

PSEC Documentation

Semester & Academic Year: **1B / 2021 - 22**

Module Code: **ST2421**

Module Name: **Programming in Security**

Assessment: **CA1 Individual Assignment**

Class: **DISM/FT/1B/05**

Scripts: **user.py, admin.py, DBcom.py**

Files: **jsonPython/db/...**

Student (Full Name): **AAI XUN EN**

Admin No: p2104089

Lecturer (Full Name): **Mr Chee Siong Quek**

1. Introduction

1.1 Program goal:

Design a Quiz Application for BetterTutors Pte Ltd to administer an online quiz for their students.

The application displays the questions one by one. It allows the user to enter his/her answers and submit the quiz. The application will then display the results of the quiz.

Admin will have the power to change settings in the question, quiz and user data.

1.2 Index:

Files: jsonPython/db, results.csv

Script: user.py, admin.py, DBcom.py

1.2.1 user.py:

Given to the user to register or login to take the Quiz function as well as viewing past results.

1.2.2 admin.py:

Used by the admin to oversee the quiz settings, question creation and modification as well as user data.

1.2.3 DBcom.py:

Used to manage the creation, reading, updating and deleting of data in the File system database(jsonPython/).

1.2.4 results.csv:

Used to record past user results. Stores the questions, user answer, correct answer, time taken to complete quiz, Score and date of the user attempt.

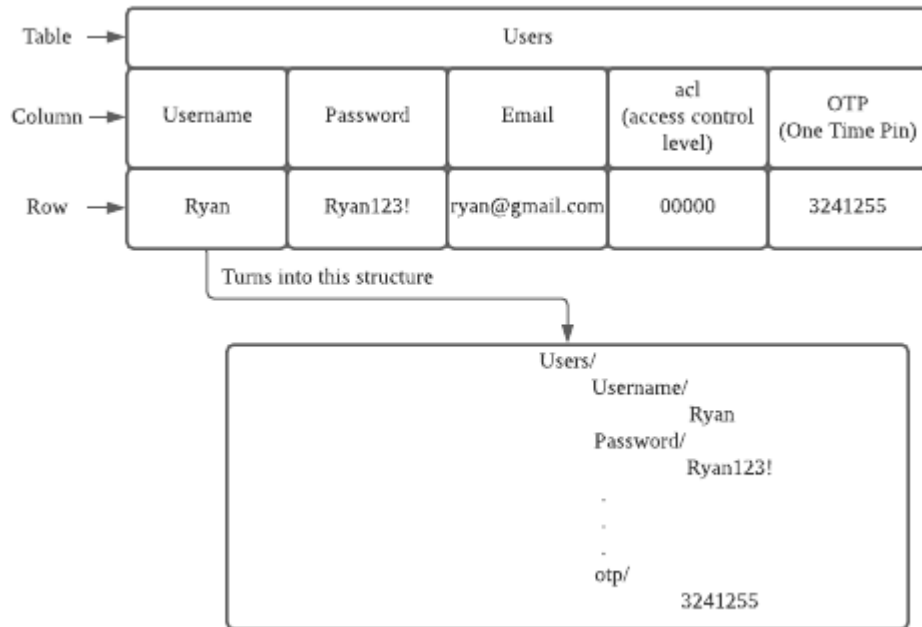
1.2.5 jsonPython/db:

Used to store data of all users as well as questions.

2. Database commands script

2.1 Information:

This application uses a file management system to store the data in filenames in rows and columns.



The database commands include:

1. Create
2. Find
3. Update
4. DeleteCol
5. DeleteUser

2.2 Usage:

Note: Not necessary for running admin and user scripts with no changes to either file.

Create - Creates a new table with a column and row data returns a unique row id

Find - Finds matching data when provided a rowid in the table/column

Update - Update data when provided row id in table/column returns True/False

DeleteCol - Deletes column when provided row id in table/column returns True/False

DeleteUser - Deletes data when provided row id in table/column returns True/False

2.3 Advantages of this System:

- Easily scalable
- Easy to use
- Provides high concurrency and low latency when reading data, only limited by the file system
- .
- .
- .

2.4 EXAMPLE OF SYNTAXES:

2.4.1 Create:

```
def create(tableName, colName, colType, localrowid, data)

DBcom.UserDB.create('users', 'password', 's', localrowid, password)
```

- Creates a password according to the local rowid into columns

2.4.2 Find:

```
def find(tableName, colName, searchPart, searchMethod, returnType,
data)

DBcom.UserDB.find('users', 'password', 'id','', 'raw', localrowid)
```

- Finds the data according to the local rowid provided and returns it to the user.

