Streamer

Information Technology Practical Assessment Task – Phase 2

Mweya Ruider

# Database Structure

|  |  |  |
| --- | --- | --- |
| Titles | | |
| Field Name | Data Type | Field Size |
| MovieID | AutoNumber | from -2,147,483,648 to +2,147,483,647 |
| Title | Long Text | **4,294,967,295** |
| Thumbnail | OLE Object | n/a |
| OnDisk | Yes/No | n/a |
| Filename | Long Text | **4,294,967,295** |
| GenreID | Number | 0 to 255 |
| Length | Number | 0 to 255 |
| Director | Long Text | **4,294,967,295** |
| Cast | Long Text | **4,294,967,295** |
| Ratings | Number | 0 to 255 |

|  |  |  |
| --- | --- | --- |
| MovieRatings | | |
| Field Name | Data Type | Field Size |
| RatingID | AutoNumber | from -2,147,483,648 to +2,147,483,647 |
| UserID | Number | 0 to 255 |
| MovieID | Number | 0 to 255 |
| RatingValue | Number | 0 to 255 |

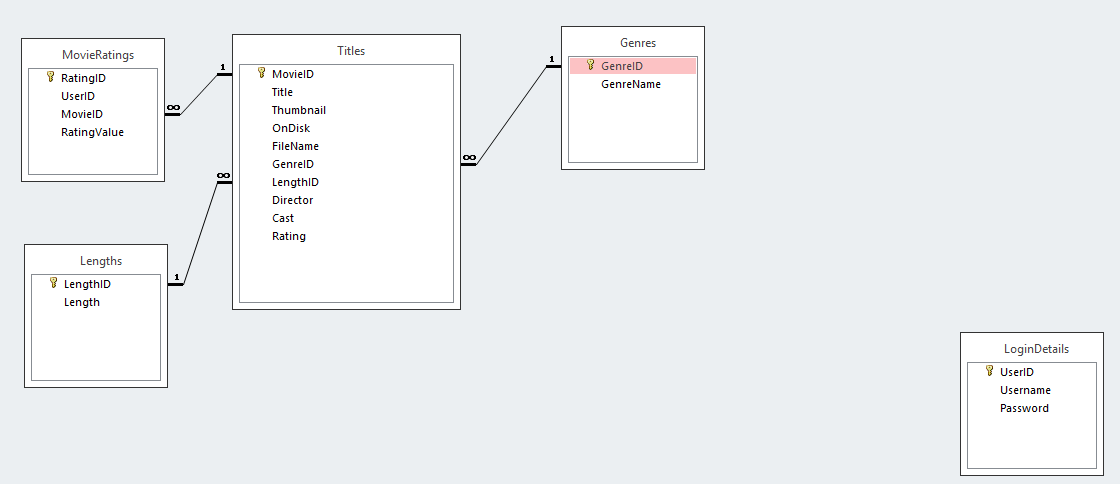
|  |  |  |
| --- | --- | --- |
| LoginDetails | | |
| Field Name | Data Type | Field Size |
| UserID | AutoNumber | from -2,147,483,648 to +2,147,483,647 |
| Username | Long Text | **4,294,967,295** |
| Password | Long Text | **4,294,967,295** |
| Dev | Yes/No | **n/a** |

|  |  |  |
| --- | --- | --- |
| Lengths | | |
| Field Name | Data Type | Field Size |
| LengthID | AutoNumber | from -2,147,483,648 to +2,147,483,647 |
| Length | Short Text | 255 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Genres | | |
| Field Name | Data Type | Field Size |
| GenreID | AutoNumber | from -2,147,483,648 to +2,147,483,647 |
| GenreName | Short Text | 255 |

Note that the field sizes are this big due to the fact that these are the default options and that there does not seem to be a justifiable reason to shorten them. (Error handling due to incorrect data being submitted can be handled in the Delphi code).

