Task completed:

| Date started | Date completed |
| --- | --- |
| 03/05/2019 |  |

Contents

[Task completed: 1](#_Toc535494764)

[Analysis 1](#_Toc535494765)

[Design 2](#_Toc535494766)

[Test design 3](#_Toc535494767)

[Development 4](#_Toc535494768)

[Testing 5](#_Toc535494769)

[Evaluation 6](#_Toc535494770)

[References/Resources Used: 7](#_Toc535494771)

Analysis

Key success criteria for your program.

**Success Criteria:**

1. Login System:

A User can enter user name

B User can enter password

C Script can check if username exists

D1 Script rejects username cause it doesn’t exist

D2 Script Accepts username

E Script checks password connected to username

F1 Script rejects password

F2 Script Accepts password

G Runs next part of script

2. Read songs/artists/accounts from an external file

3. Script is able to use/edit those files

4. Script takes first letter of artist and song and puts then im a question format

5. User Guesses

6. Answer checking:

A Compares user input to external file string

B1 Rejects input due to not being the same

B2 Accepts input runs next script

7. Adds score to that account in external file

8. Repeats asking 10 questions

9. Script reads external scores

10. Script displays your score

11. Script displays top 5 scores

12. Final:

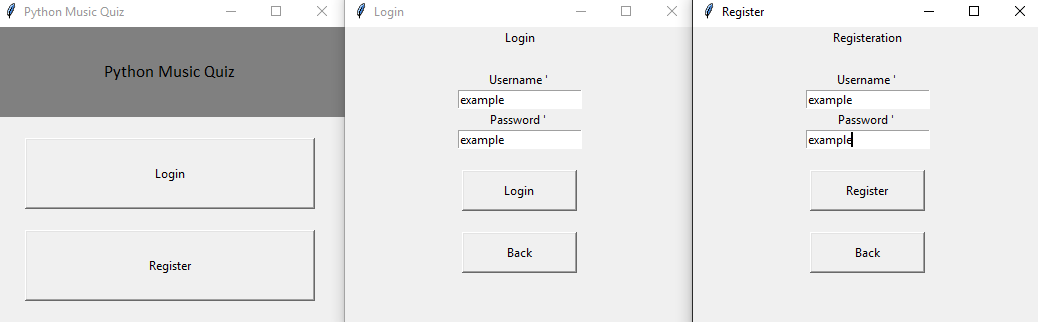
A Gives user options

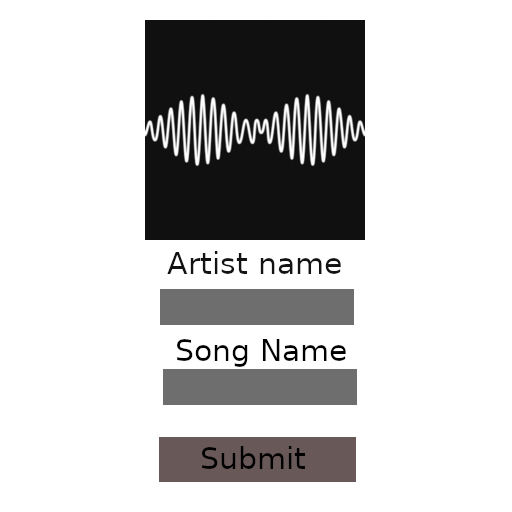
B1 User quits

B2 User inputs retry and starts code again at step 2

Design

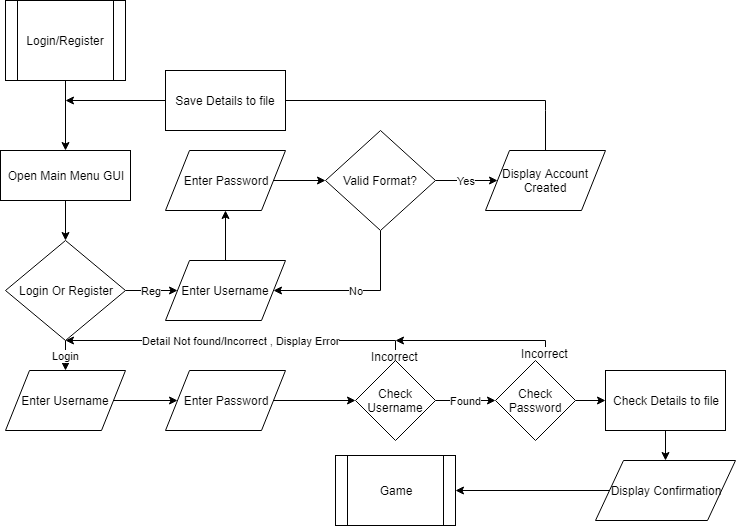
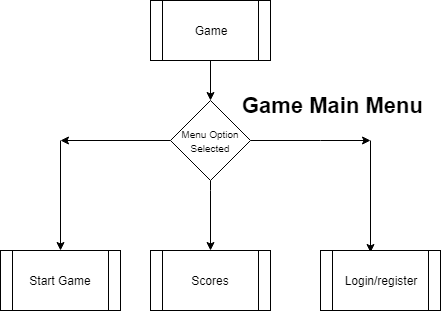
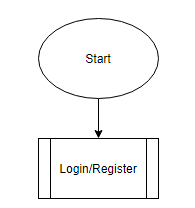
* *You may like to create a flow chart which will show broadly how your program will work. If so include your flow chart in this section.*
* *You must create pseudocode for a part of your program (minimum of 15 lines)*

Concept/mock-up of login system (made a test)

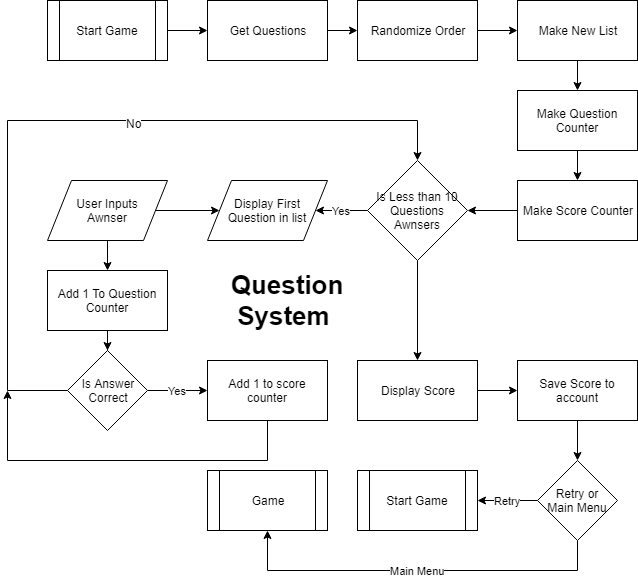
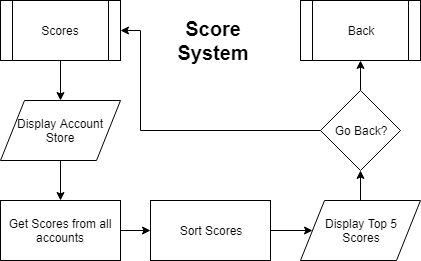


Concept for quiz UI

(Made in gimp)



