

Class Database

Location: /data/database.py

Remarks:

Database is a class that operates an SQLite database. Database object must be located in the same directory as the script

Variables:

- *name* : name of the database loaded
- *db* : database object

Functions:

- *create_database(database_name)* -> loads up the database or creates new one if the database doesn't exists.
- *get_field_by_parameter(table_name,field_name,parameter,value)* -> searches for a value at the field_name in table name that passes the parameter condition.
- *register_user(login,email,password)* -> registers user if data is valid
- *Login_user(login,password)* -> logins user if data is valid
- *get_users()* -> get all users from database
- *get_movies()* -> get all movies from database
- *get_movies_by_parameter(parameter,value)* -> returns movies that passes parameter condition
- *get_user_reviews(user)* -> get all reviews committed by the user
- *get_movie_reviews(movie)* -> get all reviews for the movie
- *add_movie(movie)* -> adds movie to the database
- *rate_movie(user,movie,score)* -> calls add_review and updates avg rate for the movie
- *add_review(user_id,movie_id,score)* -> adds review to the database
- *clear()* -> clears the datatable , calls SQL "DROP TABLE"

Class AppInstance

Location: /app_instance.py

Remarks:

AppInstance is a class that is overseeing the whole application , it holds information about database , currently logged user and app data from the Qt

Class DataGenerator

Location: /data_generator.py

Remarks:

DataGenerator is a class that is generating for the database

Functions:

- *generate_users()* -> populates users table
- *generate_movies()* -> populates movies table
- *generate_reviews()* -> populates reviews table

Class Movie

Location: /data_objects/movie.py

Remarks:

Movie is a class that represents data about it's title , director , duration ...

Variables:

- *Id*
- *Title*
- *Director*
- *Description*
- *Duration*
- *Actors*
- *Genre*
- *Icon_path*
- *Avg_rate*
- *Number_of_users*

Functions:

- *set_avg_rate(avg_rate,number_of_users)* -> sets the avg_rate for the movie
- *rate(value,user)* -> rates the movie and updates its avg_rate

Class Review

Location: /data_objects/review.py

Remarks:

Review is a class that represents data about who posted it , which movie it is related to and the score given

Variables:

- *User_id*
- *Movie_id*
- *Score*

Class User

Location: /data_objects/user.py

Remarks:

User is a class that represents data about user's login , email , password ,id ..

Variables:

- *Login*
- *Email*
- *Password*
- *Icon_path*
- *Id*

Class FilwebDialog

Location: /dialogs/filmweb_dialog.py

Remarks:

FilmwebDialog is a class that sets window icons for all of the dialogs in the application

Class AddMovieDialog

Location: /dialogs/add_movie_dialog.py

Inheritance: FilmwebDialog

Remarks:

AddMovieDialog is a class that is responsible for add movies in a way that is friendly to the user

Functions:

- *get_movie_details(parent)* -> static method that creates the dialog and returns the data about the movie
- *show_fields()* -> creates line edit for each category (title , director , description , duration ,actors , genre)
- *get_data()* -> returns movie object from the data from the line edits

Class DetailsDialog

Location: /dialogs/details_dialog.py

Inheritance: FilmwebDialog

Remarks:

DetailsDialog window is a class that hosts informations about given movie.

Functions:

- `get_movie_details(movie,parent)` -> static method that creates the dialog and displays it
- `get_ratings(movie)` -> gets user ratings for a movie
- `get_details(movie)` -> gets details about the movie

Class RateDialog

Location: /dialogs/rate_dialog.py

Inheritance: FilmwebDialog

Remarks:

RateDialog is a class that is responsible for rate a movie in a way that is friendly to the user.

Functions:

- `get_rate(parent)` -> static method that creates the dialog and returns the rate
- `create_radio_buttons()` -> creates radio button for rates from 1 to 10

Class LoginDialog

Location: /dialogs/login_dialog.py

Inheritance: FilmwebDialog

Remarks:

LoginDialog is a class that is responsible for logging users in a way that is friendly for the user.

Functions:

- `get_login_password(parent)` -> static method that creates the dialog and gets the data about the login and password
- `get_login_and_password()` -> returns login and password from line edits

Class RegisterDialog

Location: /dialogs/login_dialog.py

Inheritance: FilmwebDialog

Remarks:

RegisterDialog is a class that is responsible for registering users in a way that is friendly for the user.

Functions:

- `get_login_password(parent)` -> static method that creates dialog and return login , email and password
- `get_data()` -> returns login , email and password for line edits

Class FilmwebWindow

Location: /windows/filmweb_window.py

Remarks:

FilmwebWindow is a class that creates window icons and desired size (500px x 350px) for all windows.

Functions:

- `closeEvent(event)` -> adds pop up question "Do you really want to leave" when clicking the close button

Class WidgetManager

Location: /windows/widget_manger.py

Inheritance: FilmwebWindow

Remarks:

WidgetManager is a class that controls changing between filmweb windows

Functions:

- `switch_window(window_name,**kwargs)` -> switches to MainWindow if the window name equals to "Main Window" else if there is user object specified in kwargs switches to ProfileWindow

Class ProfileWindow

Location: /windows/profile_window.py

Inheritance: FilmwebWindow

Remarks:

ProfileWindow is a class that presents user's login and reviews created by him/her. Has reference to WidgetManager.

Functions:

- `show_main_window()` -> calls WidgetManager and switches to MainWindow
- `show_reviews()` -> show reviews of the user

Class MainWindow

Location: /windows/main_window.py

Inheritance: FilmwebWindow

Remarks:

MainWindow is a class that shows latest movies , recommended movies , users and has search engine functionality for movies. Has reference to WidgetManager.

Functions:

- `set_search_key(key)` -> sets search key for search engine
- `show_movies(**kwargs)` -> show movies sorted by parameter if key and value are specified in kwargs , shows recommended movies if sort is specified in kwargs else shows latest movies.
- `add_movie()` -> opens AddMovieDialog
- `show_profile(user)` -> switches to ProfileWindow for specified user
- `rate_movie(movie,label)` -> opens RateDialog and updates the label showing average rate
- `login()` -> opens LoginDialog
- `register()` -> opens RegisterDialog
- `logout()` -> logouts user
- `set_is_logged(user)` -> changes layout based whenever user is logged or not
- `show_users()` -> show all users in the scroll box

Summary:

- I did choose sql database over simple csv files because of the superior speed and already built in function for searching wanted values.
- I did choose PyQt as my main framework because of its abundant documentation and simplicity.
- I did add simple search bar to improve user experience and make it easier to search for specified movies.
- To generate new database run script `data_generator` , it fills the database with user , movies and reviews.
- I did add icons for users and movies to improve user experience , sadly they can be only added in via database and not in the app itself as i didn't have enough time to research how to add images using file explorer in Qt.