VŠB-TUO

Fakulta informatiky

DAIS

Track.tv

Author:
Matouš Valeš

Supervisor:
Ing. Peter Chovanec

24. dubna 2016

1 Assingment specification

Let me entroduce Track.tv by answering a couple of key questions.

WHAT? Track.tv is a sophisticated solution for keeping track of all your tv shows and movies, that you like. The solution consists of a powerful database and an app, that allows the user to make a personalised library. All the user specifies are his interests in TV - our engine then takes care of providing a list of upcoming shows, suggestions and letting him know when to get hyped for a new season premiere.

HOW? The backbone of the system is a database of tv shows, their genres, ratings and so on. Once a user logs into the app, he specifies a set of tv shows that he wants to follow. The app then presents a list of TV shows and their airing dates. The user will be able to browse the whole database, following or rating tv shows with just a click of a button, thus creating a personalised library. Users will be able to take a look at their own statistics and a list through a selection of tv shows reccomended to them, based on their watching history.

WHERE? The backend is maintained by a remote server, holding all of the database information. Users can interact with the system using an app that serves as a presentation layer.

WHO? The main purpose of this system is to serve as a tool for the users. They are the only entity using the system, along with a specified group of administrators, that is keeping the shows database up to date.

WHEN? The prime time tv is specified as an interval between 18:00 to 23:00 hours, so predictably this is when the highest usage will occur. Occasionally a one or multiple day usage spike will happen, because nowadays some internet specific tv shows have all of the episodes of the current season released in a single moment during the day.

WHY? People are passionate about tv. It is still one of the main sources of enterntainment and tv show fans like to follow them very closely. This is a reason why there is a lot of potential for apps to get massive user bases, since almost everybody watches TV. The apps make the user let somebody else take care of reminding him when to turn on their TVs or computer screens. Having less stuff to keep in mind is always welcomed by users and for developers it's just another way to collect data, that can be attractive to third parties, that later use it for personalised advertising.

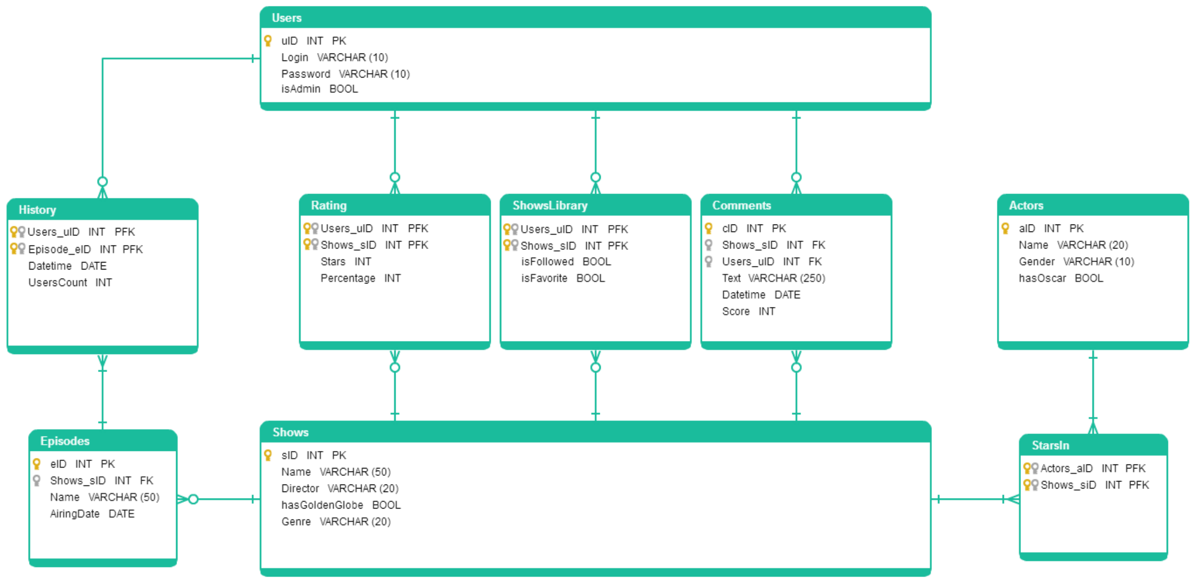
INPUTS: The main input are the specific tv shows that the user would like to follow or add to his watch history. Users are also able to give percentage ratings to individual episodes. There is also an option to leave a comment on the shows detail page.