Login System



Test design

* *Think of tests that you can carry out to see if your system works*
* *Remember to try and use normal, boundary and erroneous tests.*
* *If you wish to, you may add more tests to the table.*

**My tests:**

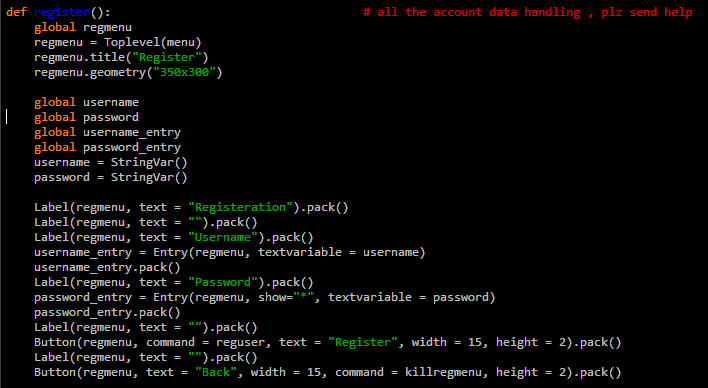
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | What am I testing? | What data will I use? | Normal/Boundary/Erroneous? | Expected Result |
| 1 | Does the Username  Check work | The username | N – Is input bigger than 0  E – Not found username | Will either reject or accept Username |
| 2 | Does the password check work | The password | N – Is input bigger than 0  E – Incorrect password | Will either reject or accept username |
| 3 | Question List Randomizer | The Question List | Erroneous | The Question list is randomized |
| 4 | Does Question score Counter work | Score count | Normal | Will it count the scores |
| 5 | Does the Question Counter Work | Amount of questions awnsered | Normal | When it hits 10 it ends the loop |

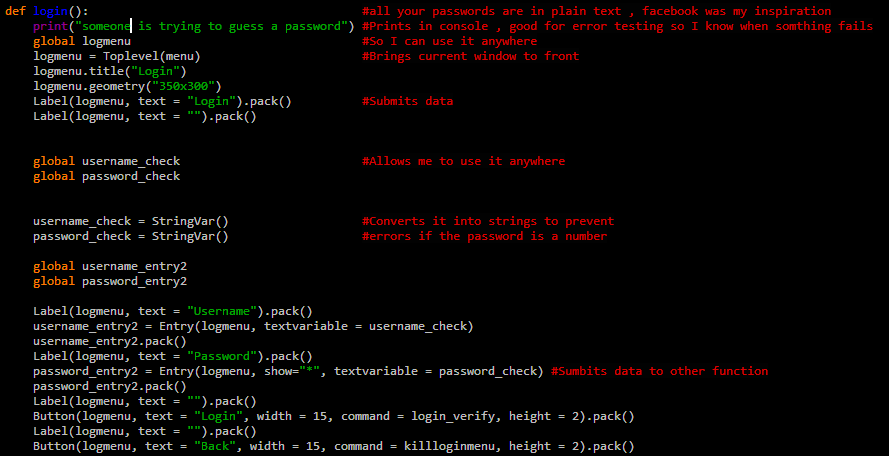
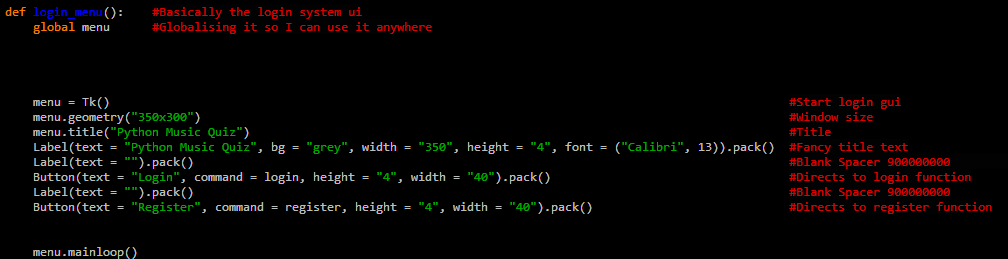
**My test screenshots:**

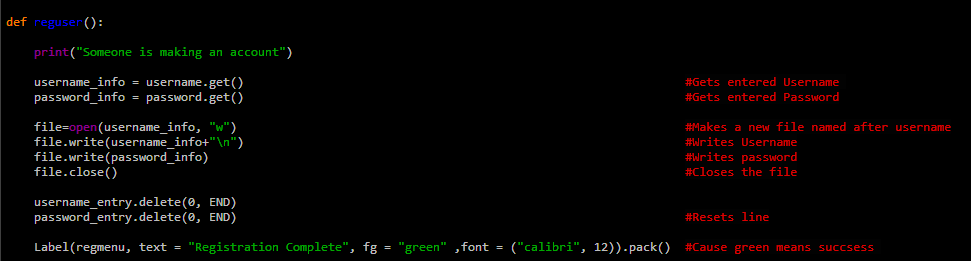
Development

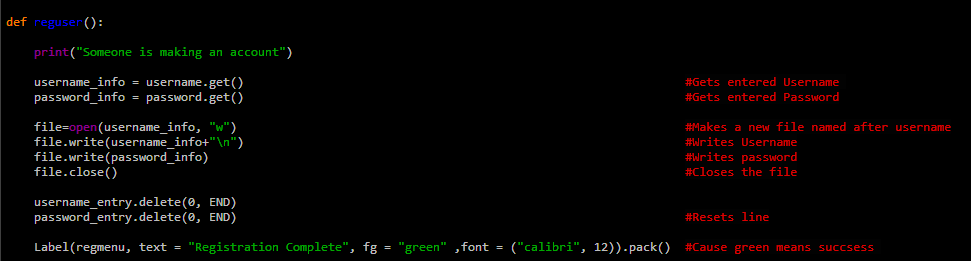
* *Copy and paste your code into this section*
* *Remember to try and add comments to your code to make it more readable!*

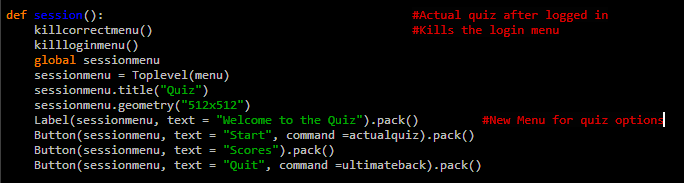
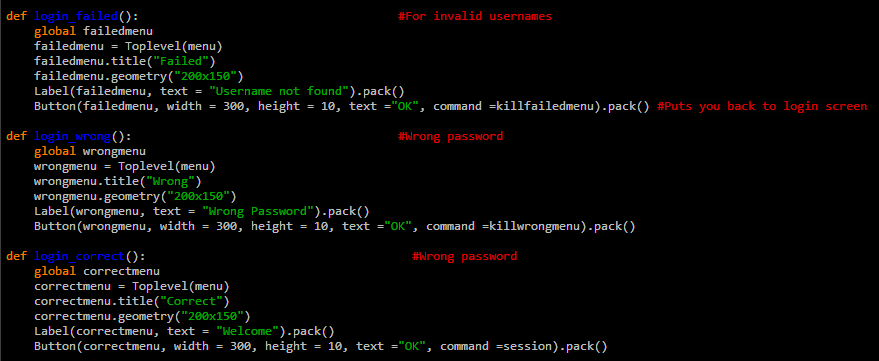
**My program code:**



