**Analysis**

**Project Description**

The program I will be creating will consist of a quiz that is software-based integrating with a database. The quiz will have 3 levels and once the user completes one level, they will unlock the next. As I intend for my users to have accounts this will be important as I will store the user's progression in a database and retrieve account data upon request. When the user starts the program a main screen in which the user logs in or registers for an account or views the leader board will appear. There will also be extra functionality for a user to save their account and sign out. The quiz questions will consist of what the capitals of various countries are. The audience of my game will be any young adult regardless of what they do or have studied with common sense.

My Project is suitable to the specifications set by you as my main technology is software development, and my secondary technology will be database design and development. The software aspect will be processing the data such as account information and will deal with inputs and outputs. my database will integrate with the software to store this account information, my 2 advanced higher concepts are the following: Concept 1 will be procedural programming with an array of records and Concept 2 will be using bubble sort. For my software, I will be using Visual Basic as my coding language. I will be using and interacting with my database through the software. I will be using SQL to execute queries once I establish a connection to the database. The queries will be used to save the users level, read in the database table, and register an account. I will use the bubble sort algorithm to filter through all the user’s accounts and find the highest-scoring users and if any users tie with the highest-scoring user they will be recognised, these results will be displayed on the leaderboard. A user will not be able to play the game unless logged in, though they will be allowed to use the leaderboard feature.

**Scope**

**(while not in the specification I felt this was necessary)**

The scope of my project will be.

1. A completed design with pseudocode for the adv h concepts and the integration and a data dictionary along with the query design.

2. test plan with a persona tested

3. fully working program integrated with a database that will store user data

4. completed test data represented appropriately

5. detailed evaluation report focusing on concepts and functional requirements

**Constraints**

The constraints my project will have will be technical, time and economic, there will be no legal:

1. My knowledge and understanding of SQL will be limited so not having done it before I will need to adjust my time accordingly to learn it.
2. My final project will run on the Visual Studio 2022 test environment and will use Microsoft Access 2022 to run my database.
3. I will ensure that my project is done and handed into my Centre by the 28th of March.
4. As for technical constraints, I will have to use laptops funded by the school as a MacBook (the computer I have at home) does not support Microsoft access this will heavily impact development and will need to be accounted for in terms of time.
5. There will be no costs as all the software I will be using for the project will be free or licensed by the school.
6. There will be no legal constraints as my product is 100% original and does not use any produced material.

**Boundaries**

**(again while not in the specification I felt this was necessary)**

Once the project is complete it will contain:

1. An input validation system where the password the user enters in the sign-up phase is validated.
2. My quiz will consist of 3 levels which the user can select to start from after completion.
3. The user must not be able to play the quiz without logging in.
4. A sign-in system where the user enters their username and password, and it is checked if those are the same as any account info on the database.
5. A fully integrated project in which the database and the software work together to output the end game.
6. User account numbers must not exceed 10.

**UML Use Case Diagram**

A diagram of software application

Description automatically generated

**Requirements Specification**

Project purpose: The purpose of this application is for a user to simply play a game and be able to create an account to save the level the user has completed; After this they should be able to view their score on the leaderboard this will be done by the user creating a username and password.

**Functional Requirements**

* The Software Application must establish a connection to the database before executing queries.
* The Software Application will need to read and write account information this will be done by using SQL to execute a query and import the data into the application upon login and registration and the leaderboard system.
* All User Inputs MUST be validated whether this is answers to questions or account information.
* The game must not be playable without logging in, login is necessary to play whether logging into an existing account or registration.
* Once a user inputs their account information this must be validated by checking the character counts for usernames and passwords are 8 characters.
* Upon registration of an account, the database connection must be opened, and the registration info must be written into the database.
* Once data is imported into the application a bubble sort algorithm will be used to find the highest-scoring users.
* A level must not be played unless that level or the previous level has been completed.
* When validated that the user information matches the login information the user must have the option to continue from the last level or play from the start.
* All user data must be stored in the database.
* When a user creates a new account, they must stay logged in with those registration details provided they have been validated.
* A user must be able to save an account once logged in.
* A user must only be able to sign out once logged in.

**End-User Requirements**

The end user requirements will be:

* Clutter-free and clear UI.
* An easy-to-navigate interface.
* Login systems requirements must be displayed.
* Simple fonts and sizes so that the user can interact easily.

**Inputs and outputs and processes**

The user inputs will be:

1. Username
2. Password
3. Whether they wish to log in or create an account or sign out or save account
4. Question Answers

The Software Applications processes will be:

1. Execute SQL queries.
2. Verify login information.
3. Save and create account information.
4. Connecting to the database
5. Validation of EVERY user input
6. Checking that the user’s answer to a level is the same as the answer
7. Retrieve the level after the user wishes to continue
8. Check a user’s level before allowing them to play it (this does not apply to level 1)
9. Using the bubble sort algorithm for the leaderboard
10. Logging out the account

The Inputs for the application will be:

1. Username information when creating or saving an account.
2. Password information when creating or saving an account.
3. Level information when creating or saving an account.

