



Bilkent University

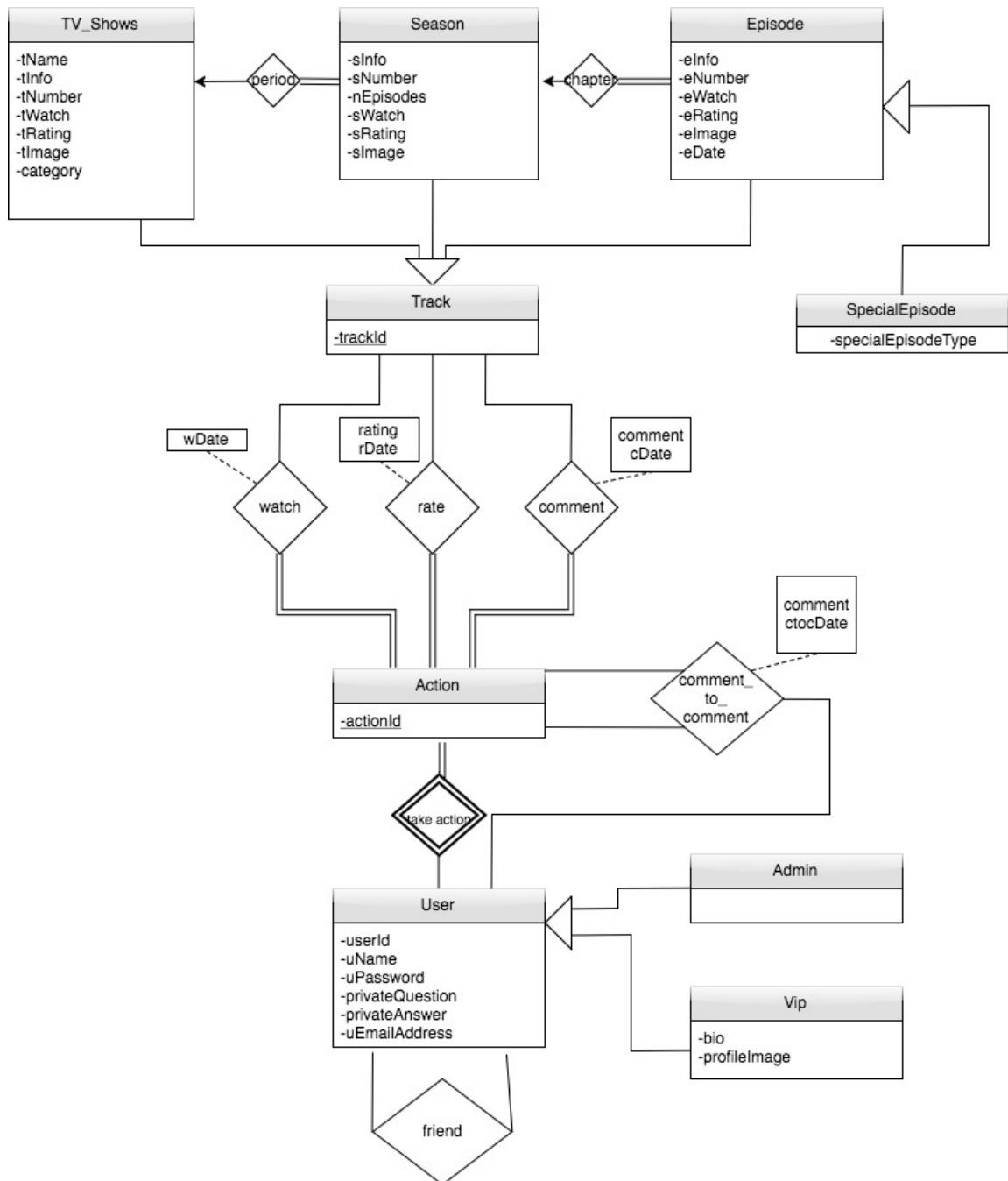
## CS 353 Final Report

Prepared by CS353 - Section 1 - Group 15

Hüseyin Eren Çalık – 21402338

Ümitcan Hasbioğlu – 21402314

Kıvanç Gümüş – 21401767



## 1.INTRODUCTION

Trip TeeVee is a web-based application developed for creating a platform that will enable users to interact with their favorite TV shows and the system's other users in various ways. The system holds information of a wide variety of TV shows, each of their seasons, each of their episodes and current users of the system.