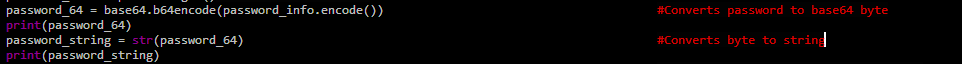


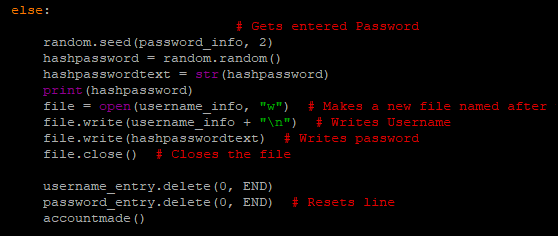




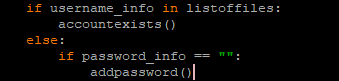


Added password encryption

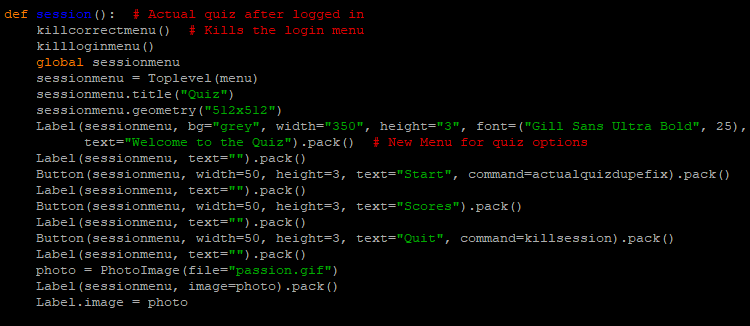




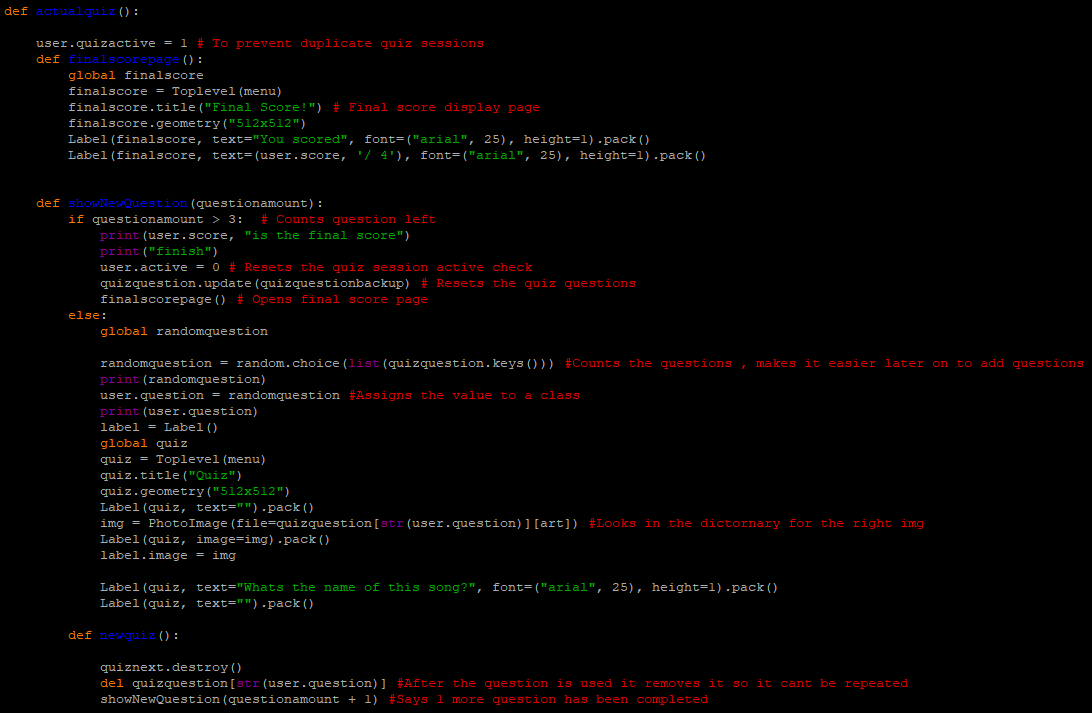
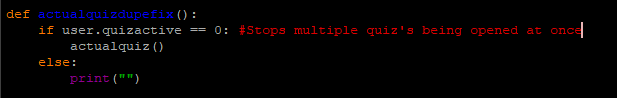
Completely redid encryption so it does not use base64 but now ACTUALLY hashes the passwords so it is secure.

