

TALLINNA TEHNIKAÜLIKOOL

Ingotehnoloogia teaduskond

Informaatikainstituut

Tarkvaratehnika õppetool

Esietenduste piletite müügile ilmumisest teatamine emaili kaudu

Bakalaureusetöö

Üliõpilane: Aleksandr Tsarjov

Üliõpilaskood: 083907IAPB

Juhendaja: Kaarel Allik

Tallinn

2015

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

(kuupäev)

(allkiri)

Annotatsioon

Terve rakendus on seotud ainult Tallinna kinoteatriga. Selle töö eesmärk on ehitada rakendus, mis võimaldab kasutajaid teavitada reaalajas interneti kaudu uue filmi esietenduse piletite müügile saabumisest ja seetõttu lahendada esietenduse piletite ilmumise probleem kinofännide seas. Rakendus võimaldab saata teateid kõigile kasutajatele piletite ilmusest ja selle filmi esietenduse kava vaatamisest e-maili kaudu. Selles töös on kirjeldatud funktsionaalseid ja ka mittefunktsionaalseid nõudeid, süsteemi ülesehitust, kasutusjuhiste diagramme, andmebaasi struktuuri ja on loodud klasside diagramme. Töös on olemas toote näide, kuidas rakendus veebilehel välja näeb. Kõik püstitatud eesmärgid said täidetud ja töö käigus loodi veebirakendus. Autori teostatud töö näitab, et veebirakendus on efektiivne vahend interneti kaudu teavitamisest piletite müügile tulemisest kinofännidele. Lõputöö on kirjutatud vene keeles.

Lõputöö on kirjutatud vene keeles ja sisaldab 34 lehekülge, 5 peatükki, 13 joonist.

Abstract

This realization is related only to Tallinn theater. The aim of this work is to create a web-application which solves the problem of movie-fans - waiting and not knowing when tickets for premiere of desired movie will appear. Authors work will allow customers to select desired movie which will appear soon and subscribe for notifications which will come by mail.

In this work are described functional and nonfunctional requirements, system architecture, Use Case diagram, Data base structure and presents diagram of common classes, which were created by author.

All goals, which were put by author, where successfully achieved and in result occurred proper and user-friendly web-application which suites exactly for notifying customers about arose tickets for a movie.

This work was written in Russian and contains of 34 pages, 5 chapters, 13 pictures.

Аннотация

Данная реализация относится только к таллиннскому кинотеатру. Главной целью работы является построение веб приложения, которое способно оповещать пользователей о появлении билетов, на премьеру фильма, в продаже, что решает главную проблему кинофанатов относительно ожидания билетов в продаже на любимый фильм. Реализация позволяет выбрать желаемый фильм и подписаться на оповещение при наличии билетов по почте.

В данной работе были описаны функциональные и нефункциональные требования, архитектура системы, структура базы данных и иллюстрация диаграммы общих классов. Продемонстрирован внешний вид веб-приложения.

Все цели, поставленные в ходе работы, были выполнены. В результате был создан прототип приложения. Прделанная автором работа показала, насколько данное приложение удобно для пользователя и эффективно в рамках поднятой проблемы.

Работа оформлена на русском языке. Она состоит из 34 страниц, 5 глав и 13 рисунков.

СПИСОК СОКРАЩЕНИЙ И ТЕРМИНОВ

DAO, Data Access Object - объект, который предоставляет абстрактный интерфейс к какому-либо типу базы данных или механизму хранения [4]

HTTP, HyperText Transfer Protocol - протокол прикладного уровня передачи данных[5]

JSON, JavaScript Object Notation - простой формат обмена данными [7]

Spring framework - универсальный фреймворк с открытым исходным кодом для Java-платформы[8].

Веб-приложение, Web application клиент - серверное приложение, в котором клиентом выступает браузер, а сервером - веб-сервер[9].

Инфо-система - Information system. Совокупность технического, программного и организационного обеспечения.[10]

Нефункциональные требования - Non-functional requirements. Требования, которые определяют критерии работы системы в целом, а не отдельные сценарии поведения.[11]

ОС - Operating system. Операционная система .[16]

Use case - обобщённые сведения о некоторых отношениях между вариантами использования, субъектами и системами.[13]

Фреймворк - Software framework. Структура программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.[14]

Функциональные требования - Functional requirement. Требования, объясняющие задачи или действия, которые должны быть выполнены.[15]

MVC - это фундаментальный паттерн, который нашел применение во многих технологиях, дал развитие новым технологиям и каждый день облегчает жизнь разработчикам.[12]

База данных - набор сведений, хранящихся некоторым упорядоченным способом. Можно сравнить базу данных со шкафом, в котором хранятся документы.[17]

Класс - это логическая структура, позволяющая создавать свои собственные пользовательские типы путем группирования переменных других типов, методов и событий.[18]

JSP, JavaServer Pages - это одна из технологий J2EE, которая представляет собой расширение технологии сервлетов для упрощения работы с Web-содержимым.[19]

Postgresql - это полноценная SQL СУБД с большим списком возможностей и огромным количеством людей по всему миру, которые используют и разрабатывают эту СУБД.[20]

Список иллюстраций

Рисунок 1. Требования, прецеденты.....	11
Рисунок 2. Диаграмма прецедентов системы.....	12
Рисунок 3. Общая архитектура данного веб-приложения.....	15
Рисунок 4. Диаграмма классов пакета «Service».....	16
Рисунок 5. Диаграмма классов пакета «DAO».....	17
Рисунок 6. Диаграмма классов пакета «Entity».....	18
Рисунок 7. Диаграмма классов пакета «Web».....	19
Рисунок 8. Код jsp страницы «Регистрация».....	20
Рисунок 9. Страница регистрации.....	21
Рисунок 10. Общая диаграмма классов веб-приложения.....	22
Рисунок 11. Диаграмма классов базы данных.....	23
Рисунок 12. Страница входа в систему.....	25
Рисунок 13. Страницы просмотра фильмов.....	26

Содержание

Введение.....	11
1. Постановка проблемы.	12
1.1 Описание проблемы и обзор похожих решений.	12
1.2 Постановка задачи.....	12
1.3 Обзор работы.....	13
2. Анализ.....	13
2.1 Требования к системе.....	13
2.1.1 Функциональные требования	13
2.1.2 Нефункциональные требования	13
2.2 Прецеденты системы.....	15
2.3 Пользователи системы.....	17
3. Архитектура системы	18
3.1 Общая архитектура.....	18
3.2 Серверная часть.....	19
3.2.1 Service package.....	19
3.2.2 DAO package.....	20
3.2.3 Entity package	21
3.3 Клиентская часть	22
3.3.1 Web package	22
3.3.2 Jsp	23
3.4 Диаграмма общих классов.....	24
3.5 Структура базы данных	26
4. Реализация	27
4.1 Выбор языка программирования	27
4.2 Выбор Фреймворка	27

4.3 Выбор базы данных.....	28
4.4 Пользовательский интерфейс	28
5. Планы на будущее.....	29
Заключение.....	30
Список использованной литературы.....	33

Введение

Целью данной дипломной работы является создание веб приложения, которое решает проблему не поступающей информации относительно появления в продаже билетов на премьеры фильмов в таллиннском кинотеатре. Ходить на премьеры выгодно по двум причинам: они дешевле и выходят раньше, чем обычный показ фильма. Поскольку автор является любителем ходить на премьеры фильмов , как и его многие друзья, он часто сталкивался с такой ситуацией, когда хорошие места успевали раскупить до него. Никаких решений данной проблемы не существует в Эстонии и в мире, поэтому приходилось постоянно следить за появления билетов вручную. Именно для решения этой проблемы было придумано данное веб-приложение.

