**Оглавление**

[ВВЕДЕНИЕ 4](#_Toc252535205)

[1. Теоретические сведения 5](#_Toc252535206)

[1.1 Информационная система 5](#_Toc252535207)

[1.1.1 Классификация информационных систем по степени автоматизации 5](#_Toc252535208)

[1.1.2 Классификация информационных систем по характеру использования информации 6](#_Toc252535209)

[1.1.3 Классификация информационных систем по архитектуре 7](#_Toc252535210)

[1.1.4 Классификация информационных систем по сфере применения 7](#_Toc252535211)

[1.1.5 Классификация информационных систем по признаку структурированности решаемых задач 8](#_Toc252535212)

[1.2 Базы данных 9](#_Toc252535213)

[1.2.1 База данных 9](#_Toc252535214)

[1.2.2 Структура БД 9](#_Toc252535215)

[1.2.3 Классификация БД 10](#_Toc252535216)

[1.2.4 Этапы проектирования базы данных 10](#_Toc252535217)

[1.3 PHP 12](#_Toc252535218)

[1.3.1 Область применения 12](#_Toc252535219)

[1.3.2 История 13](#_Toc252535220)

[1.3.3 Синтаксис 15](#_Toc252535221)

[2. Бизнес-процессы 18](#_Toc252535222)

[2.1 Просмотр программы 19](#_Toc252535223)

[2.2 Добавление фильма 22](#_Toc252535224)

[2.3 Создание нового пользователя 24](#_Toc252535225)

[3. Техническое задание 26](#_Toc252535226)

[3.1 Общие сведения 26](#_Toc252535227)

[3.2 Назначение и цели создания системы 26](#_Toc252535228)

[3.2.1 Назначение 26](#_Toc252535229)

[3.2.2 Создать электронную версию телевизионной программы 27](#_Toc252535230)

[3.3 Характеристика объектов автоматизации 27](#_Toc252535231)

[3.4 Требования к системе 29](#_Toc252535232)

[3.4.1 Общие требования 29](#_Toc252535233)

[3.4.2 Требования к характеристикам взаимосвязей создаваемой системы со смежными системами, требования к ее совместимости и способы обмена информации 30](#_Toc252535234)

[3.4.3 Требования к численности персонала, его квалификации и режиму его работы: 30](#_Toc252535235)

[3.4.4 Требования к надежности 30](#_Toc252535236)

[3.4.5 Требования к безопасности 30](#_Toc252535237)

[3.4.6 Требования к эргономике и технической эстетике 31](#_Toc252535238)

[3.4.7 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению комплектов системы 31](#_Toc252535239)

[3.4.8 Требования к защите информации от несанкционированного доступа 31](#_Toc252535240)

[3.4.9 Требования к средствам защиты от внешних воздействий 32](#_Toc252535241)

[3.4.10 Дополнительные требования 32](#_Toc252535242)

[3.5 Требования к функциям системы 32](#_Toc252535243)

[3.6 Требования к видам обеспечения: 32](#_Toc252535244)

[3.6.1 Требования к информационному обеспечению 32](#_Toc252535245)

[3.6.2 Требования к программному обеспечению для сервера 32](#_Toc252535246)

[3.6.3 Требования к программному обеспечению для клиентских машин 33](#_Toc252535247)

[3.6.4 Требования к техническому обеспечению для сервера 33](#_Toc252535248)

[3.6.5 Требования к техническому обеспечению для клиентских машин 33](#_Toc252535249)

[3.6.6 Требования к составу и содержанию работ по созданию и развитию системы 33](#_Toc252535250)

[3.7 Порядок контроля и приемки системы 34](#_Toc252535251)

[3.8 Требования к документированию 34](#_Toc252535252)

[4. Инфологическая модель 35](#_Toc252535253)

[5. Даталогическая модель 38](#_Toc252535254)

[6. Описание Web-приложения 46](#_Toc252535255)

[7. Описание форм 49](#_Toc252535256)

[7.1 Форма редактирования канала 49](#_Toc252535257)

[7.2 Форма авторизации 50](#_Toc252535258)

[8. Описание системы отчетности 51](#_Toc252535259)

[8.1 Отчет о пользователях 51](#_Toc252535260)

[8.2 Отчет о каналах 51](#_Toc252535261)

[9. Описание RSS-каналов 53](#_Toc252535262)

[Заключение и выводы 54](#_Toc252535263)

[Библиографический список 55](#_Toc252535264)

[Приложение A. Руководство пользователя 56](#_Toc252535265)

[Приложение B. Руководство администратора 57](#_Toc252535266)

# ВВЕДЕНИЕ

Уже много лет многие предвещают скорую смерть ТВ и перевода вещания в глобальную паутину. Но это до сих пор не произошло – многие как и прежде смотрят телевизор, а чтобы сделать правильный выбор передачи необходимо знать программу всех телеканалов.