2.4.3 Update:

```
def update(tableName, colName, colType, localrowid, data):

DBcom.UserDB.update('users', 'username', 's', localrowid, value)
```

- Updates the new data input by the user based on the local rowid.

2.4.4 DeleteCol:

```
def delete(tableName, colName):

DBcom.UserDB.delete('questions', 'questions' )
```

- Deletes the entire column provided, containing all the rows of data.

2.4.5 DeleteUser:

```
def deleteUser(tableName, colName, localrowid):  
  
DBcom.UserDB.deleteUser('users', 'password', localrowid)
```

- Deletes specific data according to the local rowid provided.

// End of Database commands

3. User script

3.1 How to run the user script:

Both the user and admin script can be run within the Main.py script. Instructions in the Main.py script will specify which lines to uncomment to run the appropriate scripts.

Or

They can press f5 on their keyboard in the user script to run the user menu.

3.2 Main functions of the User script:

There are three main purposes of the User script. Firstly, the **creation of the User account** and Secondly, the **taking of the Quiz**, and Lastly, the **viewing of the logged-in user's results**

3.2.1 Creation of the User account:

This section includes the **Registering of User, Logging in for the application** and **password recovery**.


```

+=====+
              Login Menu
+=====+

<ENTER> to back
Please enter your username: craz
Please enter your password: craz
+=====+
Login successful craz/['1645536']
+=====+

```

- Figure 3: A successful login screen

```

+=====+
              Login Menu
+=====+

<ENTER> to back
Please enter your username: craz
Please enter your password: Craz
a. Incorrect username or password
+=====+
              Login Menu
+=====+

<ENTER> to back
Please enter your username: █

```

- Figure 4: A failed login screen

The Forget password function will make use of the **Find** function from the database functions to find a matching **email** and simulate an email sent to the user saying the password has been sent to the user's email address. The function prompts the user to key in an email, if the email matches an email in the database, the function will simulate an email to the user's email address.

```

+=====+
We have sent the password Y3Jheg== to your Email craz@gmail.com
+=====+

```

- Figure 5: Simulated email password recovery

3.2.2 Taking of the quiz:

Note: A 'Question' consists of a question, options and the correct answer.

The take quiz function will acquire the question pool from questions saved in the file system database using the **Find** function. The

function will display the questions as well as the options for the question, prompting the user for an answer. After the user answers a question, the user will be given a choice to either go to the next question, go back to the previous question or exit the quiz.

```
+=====+
                        Take Quiz
+=====+

QuestionID: 1/5
Question:
9
['9', '8', '7', '7']
a) 9
b) 8
c) 7
d) 7
+=====+
What is the correct Answer?:
[a,b,c,d]
> 
```

- Figure 6: A example question and options

```
QuestionID: 1/5
Question:
9
a) 9
b) 8
c) 7
d) 7
+=====+
What is the correct Answer?:
[a,b,c,d]
> a
Answer saved.
1
+=====+
[p]revious, [n]ext, [e]xit.[p/n/e]
> 
```

- Figure 7: options to navigate quiz

The answers that the user chooses will be forwarded to the check answer function. The function will use the **Find** function in the database commands to find the correct answer set by the admin and compare it against the user answer to check if it is correct. The results will be printed into a CSV file for recording.


```

+=====+
                Checking Answer...
+=====+

User: craz
Question 1. Correct!
Question 2. Incorrect!
Question 3. Incorrect!
Question 4. Incorrect!
Question 5. Incorrect!
Final score: 2/10 - 20.0%
1/5 questions correct.
Elapsed Time: 104 seconds
Poor. You need to work harder.
Do you want to retake the quiz?
[2/3] attempts left.
[y]es or [n]o
Please enter your choice: 

