Python3 Programming Project Report

Game of Chance

Student id: **u2091132**

**Introduction:**

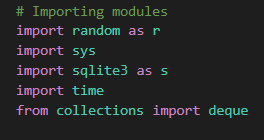
The program is intended for users who wants to play a game like roulette that relies completely on luck.

The problems I faced were mostly decision-based ones, like what modes to put and how to implement them, and how to make sure the user retains the money they won/lost in the game and how to keep it secure, and I had to choose to implement either a stack or queue so the user can see the winning/losing history of the session they are currently in.

My program solved those issues by checking the existence of the user account and if it exists it asks for the password, if not then it makes an account for the user and asks them to create a password for their account. The user automatically starts with 100£ and every game takes 10£ from them. The user gets to choose from 3 modes, the first which is High Risk- High Reward one, the second and third have similar difficulty, but yields a smaller reward. If the money in the user balance is spent they get to either top up with another 100£ or the game closes. The user also can see their session history in a queue if they choose to do so before exiting the game.

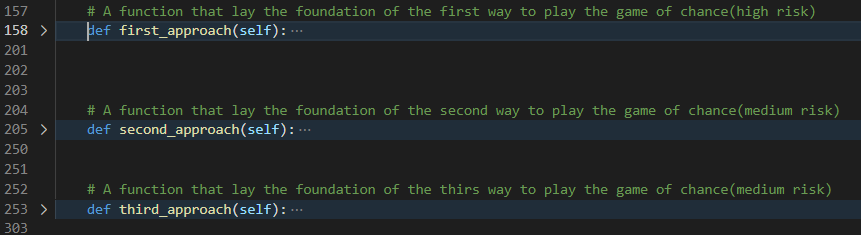
**Requirements:**

I had to import a few modules mainly random, sys, time, sqlite3 and a class from collections. I imported random as the game mainly depends on chance and on making the computer choose a random number and comparing it to the inputted player value and if the numbers match then the player’s balance increases if it does not then the player loses money. The second one I imported was sys module, I imported it as I wanted the app to close after the user lost all his money and refused to top up, when they chose to close the app or when they inputted the wrong password three times. I imported time as a supporting module to sys, as I used sys to close the app, but I wanted the app to print a message before it exits which it could not because the ***sys.exit()*** closed the app instantly, so I imported module to use the sleep function in it to delay closing the app for 20 seconds. I imported sqlite3 which is a database module that I used to create a database of all my users to check if their username exists or not, and check the password and to get their balance. The last module I imported was the deque class in collections which allowed me to use queues to make the user see their winning and losing history for this session.



**Design and Implementation:**

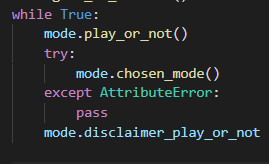
To make the program, as stated before I had to think of the design of the game itself and I decided to start my program by making the three game modes I had to do in the assessment, I called them “approaches” in my program. Those three also make the bulk of my program since they are the main game.



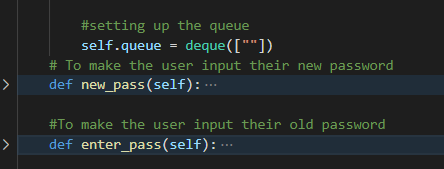
I added the user balance and set it to a fixed rate of 100 and had to see if the user wants to play the game or see their balance so I added the function below to check what the user wants, I also later added the option to see the queue in the same function. I added the disclaimer function before that is well to make sure that the user has low chances of winning.



After that I had to give the option for the user to choose which mode they want to play, which I did in the screenshot below, I also made an infinite loop so the game runs forever unless the player wants to exit or their money is lost and they do not want to top up. I also made an exception to fix a problem which occurs if the queue implemented does not have any inputted data and the user requests it.



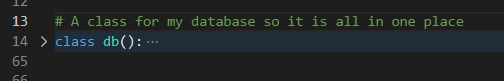
The last two functionalities I added to the game itself was the password and the queue. In the first the user needed to input the correct password to be able to play the game, if not they get to try again to a maximum of three times, if they still did not input the correct password the game exits and ask them to contact the casino. The second functionality I added was the queue functionality, where the user can see their winning and losing history for the current session they are in.