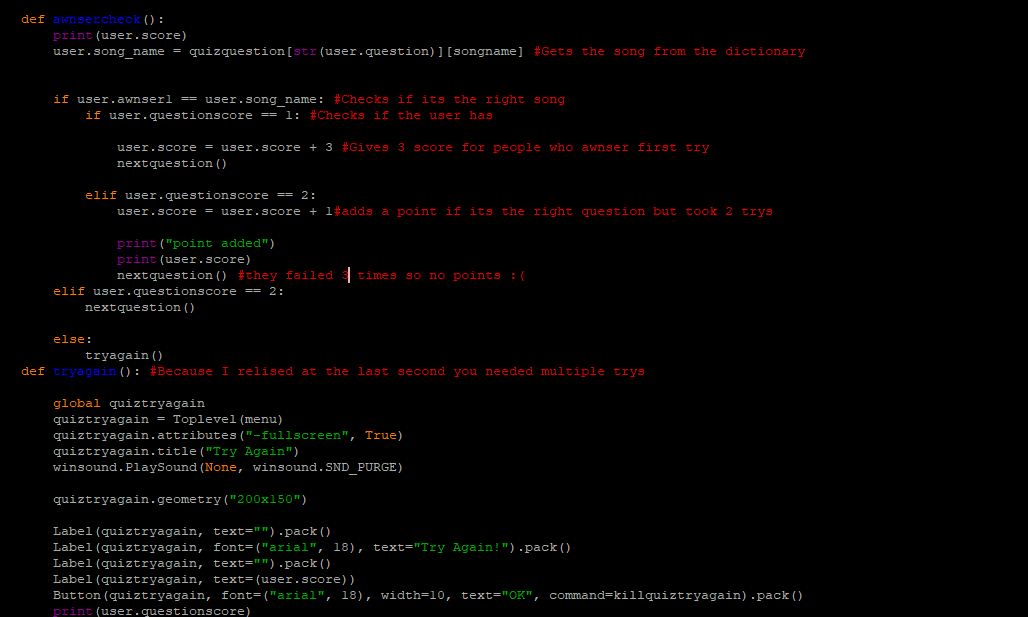


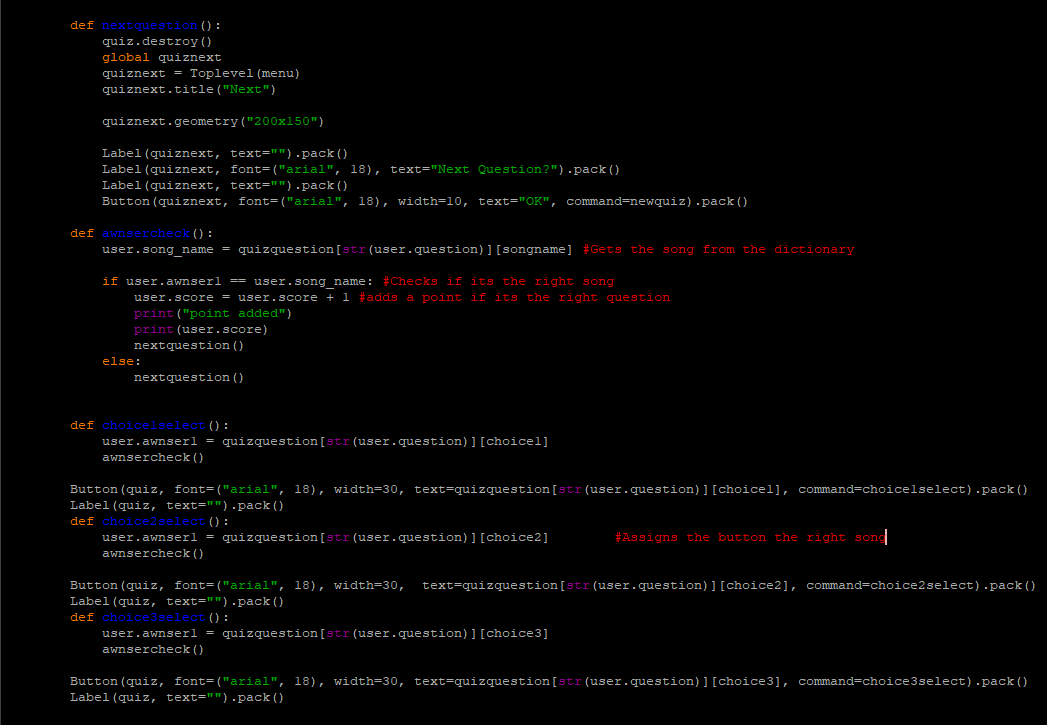
Lots of user sanitation over all of the login system to stop password less accounts and overwriting other accounts



Revamped the session menu so it looks better and has images.



 Rewrote the answer check to allow 2 chances for the user to get the right answer.

 Rewrote the answer check to allow 2 chances for the user to get the right answer.

Testing

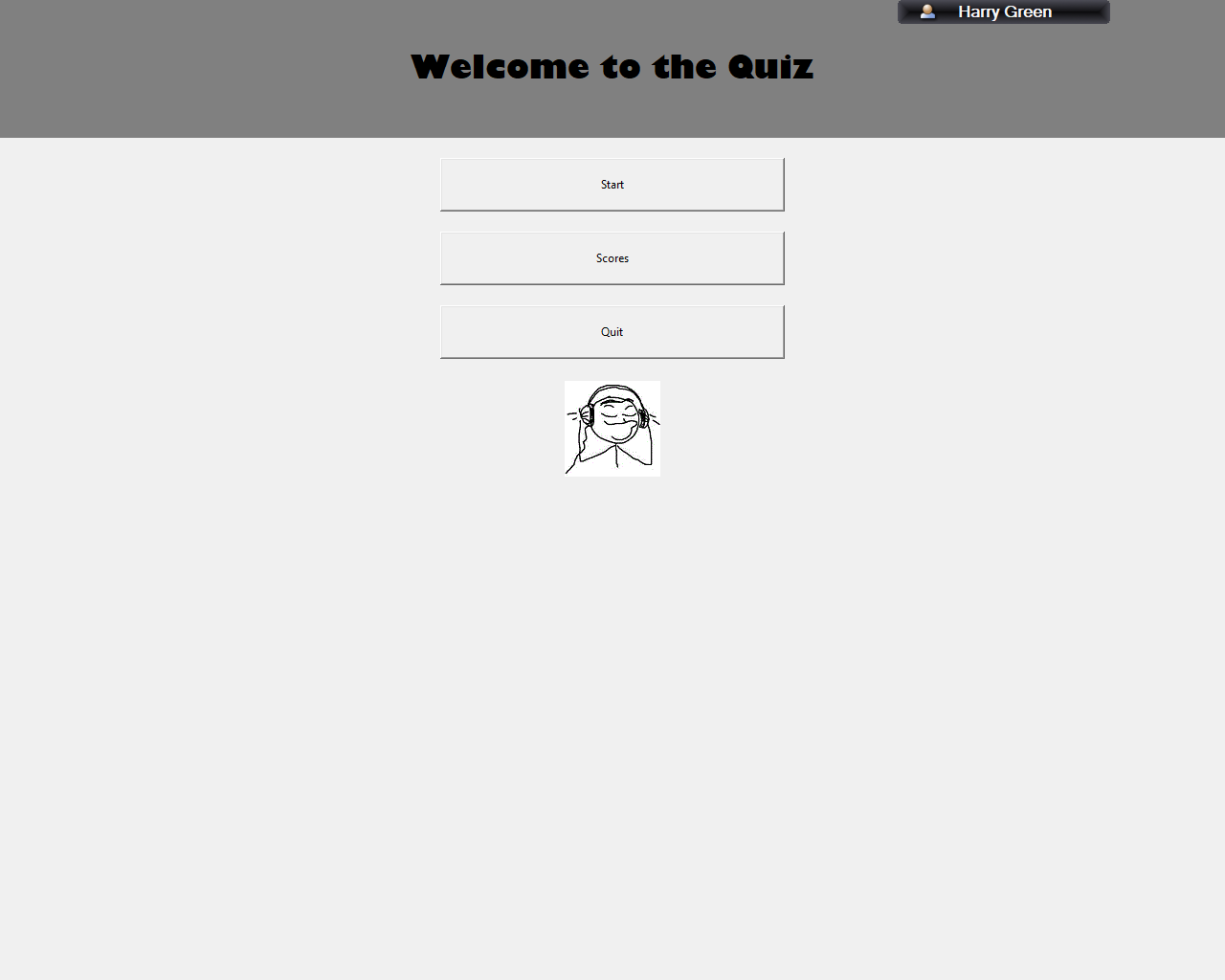
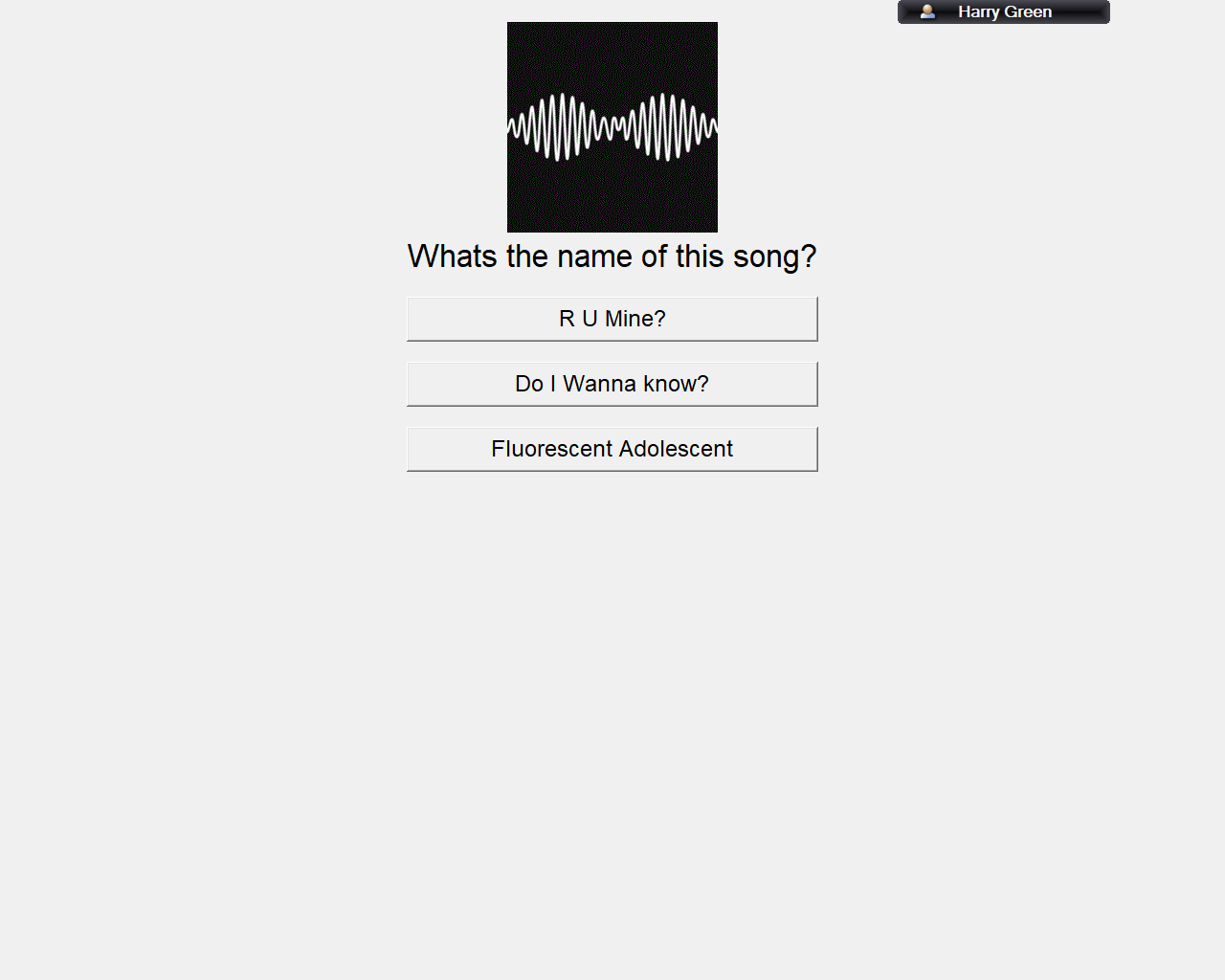
* *Show you have completed the tests you thought of*
* *Identify if you needed to make changes to your program*
* *Include the screenshots of the tests*