```

- Figure 8: Summary page

3.2.3 Displaying user results:

The User results function takes the user id and matches it to results in the file-system database. The function then displays all the results with the date the quiz was taken.

```

+=====+
                User Results Menu...
UserID: craz
+=====+

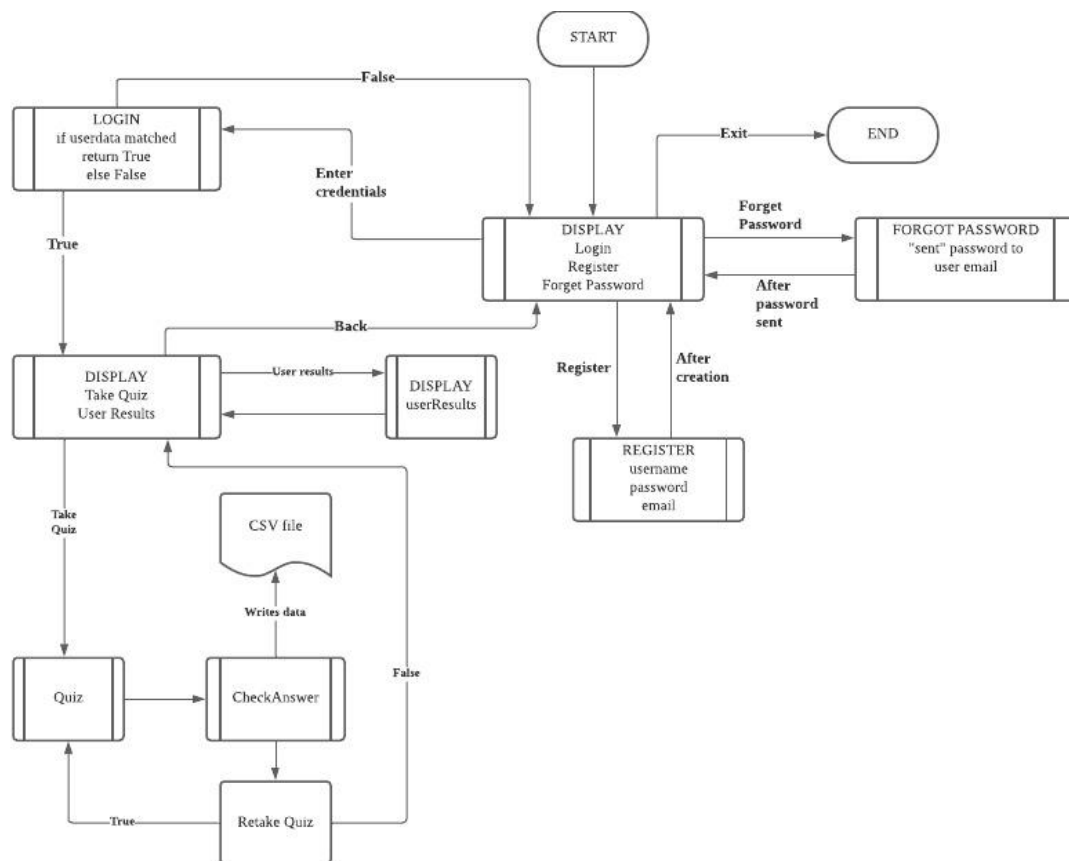
1. 2021-11-29 02:59:52 - 100%
2. 2021-11-29 03:02:31 - 100%
3. 2021-11-29 05:26:02 - 100%
4. 2021-11-29 06:09:47 - 20%
+=====+

```

- Figure 9: Result List

3.3 FlowChart:

File: user.py



- Figure 10: user script flowchart

3.4 Post-Implementation Support

In the case that there are issues with the program. Contact the admins at BetterTutors Pte Ltd.

Email:	support.btrtutors@gmail.com
Phone Number:	+65 9859 5246

3.5 Functions List:

3.3.1 Menu()

- The purpose of this menu is to provide the user with the necessary options to choose from
 1. Login
 2. Register user
 3. forgotPassword

3.3.2 generateOTP()

- Generates a unique One Time Pin when called
- Used for password Recovery

3.3.3 registerUser()

- Lets the user register a new account they can use in the Quiz
- Every user will have a unique user id to locate themselves
- Every user will be required to enter a unique Username, password, email.
- The data entered will be encoded so people with access to the database will not see the values in plaintext.
- Creates an account with the ACL(access control level) of "00000."

3.3.4 forgotPassword()

- Prompts the user for their email
- Displays 'Password has been send to your email' if the email is valid and in the database.

3.3.5 Login()

- Prompts username and password
- If valid and in the database, redirects user to doUserQuestions() for quiz taking or displaying user results

3.3.6 doUserQuestions()

- Lets the user choose to take the quiz or to view user results
 1. Take Quiz
 2. User results

3.3.7 userResults()

- Takes the user's row id, finds all the results with the same row id, and lists them to see.