The outputs for the application will be:

1. Consistent error messages throughout the whole program (whether a user is trying to do something they need to be logged in for or if the database connection fails)
2. A message saying the answer is correct or incorrect.
3. The question for the level they are at.
4. Login and create account prompts and confirmation of login.
5. The leaderboard.
6. Current logged-in username and level once logged in.

**Resources Required**

Creating my project will require me to have access to the following software:

* + Microsoft Word (latest) for documentation
  + Visual Studio (2022) for coding
  + Microsoft Access (2012) for the user database
  + Google Chrome (latest) for the UML use case diagram

**Feasibilities**

In terms of feasibilities, there will be little to no legal, economic, or technical feasibilities. As my project is an educational project and there will be no money made this should void legal issues. As my project will not be implemented on a large scale simply having a USB and a laptop is all I will need to carry this project out so there will be no economic feasibilities. As for time feasibility, I will be limited on how long I have for the project, and I will have to rigorously plan my time. I will show this by using the table below.

The table below goes off weekdays these days do not need to be during school hours as the whole project is self-study meaning no classes will be supplied by the school nor is there an advanced higher computing teacher. This allows me to work whenever (provided self-made deadlines are met) and I will be starting the project later and be more disciplined to complete it quicker because of this.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Main Task | Sub Task | Duration | Start of Task | End of task |
|  | Plan Project Idea and Read Specification | 1 day | 18th Jan | 18th Jan |
|  | Outline Problem and Identify Adv H Concepts and Integration | 0.5 days | 19th Jan | 19th Jan |
|  | Identify constraints | 0.5 days | 19th Jan | 19th Jan |
|  | Generate End -User Requirements Specification | 0.5 days | 22nd Jan | 22nd Jan |
| Analysis | Generate Functional Requirements  Specification | 0.5 days | 22nd Jan | Jan |
|  | Create UML Use Case Diagram | 1 day | 23rd Jan | 23rd Jan |
|  | Identify Resources and Timings and Identify Tasks | 1 day | 24th Jan | 24th Jan |
|  | Project Plan | 1 day | 25th Jan | 25th Jan |
|  | Design of Adv H Concepts Pseudocode | 5 days | 26th Jan | 1st Feb |
|  | Design of Integration (excluding queries) | 4 days | 2nd Feb | 7th Feb |
| Design | User interface design | 3 days | 8th Feb | 12th Feb |
|  | Learn about and design SQL queries | 3 days | 13th Feb | 15th Feb |
|  | Start Implementing leaderboard system and general methods along with record structure | 5 days | 16th Feb | 22nd Feb |
| Implementation | Implement database connections and create database table | 7 days | 23rd Feb | 4th Mar |
|  | Implemented UI | 3 days | 5th Mar | 7th Mar |
|  | Description of new Skills and or Knowledge | 1 days | 8th Mar | 8th Mar |
|  | Log of Ongoing Tests | This will be logged throughout implementation |  |  |
|  | Full detailed test plan | 3 days | 11th Mar | 13th Mar |
|  | Persona testing and requirements testing | 1 day | 14th Mar | 14th Mar |
| Testing | Evidence of test plan completed | 2 days | 15th Mar | 18th Mar |
|  | Document problems faced by testing | 3 days | 19th Mar | 21st Mar |
|  | Evaluation of fitness for purpose | 1 days | 22nd Mar | 22nd Mar |
| Evaluation | Evaluation of Robustness and Maintainability | 2 days | 25th Mar | 26th Mar |
| Gather Evidence | Simply print documents | 1 day | 27th Mar | 27th Mar |

**Design**

**User interface Design:**

**A screenshot of a computer program

Description automatically generated**



1. List box 1 will display ONLY leaderboard rankings.
2. A button to run the bubble sort which will make the leaderboard list box display the rankings.
3. The List box 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 user’s information on the database and puts the new user’s 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 aren’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 list box will show the users username once logged in.
10. The level at login list box 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 every time 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 input box.
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 etc..

**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 buttons that calls upon other methods here I will make use of local variables and global variables.

