**Design**

**User interface Design:**

**A screenshot of a computer program

Description automatically generated**



1. Listbox 1 will display ONLY leaderboard rankings
2. A button to run the bubble sort which will make the leaderboard listbox display the rankings
3. The Listbox 2 will display error messages and when the user gets a question right all messages that don’t require a user input right after will go in this box
4. A button that starts the quiz checking if the user is logged in and if so then asks what level they want to play or if they would like to sign out
5. The login button will call upon methods that will control the whole login system once logged in the user will then play the game
6. This button checks if the user is logged in and if so it deletes the users information on the database and puts the new users level and details in its place
7. The registration button will call methods that asks the user their account details and inputs that into the database if there isn’t more than 10 users
8. The sign out button will simply set the login variable to False meaning the user cannot continue with the game by pressing the start quiz button
9. The username listbox will show the users username once logged in
10. The level at login listbox will show the users level once logged in

**All user input that is NOT a button will be through input boxes and they will look like**

**This will pop up everytime user input in the form of not a button shows up**

A computer screen shot of a box

Description automatically generated

1. This is where the user will input all inputs besides button clicks
2. The ok button serves as an enter button
3. The cancel button will be used to close the inputbox
4. The question for the user does not just mean quiz questions but simply if they would like to continue level or what their password and username is ect.

**The array of records pseudocode**

For this section of the design phase, I will think about what variables will be held in my Array of Records.

Declare Structure userInformation

Declare id As String

Declare username As String

Declare password As String

Declare level As String

End Structure

I will then declare the array

Declare allUsers(9) As userInformation

**Procedural Programming**

Throughout my program, I will use multiple methods that will add functionality. I will have 1 main method that calls upon all the other methods here is where I will declare my variables for the logged-in account and all the user details from the database.

**Bubble Sort algorithm pseudocode:**

Method bubbleSort(pass by reference users() As userInformation)

Declare swaps As boolean = True

Declare outer As integer = 8

Declare placeHolder As userInformation

Loop while swaps = True and outer >= 0

swaps = False

loop inner from 0 to outer

if users(inner).level > users(inner + 1).level then

placeHolder = users(inner + 1)

users(inner + 1) = users(Inner)

users(inner) = placeHolder

swaps = true

end if

end for loop

Subtract 1 from outer

End while loop

Listbox1(highest scoring username and their level)

for x As integer = 0 to 8 step 1

if users(x).level = users(9).level then

display(usernames of users tied with the highest scoring user)

end if

end for

end method

**Integration Pseudocode**

**Read in database Connection**

Method readDatabaseConnection(pass by reference users() As userInformation)

Declare counter As integer = 0

Try

Declare SQLReader As OleDbDataReader

Declare connectionType As string = "Provider=Microsoft.ACE.OLEDB.12.0;"

Declare fileLocation As string = source

Declare conn As OleDbConnection

conn = New OleDbConnection(connectionType & fileLocation)

conn.Open()

Declare query As string = "SELECT \* FROM userInformation"

Declare command As New OleDbCommand(query, conn)

SQLReader = command.ExecuteReader

If SQLReader has rows then

while SQLReader is reading

users(counter).User ID = SQLReader(User ID FIELD)

users(counter).username = SQLReader(Username FIELD)

users(counter).pasword = SQLReader(Password FIELD)

users(counter).level = SQLReader(Level FIELD)

add 1 to counter

end while

else

Display(no results message)

End if

Catch error As exception

Listbox2(error)

End try

End method

**Save Account Connection**

This is a method the user will call upon to save their account this will be done by clicking the save button.

Method saveAccount(pass by value user As userInformation, pass by value users() As userInformation)

Try

Declare SQLReader As OleDbDataReader

Declare connectionType As String = "Provider=Microsoft.ACE.OLEDB.12.0;"

Declare fileLocation As String = source

Declare conn As OleDbConnection

conn = New OleDbConnection(connectionType & fileLocation)

conn.Open()

Declare query As String = "INSERT INTO userInformation (User ID, Username, Password, Level) VALUES (" & user.id & ", '" & user.username & "', '" & user.password & "', " & user.level & ");"

Declare query2 As String = "DELETE FROM userInformation WHERE username = '" & user.username & "';"

Declare command2 As New OleDbCommand(query2, conn)

SQLReader = command2.ExecuteReader()

Declare command As New OleDbCommand(query, conn)

SQLReader = command.ExecuteReader()

Catch error As Exception

ListBox2(error)

End Try

End Method

**Registration Connection**

method registrationDatabaseConnection(pass by value tempUsername as string, pass by value tempPassword as string, pass by value user As userInformation, pass by value max As integer)

If max <= 10 Then

Try

Declare SQLReader As OleDbDataReader

Declare connectionType As String = "Provider=Microsoft.ACE.OLEDB.12.0;"

Declare fileLocation As String = source

Declare conn As OleDbConnection

conn = New OleDbConnection(connectionType & fileLocation)

conn.Open()

Declare query As String = "INSERT INTO userInformation (User ID, Username, Password, Level) VALUES (" & (max + 1) & ", '" & tempUsername & "', '" & tempPassword & "', " & 0 & ");"

Declare command As New OleDbCommand(query, conn)

SQLReader = command.ExecuteReader()

user.id = max + 1

user.username = tempUsername

user.password = tempPassword

user.level = 0

login = True

Catch error As Exception

ListBox2.Items.Add(error)

End Try

ElseIf max > 10 Then

ListBox2(message saying too many accounts)

End If

End method

**Entity Relationship Diagram**

As I only have one table this is not required

**Query Design**

**Create Account Query**

|  |  |
| --- | --- |
| INSERT | Max id + 1, user entered Username, user entered Password, Level(default 0) |
| TABLE | The userInformation table |
| VALUES | User ID, Username, Password, Level (every field that is in my database) |

**Login Query**

|  |  |
| --- | --- |
| SELECT | Everything in the table using \* |
| FROM | The userInformation table |

**Delete Query**

|  |  |
| --- | --- |
| DELETE | Whole record |
| FROM | The userInformation table |
| WHERE | Username from username field = the logged in users username |

**Data Dictionary:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UserInformation Table** |  |  |  |  |
| **Field name** | **Field Type** | **Field Size** | **Key** | **Required** |
| User ID | AutoNumber |  | Primary Key | Yes |
| Username | Short text(8) | 8 |  | Yes |
| Password | Short text(8) | 8 |  | Yes |
| Level | Int |  |  | Yes |