OUTPUTS: The system is able to give the user his watchlist, usage history, ratings and a personalised selection of tv shows, that could be of interest to him, based on his watching behavior. Using the stored information, the system is always ready to give the user his statistics.

FUNCTIONS: The system has to keep track of individual watchlists of every registered user and show all of their information on demand. It needs to calculate suggestions based on collected information and show currently running tv shows based on the current date, store ratings input by users and display all of this information in a simple, yet easy to use package.

2 Conceptual Model

2.1 ER Diagram



Obrázek 1: Conceptual model of the database

2.2 Entity types

Legend: **Table**, Primary Key, *Foreign Key*, attribute

1. **Users**(uID, Login, Password)
2. **Comments**(cID, *Shows_sID*, *Users_uID*, Text, Datetime, Score)
3. **ShowsLibrary**(Users_uID, Shows_sID, isFollowed, isFavorite)
4. **Rating**(Users_uID, Shows_sID, Stars, Percentage)
5. **History**(Users_uID, Episode_eID, Datetime, UsersCount)
6. **Episodes**(eID, *Shows_sID*, Name, AiringDate)
7. **Shows**(sID, Name, Director, hasGoldenGlobe, Genre)
8. **StarsIn**(Actors_aID, Shows_sID)
9. **Actors**(aID, Name, Gemder, hasOscar)

3 Data models

The following is an overview of each individual tables.

- PK - Primary Key
- FK - Foreign Key
- PFK - Primary Foreign Key

3.1 Users

Attribute	Data type	Length	Key	Null	Index	IC	Desription
uID	INT		PK	N	Y		
Login	VARCHAR	10		N			Users login
Password	VARCHAR	10		N			Users password
isAdmin	BOOL			N			Indentification of system admin

3.2 Rating

Attribute	Data type	Length	Key	Null	Index	IC	Desription
Users_uID	INT		PFK (Users)	N	Y		
Shows_sID	INT		PFK (Shows)	N	Y		
Stars	INT			N		1	Rating in form of 1 to 5 stars
Percentage	INT			N		2	Percentage rating

3.3 ShowsLibrary

Attribute	Data type	Length	Key	Null	Index	IC	Desription
Users_uID	INT		PFK (Users)	N	Y		
Shows_sID	INT		PFK (Shows)	N	Y		
isFollowed	BOOL			N			Information whether the user is following given show
isFavorite	BOOL			N			Information whether the user has given show in his favorites

3.4 Comments

Attribute	Data type	Length	Key	Null	Index	IC	Desription
cID	INT		PK	N	Y		
Users_uID	INT		FK (Users)	N	Y		
Shows_sID	INT		FK (Shows)	N	Y		
Text	VARCHAR	250		N			Text of the comment
Datetime	TIMESTAMP			N			Time of its posting
Score	INT			N			Relevancy as voted by the users

3.5 History

Attribute	Data type	Length	Key	Null	Index	IC	Desription
Users_uID	INT		PFK (Users)	N	Y		
Episode_eID	INT		PFK (Episodes)	N	Y		
Datetime	TIMESTAMP			N			Time of adding an episode into history

3.6 Episodes

Attribute	Data type	Length	Key	Null	Index	IC	Desription
eID	INT		PK	N	Y		
Shows_sID	INT		FK (Shows)	N	Y		
Name	VARCHAR	50		N			Name of the episode
AiringDate	DATE			N			Airing date of the episode

3.7 Shows

Attribute	Data type	Length	Key	Null	Index	IC	Desription
sID	INT		PK	N	Y		
Name	VARCHAR	50		N			Name of the show
Director	VARCHAR	20		N			Name of the director
hasGoldenGlobe	BOOL			N			Information whether the show was awarded a Golden Globe
Genre	VARCHAR	20		N			Genre of the show