**Bubble Sort algorithm pseudocode:**

Method bubbleSort reference users As userInformation

Loop while swaps = True and outer is not negative

swaps = False

loop from 0 To outer

If users level is greater than users + 1 level Then

placeholder = users inner + 1

users inner + 1 = users

users inner = placeholder

swaps = true

end if

end fixed loop

Subtract 1 from outer

End while loop

Display max value of array username and their level

loop from 0 to 8

if a users level is the same as the highest scoring users level then

display usernames of users tied with the highest scoring user

end if

end loop

end method

**Integration Pseudocode**

**Read in database Connection**

Method readDatabaseConnection reference users As userInformation

Try

Declare SQLReader As OleDbDataReader

Declare connection type As string = OleDb provider

Declare file Location As string = source

Declare conn As OleDbConnection

conn = New OleDbConnection(connection type & file location)

Open connection

Declare query As string = Select all from userInformation

Declare command As New OleDbCommand(query, conn)

SQLReader = executed command

If SQLReader has rows that can be read then

Loop when SQLReader is reading

users User ID = User ID FIELD

users username = Username FIELD

users password = Password FIELD

users level = Level FIELD

add 1 to counter

end loop

else

Display no results message

End if

Catch error

Display the 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 value user As userInformation value users() As userInformation

Try

Declare SQLReader As OleDbDataReader

Declare connection type As String = OleDb provider

Declare file Location As String = source

Declare conn As OleDbConnection

conn = New OleDbConnection(connection type & file location)

open connection

Declare query As String = Insert into userInformation fields(User ID, Username, Password, Level) populate fields with values user id & user username & user password & user level

Declare query2 As String = Delete row where username = username field at table userInformation

Declare command2 As New OleDbCommand(query2, conn)

SQLReader = command executed

Declare command As New OleDbCommand(query, conn)

SQLReader = command executed

Catch error

Display 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 is less than 11 Then

Try

Declare SQLReader As OleDbDataReader

Declare connection type As String = OleDb provider

Declare file Location As String = source

Declare conn As OleDbConnection

connection = New OleDbConnection(connection type & file location)

open connection

Declare query As String = Insert into userInformation fields(User ID, Username, Password, Level) populate fields with values (max + 1 & tempUsername & tempPassword & 0)

Declare command As New OleDbCommand(query, connection)

SQLReader = command executed

user username = tempUsername

user password = tempPassword

user level = 0

login = True

Catch error

Display error message

End Try

ElseIf max is more than the maximum amount of accounts Then

Display 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 |

**Implementation**

**Source code**

Imports System.Data.OleDb

Imports System.Runtime.Intrinsics.X86

Imports Microsoft.VisualBasic.ApplicationServices

Imports Microsoft.VisualBasic.Logging

Public Class Form1

Dim user As userInformation 'records for users

Dim allUsers(9) As userInformation

Dim tempId As String = "0" 'variables used for registration

Dim tempUsername As String = ""

Dim tempPassword As String = ""

Dim login As Boolean = False 'variables for program operation

Dim max As Integer = 0

Dim optionInputted As String = ""

Dim loginCreate As Integer = 0

Dim locked As Boolean

Structure userInformation 'define a record structure that will store all users information from the database table along with the current logged in user

Dim id As String

Dim username As String

Dim password As String

Dim level As String

End Structure

Sub startProgram()

If login <> True Then 'this if statement is used to verify if the user is logged in or not

ListBox2.Items.Add("login to an existing account or create a new account to play the main game")

ElseIf login = True Then

homeScreen(user, allUsers, login)

Else

ListBox2.Items.Add("error")

End If

End Sub

Sub findHighestId(ByVal users() As userInformation, ByRef max As Integer, ByRef locked As Boolean)

For y = 0 To 9 Step 1 'loop ends at 10 as if there is an 11th id it will account for this

If users(y).id > max Then

max = users(y).id

End If

Next

If max = 11 Then 'simply checks if there is more than 10 users and if there is displays an error message

MsgBox("there is too many users")

locked = True

ElseIf max <= 10 Then

MsgBox("there is " & max & " users")

locked = False

End If

End Sub

Sub inputAccountInfo(ByRef Password As String, ByRef Username As String) 'this method is used for any inputs of user information this will validate whether any usernames or passwords entered into the program are 8 characters

Dim sign As Boolean ' variable to keep loop if username and password is not 8 characters

Dim characterCount1 As Integer 'variables to hold the character count for username and passwords

Dim characterCount2 As Integer

While sign <> True

Username = InputBox("please enter your username")

Password = InputBox("please enter your password")

characterCount1 = Len(Password)

characterCount2 = Len(Username)

If characterCount1 = 8 And characterCount2 = 8 Then

sign = True

MsgBox("password and username is 8 characters")

ElseIf characterCount1 <> 8 And characterCount2 <> 8 Then

sign = False

MsgBox("password and username is not 8 characters long")

ElseIf characterCount1 = 8 And characterCount2 <> 8 Then

sign = False

MsgBox("username is not 8 characters")

ElseIf characterCount1 <> 8 And characterCount2 = 8 Then

sign = False

MsgBox("password is not 8 characters")

Else

MsgBox("error")

End If

End While

End Sub

Sub homeScreen(ByRef user As userInformation, ByVal allUsers() As userInformation, ByRef loginStatus As Boolean) 'this method manages what level is being played and is a homescreen for signing out and for selecting a level and signing out

Dim desiredLevel As String

Dim userInput1 As String

Dim x As Boolean = False

userInput1 = InputBox("would you like to select a level or play from the start or sign out")

While x <> True

If userInput1 = "select" Then

desiredLevel = InputBox("which level would you like to continue from?")