**My tests:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | What am I testing? | Expected result | Pass/Fail | Do I need to change my program? If so, how? |
| 1 | If multiple quiz’s can run at once | It would run fine | Fail , it broke everything | Added user sanitation for it |
| 2 | If my user sanitation works | It stops errors due to users breaking the script | Pass | There might be places where the user can break the program BUT I have fixed most of them |
| 3 | The multiple chance system | You can awnser questions multiple times and receive the correct score | Fail, It only worked the first time , then it counted incorrectly | Reset the classes for each question |
| 4 | Duplicate accounts | Accounts cant be overwritten via registration | Fail , accounts could be over written | Add checks the account does not already exsist |
| 5 | Accounts with no passwords | It won’t allow accounts with no passwords | Fail , it allowed accounts with no passwords , it also caused errors cause you can’t hash nothing | Add a check that the string is longer or equal to 1 |

**My test screenshots:**

**Test 1**

T 

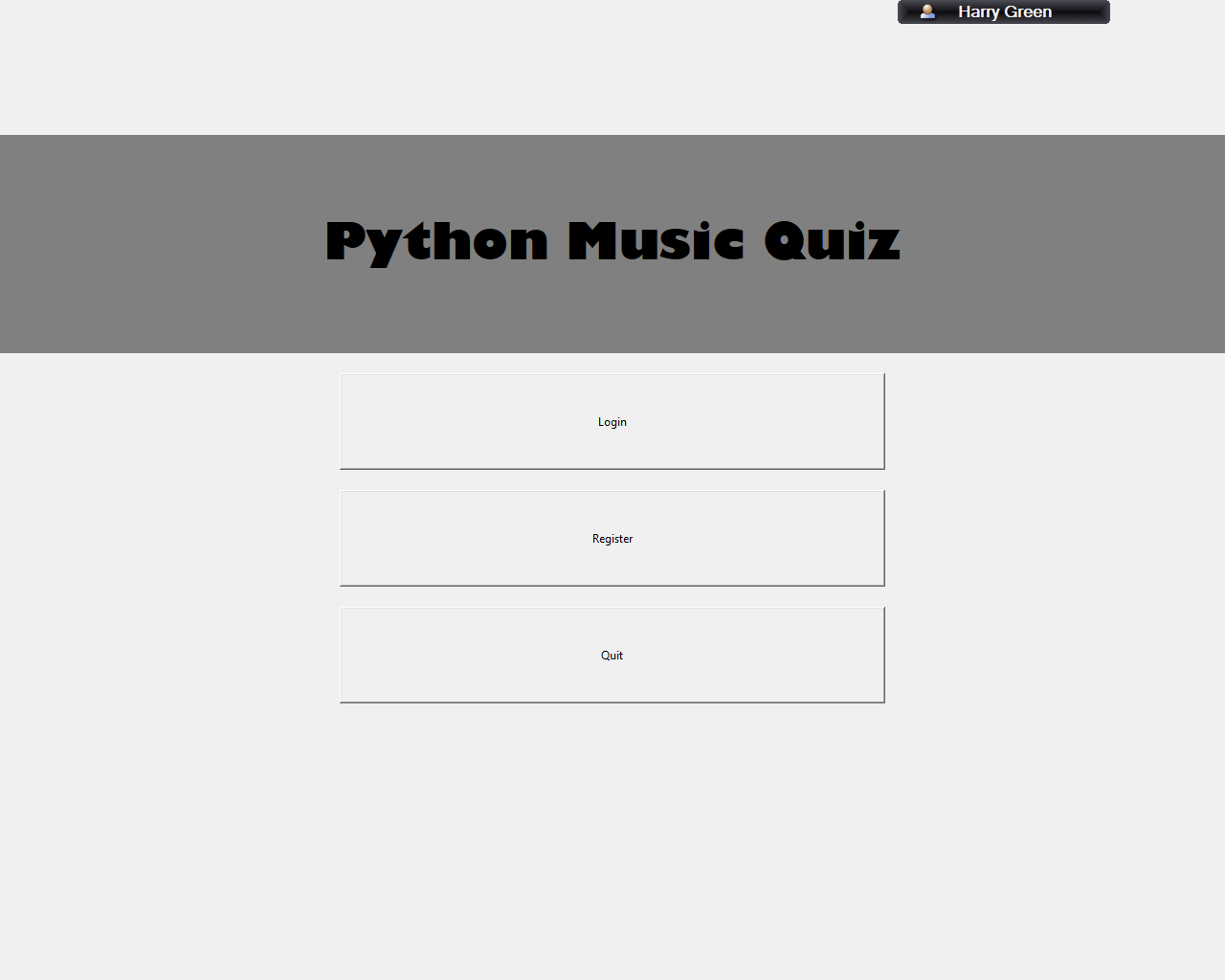
Due to it being full screen they cant open multiple!