3.8 StarsIn

Attribute	Data type	Length	Key	Null	Index	IC	Desription
Actors_aID	INT		PFK (Actors)	N	Y		
Shows_sID	INT		PFK (Shows)	N	Y		

3.9 Actors

Attribute	Data type	Length	Key	Null	Index	IC	Desription
aID	INT		PK	N	Y		
Name	VARCHAR	20		N			Name of the actor
Gender	VARCHAR	10		N			Gender of the actor
hasOscar	BOOL			N			Information whether the actor was awarded an Oscar

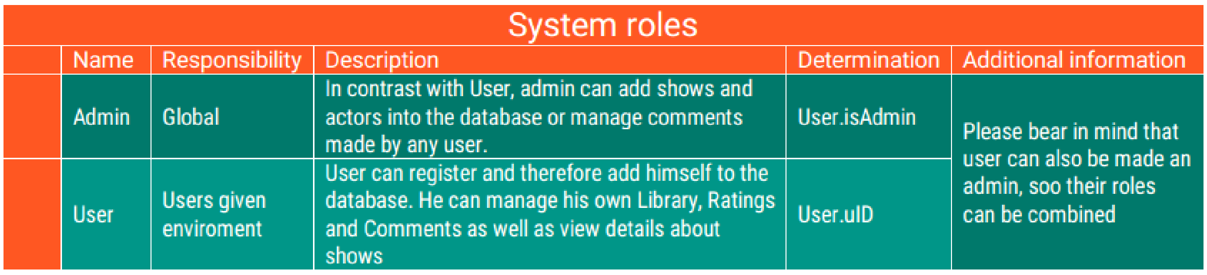
3.10 Integrity Constraints

1. **Stars:** The integer is only valid on a scale from 1 to 5.
2. **Percentage:** The value is valid only from 1 to 100.

4 Functional analysis

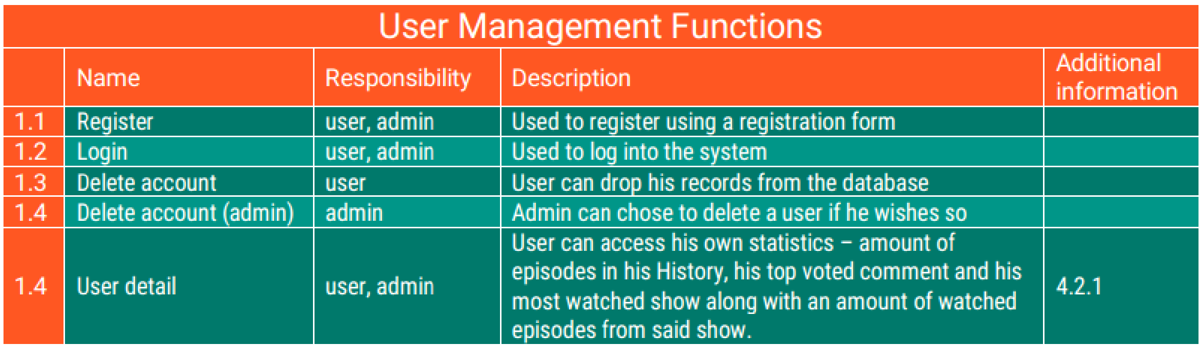
4.1 List of functions

It this following section you will find a brief description of every function in the system. Additionally I will be giving a thorough description of more complicated functions in the following chapter.



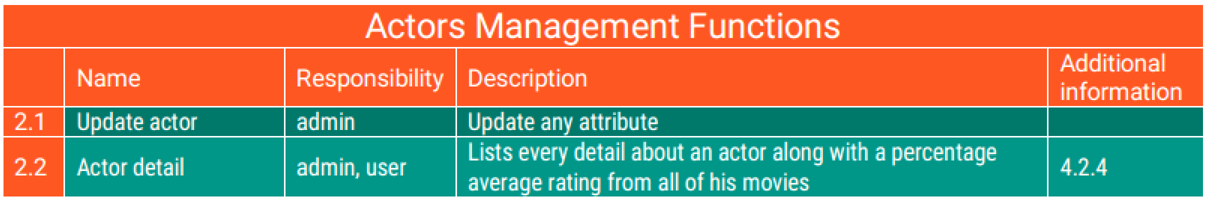
System roles

	Name	Responsibility	Description	Determination	Additional information
	Admin	Global	In contrast with User, admin can add shows and actors into the database or manage comments made by any user.	User.isAdmin	Please bear in mind that user can also be made an admin, soo their roles can be combined
	User	Users given enviroment	User can register and therefore add himself to the database. He can manage his own Library, Ratings and Comments as well as view details about shows	User.uID	