There are 4 types of different user groups which are; guests, regular users, Vip users and admins accordingly. Guests are users that haven't registered yet. Each user can view TV shows, seasons and episodes. To be able to initiate with the system, each users first have to login.

Regular users can mark TV shows, seasons and episodes as watched, give rating each of them out of 10 points and comment on each of them. They can also add other users as friends, display their activity on a separate page in which, the activities are ordered by their date and comment on others activity.

Admins are obliged to maintain the well-being of the application. They can view every user and their comments, modify or delete these comments depending on their suitability. Admins are also able to modify direct databases by adding or deleting TV shows, seasons and episodes.

Vip users, additional to regular users are able to search and view seasons/ episodes and view each item on a calendar. They can also add a biography(brief information about themselves) and a profile image which can be displayed by other users.

To sum up, Trip TeeVee is an online application that is designed for everyday-life users that provides them an environment of easily accessible TV shows and a place for them to interact with people with same interests.

## 2. Relational Schemas

### 2.1. TV SHOWS

**Relational Model:**

Tv\_Shows(trackId, tName, tInfo, tWatch, tRating, tImage, category)

FOREIGN KEY: trackId references track (trackId)

### 2.2. SEASON

**Relational Model:**

Season(trackId, tId, sInfo, sNumber, nEpisodes, sWatch, sRating, sImage)

FOREIGN KEY: trackId references track (trackId)

FOREIGN KEY: tId references track (trackId)

### 2.3. EPISODE

**Relational Model:**

Episode(trackId, sId ,eNumber,eInfo,eImage,eWatch,eRating,eDate)

FOREIGN KEY: trackId references track (trackId)

FOREIGN KEY: sId references track (trackId)

### 2.4. ACTION

**Relational Model:**

Action(actionId, userId)

FOREIGN KEY: userId references user (userId)

## **2.5. *USER***

**Relational Model:**

User(userId, uName, uPassword, privateQuestion, privateAnswer, uEmailAddress)

## **2.6. *SPECIAL EPISODE***

**Relational Model:**

Special\_Episode(trackId)

FOREIGN KEY: trackId references episode(trackId)

## **2.7. *ADMIN***

**Relational Model:**

Admin(userId)

FOREIGN KEY: userId references user (userId)

## **2.8. *VIP***

**Relational Model:**

Vip(userId, bio, profileImage)

FOREIGN KEY: userId references user (userId)

## **2.9. *TRACK***

**Relational Model:**

Track(trackId)

## **2.10. WATCH**

**Relational Model:**

watch(actionId, userId, trackId, wDate)

FOREIGN KEY: actionId references action(actionId)

FOREIGN KEY: userId references user(userId)

FOREIGN KEY: trackId references track(trackId)

## **2.11. RATE**

**Relational Model:**

rate(actionId, userId, trackId, rating, rDate)

FOREIGN KEY: actionId references action(actionId)

FOREIGN KEY: userId references user(userId)

FOREIGN KEY: trackId references track(trackId)

## **2.12. COMMENT**

**Relational Model:**

comment(actionId, userId, trackId, comment, cDate)

FOREIGN KEY: actionId references action(actionId)

FOREIGN KEY: userId references user(userId)

FOREIGN KEY: trackId references track(trackId)

## **2.13. COMMENTTOCOMMENT**

**Relational Model:**

commentToComment(actionId, userId, friendId, fActionId, comment, ctocDate)

FOREIGN KEY: actionId references action(actionId)

FOREIGN KEY: userId references user(userId)

FOREIGN KEY: friendId references user(userId)

FOREIGN KEY: fActionId references action(actionId)

**2.14. FRIEND**

**Relational Model:**

Friend(userId ,friendId)

FOREIGN KEY: userId references user(userId)

FOREIGN KEY: friendId references user(userId)

### 3. IMPLEMENTATION DETAILS

#### Tools We Used

We used MariaDB for database management and InnoDB as our database engine.

We used Java to create the tables. For back-end development we used PHP with PHPStorm framework. For front-end development we used jQuery JavaScript framework, CSS and HTML.