# Relationship Diagram (ERD)



# Function of the Database

The purpose of this database is to store information about videos on the system and to store usernames and passwords for login purposes (the idea is to eventually move authentication to an online solution to make it easier to manage accounts, but that might be out of the scope of this project). As soon as the program is started, the program should crawl the computer for videos before indexing them in the database. The program should then read through the list of videos stored locally and display them on the Browse Movies screen. The Browse Movies screen should only be accessible if the user is logged in and if the database has been refreshed relatively recently. The database should store information on the genres of the movies, the titles of the movies, if the movies are stored on the disk or not, a thumbnail for the movies. The database should also keep a record of the ratings of the individual ratings of the movies so an average could be determined. All tables except for the username table have been linked so that a list of movies that satisfy a certain requirement (Like all movies action movies for example) can be selected to be browsed

# Input Process Output Chart

|  |  |  |
| --- | --- | --- |
| **Input** | **Processing** | **Output** |
| User clicks on Log In | Program makes Login related panels visible | Login dialog is shown |
| User submits login data | Program checks to see if the user is correct | Caption changes to greet user. |
| User tries to log in again | Program checks if user is already logged in. | A warning message is displayed. The error is logged. |
| User clicks on Browse Movies | If the user is logged in, the program queries the database for a movie to display. | Program displays the movies downloaded on the Browse screen. |
| User clicks on Help | Program displays a dialog to ask the user if the user would like to view the readme site | The program either closes the dialog (if cancelled) or takes the user [here](https://mweya.ddns.net/Portfolio/Streamer/readme.html). |
| User clicks on Log out | Program checks to see if the user is logged in or not. If the user is logged in, the program sets the user state to logged out | If the user was logged in, the user is logged out and the greeting caption is reset. Otherwise an error message is displayed and the error is logged. |
| User clicks on Log out again | Program checks if user is already logged out | A warning message is displayed. The error is logged. |
| User clicks on Exit | Program frees the database, deletes the lock file and terminates itself | Program closes |
| User presses Esc | Program force quits by deleting the lock file and terminating itself without freeing the database | Program closes |
| User clicks on Logo | Program opens my site with default internet browser | My website is opened |
| User clicks Left/Right shoulder buttons | Program cycles between movies by reading through the database | New movie title and matching thumbnail and miscellaneous metadata is shown |

# Task Object Event Chart

|  |  |  |
| --- | --- | --- |
| **Task** | **Object** | **Event** |
| Set Confirmation Message | Cmessage | None |
| Set Error Message | Emessage | None |
| Refresh Database | RefreshButton | Click |
| Browse Movies | BrowseButton | Click |
| Show Settings | SettingsButton | OnCreate |
| Show Help | HelpButton | Click |
| Close Error Message | ErrorClose, ErrorOkButton | Click |
| Close Confirmation Message | ConfCancel, ConfClose | Click |
| Agree to Confirmation Message | ConfOK | Click |
| Cycle Movies | MoveLeft, MoveRight | Click |
| Close Login Dialog | LoginCancel | Click |
| Submit Login Details | SubmitLoginButton | Click |
| Redirect Curious Users to my Site | StreamerLogo | Click |
| Move panels to right location, fingerprint PC, build path, change screen resolution, connect to database, start logging, build working directory, search for database, create fingerprint files | Background | Application Start (OnCreate method) |
| Show/Hide sidebar | Background, StreamerHeader, BrowsePanel | Click |
| Exit application | ExitButton | Click |
| Exit application | Esc Key (Physical key), Alt + F4 | Press |
| Exit application | TDataModule1 | Close |
| Log debug information | Msg, msgtype, PrintDebug() | None |
| Get video title | getTitle(), BrowseButton | Click |

# The Graphic User Interface

# C:\Users\mweya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\0x001.png

Click anywhere except for the Logo to open the sidebar

Logo

Welcome caption

Figure 1 - Splash Screen

Browse all available movies

Exit Button

Refresh Database

Log In Button

Log Out Button

Settings Button

Help Dialog

Navigation Sidebar



Figure 2 - Splash Screen with sidebar extended

Hovering over the Help button highlights it



Figure 3 - Hovering over the Help Button

Takes you to the [readme site](https://mweya.ddns.net/Portfolio/Streamer/readme.html).

Cancels the operation.