User Management Functions

	Name	Responsibility	Description	Additional information
1.1	Register	user, admin	Used to register using a registration form	
1.2	Login	user, admin	Used to log into the system	
1.3	Delete account	user	User can drop his records from the database	
1.4	Delete account (admin)	admin	Admin can chose to delete a user if he wishes so	
1.4	User detail	user, admin	User can access his own statistics – amount of episodes in his History, his top voted comment and his most watched show along with an amount of watched episodes from said show.	4.2.1



Actors Management Functions

	Name	Responsibility	Description	Additional information
2.1	Update actor	admin	Update any attribute	
2.2	Actor detail	admin, user	Lists every detail about an actor along with a percentage average rating from all of his movies	4.2.4

Shows Management Functions				
	Name	Responsibility	Description	Additional information
3.1	Add show	admin	Used to add a new row into the Shows table	
3.2	Update show	admin	Update any attribute	
3.3	Add actor	admin	Adds a new actor into the Actors table and links his Actors.aID with the currently selected Shows.sID using the StarsIn table	
3.4	Update actor	admin	Update any attribute	
3.5	Add episode	admin	Adds a new episode into the Episodes table and links its Episode.eiD with the currently selected Shows.sID.	
3.6	Show detail	admin, user	Lists every detail about a show (all of its actors, episodes, comments and ratings)	
3.7	Top 10	admin, user	Lists top 10 shows based on their percentage rating	
3.8	Follow show	admin, user	Adds the Show.sID into the ShowsLibrary along with the currently logged in user's User.uID and sets the isFollowed boolean to true	
3.9	Mark show as favorite	user, admin	Adds the Show.sID into the ShowsLibrary along with the currently logged in user's User.uID and sets the isFavorite boolean to true	
3.10	Get suggested shows	user, admin	Returns a list of suggested shows based on currently logged in user's ShowsLibrary	4.2.2
3.11	Post comment	user, admin	Adds a new entry into the Comments table, links it with the currently logged in user's User.uID and with the Show.sID atribute of the commented movie.	
3.12	Delete comment	admin	Admin can chose to set the Text attribute of a selected comment to whatever he wants to, in most cases to „Comment deleted"	
3.13	Rate show (stars)	user, admin	Adds a new entry to the Rating table, links the primary foreign keys and sets the Rating.Stars attribute to the input value.	
3.14	Rate show (percentage)	user, admin	Adds a new entry to the Rating table, links the primary foreign keys and sets the Rating.Percentage attribute to the input value.	
3.15	Rate comment	user, admin	Raises or lowers the Comments.Score attribute of a selected comment base on the Comments.cID	
3.16	Get the best actor	user, admin	Displays the actor, whose average percentage made out of all shows he's starred in, is the highest	4.2.3

Watchlist Module				
	Name	Responsibility	Description	Additional information
4.1	List upcoming episodes	user, admin	Lists all episodes, that will be airing the following week	4.2.5
4.3	Mark as watched	admin, user	Adds a selected Episode.eID into the History table and links it with the currently logged in user's User.uID	

Statistics Module				
	Name	Responsibility	Description	Additional information
5.1	List History	user, admin	Shows every entry from the History table for the currently logged in user	
5.2	Most watched actor	user, admin	Displays an actor, that has starred in the most episodes in the user's history based on the UsersCount attribute	4.2.6
5.3	Comment with the highest score	user, admin	Displays the currently logged in user's highest rated comment	4.2.1
5.4	Most watched show	user, admin	Display the most watched show along with a number of episodes.	4.2.1

4.2 Function descriptions

The system uses a lot of trivial CRUD based functions. However there are a couple functions, which can be a little bit more difficult to understand, therefore I will try to explain their logic in the following chapter. Even though these functions are only reading data from the database, in the case of "Get best actor"and "Actor detail"transactions are used, to always provide the most up to date data. Transactions are also used in functions that input data to the database, for example procedures for rating shows or posting comments.

