Text

Description automatically generated

Text

Description automatically generated

Adding Question Validations (Range Check for marks and correct answer selection)

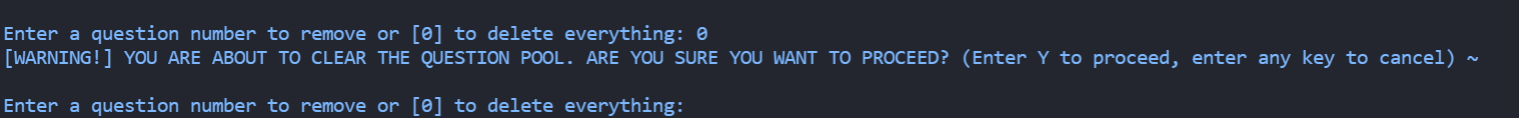
Text

Description automatically generated

Text

Description automatically generated

Removing Question SelectionText

Description automatically generated

Select Question to Edit (Range and Type Check)

Text

Description automatically generated

Validation for editing correct answer and marks like validation for adding question

Clears and no changes made to question if edit selections are invalid

Text

Description automatically generated

Quiz Configurations Validation for Time Limit and Number of Questions to test

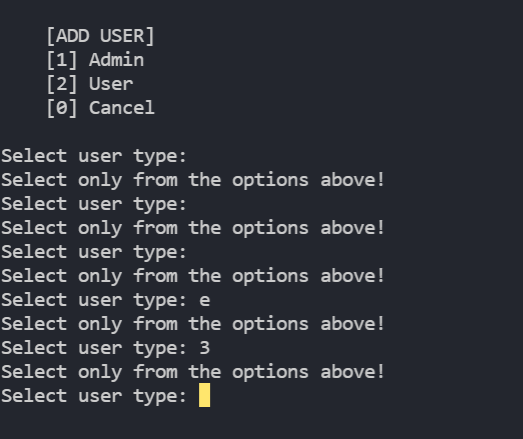
Text

Description automatically generated

Text

Description automatically generated

Adding User Validation (Choose user privileges to give)



Username Validation

Text

Description automatically generated

Password Validation

Text

Description automatically generated

Remove User Validation (Index Check and y/n Validation)

Text

Description automatically generated

Edit User Validation

Text

Description automatically generated

Text

Description automatically generated

View Past Attempts (Range and Type Check)

Text

Description automatically generated

Clearing Attempts Validation

Text

Description automatically generated

## User.py Script

Login Page Validation

Text

Description automatically generated

Text

Description automatically generated

Change Password Validation

Text

Description automatically generated’

Login Validation

Text

Description automatically generated

Quiz Option Validation

Text

Description automatically generated

Text

Description automatically generated

When time is up:

Text

Description automatically generated

Summary Page Validation

Text

Description automatically generated

Results Page (Continue Prompt Validation)

Text

Description automatically generated

END OF SYSTEM DOCUMENTATION