

# FlixKiller – Elokuvasivusto

ver 22.05.2014

Marko Mäkinen  
markoma@iki.fi

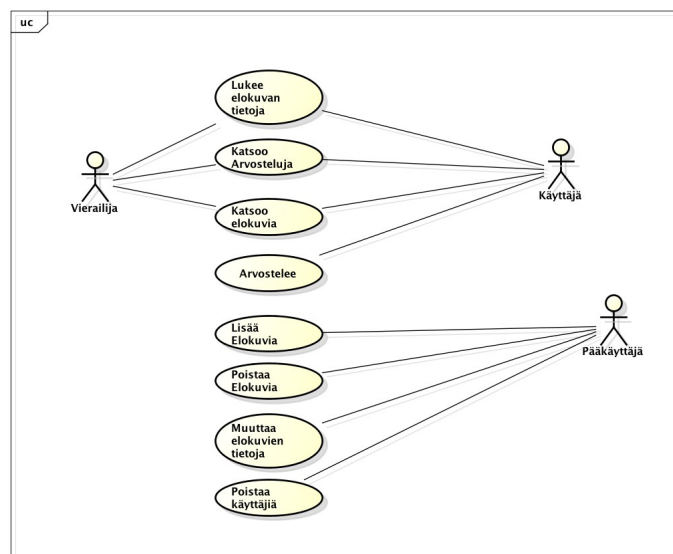
## 1. Johdanto

**FlixKiller** on yksinkertainen videopalvelu, jossa käyttäjät voivat katsoa ja arvostella elokuvia. Ohjelma ei itsessään tallenna elokuvia, vaan harjoitustyössä on ideana pitää yllä vain linkkiä elokuvaan. Esimerkkielokuvina käytetään open source -statuksen omaavia elokuvia, kuten Big Buck Bunny ja Sita Signs the Blues.

Tavoitteena on luoda edesmenneen Popcorn time:n tyylinen sivusto, jossa on elokuvan luokkaan/tagiin perustuva suodatus. Ohjelmointityö toteutetaan Scalalla Play Frameworkia käyttäen. Ohjelmiston demo julkaistaan Herokussa, joka tukee kieltä ja alustaa hyvin. Tietokantana käytetään Postgressiä. Ulkoasusta vastaa custom bootstrap.css.

## 2. Yleiskuva järjestelmästä

Jokaisella elokuvalla on yleiset tiedot tietokannassa, kuten nimi, julkaisuvuosi, kuvaus, linkki elokuvaan sekä luokka. Ohjelmalla on rekisteröityneitä käyttäjiä, jotka voivat arvostella elokuvia, sekä kerätä elokuvia katselulistalle. Jotkut käyttäjistä voivat olla pääkäyttäjää. Pääkäyttäjät voivat lisätä ja poistaa elokuvia ja käyttäjiä sekä muokata elokuvien tietoja.



## KÄYTTÄJÄRYHMÄT

### *Vierailija*

Tarkoitetaan ketä tahansa, joka käy sivustolla.

### *Käyttäjä*

Käyttäjällä tarkoitetaan rekisteröitynyttä vierailijaa. Käyttäjät voivat arvostella elokuvia.