4.2.1 Show user detail

User detail is a function to load data for the My Account pane, which I will explain in the User interface chapter. The purpose of this function is to get a personalized statistics page for each user. The user will be presented with the amount of entries (watched episodes) in his history, his top voted comment and his most watched show along with an amount of watched episodes from said show.

Input: $userID

```
SELECT Users.uID, Users.Login, his.sumofEpisodes, Comments.cID, Comments.Score,
    y.show, y.count
FROM(
    SELECT Users_uID AS usID, COUNT(*) AS sumOfEpisodes
    FROM History
    GROUP BY Users_uID
    HAVING Users_uID = "$userID"
    ) his
JOIN Users ON Users.uID = his.usID
JOIN Comments ON Users.uID = Comments.Users_uID
JOIN(
    SELECT Users_uID AS ID, Shows_sID AS show, COUNT(Shows_sID) AS count
    FROM History JOIN Episodes ON History.Episodes_eID = Episodes.eID
    GROUP BY Shows_sID, Users_uID
    HAVING Users_uID = "$userID" ORDER BY count DESC
    ) y ON Users.uID = y.ID
WHERE Comments.SCORE = (
                        SELECT MAX(Score)
                        FROM Comments
                        GROUP BY Users_uID
                        HAVING Users_uID = "$userID"
                        ) AND rownum = '1'
;
```

4.2.2 Get suggested shows

The system can introduce the user to new shows by determining which genre is the most common in the users personal library.

Once it finds out which is the user's most favorite genre, it presents him with a list of TV shows from the same genre. The presented shows are only those, which are not yet in the user's library and because we want our users to watch only critically acclaimed shows, the system only presents those which have been rewarded by a Golden Globe award.

Input: $userID

```
SELECT Name
FROM Shows
WHERE genre = (
              SELECT y.genre
              FROM(
                  SELECT Shows.genre AS genre, COUNT(*) AS amount
                  FROM ShowsLibrary
                  JOIN Shows ON ShowsLibrary.Shows_sID = Shows.sID
                  GROUP BY Shows.genre, ShowsLibrary.Users_uID
                  HAVING ShowsLibrary.Users_uID = "$userID" ORDER BY amount
                     DESC
                  ) y
              WHERE rownum = '1'
             )
AND HASGOLDENGLOBE = '1'
AND Name NOT IN(
               SELECT Shows.Name
               FROM Shows
               INNER JOIN ShowsLibrary ON ShowsLibrary.Shows_sID = Shows.sID
               WHERE ShowsLibrary.Users_uID = "$userID"
               )
;
```

4.2.3 Get best actor

The following procedure has no input parameters and only serves as a quick way to select the best actor based on the percentage average rating from all of the shows they starred in. The procedure takes into account ratings from all users across the platform. This data is then displayed in special box, about which you can read in the User Interface chapter.

```
SELECT y.average, y.ActorName
FROM(
     SELECT AVG(Rating.Percentage) as average, Actors.Name as ActorName
     FROM Rating
     JOIN Shows ON Shows.sID = Rating.shows_sID
     JOIN StarsIn ON StarsIn.Shows_sID = Shows.sID
     JOIN Actors ON Actors.aID = StarsIn.Actors_aID
     GROUP BY Actors.Name order by average desc
     ) y
WHERE rownum = '1'
```

4.2.4 Actor detail

Actor detail is a function to load data for the Actors pane, which I will explain in the User interface chapter.

The purpose of this function is to get interesting data about an actor, that will be presented to the user. The procedure will return a list of the actors shows and their average percentage rating. Along with this list, information about the actor (Name, Gender and whether he has been awarded with an Oscar) will be loaded.