#### Problems We Faced

- While filling tables with data, it was seen that the tables that included at least one foreign key created some issues. One of these issues was that mysql did not let us insert rows before inserting the specific foreign key column into the owner table of the foreign key. We solved this problem by dropping the foreign key constraint by using ALTER TABLE then doing our operations and adding the foreign key constraint back by using ALTER TABLE again.

```
ALTER TABLE tvshows DROP FOREIGN KEY trcktv;
INSERT INTO tvshows ...
ALTER TABLE tvshows ADD CONSTRAINT trcktv FOREIGN KEY
(trackId) REFERENCES track(trackId);
```

- We tried to defined triggers in our system for insertion operations. For instance, we tried to create a trigger which inserts the trigger into track table after the inserting rows into tvshows, season and episode tables. However, mysql did not let us use max() function on the triggered table since it is already used by the statement which invoked the trigger.

delimiter #

```
CREATE TRIGGER trigtrack AFTER INSERT ON tvshows FOR EACH ROW
BEGIN INSERT INTO track(trackId) SELECT MAX(trackId)+1 FROM
track; END#
delimiter ;
```

This trigger uses trackId and during insertion into the tvshows using track(trackId) we get an error. We have the correct code on the Triggers part.

## 4. ADVANCED DATABASE FEATURES

### SECONDARY INDICES

Since TV Shows can be searched using their names numerous times, we decided to add secondary indices to the tvshows table.

```
CREATE INDEX tvshows_index USING BTREE ON tvshows(tName);
```

### VIEWS

Displaying non-admin users on admin page.

```
CREATE VIEW user_view(id, name, email)
AS SELECT userId, uName, uEmailAddress
FROM user u
WHERE NOT EXISTS (SELECT userId FROM user NATURAL JOIN admin
a WHERE a.userId=u.userId)
```

Displaying actions of friends.

```
@id
CREATE VIEW friend_actions(actionType, userName,
actionText, date)
AS SELECT actionType, uName, actionText, date
CASE
WHEN wDate IS NOT NULL (SELECT "Watched", uName, "", w.wDate
FROM friend f, user a WHERE @id=f.userId AND
f.friendId=a.userId)
WHEN rDate IS NOT NULL (SELECT "Rated",
uName, CONVERT(varchar(10), r.rating), r.rDate FROM friend f, user
a WHERE @id=f.userId AND f.friendId=a.userId)
WHEN cDate IS NOT NULL (SELECT "Commented",
uName, CONVERT(varchar(500), c.comment), c.cDate FROM friend
f, user a WHERE @id=f.userId AND f.friendId=a.userId)
END
FROM action NATURAL JOIN watch w NATURAL JOIN rate r NATURAL
JOIN comment c
```

### TRIGGERS

-After deleting a tv show, seasons or episode row, the corresponding trackId entry is deleted from track table.

delimiter #

```
CREATE TRIGGER deletetvshows
AFTER DELETE ON tvshows FOR EACH ROW
BEGIN DELETE FROM track WHERE track.trackId = old.trackId;
END#
```

```

CREATE TRIGGER deleteseason
AFTER DELETE ON season FOR EACH ROW
BEGIN DELETE FROM track WHERE track.trackId = old.trackId;
END#
CREATE TRIGGER deleteepisode
AFTER DELETE ON episode FOR EACH ROW
BEGIN DELETE FROM track WHERE track.trackId = old.trackId;
END#
delimiter ;

```

-After inserting a new row into tvshows trackId's of the added rows are inserted into track table. Foreign key constraint problem is solved by dropping constraint.

delimiter #

```

CREATE TRIGGER trigtrack AFTER INSERT ON tvshows FOR EACH ROW
BEGIN
INSERT INTO track VALUES(new.trackId);
END#
delimiter ;