### *Pääkäyttäjä*

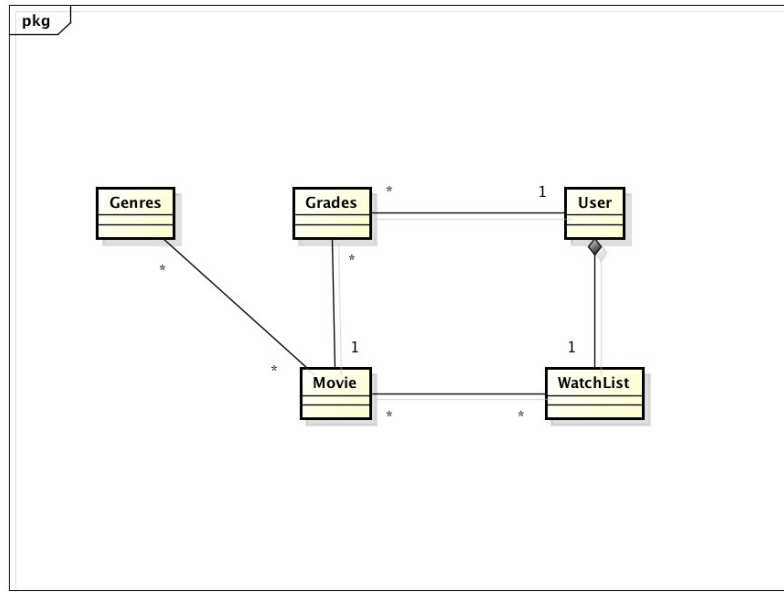
Tarkoitetaan käyttäjää, joka voi lisätä ja poistaa elokuvia ja käyttäjiä, sekä muokata elokuvien tietoja.

Tavoitteena on luoda mahdollisimman pitkälle yhden sivun järjestelmä. Sivun vasemmassa laidassa on suodatustoiminnot. Sivun ylälaidassa on klassinen navigointipaneeli, jossa on logo, hakupalkki, rekisteröityneen käyttäjän kirjautumistoiminnot, sekä linkki sivulle jossa uusi käyttäjä voi rekisteröityä. Sivun pääasiallisena sisältönä on responsiivisiin grideihin jakautunut elokuvien selausnäkymä, jossa jokaisesta elokuvasta on esillä nimi ja kansikuva. Klikkaus vie elokuvankatselusivulle, jossa on tarkemmat tiedot elokuvasta, kuten nimi, kuvaus, arvostelu sekä tietysti elokuvasoitin.

Toivottuina lisäominaisuuksina voisi olla rekisteröityneen käyttäjän mahdollisuus lisätä elokuva katselulistalle sekä mahdollisuus arvostella elokuva.

### 3. Järjestelmän tietosisältö

#### KÄSITEKAAVIO



#### TIETOKOHDE: USERS

userid (Int)	Username (String)	Userpassword (String)	Admin (Boolean)
1	'Admin'	'1234'	true
2	'PeterJackson'	'dolly'	false

Järjestelmä tallentaa tiedon käyttäjästä, jossa 'userid' on tietokannan luoma uniikki tunniste, 'username' on käyttäjän antama käyttäjänimi, 'userpassword' on käyttäjän antama salasana, 'admin' on boolean arvo joka kertoo millaiset oikeudet käyttäjällä on.

#### TIETOKOHDE: GRADES

users_userid (Int)	movies_movieid (Int)	Stars (Int)
2	1	6
2	2	8

Grades on monimoneen yhteys taulujen 'users' ja 'movies' välillä, jossa on lisäsarakeena 'Stars'. Taulun avulla voidaan tallentaa käyttäjän antama arvio elokuvasta asteikolla 0-10.

## TIETOKOHDE: MOVIES

movie id (Int)	Movietitle (String)	Year (Int)	Plot (String)	Lenght (String)	Addingdate (date)	Director (String)	Writer (String)	Actors (String)	Link (String)	Coverimg (String)
1	'Hobitti'	2013		3:15	current_date	'Peter Jackson'			Www.linkToMovie.com	Www.LinkToCover.com

'movies' on laaja taulu, joka tallentaa kattavat tiedot elokuvasta. Elokuviin liittyy luokkia ('genres'), elokuva voi kuulua käyttäjän katselulistaan ('watchlist') ja elokuvalla voi olla käyttäjän antava arvostelu('grades').