Also there is a check so if they spammed the quiz button quickly it will only open 1.

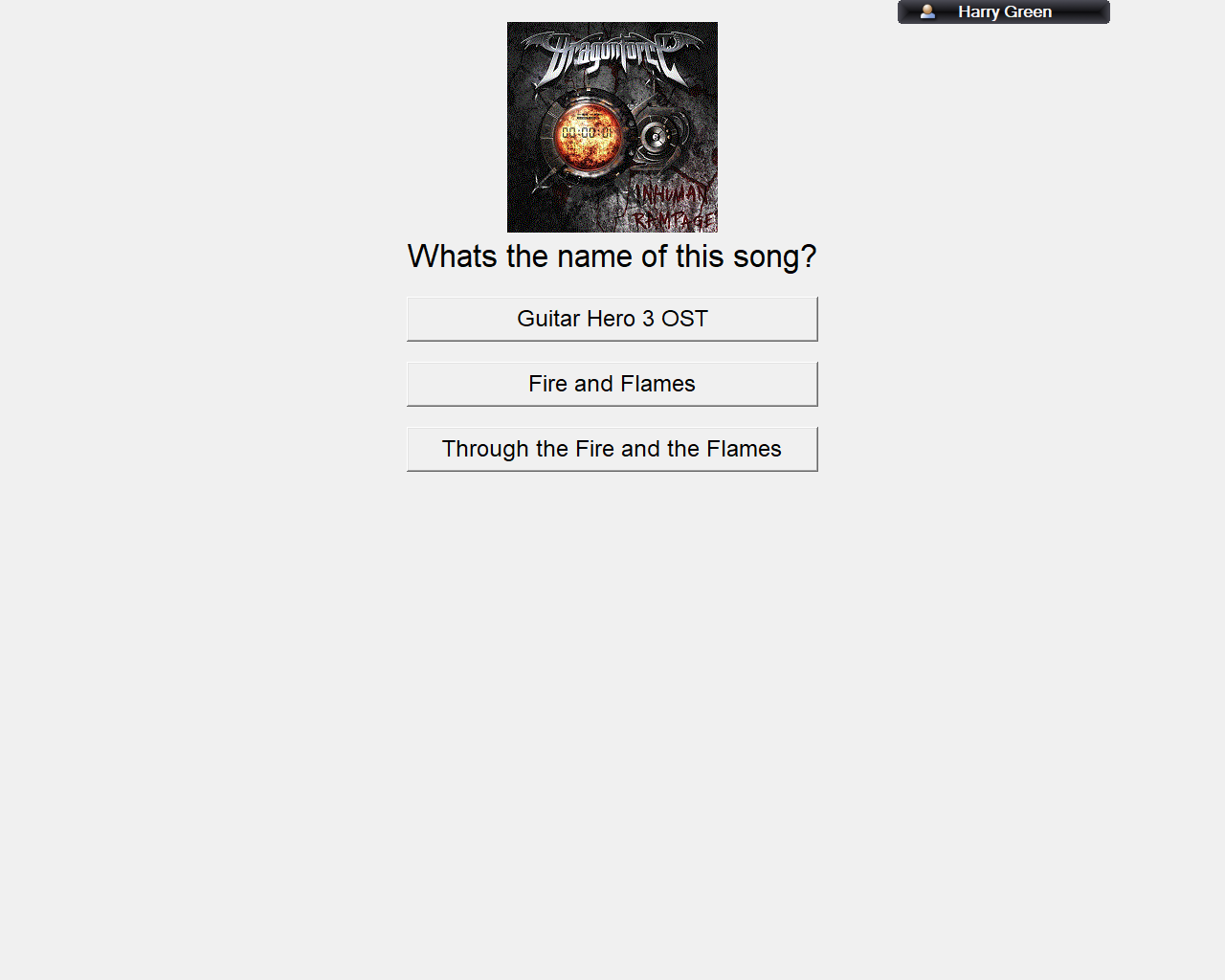
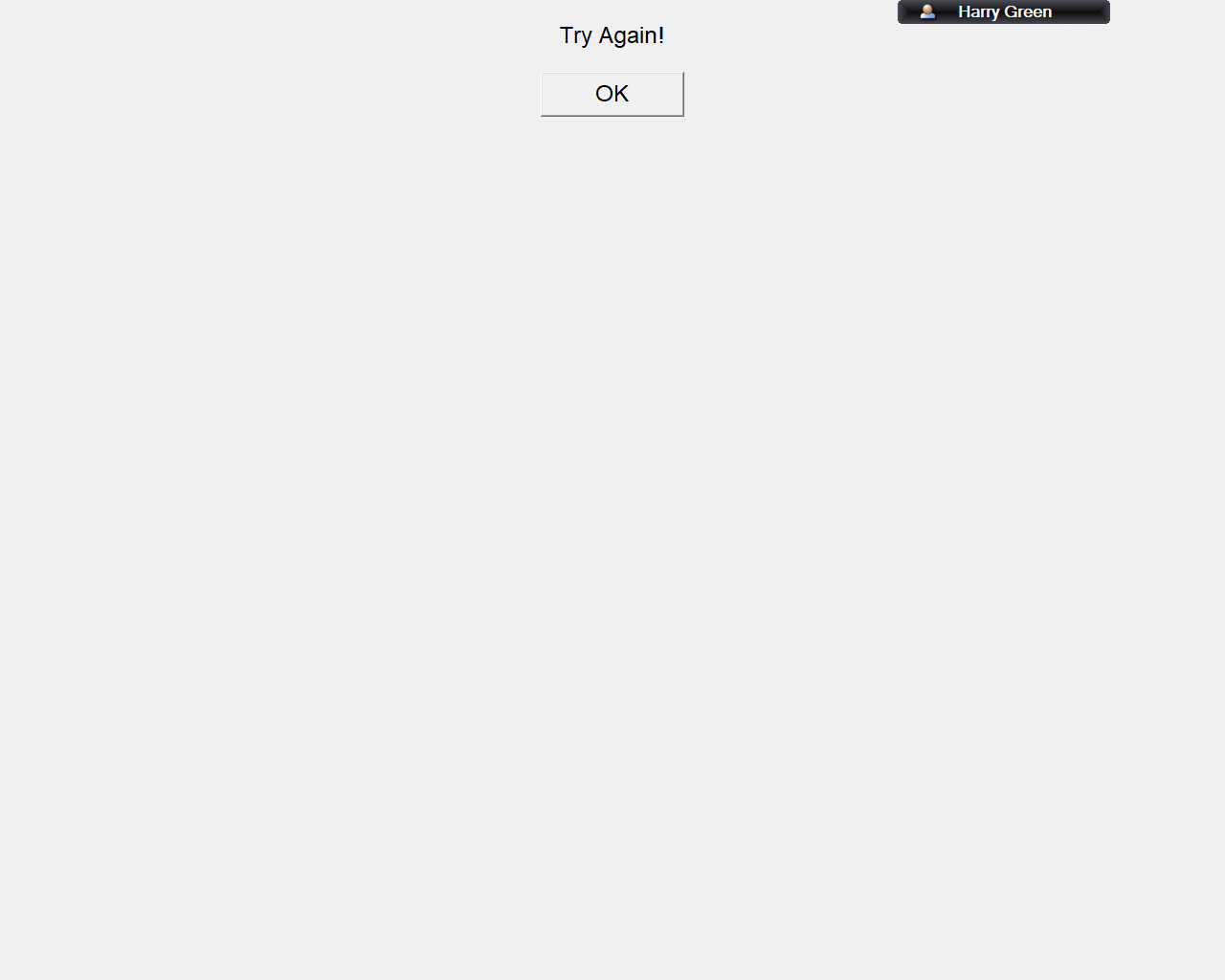
TEST 2

My user sanitation was mainly resolved by making tikinter run in full screen.