3.3.8 takeQuiz()

- Displays the number of questions the admin selected
- Lets the user select options a, b, c, d.
- Gives user answers to checkAnswer() to display results

3.3.9 checkAnswer()

- Takes in the user answers from takeQuiz() and compares it against model answers saved in the database along with the questions
- Saves the username, Question number, user answer, correct answer, time taken to do the quiz, score, and date into a CSV file

3.5 Class List:

3.4.1 colours

- Give print statements colours in the terminal.

4. Admin script

4.1 How to run the Admin script:

Both the user and admin script can be run within the Main.py script. Instructions in the Main.py script will specify which lines to uncomment to run the appropriate scripts.

Or

They can press f5 on their keyboard in the user script to run the admin menu.

4.2 Main functions of the Admin script:

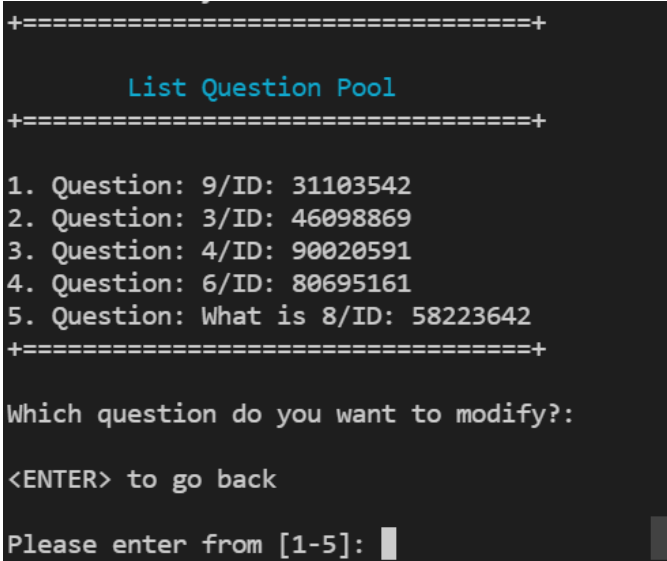
4.2.1 Question settings:

4.2.1.1 Create a new question pool

Deletes the current question pool and ask the admin how many questions they want to add. The questions should consist of a question, 4 options and a correct answer. The questions are then added to the question pool in the file system database.

4.2.1.2 List Question pool to update/delete

Lists all the existing questions in the question pool and lets the admin select which question they want to modify. Admin is able to select if they want to select to change questions, options or the correct answer. Admin is also able to delete the questions if the option is selected.



```
+=====+
      List Question Pool
+=====+

1. Question: 9/ID: 31103542
2. Question: 3/ID: 46098869
3. Question: 4/ID: 90020591
4. Question: 6/ID: 80695161
5. Question: What is 8/ID: 58223642
+=====+

Which question do you want to modify?:

<ENTER> to go back

Please enter from [1-5]: █
```

- Figure 11: List all questions in question pool

4.2.1.3 Add questions to the existing question pool

Ask the admin how many questions they want to add. Adds the questions to the file system database.

```

Creating Question 4
What is the question?
> what is a dog
What is the option a?
> cat
What is the option b?
> wolf
What is the option c?
> horse
What is the option d?
> dog
What is the correct answer?
[a, b, c ,d]
> d
['cat', 'wolf', 'horse', 'dog']
what is a dog
d
Question 4 created successfully

```

- Figure 12: Question creation

4.2.1.4 Randomize questions

Uses **random.shuffle** to change the question pool order. So when the questions are listed out in the take quiz function, they will be randomized.

```

+=====+
                    Randomizing Questions...
+=====+

+=====+

Questions randomized successfully!
+=====+

```

- Figure 13: randomized question pool successfully

4.2.1.5 Select the number of questions in the quiz

Allows the admin to select the number of questions that will be tested in the quiz. Also changes the header so that the number of questions matches the number of questions tested in the quiz

```

+=====+
          Select Number of Questions
+=====+

How many questions should the Quiz have?

<ENTER> to go Back
Current number of questions: 5
Please enter your choice: 6
+=====+

Number of Questions set successfully!
+=====+