Input: $actorsName

```
SELECT y.averageRating, y.ShowName, y.ActorName, Actors.Gender, Actors.hasOscar
FROM(
     SELECT AVG(Rating.Percentage) AS AverageRating, Shows.Name AS ShowName,
     Actors.Name AS ActorName
     FROM Rating
     JOIN Shows ON Shows.sID = Rating.shows_sID
     JOIN StarsIn ON StarsIn.Shows_sID = Shows.sID
     JOIN Actors ON Actors.aID = StarsIn.Actors_aID
     GROUP BY Actors.Name, Shows.Name
     HAVING Actors.Name = "$actorsName" ORDER BY averagerating desc
     ) y JOIN Actors ON Actors.Name = y.ActorName
;
```

4.2.5 List upcoming episodes

This procedure is actually very simple, but I'm including it, because its functionality is absolutely crucial. It is used to display shows airing the following week. This functionality is included in the Watchlist pane of the UI.

Input: $userID

```
SELECT episodes.airingdate
FROM episodes
JOIN Shows on Shows.sID = episodes.Shows_sID
JOIN ShowsLibrary on ShowsLibrary.Shows_sID = Shows.sID
WHERE ShowsLibrary.USERS_UID = "$userID"
AND ShowsLibrary.ISFOLLOWED = '1'
AND episodes.AIRINGDATE >= TRUNC(SYSDATE)
AND episodes.AIRINGDATE <= TRUNC(SYSDATE) + 7
```