1. Постановка проблемы.

1.1 Описание проблемы и обзор похожих решений.

В наше время многие люди ходят в кинотеатры и смотрят фильмы. На сайтах кинотеатров есть разделы, где можно увидеть список фильмов, которые в скором времени выйдут. Так же в социальных сетях имеются страницы кинотеатров, где выкладывается информация относительно данных заведений. Бывают известия о том, что новый фильм появился в кино и его можно посмотреть. Половина людей, посещающих кинотеатры, знают, что существуют премьеры фильмов, которые можно посетить немного раньше, чем дата появления фильма. Билеты на такие показы значительно дешевле. Но эти билеты появляются в неопределённое время, и часто случается, что для того, чтобы успеть купить такой билет, приходится постоянно проверять страницу желаемого фильма на наличие билетов. Не всегда есть время этим заниматься. В Эстонии нет программы, которая могла бы оповещать пользователей в реальном времени о том, что билеты появились и можно их купить. Автор поискал в интернете приложение такого типа и тоже ничего не нашёл.

1.2 Постановка задачи

Целью данной дипломной работы является создание веб-программы, которая решит проблему нехватки актуальной информации относительно выхода в продажу билетов на премьеры фильмов в таллиннском кинотеатре. При помощи данной программы будет возможно не только получать оповещение о появлении билетов, но так же, в удобном для пользователя веб интерфейсе, выбирать фильмы, о которых пользователь хочет получить оповещение.

Преимущество данной программы в том, что она очень проста в использовании. Нужно пройти лёгкую регистрацию и подписаться на понравившиеся фильмы.

Таким образом, задачами данной работы являются:

- Обзор требований системы
- Описание общей архитектуры приложения
- Написание прототипа программы

1.3 Обзор работы

Данная работа разделена на 3 части.

В первой части проводится обзор требований веб программы. Описываются прецеденты системы, функциональные и нефункциональные требования к системе.

Во второй части будет рассматриваться структура программы, структура базы данных и иллюстрироваться диаграмма классов.

В третьей части будут описаны использованные технологии и иллюстрирован макет веб приложения.

2. Анализ

2.1 Требования к системе

2.1.1 Функциональные требования

- Регистрация пользователей.
- Возможность восстановление пароля.
- Просмотр фильмов, не имеющих билетов в продаже.
- Возможность подписаться на оповещение фильма.
- Возможность отписаться от получения оповещения фильма.
- Рассылка оповещений.
- Переход на страницу фильма в кинотеатре.

2.1.2 Нефункциональные требования

- Аутентификация.

- Пользовательский интерфейс в браузере.
- Язык интерфейса - английский

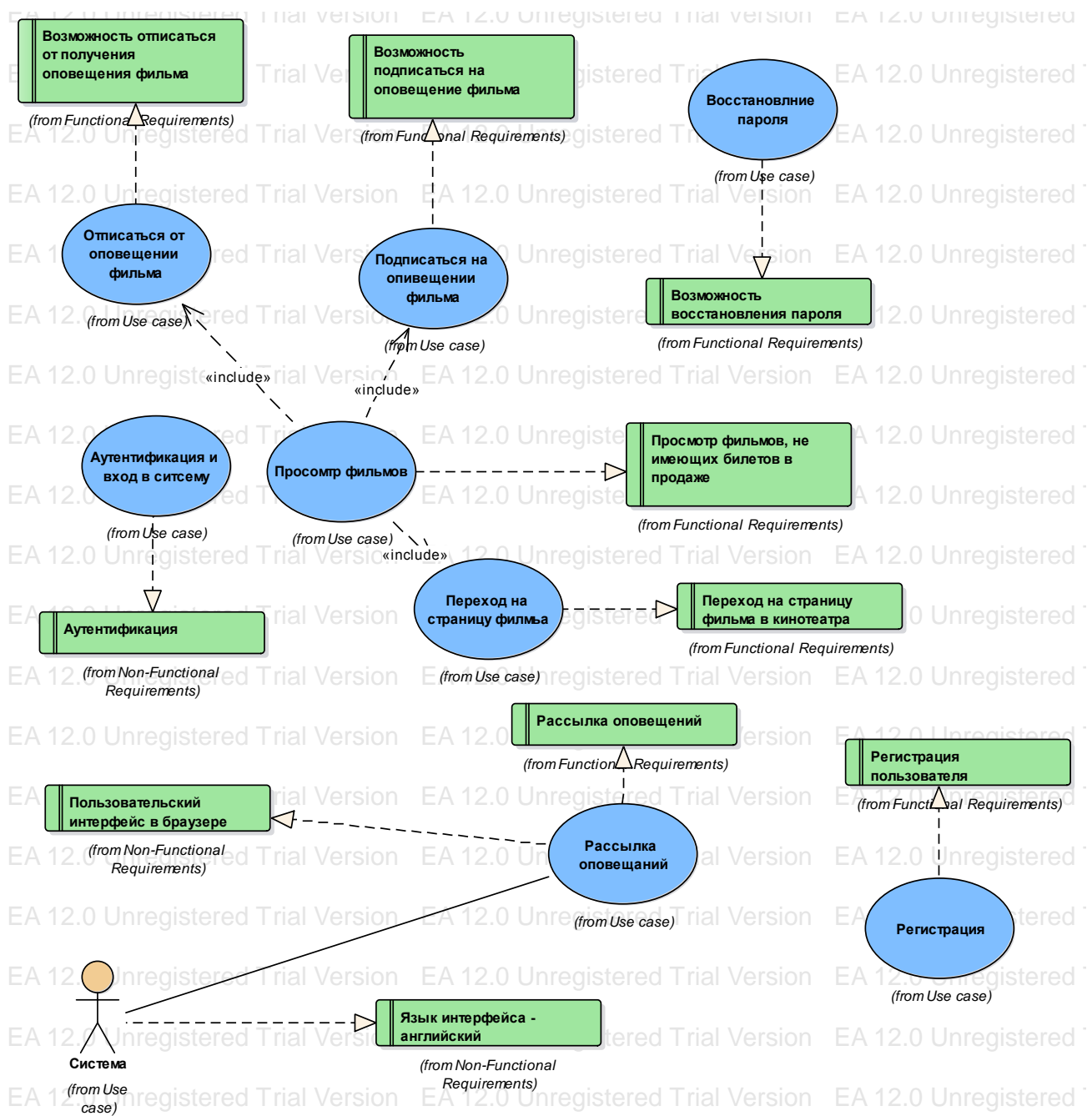
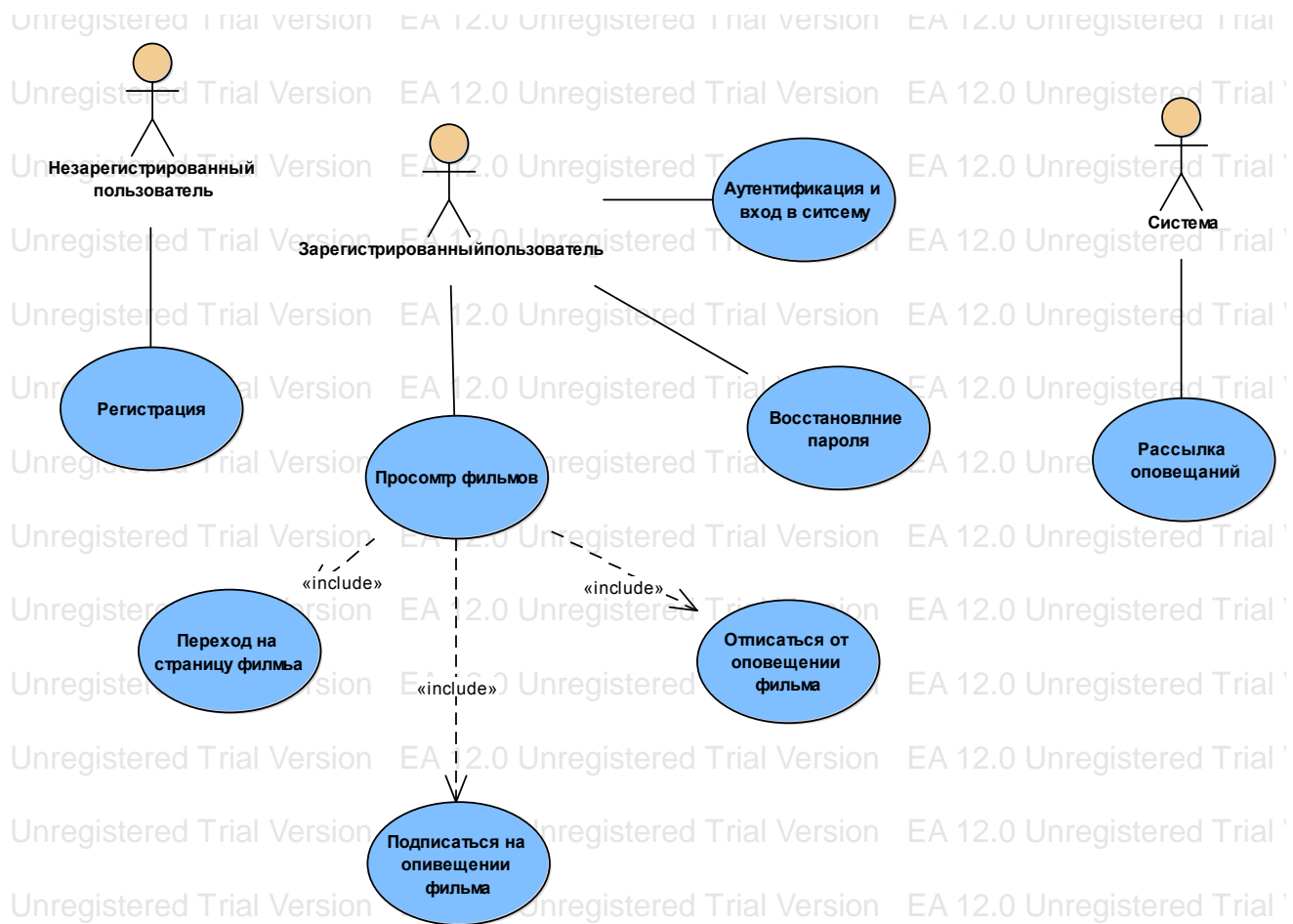