```

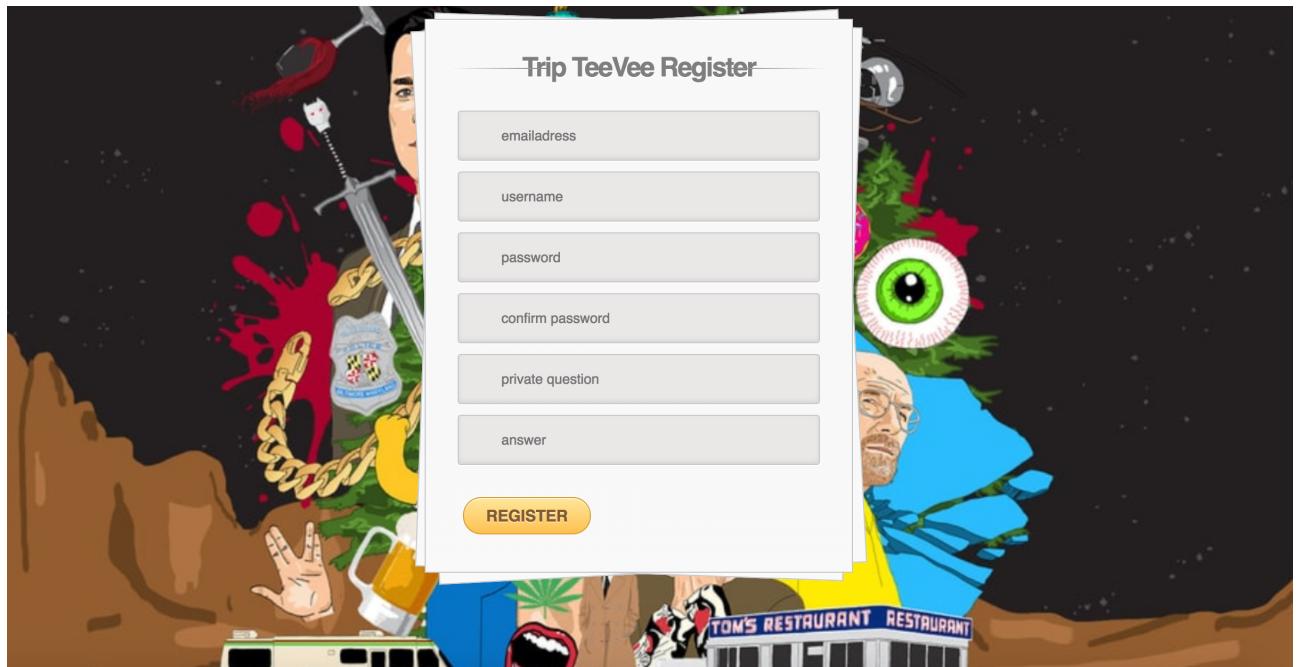
## 5. USER MANUAL

### Main Page

The homepage of the Trip TeeVee is opening page for all type use. They can see the most watched TV Show and the watch number of that particular show. Also, a registered user can sign in using the sign in button on the right corner and being directed to login page. If the user is not registered, by using join now button s/he can register to Trip TeeVee and being directed to register page.



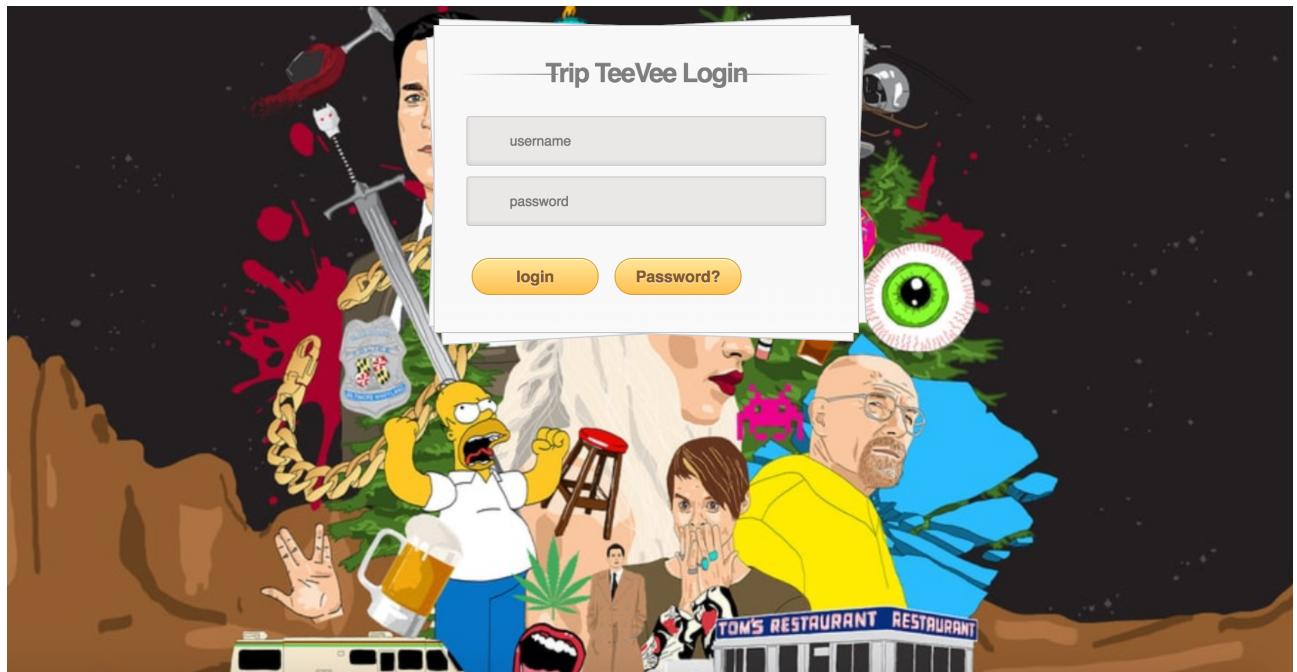
## Registering Page



After clicking on Join Now button, non-registered users can create an account by filling e-mail address, user name, password, confirm password, private question and the answer. The password and confirm password must be the same for successful register.