## TIETOKOHDE: GENRES JA VÄLITAU LU GENRES\_HAS\_MOVIES

Genreid (Int)	Genrename (String)
1	'action'
2	'comedy'
3	'documentary'

genres_genreid (Int)	movies_movieid (Int)
1	1
2	1
3	2

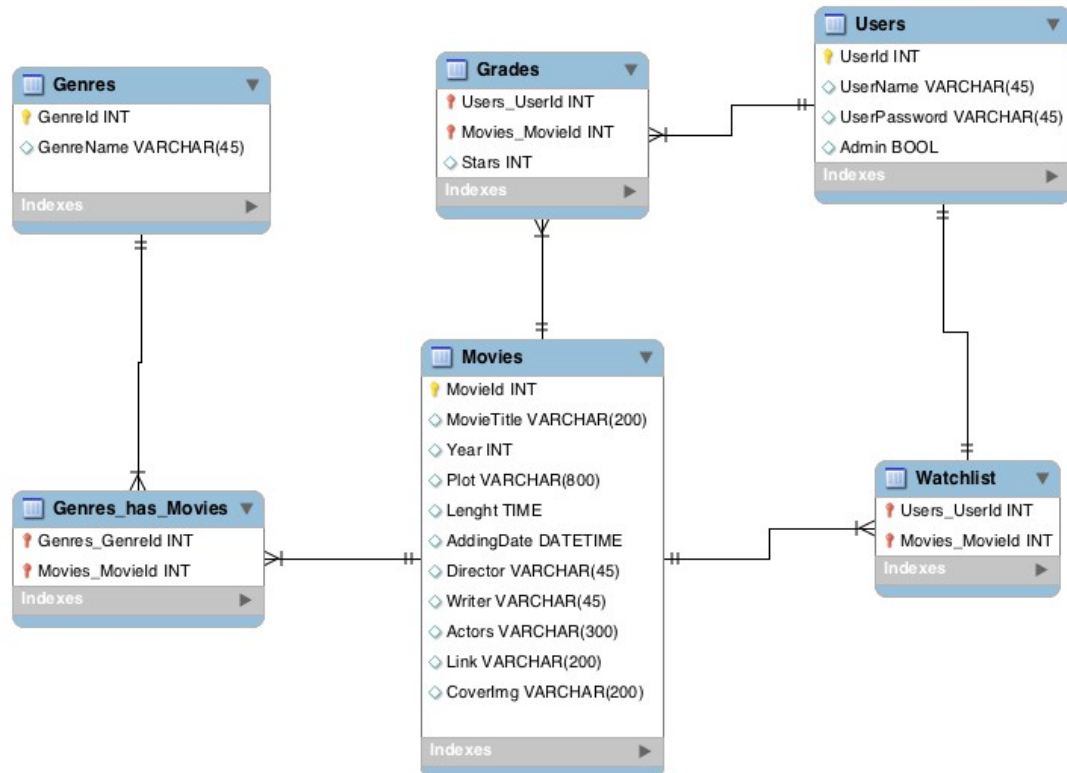
Jokaisella elokuvalla voi olla luokkia, ja jokainen luokka voi kuulua moneen elokuvaan. Luokat luodaan 'genres' tauluun ja yhdistetään elokuviin 'genres\_has\_movies' taulun avulla.

## Tietokohde: watchlist

users_userid (Int)	movies_movieid (Int)
1	1
1	2

Käyttäjällä voi olla katselulista, johon käyttäjä voi tallentaa elokuvia, joita hän haluaa katsella tulevaisuudessa tai on katsellut. On vielä hieman epäselvää, onko käyttäjällä yksi vai monia katselulistoja. Projektin alussa vain yksi.

#### 4. Relaatiotietokantakaavio



## 5. Järjestelmän yleisrakenne

Ohjelma toteutetaan Scala play -framework'in tuottamalla valmiilla ohjelmapohjalla, joka on MVC-mallin mukainen.

Rakenne on seuraava:

app/

controllers/

## Admin.scala (pitää sisällä ylläpidon funktiot)

Application.scala (ohjaa kirjautumissivulle/etusivulle)

Auth.scala (käyttäjän tunnistautuminen)

Movies.scala (elokuva sivut)

models/

Aspirant.scala (ei hyväksytty käyttäjä)

Genre.scala

## Movies.scala

User.scala

views/

main.scala.html (tämä on kaiken html renderöinnin pohjana)

movies/

-kaikki elokuvaluokan näkymät

Admin/

- kaikki elokuvien ja käyttäjien muokkaukseen ja editointiin liittyvät näkymät

conf – hakemistossa on tietokantayhteyteen liittyvät tiedot, sekä routes -tiedosto.