Рис.1 Требования, прецеденты.



2.2 Прецеденты системы

Рис.2 Диаграмма прецедентов системы

Название: Регистрация

Действующие лица: Незарегистрированный пользователь

Описание: Незарегистрированный пользователь может воспользоваться простым и удобным для использования окном регистрации, где необходимо указать электронный почтовый адрес пользователя и пароль, которые будут использоваться при входе в систему.

Название: Восстановление пароля

Действующие лица: Зарегистрированный пользователь

Описание: В случае если зарегистрированный пользователь забыл свой пароль, он имеет возможность восстановить его в окне восстановления пароля, указав свой электронный

почтовый адрес. Если адрес будет верным, то на него будет отправлено письмо со ссылкой на изменение пароля.

Название: Аутентификация и вход в систему.

Действующие лица: Зарегистрированный пользователь

Описание: Для того чтобы пользоваться функционалом данного веб-приложения, пользователь должен войти в систему с помощью окна входа, где он должен ввести электронный почтовый адрес и пароль. Процесс аутентификации проверит личность на подлинность и в случае если проверка завершится успешно, пропустит пользователя дальше.

Название: Просмотр фильмов

Действующие лица: Зарегистрированный пользователь

Описание: Аутентифицированный пользователь имеет возможность просматривать список фильмов, чьи билеты ещё не вышли в продаже. Так же он может просматривать краткую информацию о фильме.

Название: Переход на страницу фильма

Действующие лица: Зарегистрированный пользователь

Описание: Если пользователь захочет войти на страницу сайт кинотеатра с этим фильмом, то ему нужно просто нажать на название понравившегося фильма из списка.

Название: Подписаться на оповещение фильма.

Действующие лица: Зарегистрированный пользователь

Описание: Зарегистрированный пользователь имеет возможность получать оповещение на электронный почтовый ящик пользователя о том, что на желаемый фильм появились билеты. Подписка происходит при нажатии на кнопку подписаться около желаемого фильма.

Название: Отписаться от оповещения фильма.

Действующие лица: Зарегистрированный пользователь.

Описание: Зарегистрированный пользователь может отписаться от оповещения конкретного фильма при желании. Кнопка отписаться будет находиться рядом с фильмом, на который пользователь подписался.

Название: Рассылка оповещений

Действующие лица: Система

Описание: Система производит, в установленное время, поиск билетов на фильмы, которые ещё не вышли, по кинотеатру. В случае наличия таких билетов, система посылает оповещения на почтовые ящики тех пользователей, которые подписались на эти фильмы.

2.3 Пользователи системы

Данное веб приложение поддерживает два вида пользователей:

- Зарегистрированный пользователь
- Незарегистрированный пользователь

Незарегистрированный пользователь имеет единственную возможность – зарегистрироваться и тем самым получить доступ к функционалу веб-приложения.

Зарегистрированный пользователь имеет все возможности данного веб приложения, кроме регистрации.

3. Архитектура системы

3.1 Общая архитектура

Данная архитектура использует принцип MVC «модель-представление-контроллер»[16], что позволяет удобно разделить приложения на три части. Главный плюс данного способа создания программы состоит в том, что при изменении одной части приложения, другие части не подвергнутся тотальным изменениям.