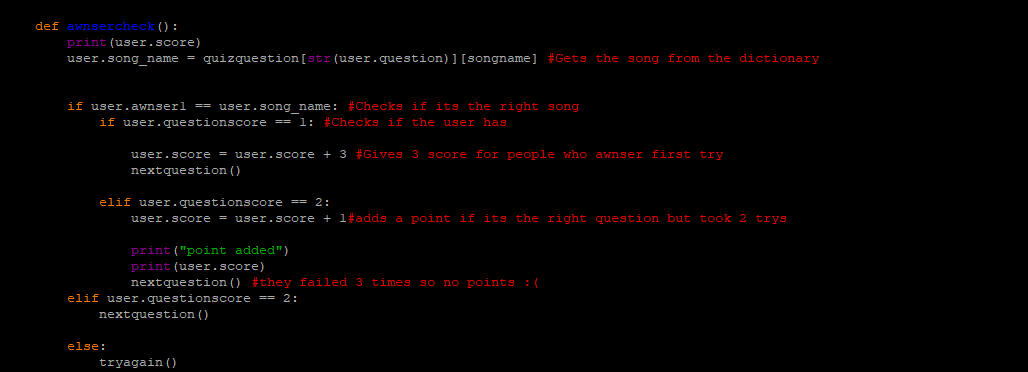
I also added checks so they cant spam buttons to open up multiple scripts at once.