В этой курсовой работе я попытаюсь представить свое видение «web» приложения для просмотра и администрирования базы данных ТВ программы.

# Теоретические сведения

## Информационная система

Информационная система (ИС) — это система, реализующая информационную модель предметной области, чаще всего — какой-либо области человеческой деятельности. ИС должна обеспечивать: получение (ввод или сбор), хранение, поиск, передачу и обработку (преобразование) информации 1).

Информационной системой (или информационно-вычислительной системой) называют совокупность взаимосвязанных аппаратно-программных средств для автоматизации накопления и обработки информации. В информационную систему данные поступают от источника информации. Эти данные отправляются на хранение либо претерпевают в системе некоторую обработку и затем передаются потребителю.

Между потребителем и собственно информационной системой может быть установлена обратная связь. В этом случае информационная система называется замкнутой. Канал обратной связи необходим, когда нужно учесть реакцию потребителя на полученную информацию.

Информационная система состоит из баз данных, в которых накапливается информация, источника информации, аппаратной части ИС, программной части ИС, потребителя информации.[[1]](#footnote-1)

### Классификация информационных систем по степени автоматизации

* Ручные информационные системы характеризуются отсутствием современных технических средств переработки информации и выполнением всех операций человеком. Например, о деятельности менеджера в фирме, где отсутствуют компьютеры, можно говорить, что он работает с ручной ИС.
* Автоматизированные информационные системы (АИС) — наиболее популярный класс ИС. Предполагают участие в процессе накопления, обработки информации баз данных, программного обеспечения, людей и технических средств.
* Автоматические информационные системы выполняют все операции по переработке информации без участия человека, различные роботы. Примером автоматических информационных систем являются некоторые поисковые машины Интернет, например Google, где сбор информации о сайтах осуществляется автоматически поисковым роботом и человеческий фактор не влияет на ранжирование результатов поиска.

Обычно термином ИС в наше время называют автоматизированные информационные системы.

### Классификация информационных систем по характеру использования информации

* Информационно-поисковые системы — система для накопления, обработки, поиска и выдачи интересующей пользователя информации.
* Информационно-аналитические системы — класс информационных систем, предназначенных для аналитической обработки данных с использованием баз знаний и экспертных систем.
* Информационно-решающие системы — системы, осуществляющие накопление, обработку и переработку информации с использованием прикладного программного обеспечения.
* управляющие информационные системы с использованием баз данных и прикладных пакетов программ.
* советующие экспертные информационные системы, использующие прикладные базы знаний,
* Ситуационные центры (информационно-аналитические комплексы)

### Классификация информационных систем по архитектуре

* Локальные ИС (работающие на одном электронном устройстве, не взаимодействующем с сервером или другими устройствами)
* Клиент-серверные ИС (работающие в локальной или глобальной сети с единым сервером)
* Распределенные ИС (децентрализованные системы в гетерогенной многосерверной сети)

### Классификация информационных систем по сфере применения

* Информационные системы организационного управления — обеспечение автоматизации функций управленческого персонала.
* Информационные системы управления техническими процессами — обеспечение управления механизмами, технологическими режимами на автоматизированном производстве.
* Автоматизированные системы научных исследований — программно-аппаратные комплексы, предназначенные для научных исследований и испытаний.
* Информационные системы автоматизированного проектирования — программно-технические системы, предназначенные для выполнения проектных работ с применением математических методов.
* Автоматизированные обучающие системы — комплексы программно-технических, учебно-методической литературы и электронные учебники, обеспечивающих учебную деятельность.
* Интегрированные информационные системы — обеспечение автоматизации большинства функций предприятия.
* Экономическая информационная система — обеспечение автоматизации сбора, хранения, обработки и выдачи необходимой информации, предназначенной для выполнения функций управления.

### Классификация информационных систем по признаку структурированности решаемых задач

* Модельные информационные системы позволяют установить диалог с моделью в процессе ее исследования (предоставляя при этом недостающую для принятия решения информацию), а также обеспечивает широкий спектр математических, статистических, финансовых и других моделей, использование которых облегчает выработку стратегии и объективную оценку альтернатив решения. Пользователь может получить недостающую ему для принятия решения информацию путем.
* Использование экспертных информационных систем связано с обработкой знаний для выработки и оценки возможных альтернатив принятия решения пользователем. Реализуется на двух уровнях:

Первый уровень (концепция «типового набора альтернатив») — сведение проблемных ситуаций к некоторым однородным классам решений. Экспертная поддержка на этом уровне реализуется созданием информационного фонда хранения и анализа типовых альтернатив. Второй уровень — генерация альтернативы на основе правил преобразования и процедур оценки синтезированных альтернатив, используя базу имеющихся в информационном фонде данных.

Экспертные системы представляют совокупность фактов, сведений и данных с системой правил логического вывода информации на основании логической модели баз данных и баз знаний. Базы данных содержат совокупность конкретных данных, а базы знаний — совокупность конкретных и обобщенных сведений в рамках логической модели базы знаний.

## Базы данных

### База данных

База Данных (БД) — информационная модель, позволяющая в упорядоченном виде хранить данные о группе объектов с одинаковым набором свойств или поименованную совокупность структурированных данных. (Поименованная совокупность структурированных данных предметной области).

«Базой данных» часто упрощённо или ошибочно называют Системы Управления Базами Данных (СУБД). Нужно различать набор данных (собственно БД) и программное обеспечение, предназначенное для организации и ведения базы данных (СУБД) 1).

### Структура БД

Организация структуры БД формируется исходя из следующих соображений:

1. Адекватность описываемому объекту/системе — на уровне концептуальной и логической модели.
2. Удобство использования для ведения учёта и анализа данных — на уровне так называемой физической модели.

Виды концептуальных (инфологических) моделей БД: «сущность-связь», семантические, графовые

Виды логических (даталогических) моделей БД:

* Документальные (архивы) — ориентированные на формат документа, дескрипторные, тезаурусные.
* Фактографические (картотеки)
* теоретико-графовые: иерархическая модель, сетевая модель.
* теоретико-множественные: реляционная модель ([[2]](#footnote-2)ER-модель), многомерная модель.
* объектно-ориентированные: объектная модель.
* основанные на инвертированных файлах.

### Классификация БД

По модели представления данных БД классифицируются:

* Картотеки
* Иерархические
* Сетевые
* Реляционные
* Многомерные
* Объектно-ориентированные
* Документоориентированные

На уровне физической модели электронная БД представляет собой файл или их набор в формате TXT, CSV, Excel, DBF, XML либо в специализированном формате конкретной СУБД. Также в СУБД в понятие физической модели включают специализированные виртуальные понятия, существующие в её рамках — таблица, табличное пространство, сегмент, куб, кластер и т. д.

В настоящее время наибольшее распространение получили реляционные базы данных. Картотеками пользовались до появления электронных баз данных. Сетевые и иерархические базы данных считаются устаревшими, объектно-ориентированные пока никак не стандартизированы и не получили широкого распространения. Некоторое возрождение получили иерархические базы данных в связи с появлением и распространением XML.

### Этапы проектирования базы данных

1. Концептуальное проектирование — сбор, анализ и редактирование требований к данным. Для этого осуществляются следующие мероприятия:
   * обследование предметной области, изучение ее информационной структуры
   * выявление всех фрагментов, каждый из которых характеризуется пользовательским представлением, информационными объектами и связями между ними, процессами над информационными объектами
   * моделирование и интеграция всех представлений

По окончании данного этапа получаем концептуальную модель, инвариантную к структуре базы данных. Часто она представляется в виде модели «сущность-связь».

1. Логическое проектирование — преобразование требований к данным в структуры данных. На выходе получаем СУБД-ориентированную структуру базы данных и спецификации прикладных программ. На этом этапе часто моделируют базы данных применительно к различным СУБД и проводят сравнительный анализ моделей.
2. Физическое проектирование — определение особенностей хранения данных, методов доступа и т. д.

## PHP

PHP (англ. PHP: Hypertext Preprocessor — «PHP: препроцессор гипертекста», англ. Personal Home Page Tools (устар.) — «Инструменты для создания персональных веб-страниц») — скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект не является свободным и распространяется под собственной лицензией.