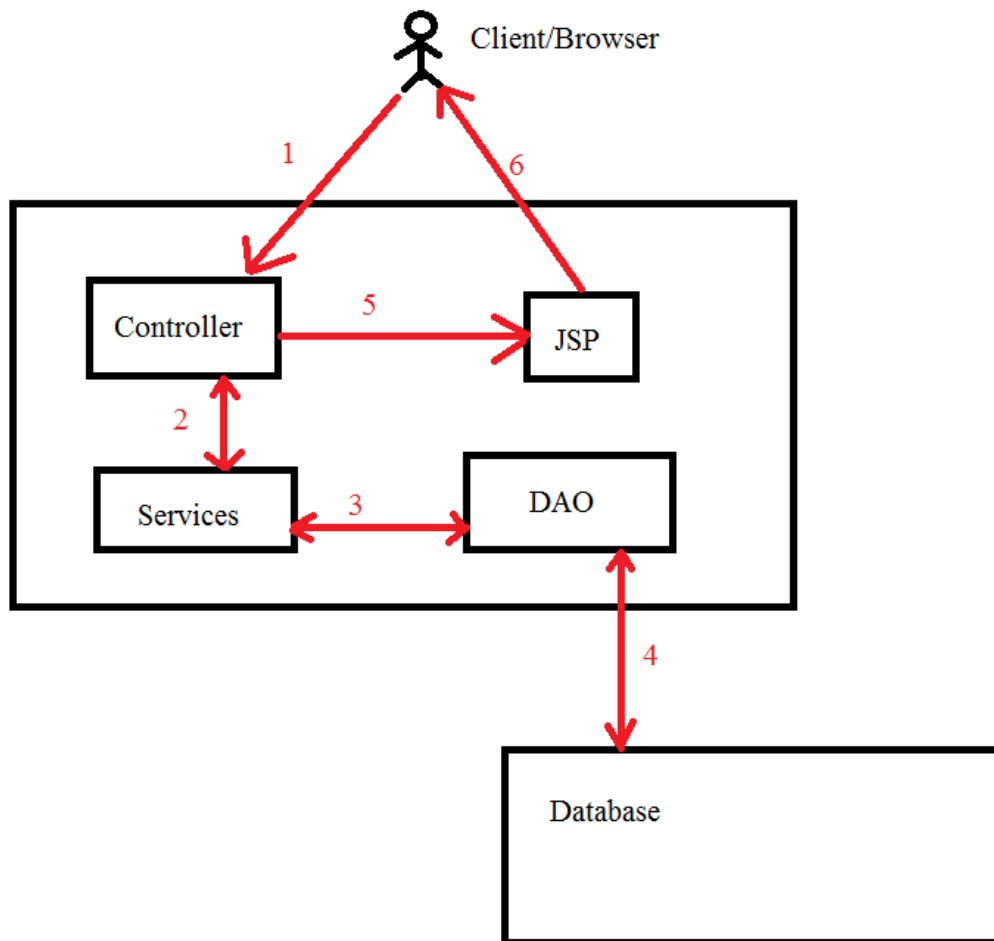


Рис.3 Общая архитектура данного веб-приложения

3.2 Серверная часть

3.2.1 Service package

В пакете “Service” находятся разные сервисы, связанные с использованием объектов базы данных[17]. Так, например, сервис “FilmService” является связующим звеном между контроллера „ContactController“ и “FilmDAO”, который отвечает за все действия, связанные с фильмами.

“PersonService” является связующим звеном между контроллера „ContactController“ и “PersonDAO”, который отвечает за все действия, связанные с фильмами.

“MailService” реализует отсылку сообщений на почтовые ящики пользователей.

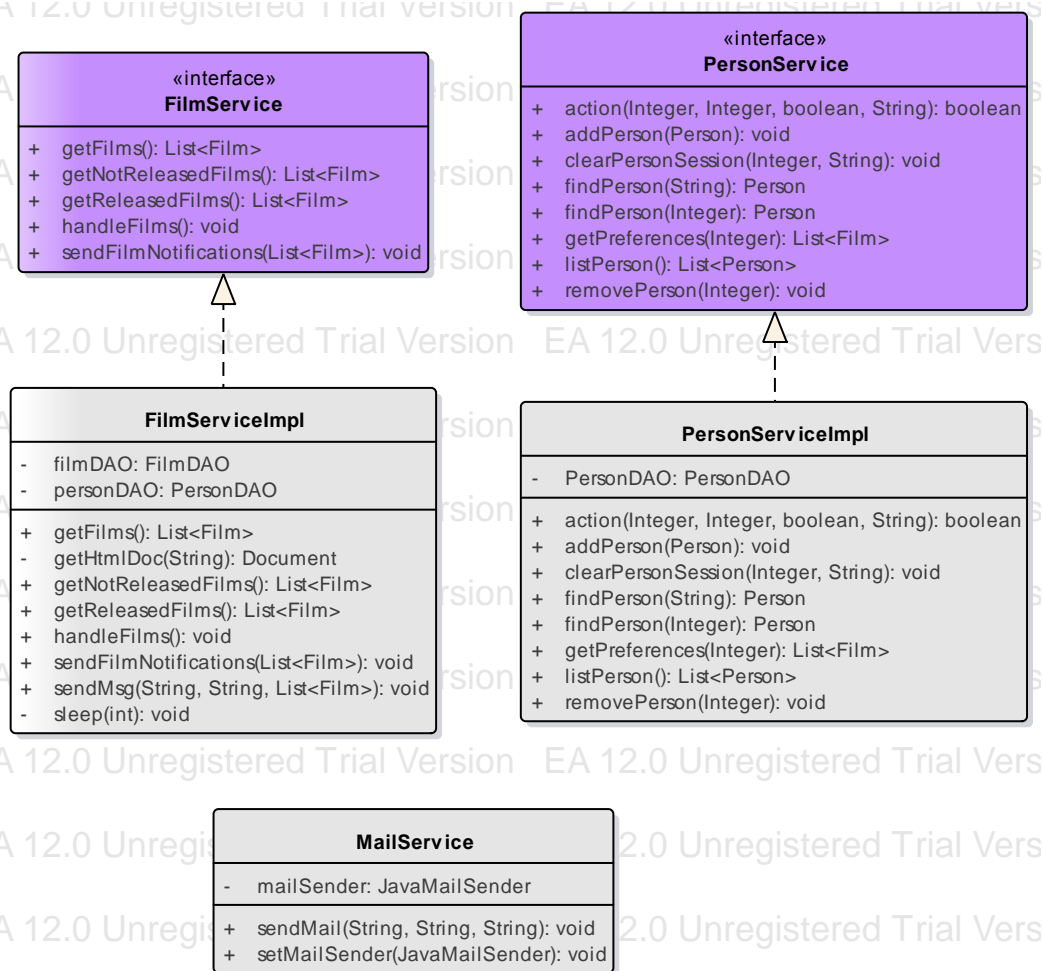


Рис.4 Диаграмма классов пакета «Service»

3.2.2 DAO package

Классы[18] используемые в пакете “DAO” используются для получения данных из базы данных, их сохранения и удаления, а так же других манипуляций ими.

Например, с помощью класса “PersonDAO” можно удалить пользователя, получив его первичный ключ или найти пользователя, используя тот же первичный ключ. Так же можно получить список всех пользователей.

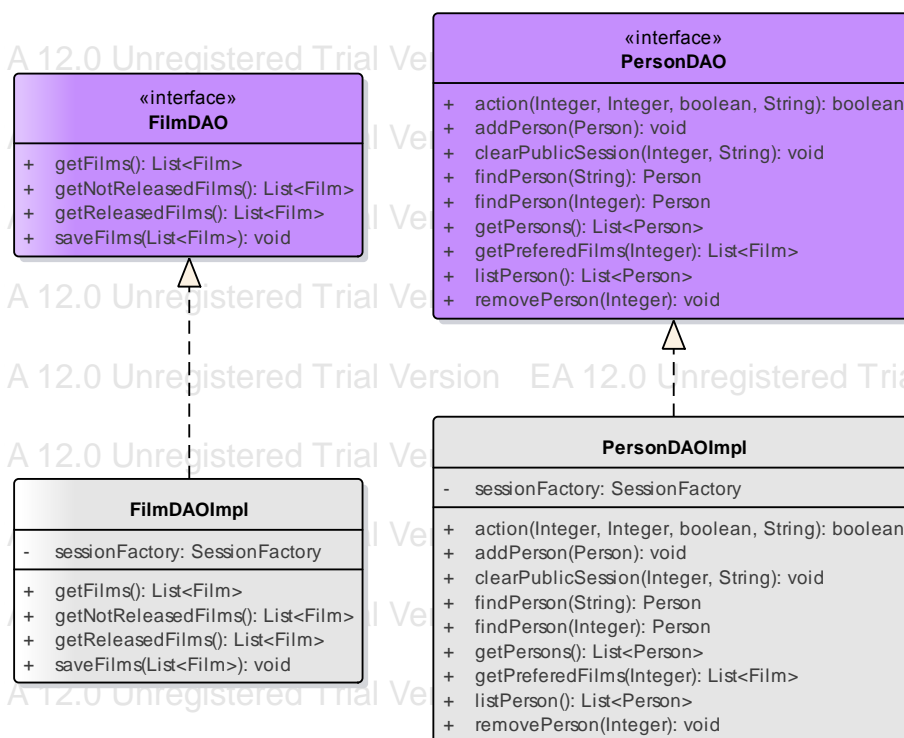


Рис.5 Диаграмма классов пакета «DAO»

3.2.3 Entity package

Пакет “Entity” отображает в java коде таблицы базы данных, тем самым преобразуя их в объекты. Это значительно облегчает работу с используемой базой данных.