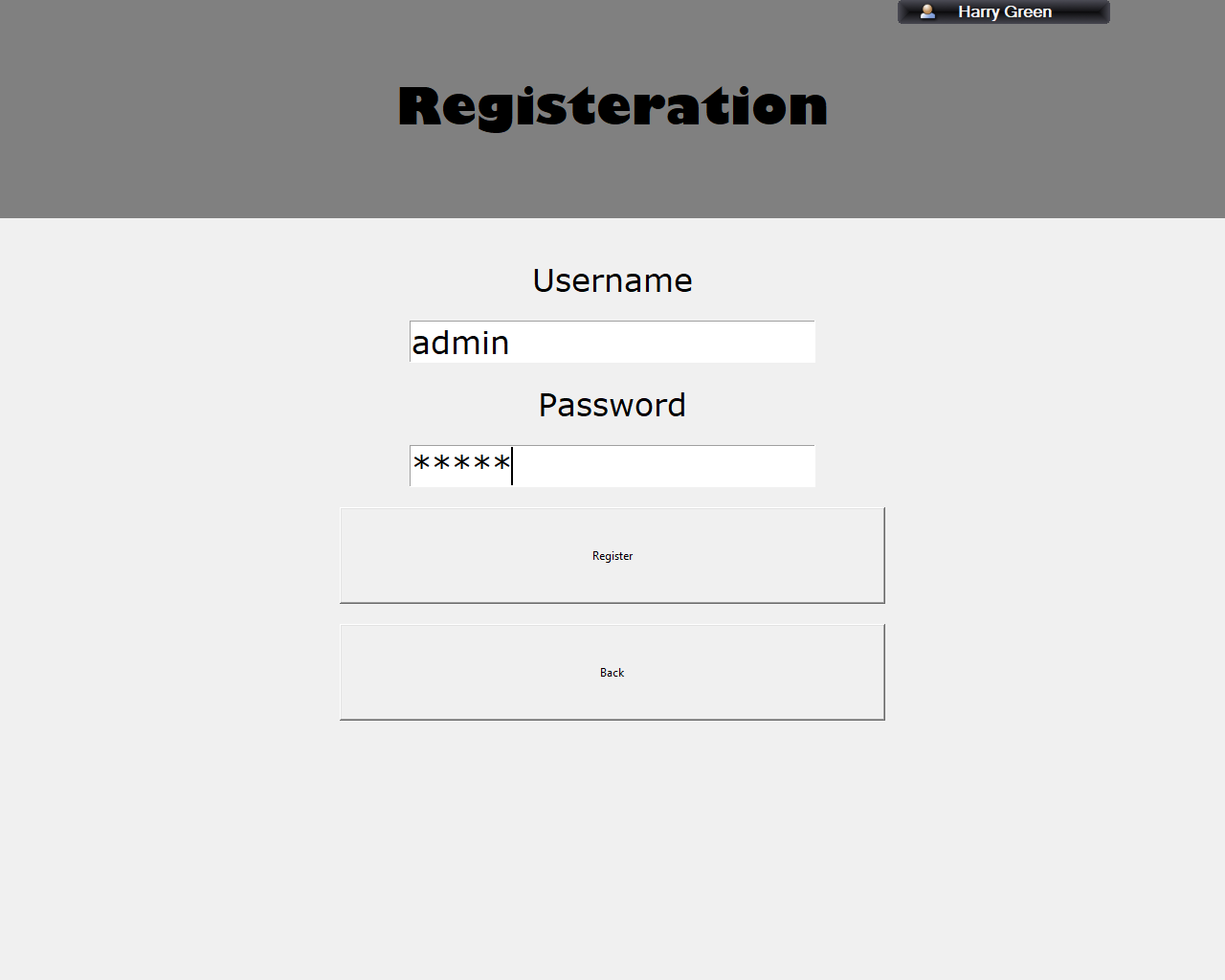
Test 3

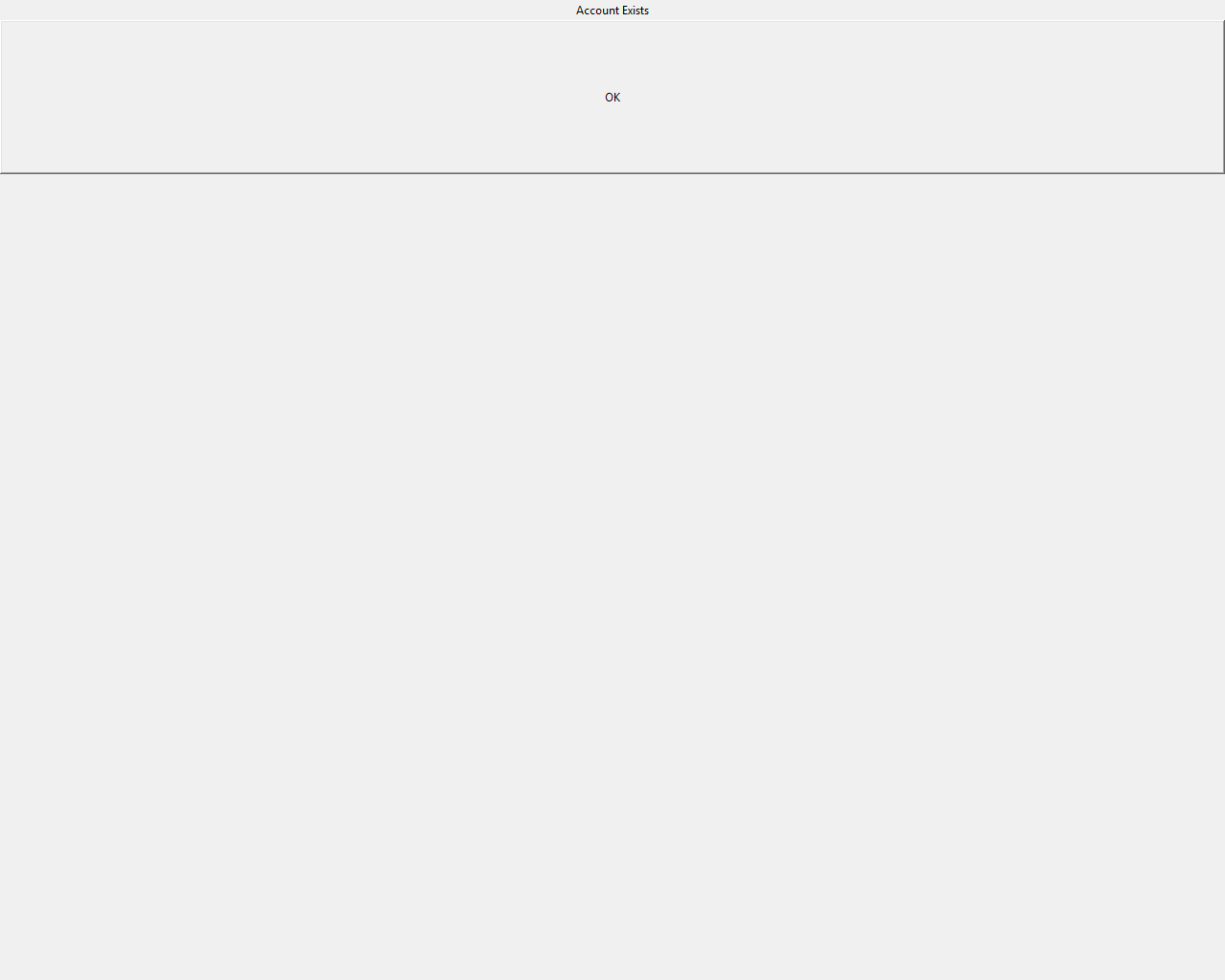
 

When you get the question wrong it lets you try again



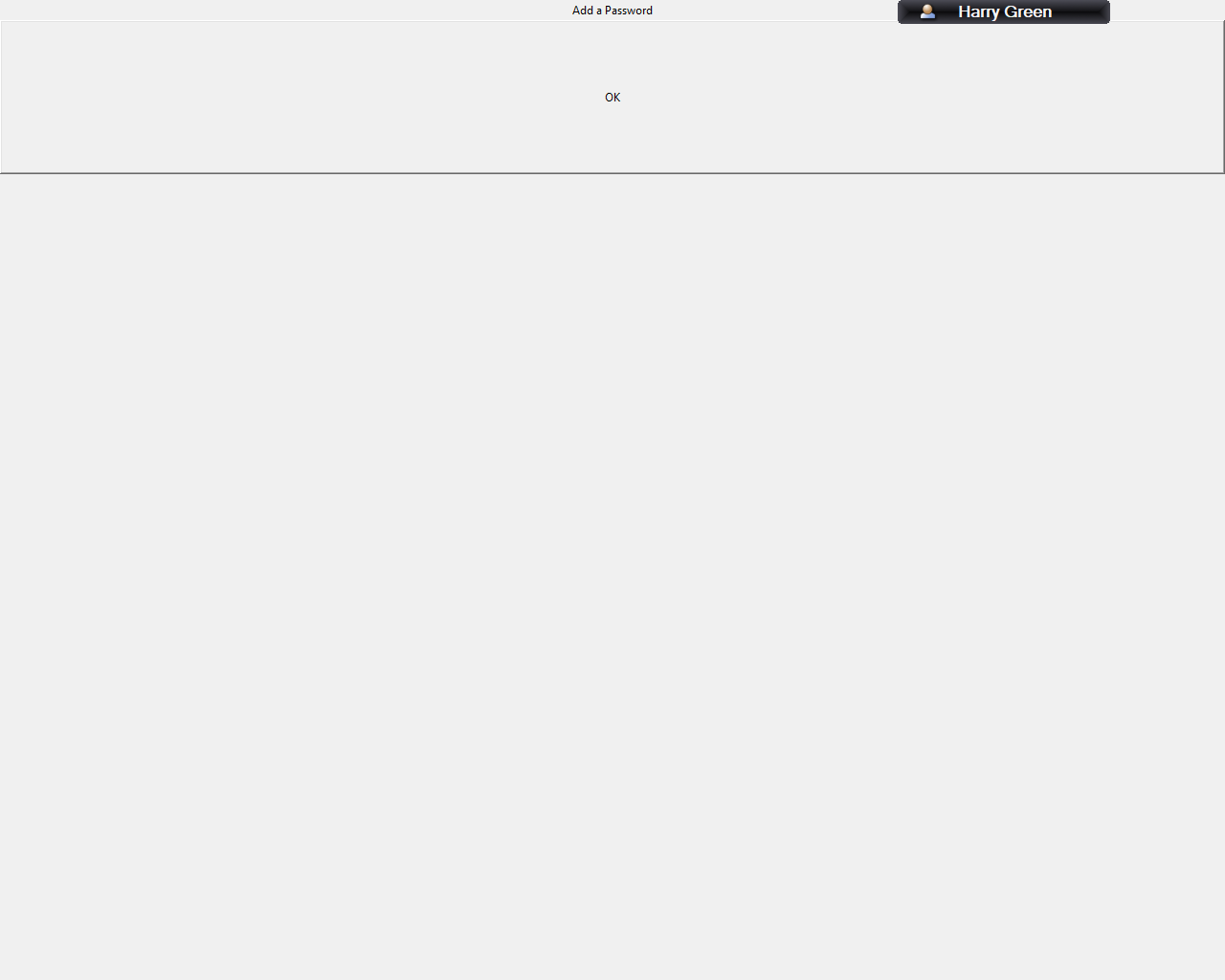
TEST 4





It gives a error if the account exists and allows you to retry

Test 5



Added a check if the passwords were blank.

Evaluation

* Evaluate how successful your program was. You may find it useful to refer back to your tests
* You should reflect on any new skills you have developed

This section should be approximately 200-500 words.

**How successful was my program?**

**T**

**What new skills have I developed?**

References/Resources Used:

|  |  |
| --- | --- |
| Resource Used | Reason |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |