



UNIVERSITY OF CYPRUS  
DEPARTMENT OF COMPUTER SCIENCE



# **EPL 342 Databases**

## **Project Social Network Data Management System**

**Ηροδότου Στυλιανός  
Κωστή Χριστόδουλος  
Χρίστου Πανίκος**

**12.04.2020**

## Table of Contents

User Manual:.....	2
Problems running? .....	5
Indexes: .....	5

## User Manual:

When the program starts, it displays the login in view. There the user has the options to sign up or sign in by pressing the appropriate button. By pressing the sign-up button a new view is presented where the user is able to provide the necessary information to register to the system. By pressing the create account button and there is no username with the chosen username a new account is created otherwise a pop up informs the user.

If the user already has an account he can choose to sign in by giving his username and password. If either is wrong, then he is denied access and he should re-enter his credentials. If the credentials are corrected, then he is successfully sign in the system.

When the sign in process is successfully done the user is greeted by the profile view. There he can see the details of his profile. From there he can

1. Edit the information of his profile by pressing the edit button. If his chooses to submit in the editing form view the new information will be saved if not the old information will remain.
2. Choose to see his items. If he chooses the show “item category” button a selection the items of that category is shown. From there he can see the details of that item by pressing the view” item category “button next to it. From there he can edit the details of the item by pressing the edit button If his chooses to submit in the editing form view the new information will be saved if not the old information will remain .
3. Choose to add an item. By providing the necessary information the user can add new item if he press submit. If he changes his mind and doesn’t press submit no new object will be created.

In friends tab the user has the ability to see some details about his friends ( people who he follows and have accepted him ) and to delete his friendship with them . From there he has the ability to see , Friends with at least the same friends as him , his or hers most popular friends , meaning she or he have the most followers , friends that have an album with more than X pictures , friends network ( depth 3 ) that have an album with more than X pictures , Friends that have the exact same interests , an option to see the least popular events and an option to see the users average age .

By choosing the tab of friend requests, a user can see friend requests that others have send to him and want to be friends with him or her . The user has the ability to add them meaning that that person will have the user as a friend ( accept their request ) , delete that request , or ignore that request making it not visible . The user has the option to see his ignored requests and chose to stop ignoring them.

In the search users tab the user is able to make search for other users based on a number of information criteria, like name, birthday, living place, education, previous employment and a lot more. The search will open a new view with the results of the search. There the user is able to see the users profile by pressing the view User. The user is able to sent a friend request to that user when he is viewing his profile. The user is able to sent at max 1 friend request to the other user unless the receiver deletes it(so he can sent again) or he accepts but then deletes him (so he can sent again).

In the search my items tab a user is able to search for his items based on the name of each item. Here we chose the following for the identifying criteria for these search

1. Album->Album Name
2. Link-> Link Name
3. Event-> Even Name
4. Picture-> Picture source
5. Video-> Video message

The search will open a new view with the results of the search. There the user is able to see the items that result from the search. By pressing the view "Category of item" button next to the item you can see the selected item.

In the search other people's items, a user is able to search for his items based on the name of each item. Here we chose the following for the identifying criteria for these search

1. Album->Album Name
2. Link-> Link Name
3. Event-> Even Name
4. Picture-> Picture source
5. Video-> Video message

The search will open a new view with the results of the search. There the user can see the items that result from the search. By pressing the view "Category of item" button next to the item you are able to see the selected item

In the Log all tabs the user can give the number of latest logs he wants to see and after providing that and pressing the search button. It will result to a new view with the results of the search. There the user can see the logs that result from the search. By pressing the view log button next to the log you are able to see the selected log.

In the Log single category tab the user can give the number of latest logs and choose the category of logs he would like to see and after providing that and pressing the search button. It will result to a new view with the results of the search. There the user can see the logs that result from the search. By pressing the view log button next to the log, you are able to see the selected log.

In The search event tab the user is able to search for an event with the characteristics he wants to see. He can provide the venue, name, location and /or other information he wants the event to have. By pressing the search item button It will result to a new view with the results of the search. There the user can see the logs that result from the search. By pressing the view event button next to the event , you are able to see the selected event. There if the users wishes too he can say that he will attend the event. He can only say id once though.

Finally in the finish tab the user can press the export, import or delete button to do this action on the database. If the delete button is choses you have to recomfirm that you with to do so.

The functions of these button is implemented in SQL but is is not implemented in java.

For our database we decided , to create tables for each combination of items where these items are dependent on each other . For example each video details are stored in a table , but its comments id is stored in another table that connects the video with that specific comment , same can be said for the user that uploaded that video . This was done so we can avoid duplicate information , for example , there is no reason to

One Problem we encountered is that despite SQL being able to support it, there was no ways to pass an array in the stored procedure. There was the option to insert it as a string and split it in the database and which we choose but it was a shame because it would have made a big difference and it would have saved as a lot debugging time!

The time limit for this Project was not realistic. It was incredibly hard for us not just to make this project working, but to learn a java UI framework to make it work. Also, the fact that we didn't have the appropriate knowledge until 2 weeks before due was an incredible challenge. We lost a lot of time because we had to change the queries from dynamic SQL to stored procedures. Finally to make changes on the database in the last days as the instructor said was in impossible task as well.

We spent an big portion of time trying to learn the UI framework. The result shows what the instructor wanted us to do, it could have been shown in a better way but thanks to the time limitations we could not achieve the most optimal choice that we wanted.

This project has a lot of potential. The next steps for this is to make the UI more presentable. Next we would have to include actual images not just text. We could also add videos. Finally we could expand the capabilities of the database by creating an API where developers could add new tabs.