На пример при получении из базы данных данные о пользователе, они автоматически сконвертируются в java объект “Person” . После чего можно будет использовать методы, данного объекта для получения информации и манипуляции ими.

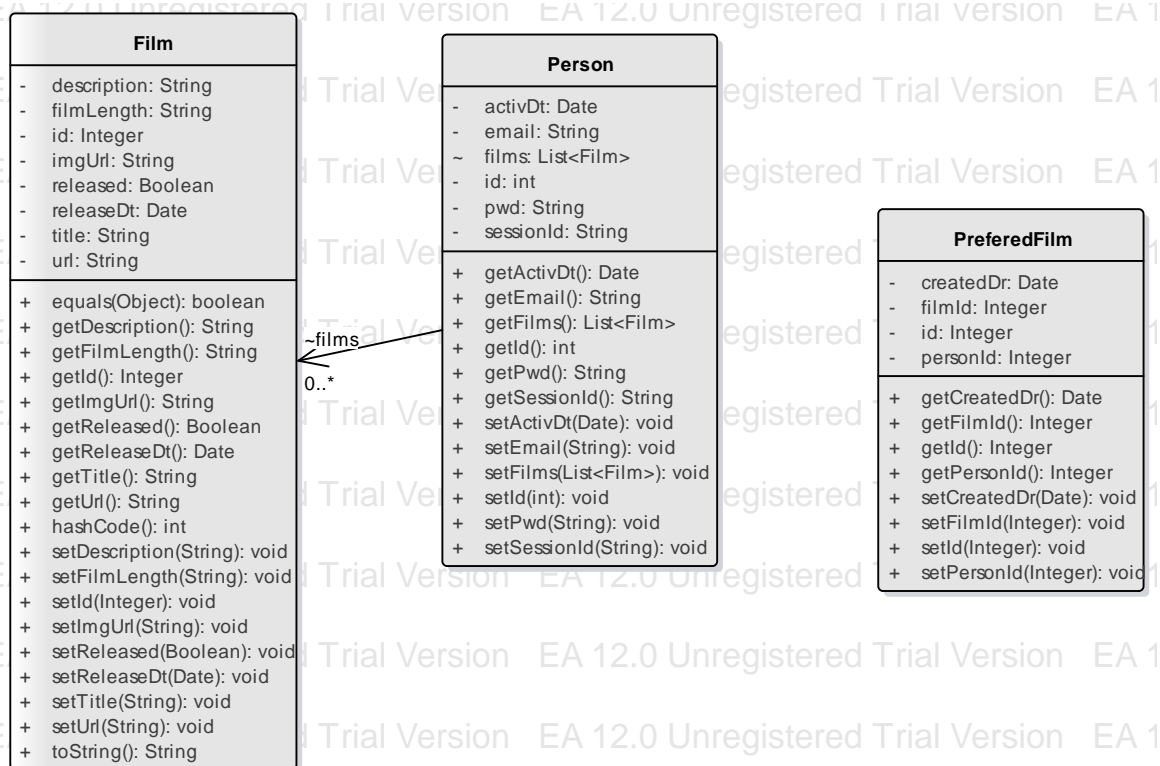


Рис.6 Диаграмма классов пакета «Entity»

3.3 Клиентская часть

3.3.1 Web package

ContactController принимает запросы, которые из браузера и исходя из запроса производит определённые операции используя классы из пакета “Service”. Получив нужные данные от сервиса, он прикрепляет их к jsp странице и отправляет в качестве ответа на запрос обратно в браузер пользователю.

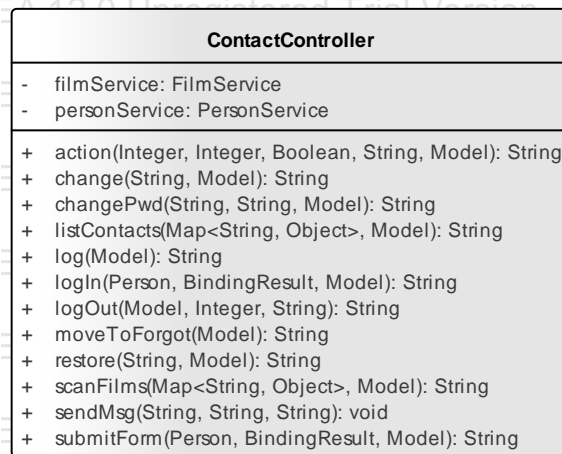


Рис.7 Диаграмма классов пакета «Web»

3.3.2 Jsp

Jsp[19] это веб страницы, содержащие как html, css, js, java код и java объекты, передаваемые контроллером.

Ниже приведено два рисунка – код страницы регистрации и то, что видит пользователь в браузере

```
1 <%@ page language="java" contentType="text/html; charset=utf8"
2   pageEncoding="utf8"%>
3 <%@taglib uri="http://www.springframework.org/tags" prefix="spring"%>
4 <%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
5 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
6
7 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
8 <html>
9 <head>
10 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
11 <script src="//jquery.com/jquery-wp-content/themes/jquery/js/jquery-1.11.2.min.js"></script>
12 <link rel="stylesheet" href="resources/css/bootstrap/bootstrap.min.css">
13 <link rel="stylesheet" href="resources/css/bootstrap/bootstrap-theme.min.css">
14 <link rel="stylesheet" href="resources/css/Login.css">
15 <title>Registration</title>
16 </head>
17 <body>
18 <div class="container">
19 <div class="jumbotron">
20   <h1>Hello, movie-fans!</h1>
21   <p>Here you can registrate your own account for our extremely comfortable notification service for cinema "Coca-Cola Plaza"
22   <p><a class="btn btn-primary btn-lg" href="Login" role="button">Log in</a></p>
23 </div>
24 <div class="form">
25 <form:form method="post" action="reg" modelAttribute="person" >
26   <div class="form-group">
27     <form:label path="email">
28       Email
29     </form:label>
30     <form:input class="form-control" type="email" path="email" /><form:errors style="color:red" path="email" />
31   </div>
32   <div class="form-group">
33     <form:label path="pwd">Password
34     </form:label>
35     <form:input class="form-control" path="pwd" /><form:errors style="color:red" path="pwd" />
36   </div>
37   <input type="submit" class="btn btn-primary" value="Submit" />
38 </form:form>
39 <p style="color:red">${exists}</p>
40 </div>
41 </div>
42 </body>
43 </html>
```

Рис.8 Код jsp страницы «Регистрация»

Hello, movie-fans!