If desiredLevel = "1" Then

level1(user, allUsers)

x = True

ElseIf desiredLevel = "2" Then

If user.level >= 1 Then

level2(user, allUsers)

x = True

ElseIf user.level < 1 Then

ListBox2.Items.Add("you have not unlocked that level yet")

Else

ListBox2.Items.Add("error")

End If

ElseIf desiredLevel = "3" Then

If user.level >= 2 Then

level3(user, allUsers)

x = True

ElseIf user.level < 2 Then

ListBox2.Items.Add("you haven't unlocked that level yet")

Else

ListBox2.Items.Add("error")

End If

ElseIf desiredLevel > 3 Or desiredLevel < 1 Then

MsgBox("there's only 3 levels")

End If

ElseIf userInput1 = "start" Then

level1(user, allUsers)

ElseIf userInput1 = "sign out" Then

ListBox2.Items.Add("please click the sign out button")

x = True

Else

ListBox2.Items.Add("error")

x = True

End If

End While

End Sub

Sub verifyLogin(ByVal users() As userInformation, ByVal userPassword As String, ByVal userUsername As String, ByRef login As Boolean, ByRef user As userInformation) 'this method is used to verify whether the username and password entered by the user is equal to any of the usernames and passwords for

For x As Integer = 0 To 9 Step 1

If users(x).username = userUsername And users(x).password = userPassword Then

login = True

user.id = users(x).id

user.username = users(x).username

user.level = users(x).level

ListBox3.Items.Add("hello " & user.username & "!")

ListBox4.Items.Add(user.level)

End If

Next

If login = True Then

MsgBox("login was verified!")

ElseIf login = False Then

MsgBox("login was not verified please enter your username and password again")

End If

End Sub

Sub bubbleSort(ByRef users() As userInformation)

Dim swaps As Boolean = True

Dim outer As Integer = 8

Dim placeHolder As userInformation

While swaps = True And outer >= 0

swaps = False

For inner = 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

Next

outer = outer - 1

End While

ListBox1.Items.Add("the player with the highest score is " & users(9).username & " and their score is " & users(9).level)

For x As Integer = 0 To 8 Step 1

If users(x).level = users(9).level Then

ListBox1.Items.Add(users(x).username & " also tied with " & users(9).username)

End If

Next

End Sub

Sub registrationDatabaseConnection(ByVal tempUsername As String, ByVal tempPassword As String, ByRef user As userInformation, ByVal max As Integer)

If max <= 10 Then

Try

Dim SQLReader As OleDbDataReader

Dim connection\_type As String = "Provider=Microsoft.ACE.OLEDB.12.0;"

Dim file\_location As String = "Data Source=C:\Users\sahil\Desktop\userInformationDatabase.accdb"

Dim conn As OleDbConnection

conn = New OleDbConnection(connection\_type & file\_location)

conn.Open()

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

Dim 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 ex As Exception

ListBox2.Items.Add(ex.Message)

End Try

ElseIf max > 10 Then

ListBox2.Items.Add("there's too many users no new account can be made")

End If

End Sub

Sub saveAccount(ByVal user As userInformation, ByVal users() As userInformation) 'this method is used to save an account by deleting the row and then inserting the information back into the row with the new level

Try

Dim SQLReader As OleDbDataReader

Dim connection\_type As String = "Provider=Microsoft.ACE.OLEDB.12.0;"

Dim file\_location As String = "Data Source=C:\Users\sahil\Desktop\userInformationDatabase.accdb"

Dim conn As OleDbConnection

conn = New OleDbConnection(connection\_type & file\_location)

conn.Open()

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

Dim query2 As String = "DELETE FROM [userInformation] WHERE username = '" & user.username & "';"

Dim command2 As New OleDbCommand(query2, conn)

SQLReader = command2.ExecuteReader()

Dim command As New OleDbCommand(query, conn)

SQLReader = command.ExecuteReader()

Catch ex As Exception

ListBox2.Items.Add(ex.Message)

End Try

End Sub

Sub readDatabaseConnection(ByRef users() As userInformation)

Dim i As Integer = 0

Try

'connection to the database executing the query and reading in the data

Dim SQLReader As OleDbDataReader

Dim connection\_type As String = "Provider=Microsoft.ACE.OLEDB.12.0;"