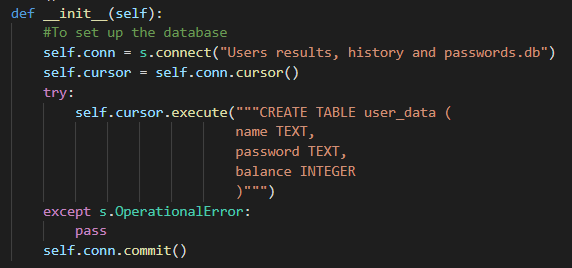
I added all of the above-mentioned functions to a single class named mode\_setup.



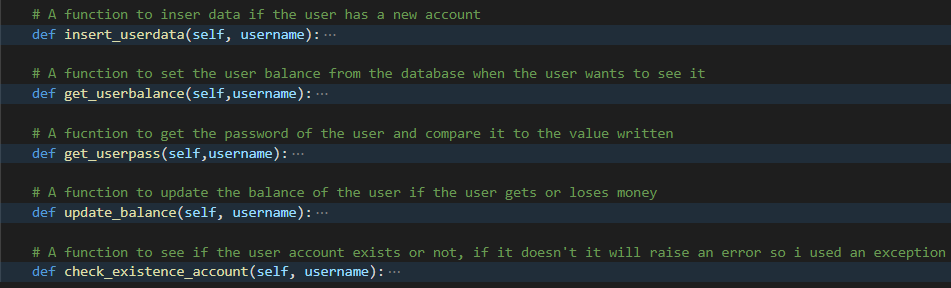
After that I decided that each of my users should have their own account with their own usernames and password, the main idea behind this was to make it more secure for users so only the intended user is able to access their own account and see their balance. I used the module sqlite3 for that and made another class for it named db.



I started the class by setting a table if it does not already exist, I added three tables to it, one for username, one for password and the third for balance.



I then made a few other functions in my database module, one to insert the user data when they register if they are new, one to get the balance of the user(100 by default) from the database, one to get the password of the user to compare it to the password inputted by the user, one to update the user balance if they win/lose a game, and the last function was to check if the user accounts exist or not by checking if the user name exists in the database or not, if it does the user will be prompted to input their password , if not they will prompted to create a new password for the their new account.



**Programming Techniques used:**

I used several Advanced programming techniques like databases to collect all the information we need about the user and store it somewhere so I can have easy access to it when needed, new users are always saved on the database when they register, and the balance is updated after every game. Using a database also allowed the information to be more secure behind the user’s password. I used loops frequently to either make sure the user only inputs a number from the range, or used them as infinite loops to make sure my apps run forever or until the user chooses to close it or they have no money left and refuse to top up. I used classes and functions to try to decrease the number of lines in the code as much as I can and make it as efficient as possible, I also made use of the arguments that can be inserted in functions or classed to use the same variable several times which I made use of in my database class by using ***username*** variable to differentiate between users. I used conditions in my if-else and try-except statements to make sure that the command happens only when a specific event occurs. I used my conditional statements and paired them with AND and OR to set several conditions in one line, which made it easier for readability and for designing the program for me later. I tried making the names as clear and as relevant as possible, I also tried adding comment lines after every major part of my code so anyone reading it can understand what I did and why. I used more than one python library that proved essential to the project I had mind, I mentioned them above in both the Requirement part and some of them in the Design and Implementation part.