Here you can registrate your own account for our extremely comfortable notification service for cinema "Coca-Cola Plaza"

Log in

Email

Password

Submit

Рис.9 Страница регистрации.

3.4 Диаграмма общих классов

Классов веб-приложения не очень много, так что скрывать методы и переменные не имеет смысла.

Данные классы сгруппированы на пакеты. Группировка происходит, основываясь на функциях классов, а так же используется архитектура Spring Framework. Пакет с классами контролерами отдельно, пакет с сервисами отдельно, пакет с классами связи с базой данных отдельно, а так же пакет с объектами базы данных отдельно.

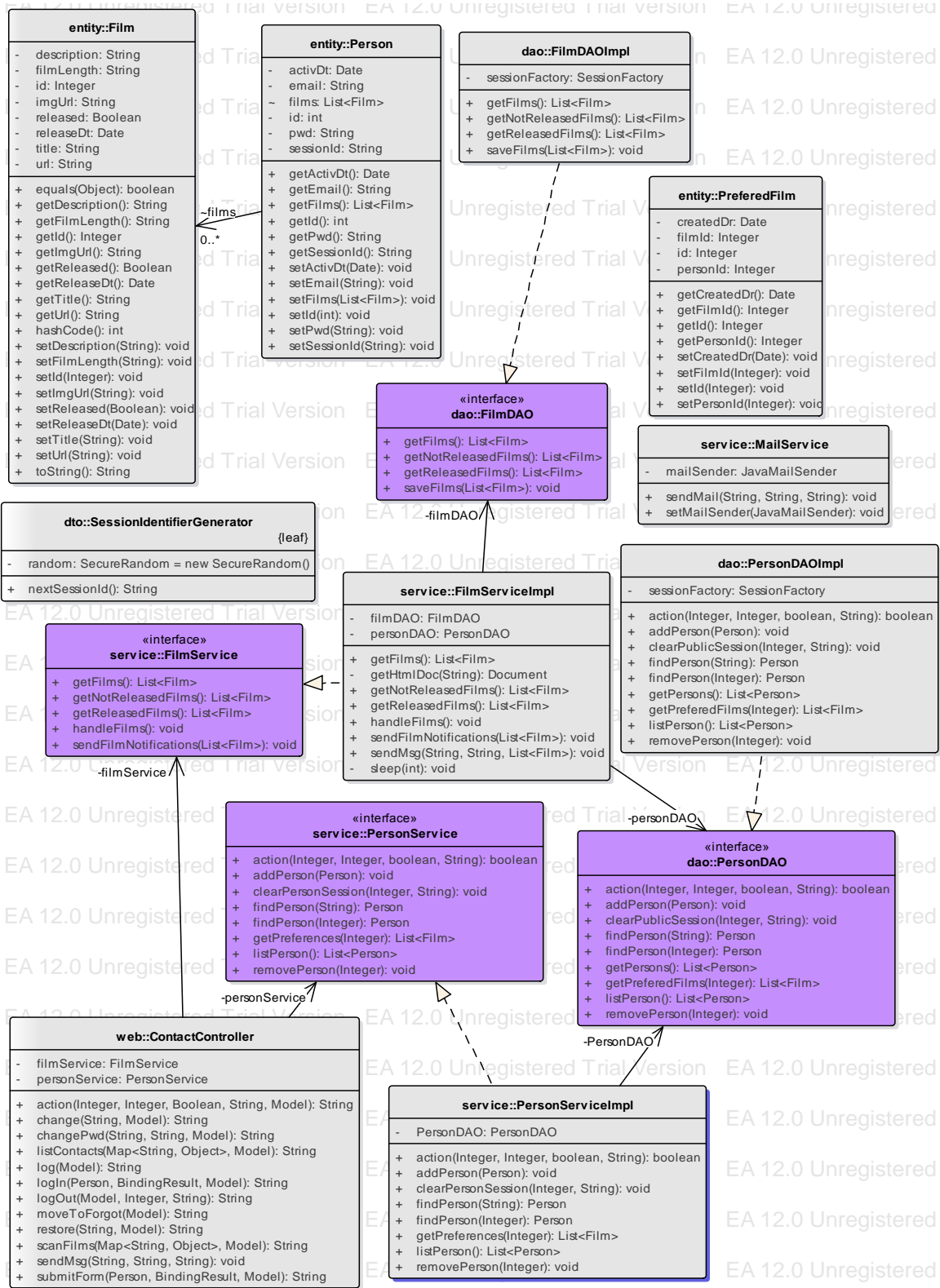


Рис.10 Общая диаграмма классов веб-приложения

3.5 Структура базы данных

Структура базы данных проста и не громоздка, так как решения, поднятой в данном проекте, проблемы оказались лёгким. В таблице «film» хранятся все фильмы, которые сканируются системой. В таблице «person» хранятся все пользователи, зарегистрированные данной системой. В таблице «preferred_film» хранятся связи пользователей и фильмов, на которые они подписались.