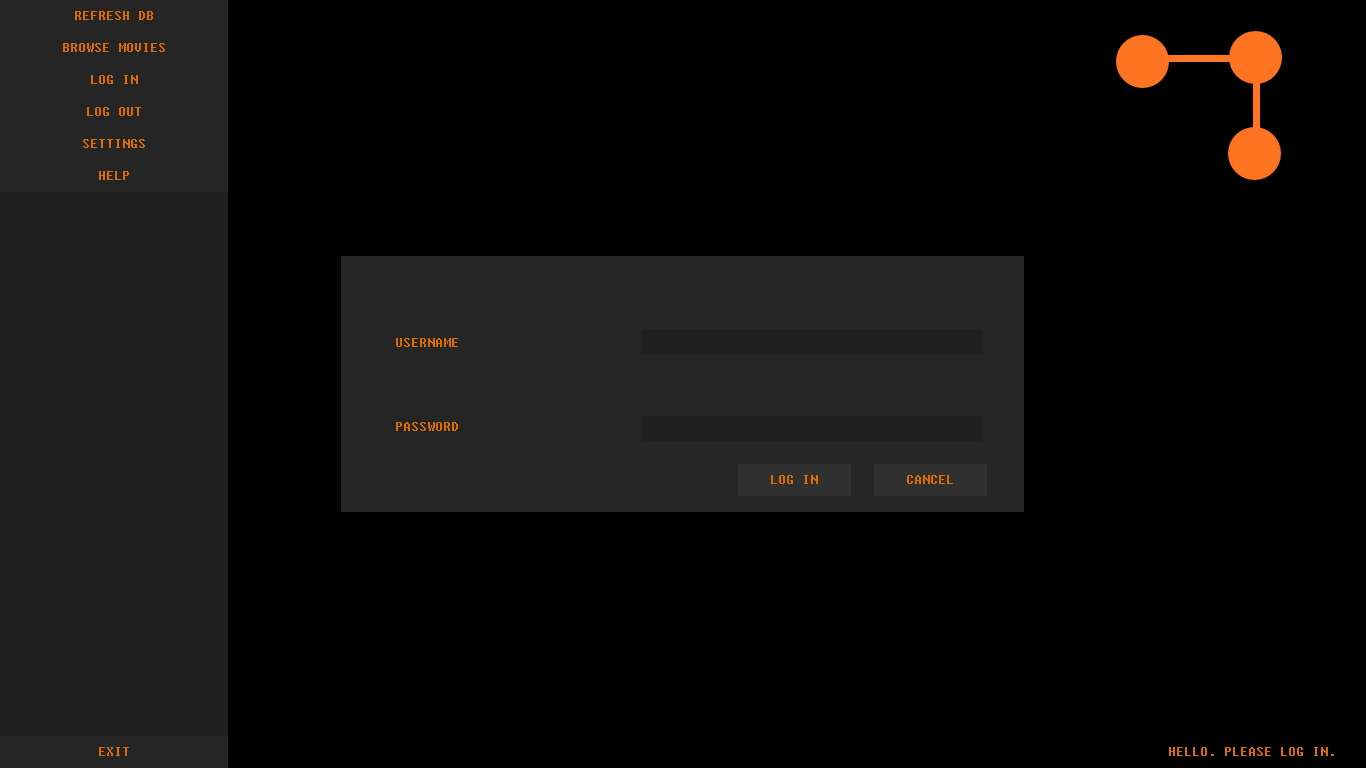


Figure 4 - The Help Context

Hovering over the Log In button highlights it.



Figure 5 - Hovering over the Log In button



Field to enter the password

Field to enter the username

Attempts to log the user in

Cancels the login attempt and closes the dialog

Figure 6 - The Login Dialog

Functionality for the Log In button has changed and the Log Out button has been enabled

