Technical Documentation

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 createds the dialog and returns the rate
* create\_radio\_buttons) -> creates radio button for reates 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 it 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 bar functionality for movies. Has reference to WidgetManager.

**Functions:**

* set\_search\_key(key) -> sets search key for search bar
* 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.( look a the arrays to change generated data)
* I did add icons for users and movies to improve user expierience , 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 PyQt.
* To run the application , run the main.py script.
* Tests cover Movie,User,Review objects and database methods.