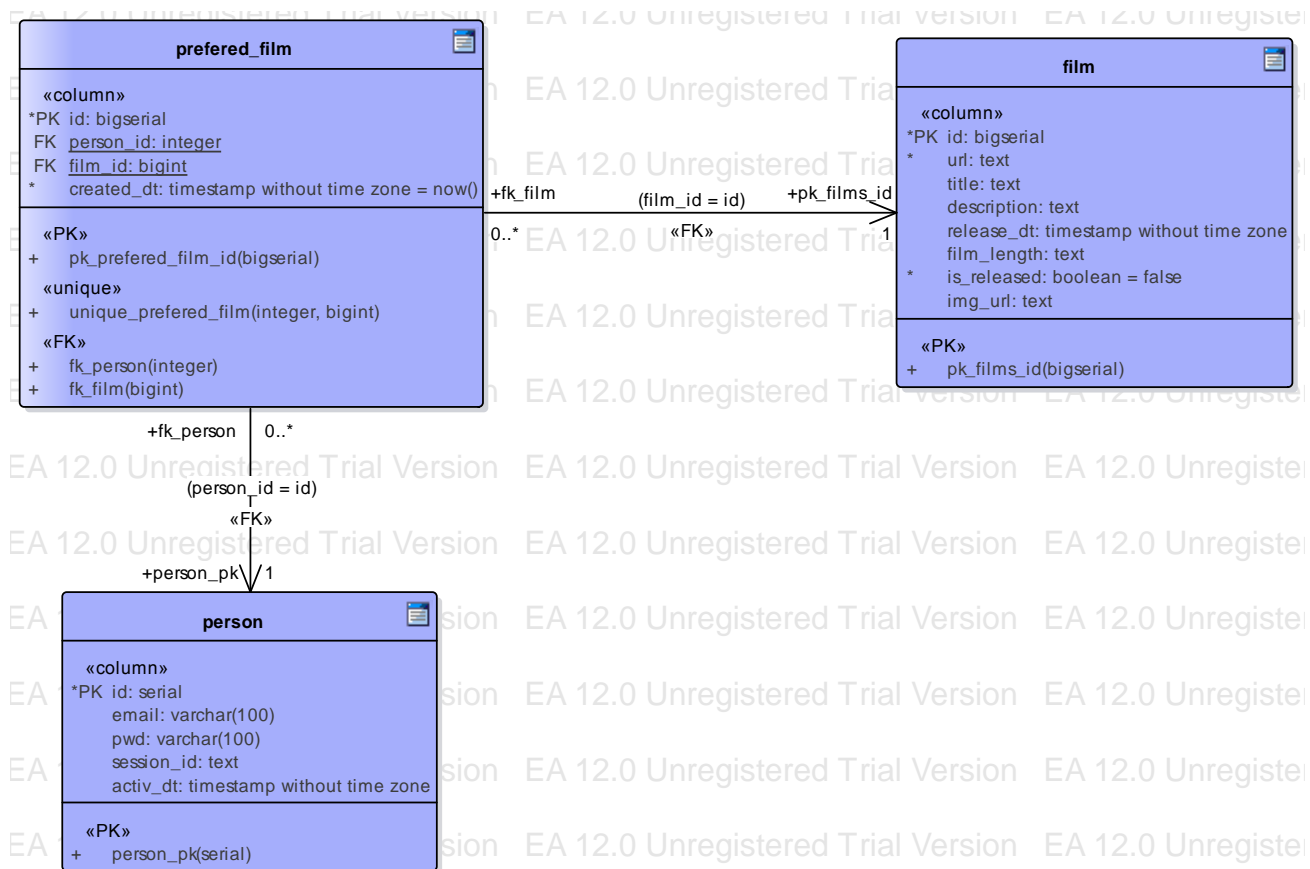


Рис.11 Диаграмма классов базы данных

4. Реализация

4.1 Выбор языка программирования

В качестве языка программирования был выбран язык Java. На данный выбор повлияло несколько факторов.

- Во время учёбы автора в Таллиннском Техническом Университете основной уклон был сделан именно на использование объектно-ориентированного языка программирования Java.
- По данным сайта «tjobe»[1] на 2015 год язык программирования Java является самым популярным языком в мире.
- Автор данной работы уже больше года работает в ИТ-фирме, где используется язык программирования Java
- Язык программирования Java очень удобен для создания веб-приложений.

4.2 Выбор Фреймворка

Выбор автора пал на framework Spring MVC и библиотек Hibernate, который идеально подходит для разработки веб-приложений с использованием баз данных.

Его целью является упрощение разработки приложений на языке Java:

- Легковесность и не насильственность благодаря применению простых Java-объектов (POJO);
- Слабое связывание посредством внедрения зависимостей и ориентированности на интерфейсы
- Декларативное программирование через аспекты и общепринятые соглашения
- Уменьшение объёма типового кода через аспекты и шаблоны.

Данный список взят из книги[2]

Общение с клиентом происходит по протоколу HTTP[5], а так же для использования AJAX [6] происходит пересылка некоторых данных в формате JSON [7].

4.3 Выбор базы данных

Выбор автора пал на PostgreSQL[20], так как эта база данных очень удобна в использовании, бесплатна и развивается. Так же она имеет необходимый функционал для реализации данного веб приложения.

4.4 Пользовательский интерфейс

Пользовательский интерфейс отображается с помощью браузер.