```

- Figure 14: Question count set successfully

4.2.1.6 Select number of quiz attempts

Allows the admin to select the number of attempts the user is allowed to retake the quiz.

```

+=====+

Welcome to the quiz attempts setting menu
+=====+

Current number of Attempts: 4
Please enter your choice: 3
+=====+

Number of Attempts set successfully!
+=====+

```

- Figure 15: Attempt count set successfully

4.2.1.7 Change header for the CSV result file

Displays the current header format for the CSV file and prompts the admin to see if they want to change the header. Then the admin is asked to enter the new header to the CSV file, updating the header.

```

Please enter your choice: 7
Current Header:
User | Question | User Answer | Model Answer | Elapsed Time | Score | Date

<ENTER> to go Back
+=====+

Are you sure you want to change the Header?[y/n]
Please enter your choice: y
Please enter the new Header:
User: █

```

- Figure 16: Change the header for csv results file

4.2.2 User settings

4.2.2.1 Create Admin account

Lets the admin register an account with admin privileges.

4.2.2.2 List users to Update/Delete

Lists out all the users and prompts the admin to select a user to modify data. When selected a user, it shows the current username, password, email, access control level(ACL) and one-time pin(OTP). The admin will then be prompted to select a value to change.

```
+=====+
      List of Users
+=====+
1. Username: craz / UserID: 1645536
2. Username: User / UserID: 99461699
<ENTER> to go Back
Please enter your choice [1-2]: █
```

- Figure 17: List of users to choose from

```
+=====+
      List of Users
+=====+
1. Username: craz / UserID: 1645536
2. Username: User / UserID: 99461699
<ENTER> to go Back
Please enter your choice [1-2]: 1
1. username:craz
2. password:craz
3. email:craz@gmail.com
4. acl:11111
5. otp:11979950
6. DELETE USER
<ENTER> to go Back
Please enter your choice [1-6]: █
```

- Figure 18: Choice of which data to modify

4.2.3 All User results

Lists all the users and prompt the admin to select a user to view results. Once selected the results will be listed along with the date of the quiz attempt.


```

+=====+
All User Results
+=====+

1. Username: craz / UserID: 1645536
2. Username: User / UserID: 99461699

<ENTER> to go Back
Please enter your choice [1-2]: 1
+=====+

All User Results
+=====+

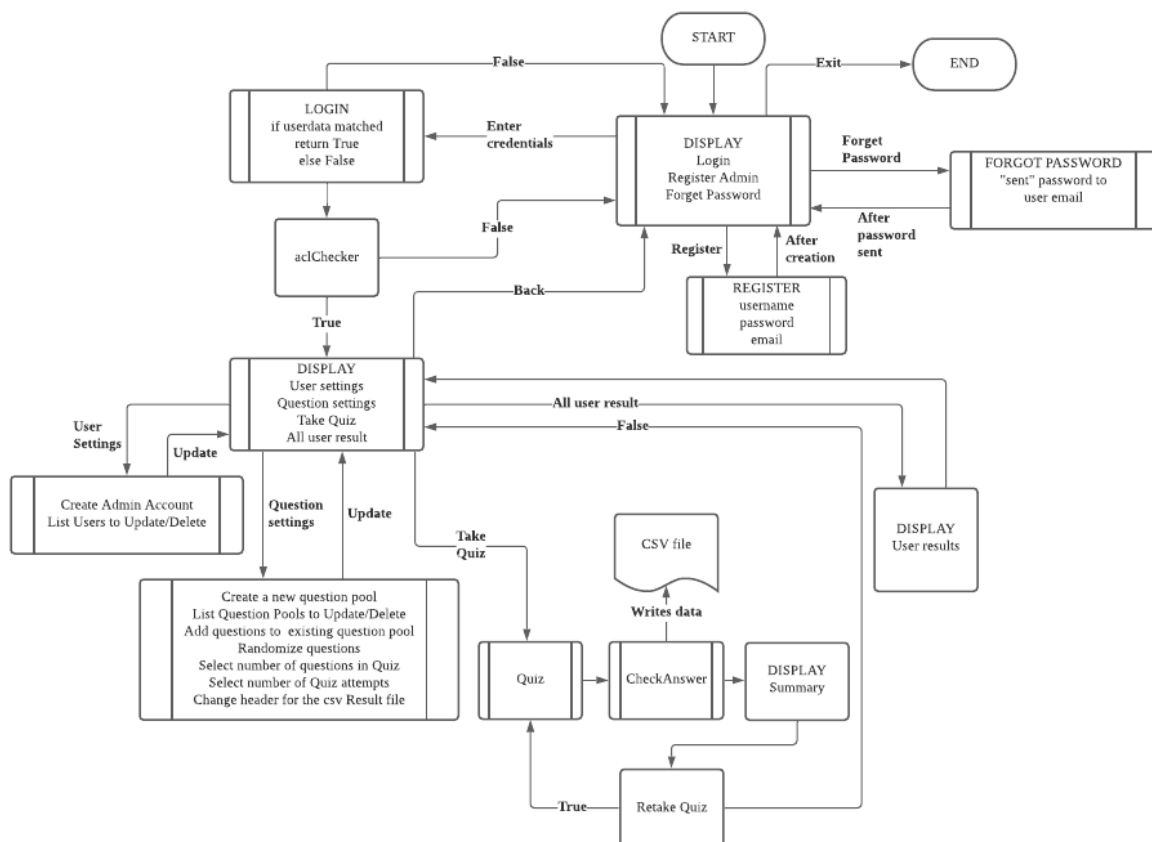
1. 2021-11-29 02:59:52 - 100%
2. 2021-11-29 03:02:31 - 100%
3. 2021-11-29 05:26:02 - 100%
4. 2021-11-29 06:09:47 - 20%
+=====+