Caption has changed, greeting the user



Figure 7 - Example of a session where a user has logged in

Hovering over the Log Out button



Figure 8 - Hovering over the Log Out button

Note that the caption has changed back to the previous state

Buttons that close the dialog

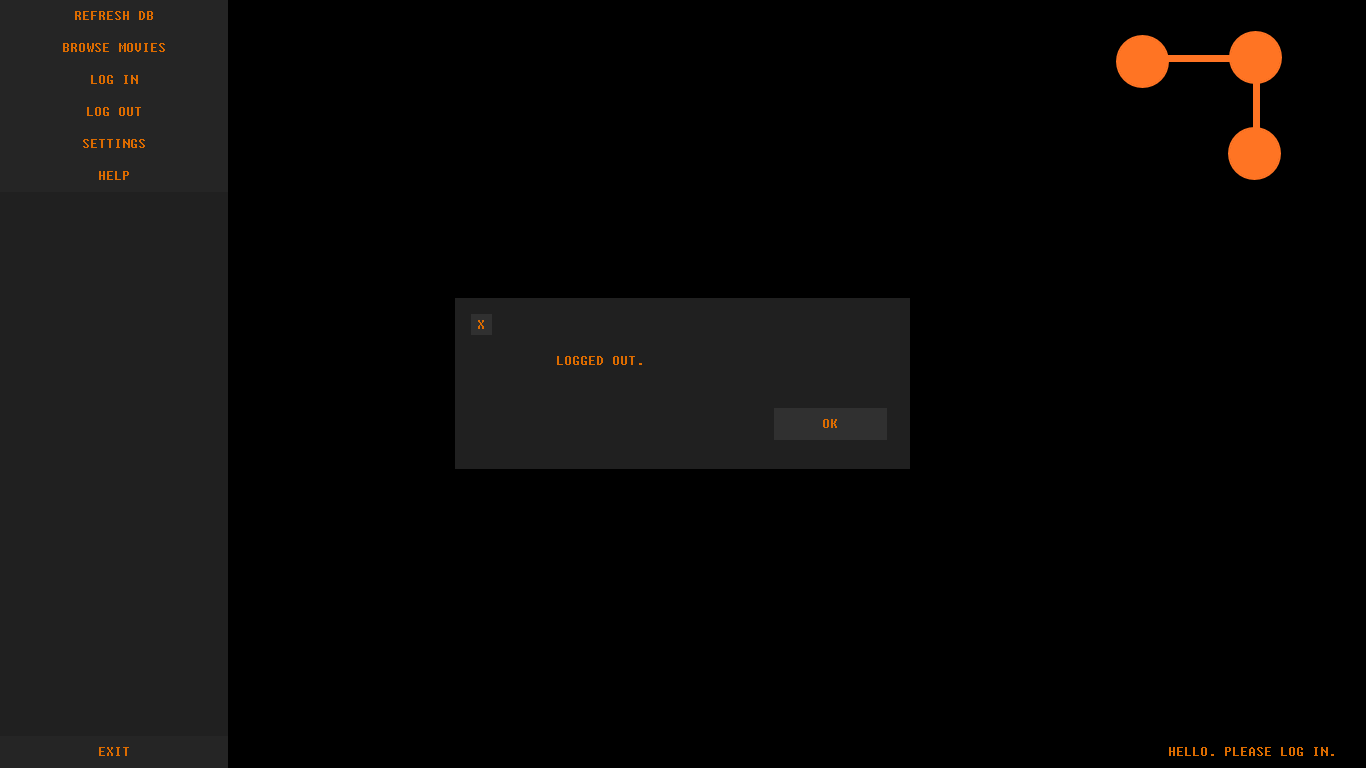
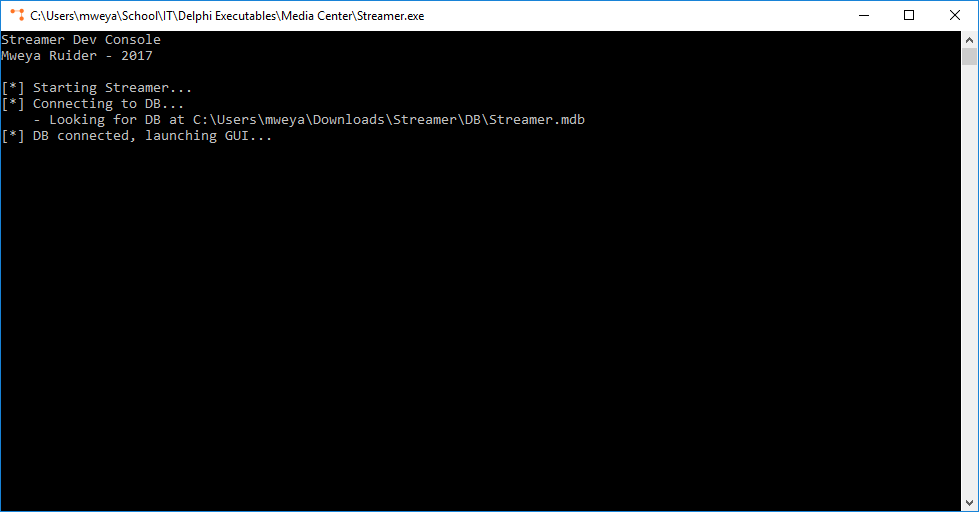