## Login Page

After clicking on sign in button, registered users can login providing e-mail address and password. Also, if the user forgot his/her password, a password saving section is available thanks to private question. After answering the private question, the current password is changed according to new password input.



## Change Password

A registered user can change their password by providing their email and answering his/her previously chosen private question.

Trip TeeVee

Most Watched TV Show  
Supernatural 1.000.002 watches

E-mail Adress  
Password  
I forgot my password

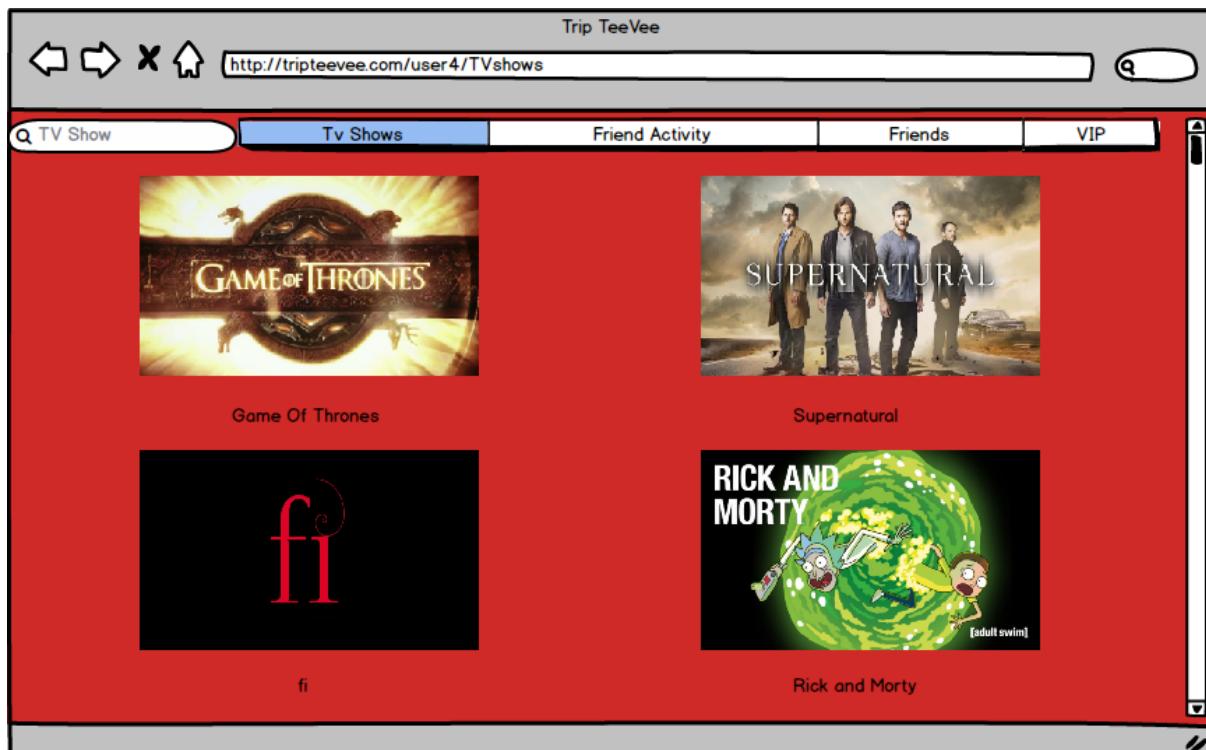
Private Question : Who is my favourite TA?  
Answer :  
New Password :

We Are Trip TeeVee  
We Are Here to Help You Schedule Your Tv Plan!