```

- Figure 19: Able to choose which user to see results

4.3 FlowChart:

File: admin.py



- Figure 20: admin script flowchart

4.4 Post-Implementation Support

If the admins and programmers at Better Tutor are unable to solve the issue.

Please contact the following email to be instructed on further assistance

Email:	ryan.support@gmail.com
Phone Number:	+65 9783 9382

4.5 Function List:

4.5.1 Menu()

- The purpose of this menu is to provide the user with the necessary options to choose from
 1. Login
 2. Register admin
 3. forgotPassword

4.5.2 generateOTP()

- Generates a unique One Time Pin when called
- Used for password Recovery

4.5.3 aclchecker()

- Check if the account used to log in has admin privileges
- If the account doesn't have permission, the user will be unable to log in

4.5.4 Login()

- Prompts username and password
- If valid and in the database, redirects user to doUserQuestions() for quiz taking or displaying user results.

4.5.5 registerUser()

- Lets the user register a new account they can use in the Quiz
- Every user will have a unique user id to locate themselves

- Every user will be required to enter a unique Username, password, email.
- The data entered will be encoded so people with access to the database will not be able to see the values in plaintext.
- The account created will have an ACL(access control level) of "11111"

4.5.6 forgetPassword()

- Prompts the user for their email
- Displays 'Password has been send to your email' if the email is valid and in the database.

4.5.7 adminMenu()

- Provides users with the necessary options to choose from
 1. User settings
 2. Question settings
 3. Take Quiz
 4. All User Results

4.5.8 doAdminResult()

- List all users with their respective user-id
- Prompts the user to select a user to view results
- Returns user-id

4.5.9 doAdminListResults()

- Takes user-id from doAdminResult() to find results with the same user-id
- Displays all the results of the user with the date of quiz taken

4.5.10 doAdminUser()

- Gives the user options to create a new account or to update settings
 1. Create new user
 2. List user to Update/Delete

4.5.11 doAdminListUsers()

- Lists username and user-id for all users

- Prompts the user to select a user to Update/Delete
- Returns the user-id

4.5.12 doAdminDelData()

- Removes the user information when provided the user-id

4.5.13 doAdminEditData()

- Prompts the user to select whether to update the username, password, acl(access control level), email or delete the user.
- Updates the data when provided the user-id

4.5.14 doAdminRandomizeQuestions()

- Randomizes the question pool for the quiz

4.5.15 doAdminQuestions()

- Shows all settings for quiz and question options
 1. Create new question pool
 2. List questions to Update/Delete
 3. Add questions into the existing question pool
 4. Randomize questions
 5. Select number of questions in quiz
 6. Select number of attempts in quiz
 7. Change header for the CSV result file

4.5.16 doAdminSelectHeader()

- Displays the current header format
- Prompts the user to enter the new header values
- Writes a new header in the CSV results file

4.5.17 doAdminSelectAttempts()

- Displays the current number of attempts
- Prompts the user to enter a new number of attempts
- Updates the current number of attempts

4.5.18 doAdminAddQuestions()