Dim file\_location As String = "Data Source=C:\Users\sahil\Desktop\userInformationDatabase.accdb"

Dim conn As OleDbConnection

conn = New OleDbConnection(connection\_type & file\_location)

conn.Open()

'example 1 - select and display results()

Dim query As String = "SELECT \* FROM [userInformation]"

Dim command As New OleDbCommand(query, conn)

SQLReader = command.ExecuteReader

If SQLReader.HasRows Then

While SQLReader.Read

users(i).id = SQLReader("User ID")

users(i).username = SQLReader("Username")

users(i).password = SQLReader("Password")

users(i).level = SQLReader("Level")

i = i + 1

End While

Else

ListBox2.Items.Add("no results returned")

End If

Catch ex As Exception

ListBox2.Items.Add(ex.Message)

End Try

End Sub

Sub level1(ByRef user As userInformation, ByVal users() As userInformation) 'level 1

Dim userAnswer As String

Dim levelComplete As Boolean

Dim [continue] As String

While levelComplete <> True

userAnswer = InputBox("what is the capital of France?")

If userAnswer = "Paris" Or userAnswer = "paris" Then

levelComplete = True

ListBox2.Items.Add("correct!")

If user.level < 1 Then

user.level = 1

End If

ElseIf userAnswer <> "Paris" Or userAnswer <> "paris" Then

levelComplete = False

ListBox2.Items.Add("incorrect")

Else

ListBox2.Items.Add("error")

End If

End While

[continue] = InputBox("would you like to continue?")

If [continue] = "yes" Then

level2(user, users)

ElseIf [continue] = "no" Then

homeScreen(user, users, True)

Else

ListBox2.Items.Add("error")

homeScreen(user, users, True)

End If

End Sub

Sub level2(ByRef user As userInformation, ByVal users() As userInformation) 'level 2

Dim userAnswer As String

Dim levelComplete As Boolean

Dim [continue] As String

While levelComplete <> True

userAnswer = InputBox("what is the capital of England")

If userAnswer = "London" Or userAnswer = "london" Then

levelComplete = True

ListBox2.Items.Add("correct!")

If user.level < 2 Then

user.level = 2

End If

ElseIf userAnswer <> "London" Or userAnswer <> "london" Then

levelComplete = False

ListBox2.Items.Add("incorrect")

Else

ListBox2.Items.Add("error")

End If

End While

[continue] = InputBox("would you like to continue?")

If [continue] = "yes" Then

level3(user, users)

ElseIf [continue] = "no" Then

homeScreen(user, users, True)

Else

ListBox2.Items.Add("error")

homeScreen(user, users, True)

End If

End Sub

Sub level3(ByRef user As userInformation, ByVal users() As userInformation) 'level 3

Dim userAnswer As String

Dim levelComplete As Boolean

While levelComplete <> True

userAnswer = InputBox("what is the capital of Scotland")

If userAnswer = "Edinburgh" Or userAnswer = "edinburgh" Then

levelComplete = True

ListBox2.Items.Add("correct!")

If user.level < 3 Then

user.level = 3

End If

ElseIf userAnswer <> "Edinburgh" Or userAnswer <> "edinburgh" Then

levelComplete = False

ListBox2.Items.Add("incorrect")

Else

ListBox2.Items.Add("error")

End If

End While

MsgBox("You have completed the game! the program will now restart you will be signed out")

saveAccount(user, users)

End Sub

End Sub

Private Sub Button5\_Click\_1(sender As Object, e As EventArgs) Handles Button5.Click 'Leaderboard Button

Dim users(9) As userInformation

readDatabaseConnection(users)

bubbleSort(users)

End Sub

Private Sub Button4\_Click\_1(sender As Object, e As EventArgs) Handles Button4.Click 'Save button

If login = True Then

saveAccount(user, allUsers)

ElseIf login = False Then

ListBox2.Items.Add("you are not logged in")

End If

End Sub

Private Sub Button2\_Click\_1(sender As Object, e As EventArgs) Handles Button2.Click 'Login Button

inputAccountInfo(tempPassword, tempUsername) 'this is where the user inputs the password and username and the program verifies they are both 8 characters

readDatabaseConnection(allUsers)

verifyLogin(allUsers, tempPassword, tempUsername, login, user) 'this is the method that checks the 10 accounts imported in from the database and compares them to what the user has inputted as their password and username in the login process

login = True

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click 'Registration button

inputAccountInfo(tempPassword, tempUsername)

readDatabaseConnection(allUsers)

findHighestId(allUsers, max, locked)

If max < 10 Then

registrationDatabaseConnection(tempUsername, tempPassword, user, max)

ListBox3.Items.Add("hello " & user.username & "!")