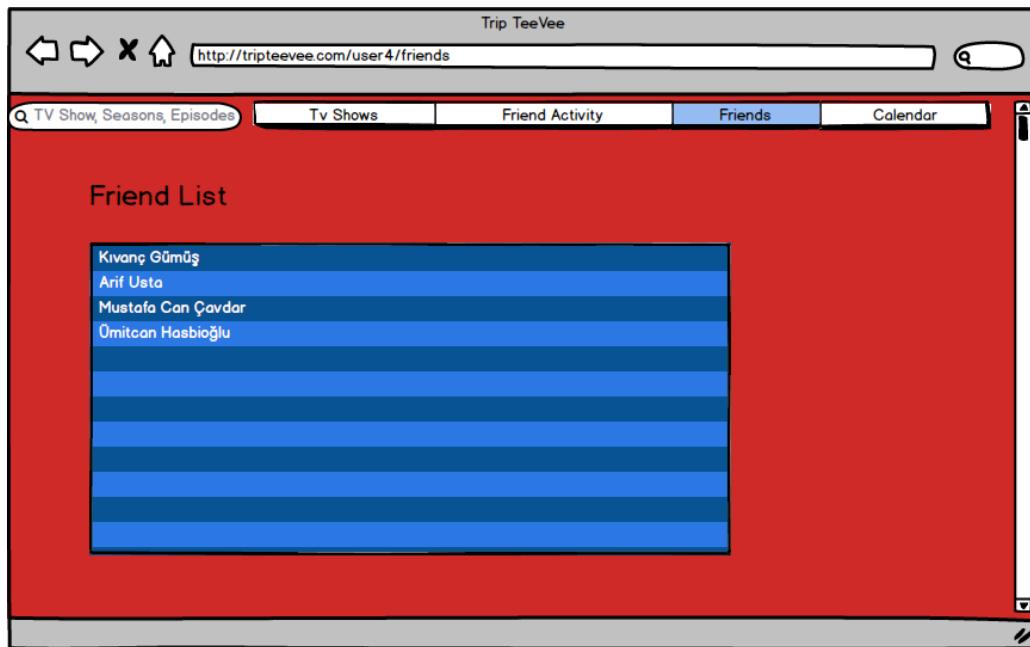
JOIN NOW >

## TV Show Page

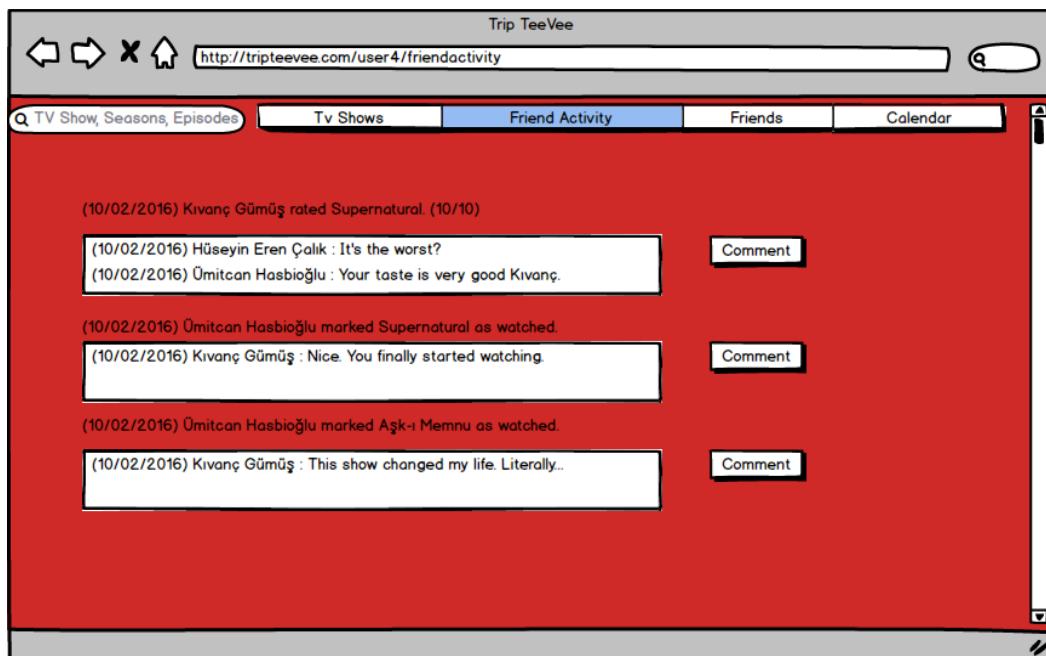
After logging into the system, Tv Show page is displayed. This page includes all the tv shows. Also, the user can search for a show via search engine on the left corner. Also, s/he can view his/her friend's activities, friends and vip page.



## Friends Page



Users can view their friends listed. By clicking a friend's name on the screen, s/he can list activities of his/her friend. Also, in that window the user can comment on that activities.