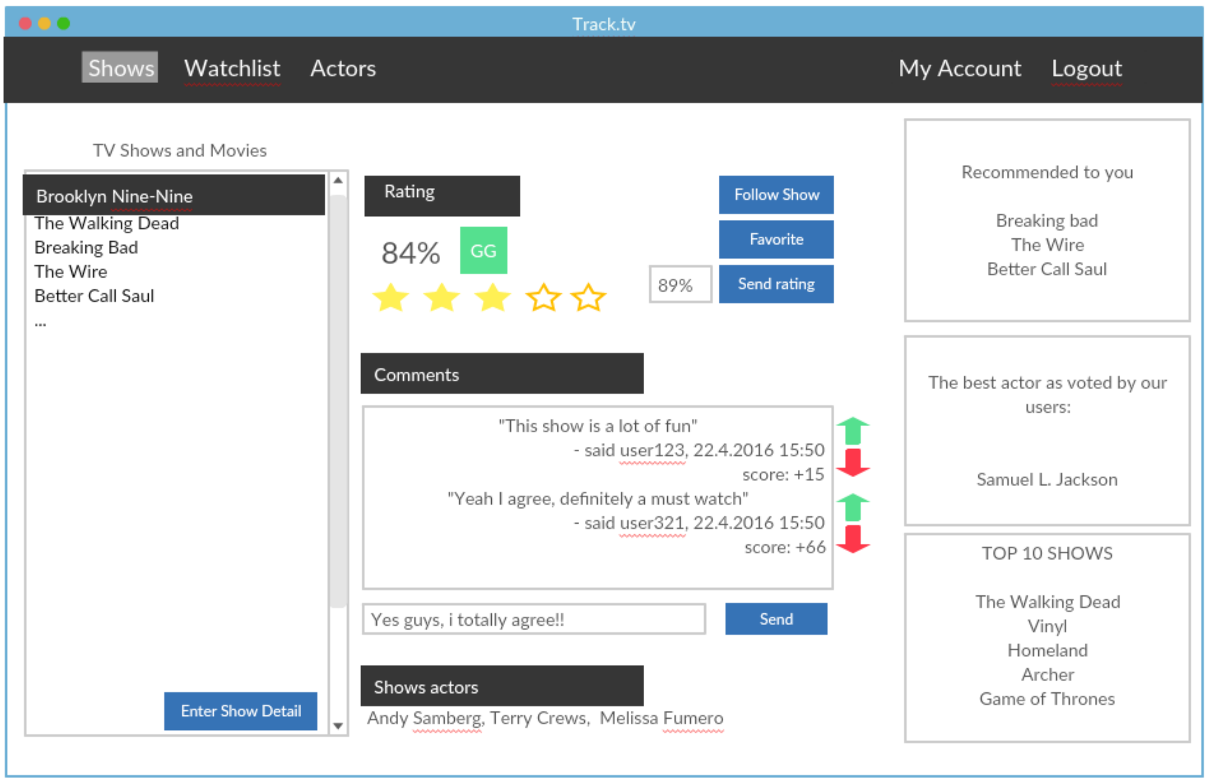
4.2.6 Most watched actor

This procedure is used to determine the currently logged in user's most watched actor. The system browses through the whole history of the user and then selects an actor with the highest occurrence. This information is later displayed in the user detail pane as a part of personalized statistics.

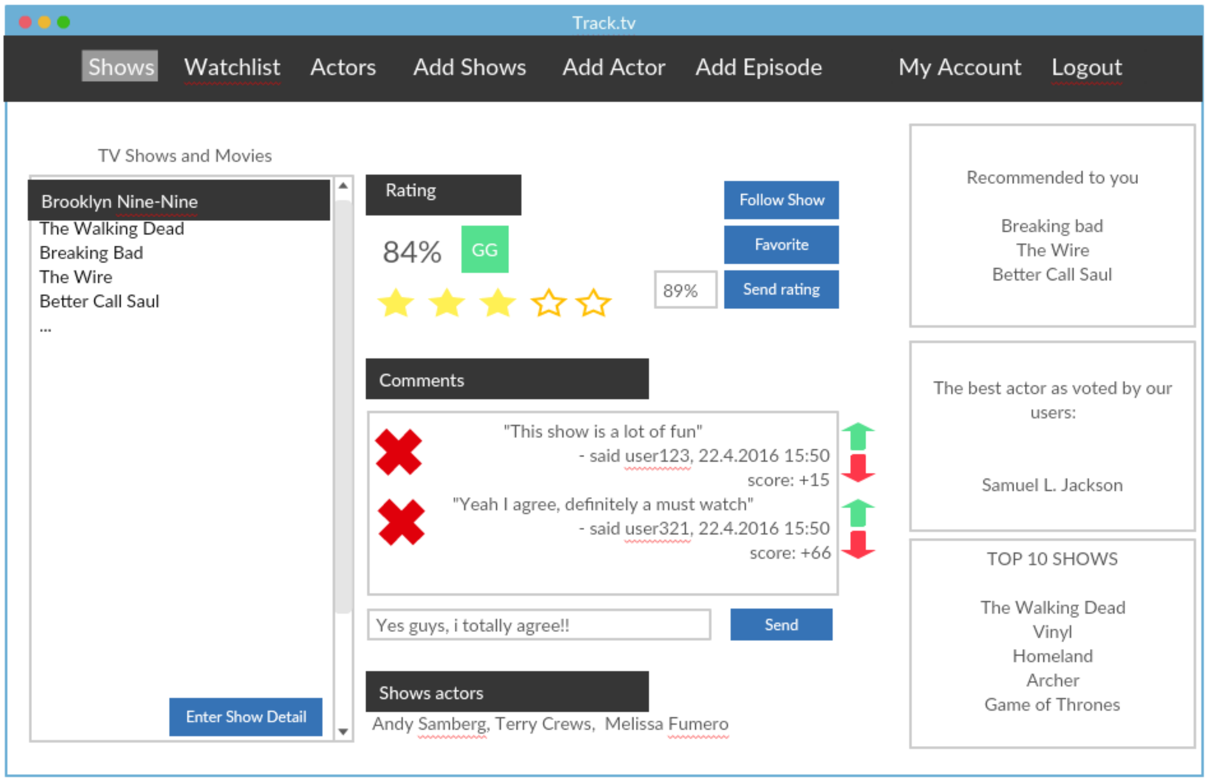
Input: $userID

```
SELECT y.ActorName, y.timesInHistory
FROM(
     SELECT Actors.Name AS ActorName, COUNT(Actors.Name) AS timesInHistory
     FROM History
     INNER JOIN Users ON Users.uID = history.users_uID
     INNER JOIN Episodes ON Episodes.eID = history.episodes_eID
     INNER JOIN ShowsLibrary ON ShowsLibrary.Shows_sID = Episodes.shows_sID
     INNER JOIN Shows ON Shows.sID = ShowsLibrary.Shows_sID
     INNER JOIN StarsIn ON StarsIn.Shows_sID = Shows.sID
     INNER JOIN Actors ON Actors.aID = StarsIn.Actors_aID
     GROUP BY ShowsLibrary.Users_uID,History.Users_uID, Actors.Name
     HAVING ShowsLibrary.Users_uID = "$userID" AND History.Users_uID = "
        $userID" ORDER BY timesInHistory DESC
     )y
WHERE rownum = '1'
```