ListBox4.Items.Add(user.level)

ElseIf max >= 10 Then

ListBox2.Items.Add("error too many accounts")

End If

End Sub

Private Sub Button6\_Click\_1(sender As Object, e As EventArgs) Handles Button6.Click 'Sign out button

If login = False Then

MsgBox("you weren't signed in")

ElseIf login = True Then

login = False

ListBox2.Items.Add("you have logged out")

End If

End Sub

Private Sub Button1\_Click\_1(sender As Object, e As EventArgs) Handles Button1.Click 'start quiz button

startProgram()

End Sub

End Class

**Ongoing Testing Document**

**Error 1**

When testing the database connection, the library would not import, and I could not use the OleDb classes.

The error looked like this:



This meant that every reference to an OleDb datatype was not installed meaning I would need to install the packages or find another way to establish a database connection.

I then researched the problem online and could not find a solution I turned to my computing teacher for this, and he did not help either. I was worried I would not find a solution to this problem and thought it would take too much time to find another way and implement it, so I stuck with OleDb.

I had contacted a tester at a company called the craneware group in Edinburgh, this is where I met Phil. Phil was a tester who managed to help me with the library issue.

Phil sent me this YouTube video showing me how to solve the issue:

<https://www.youtube.com/watch?v=yBJVHBaEtc8>

I had to install the NuGet package called “System.Data.OleDb”, I installed this package version 8.0 and my error messages disappeared like that!

Here is also when Phil introduced me to the idea of unit testing. Unit testing is the idea of isolating a method and testing it to show that it works. I used this throughout the whole project, and it was ridiculously useful. Isolating a method allows you to check if it works truly.

This Error was detrimental to the project, before speaking to Phil I had contacted my teacher and spoke with the technicians to download the libraries and they were no help. I tried to research the problem and had failed I was eventually going to resort to another method of connection but then I emailed Phil. This problem alone took a couple of weeks away from the project and heavily halted development as I was trying to research a way myself instead of simply asking people.

**Error 2**

When testing the insert query, I came across an issue saying there was a syntax error in the insert query in the registrationDatabaseConnection method. Since I had never done databases before this project, I had no idea what I was doing. I learned about it and still could not figure out what went wrong. The code on line 196 was – Dim query As String = "INSERT INTO [userInformation] (User ID, Username, Password, Level) VALUES (" & (max + 1) & “, ‘” & tempUsername & " ', '" & tempPassword & "', " & 0 & ");"

Trying to find the issue with the code was a process that took me a while as I kept searching for syntax for insert statements in SQL. This is when I caved and contacted Phil again! Phil worked his magic again and he told me he needed to see my database and solution fully. I tried to send him this, but the email was being difficult. This is when I discovered GitHub, GitHub is a platform which many coders use to work collaboratively and helps with project management. I made an account, posted my project there and sent it to Phil. Ultimately this error taught me about GitHub and project management.

Phil told me that the error was a problem with reserved words in Access, Password and Level are both reserved keywords, to get around it you add []. This tiny issue was simply resolved by adding square brackets to the field names this cost me a few days.

The new code was Dim query As String = "INSERT INTO [userInformation] ([User ID], Username, [Password], [Level]) VALUES (" & (max + 1) & ", '" & tempUsername & "', '" & tempPassword & "', " & 0 & ");"

This new code had worked and had no errors after.

While these 2 errors may be minor, they were the only 2 significant errors I had, and they were both either syntax errors and issues with the packages installed. The smallest issue in computer science can make the biggest impact.

**Log of new Skills/Knowledge developed**

I would like to say that I didn’t learn what I intended to, but I learnt a lot about things I didn’t intend to. When it came to the database connection, I not only learnt about how to use queries basically but about reserved keywords and using wildcards like “\*”

I didn’t just learn about programming but rather what else I learnt was in terms of project management and communication. Using GitHub and emailing Phil taught me a lot about how communicating in the software development cycle works. After working with the folks at Craneware Plc they recognised the initiative I took and offered me a work experience program.

I had also learned about using forms to make a user interface in visual studio as I have never done this before I had only used the console as a user interface. This took time but was honestly necessary as a console user interface is ridiculous for the final iteration of a project.