## VIP EXTRA FEATURES

In VIP user mode, s/he can display the tv shows, seasons and episodes with the help of calendar. Also, s/he can have profile which can be seen by the other users. This profile includes profile image and info. It can be changed and updated anytime by VIP.

### Selected Tv Show page

The screenshot shows the Trip TeeVee website interface for a selected TV show. At the top, there is a header bar with navigation icons (back, forward, search, etc.) and the URL <http://tripteevee.com/user4/supernatural>. Below the header, there is a navigation menu with tabs for 'TV Show' (which is active), 'Tv Shows', 'Friend Activity', 'Friends', and 'VIP'. The main content area has a red background. On the left, there is a large image of the TV show 'Supernatural' with the title 'SUPERNATURAL' overlaid. Below this image, the text 'Rate : 9.7/10' and 'Watched 1.000.002 times' is displayed. To the right of the image, there are three buttons: 'Mark Watched', 'Rate: 10/10' (with a dropdown arrow), and 'Season 3'. Below these buttons, there are three smaller images representing Season 1, Season 2, and Season 3. Underneath the show image, there is a section titled 'About The Show' with a short description and a 'Comment' button. Further down, there is a section titled 'Audience Comments' with a comment from a user named 'Mustafa Can Çavdar'. The entire interface is styled with a clean, modern look with a red and white color scheme.

After selecting a tv show, seasons of that tv show are listed. A photo of that tv show is shown on the left corner. Its overall rate and number of times watched is displayed. Also a brief information and audience comments are below that photo. The user can mark it as watched by clicking, give a rate to it via select box (numbers are between 1-10) or comment to directly it or comment to his/her friend's comment. While commenting, a pop-up screen occurs. Lastly, by clicking on a season, the system directs the season page of that tv show.

## Season Page of a Tv Show



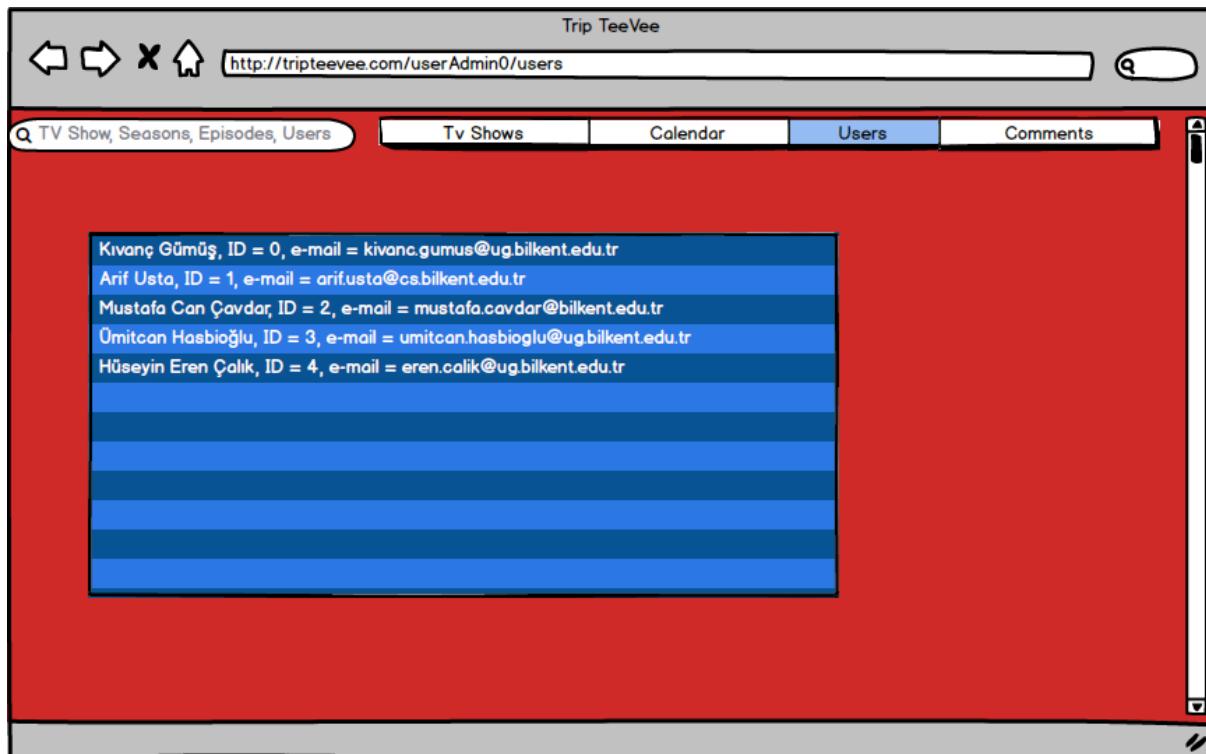
After selecting a season, episodes of that season are listed. A photo of that season is shown on the left corner. Its overall rate and number of times watched is displayed. Also a brief information and audience comments are placed below that photo. The user can mark it as watched by clicking, give a rate to it via select box (numbers are between 1-10) or comment to directly it or comment to his/her friend's comment. While commenting, a pop-up screen occurs. Lastly, by clicking on a episode, the system directs the episode page of that season.