Lib – Hakemistossa on tietokanta-ajuri: postgresql-9.3-1101.jdbc3.jar

Public – Hakemistossa on bootstrapin css tiedostot, sekä itse muokattu main.css

Juurikansiossa sijaitseva public.sbt tiedosto pitää sisällään ohjelman tarvitsemat kirjastoriippuvuudet.

## ISTUNTO

Käyttäjän kirjautuessa sivustolle luodaan istunto, jossa käyttäjän koneelle tallennetaan eväste. Eväste kertoo istunnon voimassaolosta. Istunto on voimassa, kunnes käyttäjä käynnistää selaimen uudelleen, tai ulos kirjautuu sivustolta 'log out' painikkeen kautta.

Istunnon validointi lisätään vasta projektin lopussa, sillä jatkuva kirjautuminen hidastaa kehitystyötä.

## JAVASCRIPT

Ohjelmassa käytetään vain bootstrap:n tarvitsemaa javascript -tiedostoa. Ehkä lopullisessa muokkauksessa perus jquery -pakettia.

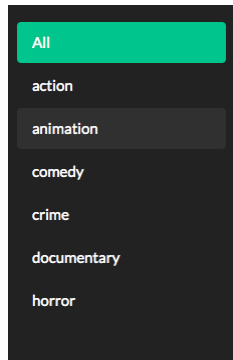
## 6. Käyttöliittymä ja järjestelmän komponentit

Käyttöliittymästä tulee hyvin yksinkertainen. Käyttöliittymän ulkoasu on toteutettu kokonaisuudessaan Bootstrap:lla. Vierailijalle ja rekisteröityneelle käyttäjälle on lähestulkoon sama näkymä. Etusivulla on yläpalkki, jossa on toiminnot kirjautumista varten. Sivun vasemmassa laidassa on elokuvien suodatustoiminnot. Muu sisältö täyttyy 'elokuvaalaatikoista', joissa on elokuvan kansikuva ja nimi. 'Elokuvalaatikko' on linkki elokuvanäkymään, jossa elokuvasta voi lukea tarkemman tiedot, arvostella elokuva, lisätä katselulistalle sekä tietysti katsoa elokuva.

