Gjøvik University College

Web Technology

EpisodeGuide

Authors:
Halvor M. Thoresen - 120915
Tommy B. Ingda - 120913
Victor Rudolfson - 120912

March 14, 2014

Contents

1 Abstract

EpisodeGuide is a web application that help users keep up to date on their favourite TV-shows. Both registered users and non-registered users can use this application, though registered users will get more features.

Features include, but are not limited to:

- Find new TV Shows.
- Keep track of episodes watched so far.
- Share TV Shows and episode data on social media (e.g Facebook).
- Login via Facebook, Twitter, Google etc.
- ... And more

What separates this application from all the others is simplicity. This was the main goal when we started developing EpisodeGuide, and is something we believe our users will benefit from in the long run. Both end users and developers should be able to use and further develop EpisodeGuide in the

future without going through several hundred pages of complicated documentation.

EpisodeGuide uses different APIs, technologies and frameworks to accomplish this task.

APIs and frameworks:

- TvDb for data about new and existing TV-shows and episodes
- Foundation as a framework for layout, box models etc.
- OpenSubtitles for subtitles

Technologies:

- PHP
- MySQL
- Javascript

It is worth mentioning that EpisodeGuide is an application in developement. This means that at this stage, some features are currently not available or implemented.

Since all the developers on our team believe that OpenSource is the best way to release and distribute new applications, EpisodeGuide will be OpenSource and available on Github for download once it is ready for a public release.

You can test EpisodeGuide live at: http://flawedlogic.pw/flawedlogic/bugfree-shame/

2 Overview

2.1 Database

2.1.1 Introduction

The database should contain all necessary tables for fetching, storing and displaying TV-shows and their episodes, as well as all attributes related to these that may be of interest to the user, such as original pilot date, amount of episodes, and details for each episode.

There must also be supporting tables, such as *country*, *roles* and *channel* which must exist for the database to be in 3NF and allow values to be atomic. To allow a *user* to mark whether he or she is actually watching a *show*, relational tables such as *user_show* must also be created, and for a future administration interface, *parameter* needs to exist for storing and managing site-wide configuration without manually editing the configuration file.

Furthermore, additional functionality which has not yet been fully implemented must be supported by the database, i.e tables *url* and *subtitle*, which are meant to contain links to where a specific episode can be watched or downloaded, as well as where subtitles may be found.

2.1.2 Logical design model

actor (actor_id, imdb_id, poster_url, name)

Primary Key actor_id

channel (id, name, country_id)

Primary Key id

Foreign Keys
country_id **references** country(id)

country (id, name, language)

Primary Key id

episode (show_id,season,episode,name,summary,date)

Primary Key show_id, season, episode

Foreign Keys
show_id **references** show(id)

parameter (id,key,value)

Primary Key id

roles (id,name,description,is_admin)

Primary Key id

show (id,imdb_id,zap2_id,channel_id,poster,pilot_date,name,summary,lang,rating,lst_update)

Primary Key id

subtitle (show_id,season,episode,url)

Primary Key show_id, season, episode

Foreign Keys
show_id **references** show(id)
season **references** episode(season)
episode **references** episode(episode)

url (show_id,season,episode,url)

Primary Key show_id, season, episode

Foreign Keys
show_id **references** show(id)
season **references** episode(season)
episode **references** episode(episode)

user (id,email,password,salt,role_id,country_id,last_activity)

Primary Key id

Foreign Keys
role_id **references** roles(id)
country_id **references** country(id)