## Episode Page of a Tv Show

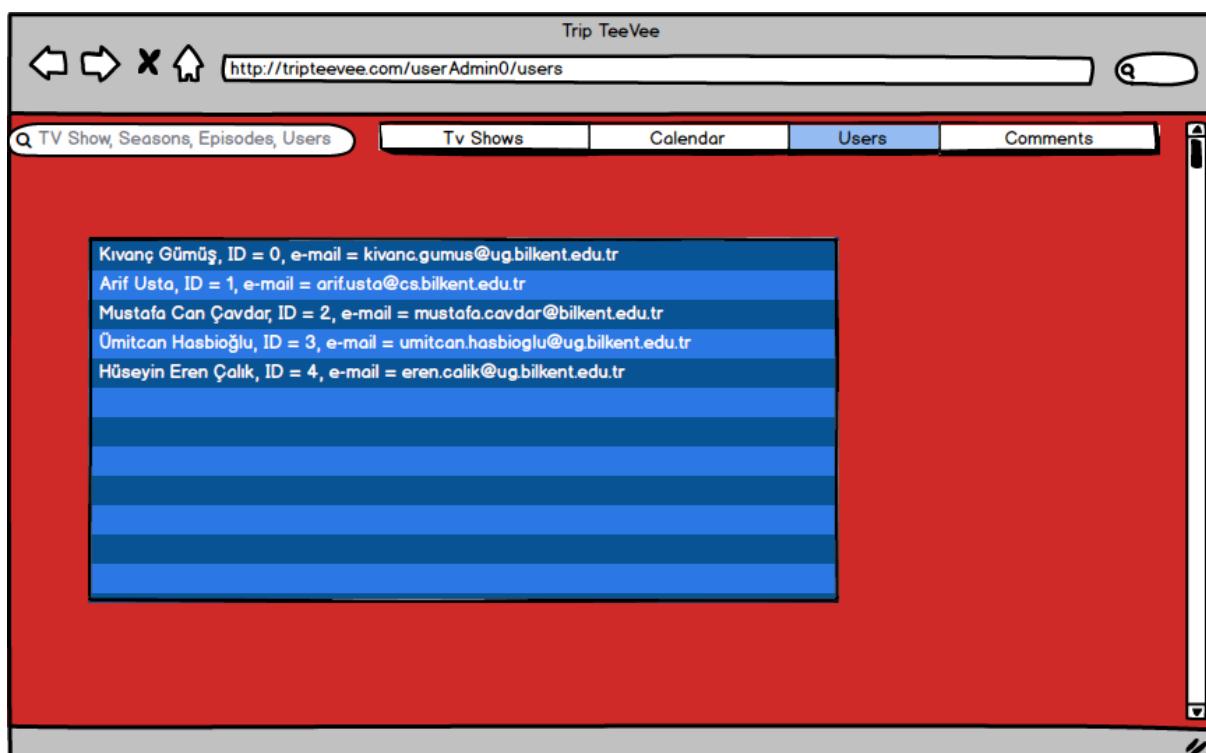


After selecting an episode, a photo of that episode is shown on the left corner. Its overall rate and number of times watched is displayed. Also a brief information and audience comments are placed below that photo. The user can mark it as watched by clicking, give a rate to it via select box (numbers are between 1-10) or comment to directly it or comment to his/her friend's comment. While commenting, a pop-up screen occurs.

## Comments Page of Admin



The admins have four sections which are tv shows, calendar, users and comments. In comment window they can view all the comments and user information of the commenters in the user window. They can also delete and modify the comments. In user window, they can view all the users and delete them.



In Calendar window, they can modify the dates of the tv shows, seasons and episodes. Lastly, in tv how section, they modify or delete tv shows, episodes and seasons.