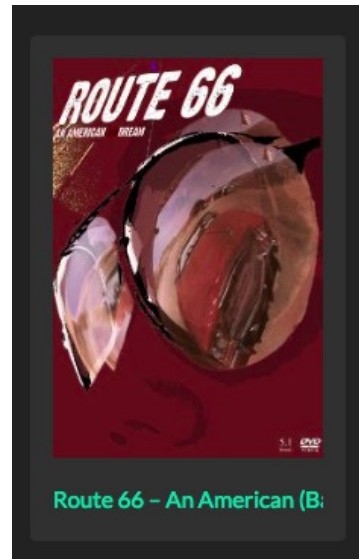
## NAVIGAATIOPANEELI JA KIRJAUTUMINEN

FlixKiller	admin	1234	Login
------------	-------	------	-------

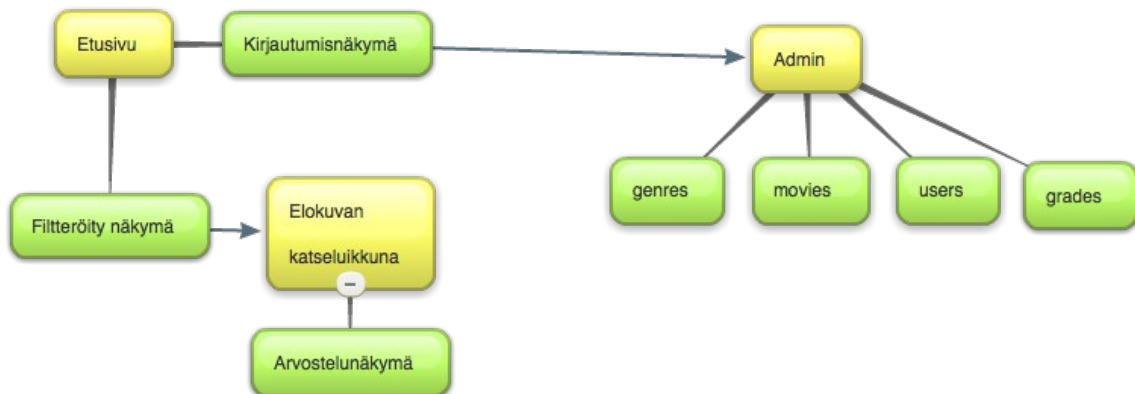
## ELOKUVIEN SUODATUS



## ELOKUVALAATIKKO



## KÄYTTÖLIITTYMÄKAAVIO



'Admin', eli pääkäyttäjän hallintapaneelissa on omat näkymät sisällön muokkaukseen, jonne pääsee vain käyttäjät joille on annettu oikeudet.

## KONTROLLERIT

CONTROLLER	SIVU	JULKISUUS
Admin.scala	Admin hallintapaneeli	Pääkäyttäjät
Application.scala	Ohjaa etusivulle	-
Auth.scala	Vastaa sisäänkirjautumisesta	Kaikki
Movies.scala	Vastaa elokuvanäkymistä	Kaikki

## 7. Asennustiedot

Järjestelmä vaatii tällä hetkellä käyttöönsä Java JDK 6 tai myöhemmän. Tämän hetkessä kehitysympäristössä on käytössä Java 1.7. Scala versiona on 2.10.3. Scalan rakennustyökalu Scala-sbt (build tool) versiona on 0.13.. Play framework:sta on käytössä tällä hetkellä versio 2.2.3.

Tietokantaohjelmistosta vastaa PostgreSQL 9.3.4. Tietokannan asennus tapahtuu conf/application.conf tiedostosta, jonne kirjataan tietokantaohjelmiston ajuri sekä osoite kantaa. Itse ajuri tallennetaan lib -hakemistoon. Tietokantaohjelmiston kirjastovaativuudet kirjataan juurihakemiston build.sbt tiedostoon.

Play:sta on hiljattain julkaistu versio 2.3, johon siirtyminen tulee ajankohtaiseksi luultavasti projektin aikana. Samalla päivittyy Scala ja Scala-SBT. Scala SBT version saa vaihdettua projects/build.properties tiedostosta.

Kehitystyö on toteutettu mac osx 10.9. käyttöjärjestelmällä ja demoversiot ohjelmistosta on julkaistu Heroku:ssa osoitteessa [flixkiller.herokuapp.com](http://flixkiller.herokuapp.com).

## 8. Asennus- ja käyttöohje

### *Asennusohje*

Kuten edellä mainittiin, on sovellus kehitetty ios ympäristössä, mutta luultavasti sama ohje pätee myös Linux laitteistossa. Tämä yleisohje koskee siis kehitysympäristöä.

- Järjestelmässä tulee olla JDK6 tai uudempi. Typesafe Activator, Scala ja Play asennusohje löytyy Playframeworkin kotisivuilta:

<https://www.playframework.com/documentation/2.3.x/Installing>

- Kloonaa Git repositorio koneellesi
- Käynnistä Play komennolla `play ~run`
- Avaa selain localhost:9000

### *Käyttöohje*

Sovellukseen pääsee kirjautumaan vasemman ylälaidan login -painikkeen kautta. Testi pääkäyttäjä tunnukseksi on admin ja salasana 1234. Ei-pääkäyttäjän tunnus on noadmin ja salasana 1234. Pääkäyttäjä ohjataan pääkäyttäjän ohjauspaneeliin, jossa on käyttäjien ja elokuvien muokkaustoiminnot. Ei-pääkäyttäjä ohjataan etusivulle, josta on mahdollisuus päästä käyttäjän tietojenmuokkaussivulle vasemmassa ylälaidassa sijaitsevasta käyttäjänimipainikkeesta.