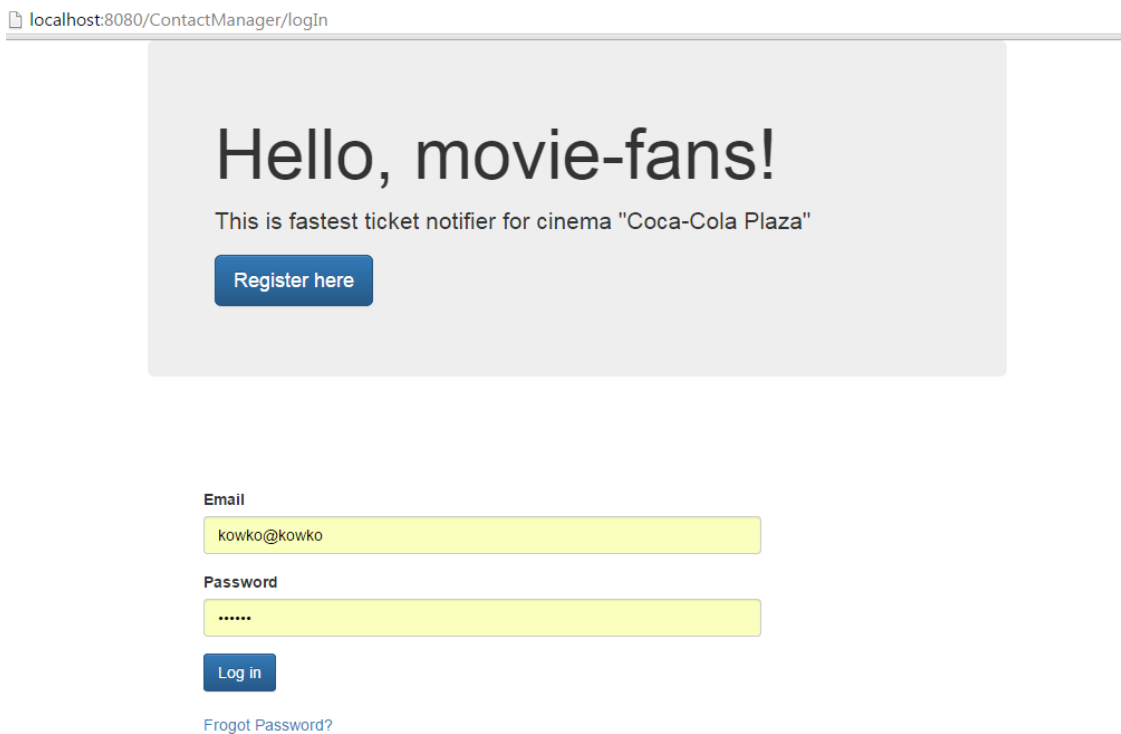


Рис.12 Страница входа в систему

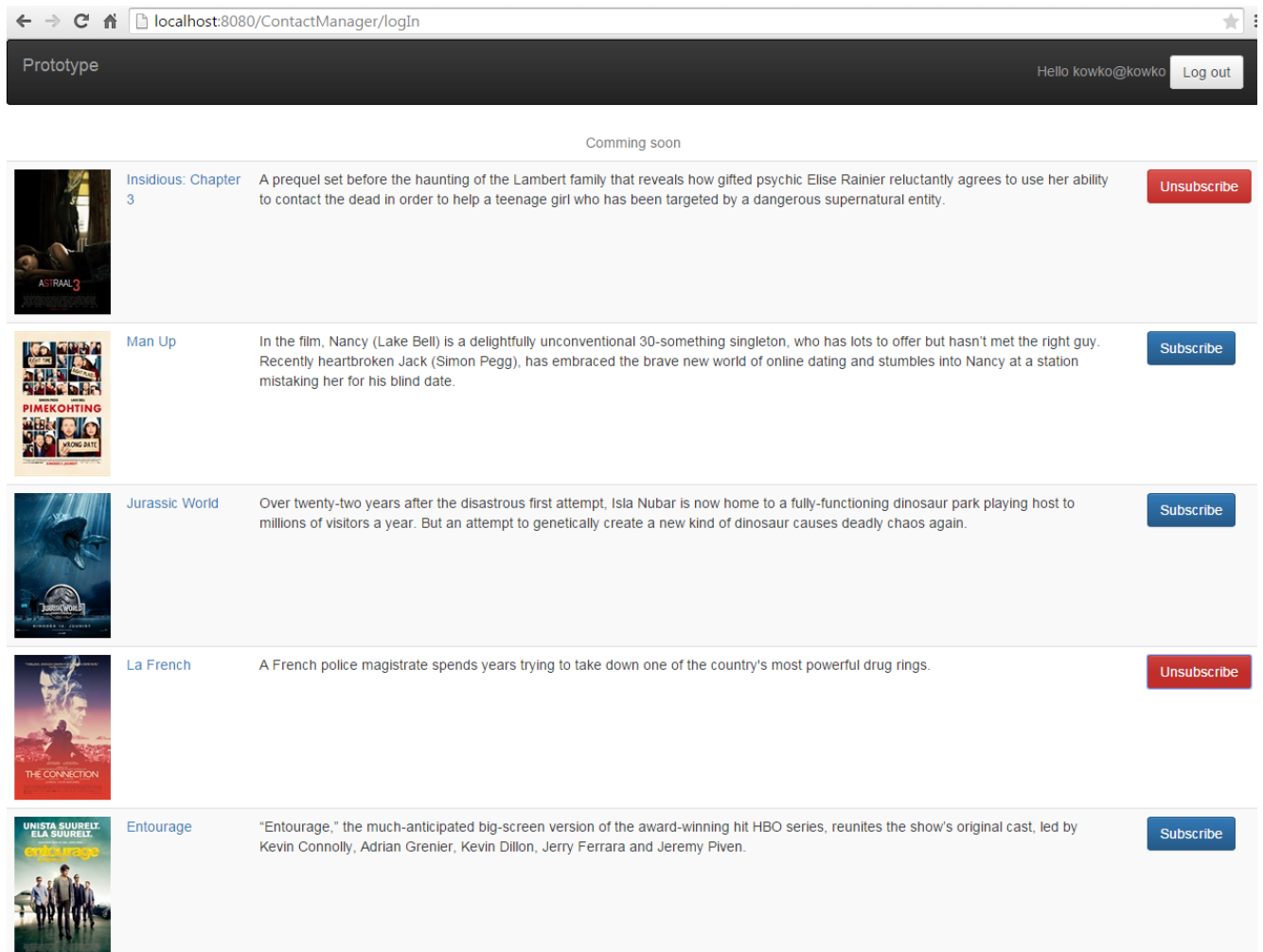


Рис.13 Страницы просмотра фильмов

5. Планы на будущее

Данное веб приложение находится на финальной фазе тестирования и в ближайшее время будет готово для запуска на сервере.

В будущем возможно:

- Добавление удобного фильтра по категориям на странице просмотра фильмов, для более быстрого поиска желаемого фильма.
- Создать аналогичное приложение для Android и iOS.

Заключение

В ходе работы все поставленные задачи были выполнены: прецеденты системы были описаны, функциональные и нефункциональные требования предоставлены, описана архитектура системы и структура базы данных. Было пояснено, почему использовались данные технологии для реализации веб-приложения. В результат было создано приложение, отвечающее всем требованиям и решающее поднятую проблему в данной работе. Программа позволяет не только просматривать фильмы, которые ещё не появились, но и подписаться на оповещение при появлении билетов в продаже в течение короткого времени. Данное приложения не является окончательным и в будущем его разработка продолжится.

Итогом работы автора является удобное и эффективное веб-приложение, которое подойдёт любителям сходить на премьеры фильмов. В ходе работы автор использовал знания, полученные во время учёбы в университете, а так же опыт, полученный на работе.

Kokkuvõtte

Kõik eesmärgid, mis olid püstitatud töökäigus, said täidetud ja saavutatud. Need olid: kasutusjuhiste kirjeldamine ja diagrammide näitamine, funktsionaalsete ja mittefunktsionaalsete nõuete püstitamine ning süsteemi ülesehitmine ja andmebaasi struktuuri näitamine. Antud töös oli nimetatud ka kasutatud tehnoloogia, kui ka antud tehnoloogia tõestamine. Tulemusena loodi veebiprogramm, mis võimaldab esietenduste piletite müügile ilmumisest teatada kasutajaid e-maili kaudu reaalajas. Hetkel ei ole see veel rakenduse lõppversioon ja seda tuleb natuke arendada. Tehtud töö näitab, et veebiprogramm on mugav ja efektiivne lahendus, mis on tõestatud selles projektis läbi püstitatud eesmärkide. Selle rakenduse tegemisel kasutas töö autor teadmisi, mida omandas ülikoolis õppimise ajal ning kogemusi, mida omandas töö käigus.

Summary

All author`s goals were achieved during this work. Functional and none-functional requirements, Use Case diagram, system architecture and database structure were shown and described. Also author described and explained why he chose exactly this technology for building this web-application. As a result, author got properly working prototype. Application became extremely simple and user-friendly. User can easily register, log in and subscribe for the desired movie and also visit home page if desired movie in Tallinn. Also sending notifications to the users is realized.

This work was a great chance to test author`s knowledge in programming, which he gained while studding in university and also to use experience which he obtained while working.

Список использованной литературы

1) «tiobe», [В Интернете]. Available:

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

[Дата обращения 24.05.2015]

2) Крейг Уоллс, Книга «Spring in Action», ДМК Пресс, 2015, 754 lk [Дата обращения 24.05.2015]

3) «spring», [В Интернете]. Available: <http://spring.io/> [Дата обращения 24.05.2015]

4) «wikipedia», [В Интернете]. Available: https://ru.wikipedia.org/wiki/Data_Access_Object
[Дата обращения 24.05.2015]

5) «wikipedia», [В Интернете]. Available: <https://ru.wikipedia.org/wiki/HTTP> [Дата обращения 24.05.2015]

6) «wikipedia», [В Интернете]. Available: <https://ru.wikipedia.org/wiki/AJAX> [Дата обращения 24.05.2015]

7) «wikipedia», [В Интернете]. Available: <https://ru.wikipedia.org/wiki/JSON> [Дата обращения 24.05.2015]

8) «wikipedia», [В Интернете]. Available: https://ru.wikipedia.org/wiki/Spring_Framework
[Дата обращения 24.05.2015]

9) «wikipedia», [В Интернете]. Available: https://en.wikipedia.org/wiki/Web_application [Дата обращения 24.05.2015]

10) «wikipedia», [В Интернете]. Available: http://en.wikipedia.org/wiki/Information_system
[Дата обращения 24.05.2015]

11) «wikipedia», [В Интернете]. Available: http://en.wikipedia.org/wiki/Non-functional_requirement [Дата обращения 24.05.2015]

- 12) «wikipedia», [В Интернете]. Available: <http://habrahabr.ru/post/215605/> [Дата обращения 24.05.2015]
- 13) «wikipedia», [В Интернете]. Available: http://en.wikipedia.org/wiki/Use_case [Дата обращения 24.05.2015]
- 14) «wikipedia», [В Интернете]. Available: http://en.wikipedia.org/wiki/Software_framework [Дата обращения 24.05.2015]
- 15) «wikipedia», [В Интернете]. Available: http://en.wikipedia.org/wiki/Functional_requirement [Дата обращения 24.05.2015]
- 16) «wikipedia», [В Интернете]. Available: http://en.wikipedia.org/wiki/Operating_system [Дата обращения 24.05.2015]
- 17) «site-do», [В Интернете]. Available: <http://www.site-do.ru/db/db1.php> [Дата обращения 24.05.2015]
- 18) «msdn», [В Интернете]. Available: <https://msdn.microsoft.com/ru-ru/library/x9afc042.aspx> [Дата обращения 24.05.2015]
- 19) «java2ee», [В Интернете]. Available: <http://www.java2ee.ru/jsp/> [Дата обращения 24.05.2015]
- 20) «postgresql», [В Интернете]. Available: http://postgresql.ru.net/docs/about_pgsql.html [Дата обращения 24.05.2015]