Figure 9 - User logged out



The console outputs information that might be needed during debugging

Figure 10 - The Dev Console for debugging



Hovering over the Browse Movies button

Figure 11 - Hovering over the Browse Movies button



Click anywhere to open the sidebar

Dark space to be used for displaying movie thumbnail

Buttons used to navigate through the database entries

Panel showing the title of the movie

Figure 12 - The Screen used to browse movies



Sidebar extended

Figure 13 - The Browse Screen with the sidebar extended

# Data Structure

|  |  |  |
| --- | --- | --- |
| ***Data Storage (variable)*** | ***Data Type*** | ***Details*** |
| PCuser | String | Used to store the username of the user to build the Path variable. |
| Path | String | Used to store the path to Streamer’s working directory |
| Background | TObject | The form the GUI is built on |
| SBarHidden | Integer | Variable used to track if the sidebar is open or not |
| LoggedIn | Integer | Used to check if the user is logged in or not |
| Username | String | Used to store the username inputted to check it against the database |
| Password | String | Used to store the password inputted to check it against the database |
| Scrwidth | Integer | Used to store the size of the screen’s width before attempting to change the screen resolution |
| Scrheight | Integer | Used to store the size of the screen’s height before attempting to change the screen resolution |
| Emessage | String | Used to change the text on the error message to whatever the error is |
| Cmessage | String | Used to change the text on the confirmation dialog to give more information on the situation |
| DBRefreshed | Integer | Used to keep track on if the database has been refreshed or not |
| Testwidth | Integer | Used to test changing the screen resolution |
| Testheight | Integer | Used to test changing the screen resolution |
| SourceFile | String | Used in an experimental downloading method to update the database if I get around to building a media repository for the streaming of online videos |
| DestFile | String | Used in an experimental downloading method to update the database if I get around to building a media repository for the streaming of online videos |
| LoginPacket | String | Encrypted string for logging into an online media repository |
| TDataModule1 | TObject | Console application used to log debugging information and for handling the database connection |
| DebugMode | String | Used to keep track of if the dev console should be open or not |
| ID | Integer | Used for SQL queries when interacting with the database |
| SQLquery | String | The full SQL query used when interacting with the database |
| Msg | String | Used to transfer information from the GUI to the console application to log data from the GUI |
| Msgtype | String | Used to specify the type of message to be logged |
|  |  |  |

# A note on validation

Validating the user’s input is only done during the login process due to the fact that the program will index videos from the user’s computer (and store them in the database) automatically. Validation during the login process is done by checking if a user exists in the database or not. If the user does not exist, the program should register the user and log them in. Otherwise, the program should check to see if the password entry matches the username. If it matches, the user should be logged in. Otherwise, an error message should be shown and the user should be taken back to the splash screen. Note that the text field should not accept potentially dangerous characters like the ones in this table:

|  |  |  |
| --- | --- | --- |
| Potentially dangerous characters | Examples of malicious use | Explanation |
| “ or ‘ | “;-- or ‘;-- | Ends the SQL statement prematurely by closing the string and commenting out the rest of the statement |
| ; | ;-- | Ends the SQL statement prematurely and comments out the rest of the statement |
| -- | ;-- | Comments out the rest of the SQL statement |
| = | “ OR 1=1;-- | Could be used to make sure the SQL statement always results in a specific result |
| \* | “ or 1=1; DROP DB \*; | Could be used in conjunction with the other potentially dangerous characters to select everything and tamper with all the data in the database |
| \ | \3b -- | Could be used to escape characters to bypass the previous checks |
| % | %3B-- | Could be used to URL encode the characters to bypass the previous checks |

All other symbols can and should be used to make sure that accounts stay as secure as possible.

# Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Input** | **Valid** | **Reaction** |
| No username or password inputted |  | No | GUI prevents username or password from being submitted |
| Correct username and password | Username: “Mweya” Password: “This is the right password” | Yes | Logs the user in |
| Incorrect username or password | Username: “Mweya” Password: “This is not my password” | No | GUI displays error dialog |
| Incorrect username and password | Username: “Shane” Password: “This is not my password” | No | GUI displays error dialog |