## 9. Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

### *Testaus ja tunnetut bugit*

Ohjelmiston toimintaa on testattu kattavasti kehityksen aikana. Erillisiä testejä ei ole kirjoitettu, sillä ohjelman toiminta on hyvin yksinkertainen.

Tiedossa on vain visuaalisia bugeja, kuten kansikuvien huono latautuminen ulkoisilta palvelimilta.



## *Puutteet*

Kaikkia alkuvaiheessa suunniteltuja toiminnallisuuksia ei ole otettu vielä käyttöön, kuten elokuvien arvostelu, katselulista ja hakutoiminto.

- Elokuvien arvosteluun löytyy hyviä Javascript kirjastoja (google: js star rating) ja niiden käyttöönotto sovellukseen on ilmeisen yksinkertaista.
- Käyttäjän katselulistaa ei ole keretty vielä toteuttamaan.
- Elokuvien kansikuvien lataaminen serverille nopeuttaisi sivujen latautumista.

## *Jatkokehitys*

*Sovelluksen jatkokehitys riippuu täysin halutusta tuloksesta. Sovellusta voitaisiin laajentaa Netflix -tyyppiseksi maksulliseksi videopalvelusivustoksi. Tällöin sivustolle pitäisi toteuttaa kattavampi käyttäjäprofiili- ja maksujärjestelmä.*

*Sovellusta voisi lähes tällaisenaan hyödyntää esimerkiksi elokuvafestivaalien trailerisivustona.*

*Uusien käyttäjien kirjautuminen on helppo toteuttaa sosiaalisella kirjautumismoduulilla <http://securesocial.ws>, jolloin käyttäjät pääsisivät nopeasti arvostelemaan ja kommentoimaan videoita.*

## **10. Omat kokemukset**

*Sovelluksen toteuttaminen Scala-Play -kokoonpanolla on ollut erityisen mielenkiintoista. Kokemukseni Scalalla ohjelmoinnista ennen projektia oli hyvin nimellistä. Kehityksen aikana Scalalle tyypilliset funktionaalisen ohjelmoinnin periaatteet tulivat hyvin tutuiksi. Funktioliteraalien käyttö on hyvin hienostunut ja tyylikäs tapa suorittaa haluttuja toimintoja, samoin mielestäni koodin ulkoasu on hyvin siistiä muutamaa suurempaa metodia lukuun ottamatta. Play frameworking oppimiskynnys on hyvin pieni ja sen perusteet on hyvin dokumentoitu.*

*Play Framework tuntui toimivan suuremmilta osin todella hyvin. Sen virheilmoitukset olivat lähes aina hyvin tulkittavia. Yksi suurimmista puutteista oli html -lomakkeen monivalintatoiminnon keuhko toteutus ja lopulta teinkin siihen oman ratkaisun. Tämä oli myös erittäin huonosti dokumentoitu Playn sivuilla. Ongelma oli monivalinnan katoaminen lomakkeen virhetilanteessa.*

*Samoin yksinkertainen käyttäjien uudelleenohjaus perustuen käyttäjärooliin oli vaikea toteuttaa ja lopputulos on hieman bruteforcea käyttävä. SecureSocial moduulissa on vastaava suodatustoiminto perustuen käyttäjärooliin. Playn dokumentaatio tietoturvasta oli myös mielestäni hyvin ohut, ja minulle jäi epäselväksi, onko cookieihin perustuva käyttäjän varmistus toimiva tapa, ja kuinka saan salasanojen salauksen toteutettua by-the-book.*

*En myöskään ymmärtänyt miksi routes ei osaa välittää parametrina muita kuin Scalan perustyyppisiä. Sovelluksessani esimerkiksi elokuvaolion välittäminen olisi ollut erittäin hyödyllistä ja säästänyt muutamia SQL -kyselyitä. Muutoin tämä reittien staattinen tyyppitys vaikutti hyvin varmalta toteutustavalta.*

*Tein kaiken ohjelmoinnin Sublime tekstieditorilla. En testannut ollenkaan Playn IDE tukea, mutta ilmeisesti on toimiva.*