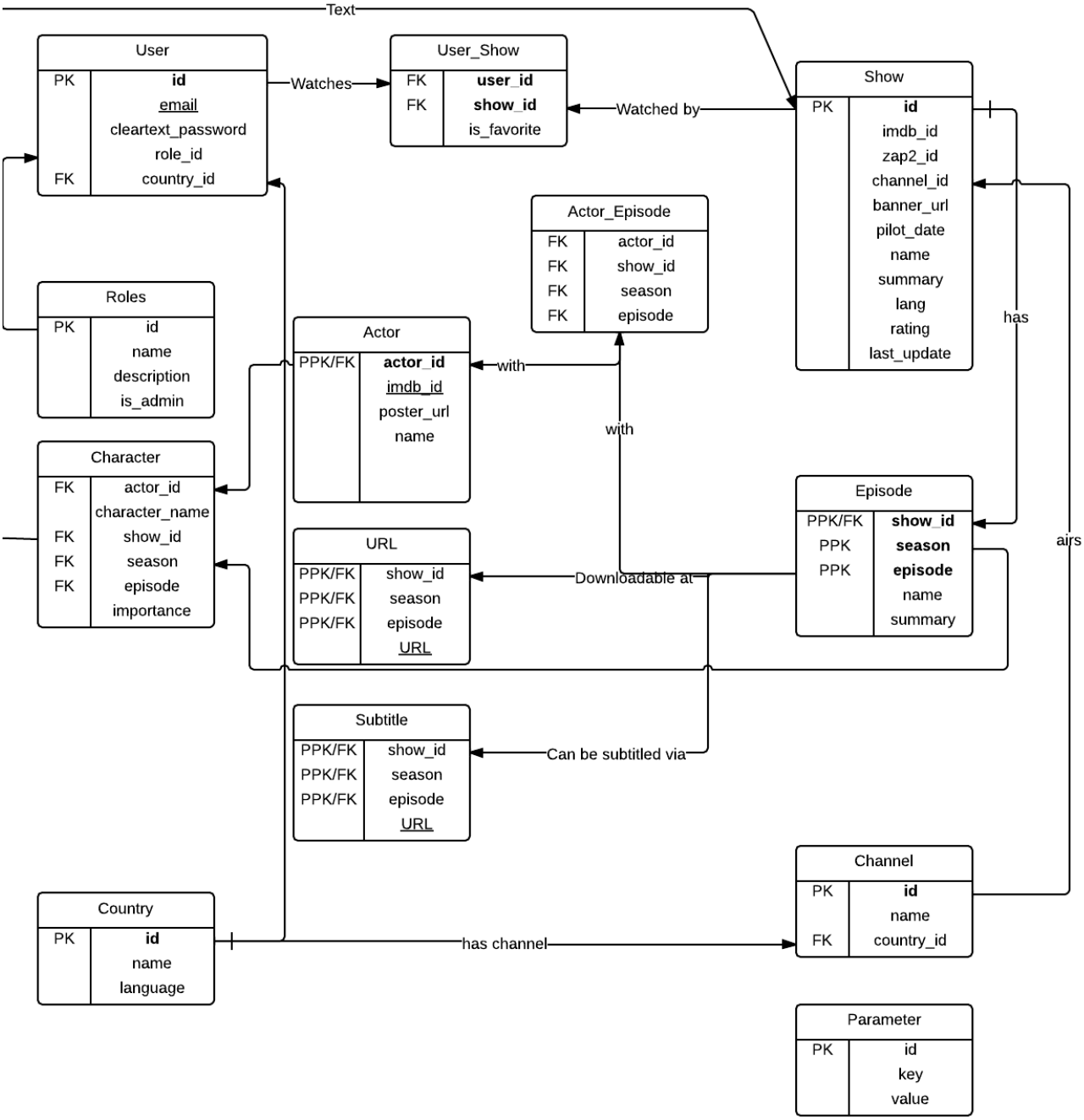
user_session (id,session_data,session_ip)

Primary Key id

user_show (user_id,show_id,is_favorite)

Foreign Keys
show_id **references** show(id)
user_id **references** user(id)

2.1.3 Entity-Relationship Diagram



3 Worklog

3.1 Week 1

The project startet off with a meeting. After about two hours of brainstorming we had a couple of ideas of what we could do. We then gave ourself until the next day to think about the alternatives.
The next day we discussed wich of the ideas would be the most appropriate for our combined skill level and the time we had. After a little while, we came to the conclusion that making a website that holds information about tv-shows would be both challenching and interesting.

The rest of the week was spent on casually brainstorming ideas about the website and writing them down. ...

3.2 Week 2

The beginning of week 2 on spent on discussing exactly what kind of functionality we wanted and what tasks that had to be done, and in what order. After we had a clear overview about what had to done, we started laying out a plan on how we would share the task between eachother. ...

3.3 Week 3

At this point we felt like we had done enough planning and was quite ready to start coding. Each of us had our tasks and worked on those task seperatly for about a week to get the most essential building stones up and running.
...

3.4 Week 4

when most of the fundemental part where done we could start coding functionality. A lot of time was spent on including a functionality and then adapting the rest of the code to work with the functionality through a lot of bug testing. We had a list of functions we wanted to implement, so we picked at will what we wanted to do. When we where done with one task, we picked another one. ...

3.5 Week 5

Week 5 were mostly spent on finishing up as many functions as we could manage, fixing as many bugs as we could and make the code as efficient as possible.

An overview over who had the most to do with most of the files:

Tommy	Halvor	Victor
configurationClass.php	tvdbClass.php	activeRecord.php
databaseClass.php	fileHandlerClass.php	channelClass.php
index.php	emailSenderClass.php	episodeClass.php
userClass.php		showClass.php
tpbClass.php		table.php
interfaces		browse.php
includeClass.php		details.php