5 User interface



Obrázek 2: User interface mockup - viewed as a User



"This show is a lot of fun"
- said user123, 22.4.2016 15:50
score: +15
"Yeah I agree, definitely a must watch"
- said user321, 22.4.2016 15:50
score: +66

Yes guys, i totally agree!!

Obrázek 3: User interface mockup - viewed as an Admin

5.1 Menu

The menu is implemented in a form of panels switched by a top menu bar. Once a user choses a section, the selected page will be highlighted. Each panel will consist of boxes, buttons and list views. Once a user flagged as admin logs in, there is a second set of menu items, which are used to maintain the database of shows.

5.1.1 Shows

The shows menu is the users main source of information about shows. Individual shows are listed in a scrollable box on the left side. Once a show is higlighted, the middle part of the pane displays shows rating, average amount of stars, comments and a set of buttons. If the show has the attribute hasGoldenGlobe set to true, then a green box with "GG"is displayed next to the rating. The funcionality is as follows:

- Enter Show Detail - action 3.6(Show detail), switches the layout to display information about the highlighted show in the middle part of the Shows pane
- Follow Show - action 3.8 (Follow show)
- Favorite - action 3.9 (Mark show as favorite)

- Send rating - action 3.14 (Rate show (percentage))
- Send comment - action 3.11 (Post comment)
- Clicking on a star - action 3.13 (Rate show (stars))

There is also another set of boxes on the right hand side, which display relevant information to the user. The functions, that are used to get the information for these boxes, are these:

- Reccomended to you - action 3.10 (Get suggested shows)
- The best actor - action 3.16 (Get the best actor)
- Top 10 shows - action 3.7 (Top 10)

If the logged user is an admin, then he has the authority to delete a comment. For that he can use the designated buttons in a form of a red cross. These use the Delete comment function (action 3.12)

5.1.2 Watchlist

The watchlist is a pane where the user will be presented with a list of episodes, that will be airing the following week. The only functionality this pane has is to inform the user about upcoming premieres and let him select a show as watched. For that the function 4.3(Mark as watched) will be called.

5.1.3 Actors

The user has an option to browse through a list of actors that are in the database. The layout is very similar to the Shows pane, where on the left side there is a list of all actors with a button, that changes the layout to show information about the actor, after it is clicked. The middle side of the pane will then show actors name, whether he has an Oscar and an average percentage rating of all shows he stars in. The function 2.2(Actor detail) will be called to get all the data.

5.1.4 My Account

The user account page will show statistics about the user and display them in boxes. The system uses the Show user detail function to load all the important data. A number of other functions are avaiable to use:

- List History - action 5.1
- Most watched actor - action 5.2
- Comment with the highest score - action 5.3
- Most watched show - action 5.4

5.1.5 Admin options

If the logged in user is an administrator, then he has a couple more panes to chose from. In the Add Shows pane, the admin is prompted to chose wheter he wants to edit a current show or add a new one. The functions used in these 2 cases are 3.1 (Add show) and 3.2 (Update show). If he's adding a new show, he will be immediately asked to input at least a single episode and one starring actor. For this, the actions 3.3 and 3.5 are used. The same actions are used in the case of selecting Add Actor or Add Episode options. In the Add Actor pane, the admin can also modify information about an actor (for example setting that he's just been awarded an Oscar) using the action 3.4 (Update actor).