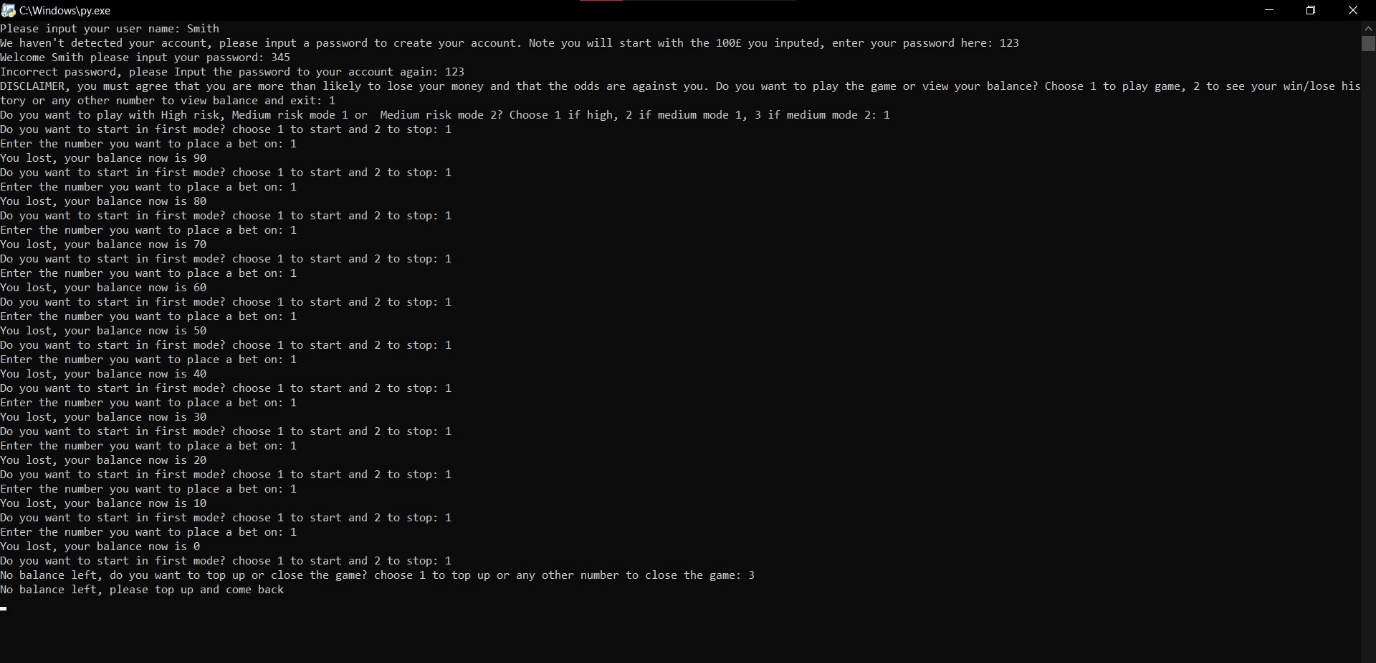
**Instructions:**

When you open the app, you get to write your username which gets used later as explained above. After writing it you get to register if your username is not found, and login if your username exists, in the former you get to choose a password and input it again, in the latter you get to input your password, if it is wrong then you get to type it three times, after that you are asked to check with the casino’s administration. Then you choose if you want to play or look at your balance, and later down the line, you can choose to see your winning and losing history. If you choose to play you get to choose the mode you want to play from the 3 mentioned above, then you get to choose the number, or group of numbers you want to bet on, you see your balance after you make your choice to see if you won or lost. After that you get to choose again if you want to play the game in the same mode, if you choose not to you will get to choose if you want to play, see your balance and exit the game, or see your history for that session.

**Testing and Maintenance:**

**Text

Description automatically generated**I tested the program several times under several circumstances myself, I had a few problems and I was able to fix most of them, I also asked some friends to try the program and asked them to tell me if they faced anything and they said it worked without any crashes or any unexpected behaviour. When I debugged it, myself I used my name as username and 123 as password, I tried all the modes till my balance became Zero to try the top up option as well. I also used visual studio debug mode which helped me figure out if I forgot something. The requirements mentioned in the criteria are all met in my program, with a few more added like the Authentication system for example. I included two instances of me debugging the program myself to see if it works as intended or not.

****

**Reflection and Next steps:**

Overall, this project was a success, I used python OOP basics and applied it in a manner it worked in, I also learnt to use a lot of important python modules and familiarised myself with them. I used a database which ended up causing a few issues that took time and a deep understanding of both, the module and python logic to fix, but I am certain it will prove beneficial in my programming career as databases are an integral part of the most important programs nowadays. Another issue I had as well was determining if I need a queue or a stack, I had to examine the language used in the assessment to see which one to choose and I decided on using a queue. I mainly used Rapid application development (RAD) as my development process depended on a lot of prototyping which offered me fast seen progress, I also cut the work I needed to do into several segments as well to help me organise my time, this is not part of RAD, but I found it beneficial nevertheless. A few ways to improve the project would be to make the balance constant through sessions, as in it does not reset to a 100£ if you close and open the game again, I will also add GUI using tkineter, I wanted to add it, but I decided that it was too much work especially with my exams around the corner.

**References / Open-Source Declarations:**

Matthes, E. (2019) *Python Crash Course, 2nd Edition : A Hands-On, Project-Based Introduction to Programming*. 2nd ed. [Online] No Starch Press: San Francisco, CA. Available from: [https://ebookcentral.proquest.com/lib/warw/detail.action?docID=6063209#](https://ebookcentral.proquest.com/lib/warw/detail.action?docID=6063209) (Accessed 01 May 2021).