**Testing**

**Comprehensive Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality to test | Time taken to test this | Tester | Issues |
| Basic playing the quiz | Unit testing as programmed | Me |  |
| Refreshing leader board | Unit testing as programmed | Me |  |
| Logging into an account | Unit testing as programmed | Me |  |
| Registering an account | Unit testing as programmed | Me | Syntax error with insert query stated in ongoing log |
| Saving account details | Unit testing as programmed | Me | Syntax error with insert query stated in ongoing log |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Time taken to test this | Tester | Issues |
| Reading quiz data into method | Unit testing as programmed | Me | Library not downloaded stated in ongoing log |
| Button to refresh leader board | Unit testing as programmed | Me |  |
| Level methods | Unit testing as programmed | Me |  |
| Inserting quiz data into database | Unit testing as programmed | Me | Syntax error with insert query stated in ongoing log |

**Requirements Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | User input | Output | Expected Output |
| Check if the user can play the game without logging in | Clicking the start quiz button |  | The program will display a message saying you must log in or create an account |
| Check user input for account input(same regardless of login or register button pressed) | The username “sahil123” and the password “sahil123” after clicking the login button        The username “sahil123” and the password “WRONG” after clicking the login button      The username “WRONG” and the password “sahil123” after clicking the login button | The output of entering username “sahil123” and password “sahil123”    The output of entering username “sahil123” and the password “WRONG”    The output of entering username “WRONG” and the password “sahil123” | The program will display a message saying which input isn’t 8 characters |
| Questions are validated correctly | I logged into the sahil123 account and played from the start it then asked me this    I responded with: A screenshot of a computer  Description automatically generated | The program then said this after I inputted the capital of France as “Paris”    It said that the answer was “correct” the previous message was there from before    It then asked me if I wanted to continue  A screenshot of a computer  Description automatically generated | The program will say the answer is correct if the answer is correct, it should then ask if I would like to continue from that level |
| Check registration info is written to database upon registration | I registered with the username “TEST1234” and the password “SQACOOL1” as these were both 8 characters | The program then told me how many users there were.    And then it showed that I had registered by showing the username and what level I am at login    I then opened up the database and went to the table here’s what I saw:  It registered the username and password perfectly and set the level to 0 as it should | The program should tell me how many users there is and then it will tell me that I have registered and logged in my showing me in the username list box.    It will update the database with my new registered information |
| User tries to sign out and save an account without being logged in | I started the program and did nothing but press the sign out button    after the program had outputted, I then closed the program.  I started it again and pressed the save account button.    after the program had outputted, I then closed the program | After I pressed sign out this is what had outputted  A screenshot of a computer error  Description automatically generated  After I pressed the save account button this is what had outputted | The program should regardless of button pressed display a message saying I was not signed in |
| Check a user’s username is the same as when they registered | Earlier when I had registered the account to check the account was written to the database I had registered with the username “TEST1234” and the password “SQACOOL1” I then logged into this account | After I logged in here’s what displayed    This shows that the username is remembered, and the username is the same as when I registered | The program will display in the listbox what the logged in username is and if the program has remembered my username, then it will show that in the listbox |
| Check a user with the highest level gets the position on the leader board | For testing purposes, I had then inputted various values into the database    After that I had then clicked the refresh leaderboard after starting the program | After I had clicked the button, here is what came up on the leaderboard  A screenshot of a computer  Description automatically generated  This clearly did what it was meant to and was a huge success! | The bubble sort should sort the highest user and whoever ties with them and then display their usernames |
| Check a user can’t make an account when there’s already 10 users | For testing purposes I needed to create 10 accounts and then try to make an 11th  Heres the 10 accounts    Next I tried to make the 11th account with the username ”testing1” and the password “notgonna” | Here’s what happened  After it told me how many users there were it then said  A blue and white screen with black text  Description automatically generated  The program would not continue as expected.    Here’s the database after    It didn’t change at all | The program MUST not write to the database if there is 10 users instead it will output a message saying there’s too many accounts |
| Check a user gets recognised for tying with the highest scorer | I pressed the refresh leaderboard button    Here was the database | It displayed in the leaderboard textbox jack1212 and suittie1 and woohoo12  This is exactly what should’ve happened and is the correct 3 that all tied together with a score of 3 | the program should find the highest users and say they are either the highest or tied with the highest |
| Not answering correctly | I then logged into the random12 user account and then continued and answered London to the question what is the capital of France | Here's what happened  The program listbox outputted this    And it then asked me the question again  A screenshot of a computer  Description automatically generated  This is exactly what I wanted it to do but this also raised the issue what happens if a user doesn’t know the answer to a question    In hindsight implementing a way for a user to exit a question would’ve been wise    Let’s say I still didn’t know it was paris and tried to exit by pressing the cancel button here’s what happens  A blue and white background with text  Description automatically generated  This is an extreme design flaw that should’ve been thought about at the requirements stage | The program should tell me the question is wrong and give me another chance to answer it |
| Check a user can continue from the last level they completed plus 1 | I logged into the TEST1234 account as its level was 2 this means I should be able to start from level 3  When prompted do I want to select a level I said yes, and it asked what level I wanted to start from I said 3 | Here’s what happened  A screenshot of a computer  Description automatically generated  It played the next level as expected | It should logically go to the next level as the last level TEST1234 had completed was level 2 |
| See what the program does when 2 highest scorers with the same name tie | For testing purposes, I changed the database so that jack was now woohoo12 the same username as another user with the same level    After that I started the program and pressed the refresh leaderboard button | Heres what happened  A blue and white box with black text  Description automatically generated  This is what was expected and there was no errors here | The program should just display a message saying they both tied with suittie1 |