We will use the data given to us by the instructors.

## Problems running?

Execution: You can run the application by running the main method in the class main. BUT it will not run because you probably don't have the java fx library, jdbc in your classpath. And you have not connected to a correct Database.

All that you will need are in the resources folder. Firstly you extract the resources to a location on your computer. Then you go to the projects classpath and click on it. Press the button that says add library and click on the user defined libraries. From there you need to click on user Libraries button.

Now you need to create the libraries we need: Click on the new and add a name and create. Then when it appears click on add external jars and add all the jars that appear on the resources. When you are done, press apply and close. Check the library you just created and press finished.

The final step you need to make is to go to the main function, right click and click on the run configurations. From there go to arguments and add the text ' --module-path "[path you extracted]\javafx-sdk-15\lib" --add-modules ' javafx.controls,javafx.fxml'

And click apply.

Hopefully you can run the program now.

In addition you will need to change the connection in the authentication model to be the connection to the DataBase you want. Lastly, that DataBase needs to have the tables that are in the Database package. There you will find scripts to create an identical DB to fully utilize (and firstly log in of course).

You can also whenever a function is called from the models (DB connectors) replace the result with a dummy result you want in the format of the item it return and it will work just as fine.

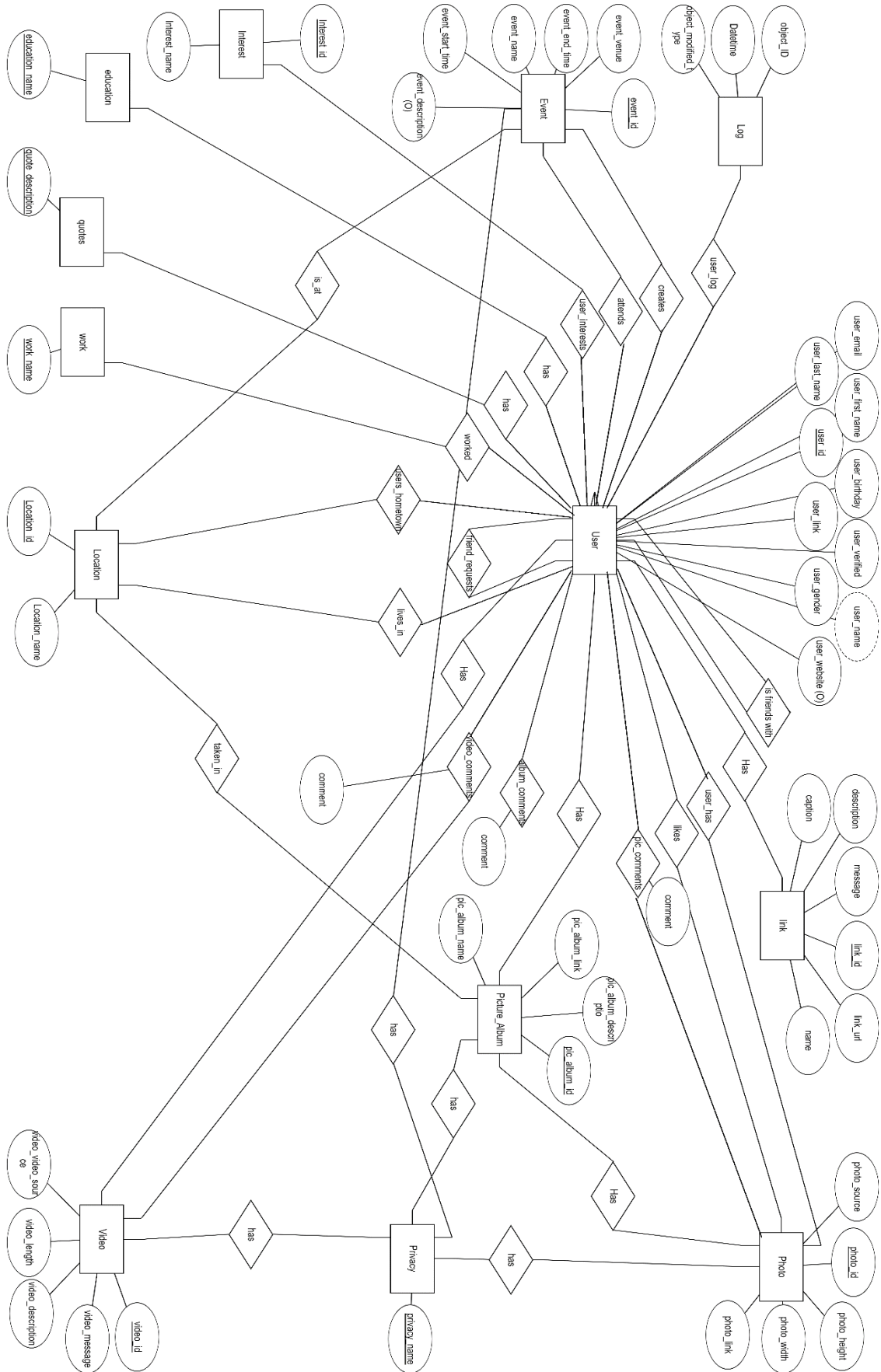
## Indexes:

We have decided to create 4 indexes.

Firstly, in the users id because we go through it all the time in almost everything.

Secondly in the first column of the Friendship because at many times use that to check the friendship. And lastly another 2 on the username and password because every time you log in you use these so it's vital the authentication happens fast.

ER: Small change in how the comments are registered



Relational: Taken all the feedback and fixed it