- Asks the user to select the number of questions to add
- Prompts the user to create a question
- A question consists of the Question, options(a, b, c, d) and the correct answer(a, b, c, d)
- Adds the question to a database

4.5.19 doAdminSelectQuestions()

- Prompts the user to input the number of questions needed in the quiz
- Creates a new header according to the number of questions set by the admin

4.5.20 doAdminCreateQuestionPool()

- Confirms with the admin if they want to delete the current question pool to create a new question pool
- If user confirmed, deletes the current question pool
- Prompts the admin to ask how many questions to add
- Creates the questions with a unique rowid

4.5.21 doAdminListQuestionPool()

- Lists all the questions in the question pool currently
- Prompts the user to ask to select a question for modification
- Returns the unique rowid of the question chosen

4.5.22 doAdminModifyQuestion()

- Finds the question, options and correct answer based on the rowid given by doAdminListQuestionPool()
- Asks user whether they want to modify username, password, email, acl or delete the user

4.5.23 takeQuiz()

- Displays the number of questions the admin selected
- Lets the user select options a, b, c, d.
- Gives user answers to checkAnswer() to display results

4.5.24 checkAnswer()

- Takes in the user answers from takeQuiz() and compares it against model answers saved in the database along with the questions
- Saves the username, Question number, user answer, correct answer, time taken to do the quiz, score, and date into a CSV file

4.6 Class List:

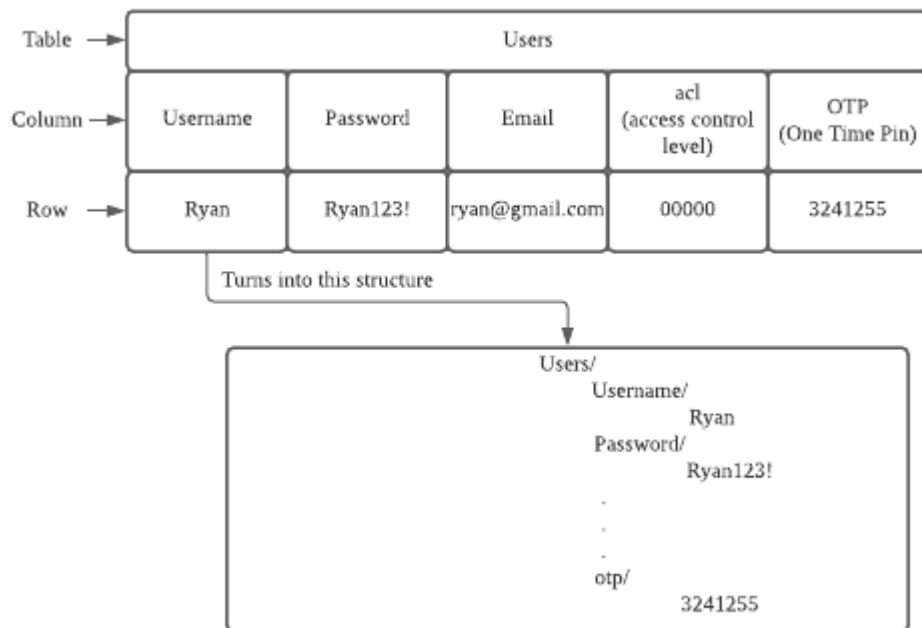
4.6.1 colours

- Give print statements colours in the terminal.

4.7 Input/Output Files:

4.7.1: Data storage:

Each data is stored in the file name as shown below:



Example:

Filename 47128222_s_b'VXNlcg=='

Row ID	Data Type	Data
47128222	s	b'VXNlcg=='
string	string	string

4.7.2 results.csv:

Username	Question1	User ans...	Correct a..	TimeTook.	Score	Date
craz	1 + 1	a	a	18.42 sec	100%	d/m/y h/min/sec
string	string	string	string	string	string	string

5. System Documentation Overview

4.1 Program Features

Features	Completion(%)
Registration of user	100%
Compliance of password	100%
Validation of input values	95%
Select X questions from the question pool	100%
View results of logged in user	100%
Modify user data	100%
View logged in user results	100%
View all user Results	100%
Creation of Admin account	100%
Creation of new Question pool	100%
Add questions to the existing question pool	100%
Select number of quiz attempts	100%
Change CSV file header*	95%
Export user results to CSV file*	95%
Randomize questions pool*	100%
aclChecker*	100%
Taking of the Quiz	90%
Checking answer	95%