**Persona test**

Persona- Michael is a 22-year-old university student who has an interest in fun little games like quizzes and puzzles he forgot about my quiz and hasn’t played it in a year! He didn’t ever complete it fully but remembers his username and password. When Michael played the game a year ago, he completed level 1 and stopped playing.

Michael remembers his username and password as the following:

Username - Michael1

Password - COOLquiz

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | User Input | Output | Expected Output |
| Michael continuing from after completing 1 level | After I entered the details it told me the characters for username and password were both 8 and then it said this    Meaning the login was verified I then tried to select level 2 | And it asked me    This was exactly what was expected that still after a year the account would be logged into and would be able to continue from level 2 | The account will have a level of 1 since he has already completed 1 and therefore, he will be allowed to continue from 2 |
| Michael saves his account after continuing | I entered London and then once I got the question right, I said no I don’t want to continue it was here that I pressed the save account button. | It had then saved Michael’s name and password in the database and his level was now 2 with no duplicates | Michael’s level should simply change to 2 |
| Michael forgets to login and clicks sign out | I clicked the sign out button after starting the program | Here’s what happened  A screenshot of a computer error  Description automatically generated  This is exactly what was expected | The program should simply tell Michael he wasn’t signed in so how you can sign out |
| Michael gets to level 3 and wants to see himself on the leader board | Michael is now level 3 and refreshes the leaderboard Here’s what the database looked like.  A screenshot of a computer  Description automatically generated | Michael is now on the leaderboard!  A blue and white box with black text  Description automatically generated  This is the expected outcome with him tied in the leaderboard | Michael should be on the leaderboard now with anybody else that is level 3 aswell |
| Michael starts from the start when logging in | Michael types “start” after clicking the start quiz button  A screenshot of a computer  Description automatically generated | Heres what happened  A screenshot of a computer  Description automatically generated  As expected, level 1 is what is outputted | It should start level 1 |

**Evaluation**

**Fitness for purpose**

My initial idea for this project was to create a software application that stores account information in the database. In my program I have created this by using SQL queries executed by my program when needed. The user can register for an account, play the game, and then log back into that account to have all their progress. The user can also use the leader board system backed by a bubble sort algorithm to filter for the highest scoring users in the database. This gives incentive for the user to keep the same account and use the login sign out save account system.

I believe my solution is fit for purpose as it meets all the requirements set in the requirements specification. Adding extra functionality to the project in terms of using emails to verify logins (otherwise known as 2FA) was always something I strived for as this is how literally everything works these days. As a result, from the extreme delays from the database connection and not having the support to fix this I then emailed a tester from a company, and he helped me. Working with Phil the tester took a while and was not as efficient as working with an advanced higher teacher also explaining further delays. I was stuck between either trying to quickly get an authentication system in and potentially rush testing and evaluation or to drop it. I removed this from the project.

I think that my end user test evidently shows that this project is fit for solution. The fact that a user can log in and have their level saved and the program output that shows that the login system works.

**Results of testing**

Through testing, my program does store the information in the database is retrieved upon request by use of SQL queries after connecting. Users can register an account complete a level or two and then log back into that account and play the same level that they logged out on. A user can also only play the game when logged in. User inputs are validated as shown for the login system, the register system the level system and you could also not sign out or save an account when not logged in. The username and password a user enter in is always 8 characters otherwise an error message is sent out. The queries for inserting and deleting and reading all work as they should. When user clicks save account, it deletes their account and then inserts, this was something that seemed tricky as I was executing 2 queries one after another and it worked. I checked that the leader board works by entering several different instances of databases and it outputted the right answer every time.

The major issue I found in testing was if a user doesn’t know the answer to a question, then how can they play the game, in future development this will need to be considered.

**Maintainability of the code**

The use of SQL queries in my code makes it extremely maintainable. If a developer wanted the query to do something else like sort the table, then they could do this. The use of an array of records also proves maintainability as if a developer wished to add another function such as a life bar, then they could do this by adding another variable to the structure.

**Robustness of code**

In terms of robustness my program is not at all robust as it could be, there isn’t a user input that is not validated and if the program has an error with a process such as reading data or a level on a database somehow being higher than 3 then processes are put in place to stop the program and say error. I wanted to add validation so that users could use different languages or characters but because of the time situation I wasn’t able to.