### Область применения

В области программирования для Сети PHP — один из популярнейших скриптовых языков (наряду с JSP, Perl и языками, используемыми в ASP.NET) благодаря своей простоте, скорости выполнения, богатой функциональности, кроссплатформенности и распространению исходных кодов на основе лицензии PHP.

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств для разработки веб-приложений. Основные из них:

Автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределенные массивы;

Файловые функции успешно обрабатывают как локальные, так и удаленные файлы;

* Автоматическая отправка HTTP-заголовков;
* Работа с cookies и сессиями;
* Обработка файлов, загружаемых на сервер;
* Работа с HTTP заголовками и HTTP авторизацией;
* Работа с XForms;
* Работа с удаленными файлами и сокетами

В настоящее время PHP используется сотнями тысяч разработчиков. Согласно рейтингу Tiobe, базирующемуся на данных поисковых систем, в декабре 2009 года PHP находится на 3 месте среди языков программирования (уступая Java и C), поднявшись за год на одну позицию.[7] К крупнейшим сайтам, использующим PHP, относятся Facebook, В контакте, Wikipedia.

Входит в LAMP — распространённый набор для создания веб-сайтов (Linux, Apache, MySQL, PHP).

### История

В настоящий момент существует единственная реализация PHP, ни одна сторонняя компания не поддерживает исполняемых модулей, отличных от официальной сборки. Такое положение вещей, с одной стороны, позволяет быстро внедрять и распространять нововведения среди сообщества разработчиков, с другой стороны, разрабатывать язык программирования в условиях отсутствия стандарта, так как единая реализация обеспечивает его по факту. В таких условиях большое значение приобретает версия интерпретатора, определяющая текущую функциональность (обратная совместимость между версиями интерпретатора не соблюдается строго).

В 1994 году датский программист Расмус Лердорф создал набор скриптов на Perl/CGI для вывода и учёта посетителей его онлайн-резюме, обрабатывающий шаблоны HTML-документов. Лердорф назвал набор Personal Home Page (Личная Домашняя Страница). Вскоре функциональности и быстроты Perl — интерпретатора скриптов — перестало хватать, и Лердорф разработал с использованием языка C новый интерпретатор шаблонов PHP/FI (англ. Personal Home Page / Forms Interpreter — «Личная Домашняя Страница / Интерпретатор форм»).

В 1997 году после длительного бета-тестирования вышла вторая версия обработчика, написанного на C — PHP/FI 2.0. Её использовали около 1 % (приблизительно 50 тысяч) всех интернет-доменов мира.

Версия PHP 3.0 подверглась значительной переработке, определившим современный облик и стиль языка программирования. В 1997 году два израильских программиста, Энди Гутманс и Зээв Сураски, полностью переписали код интерпретатора. PHP 3.0 был официально выпущен в июне 1998 года.

Одной из сильнейших сторон PHP 3.0 была возможность расширения ядра дополнительными модулями. Впоследствии интерфейс написания расширений привлёк к PHP множество сторонних разработчиков, работающих над своими модулями, что дало PHP возможность работать с огромным количеством баз данных, протоколов, поддерживать большое число API. Большое количество разработчиков привело к быстрому развитию языка и стремительному росту его популярности. Также язык был переименован в PHP.

К зиме 1998 года, практически сразу после официального выхода PHP 3.0, Энди Гутманс и Зээв Сураски начали переработку ядра PHP. В задачи входило увеличение производительности сложных приложений и улучшение модульности базиса кода PHP. Новый движок, названный Zend Engine, успешно справлялся с поставленными задачами и впервые был представлен в середине 1999 года. PHP 4.0, основанный на этом движке и принёсший с собой набор дополнительных функций, официально вышел в мае 2000 года. В дополнение к улучшению производительности, PHP 4.0 имел ещё несколько ключевых нововведений, таких как поддержка сессий, буферизация вывода, более безопасные способы обработки вводимой пользователем информации и несколько новых языковых конструкций.

Пятая версия PHP была выпущена разработчиками 13 июля 2004 года. Изменения включают обновление ядра Zend (Zend Engine 2), что существенно увеличило эффективность интерпретатора. Введена поддержка языка разметки XML. Полностью переработаны функции ООП, которые стали во многом схожи с моделью, используемой в Java. В частности, введён деструктор, открытые, закрытые и защищённые члены и методы, окончательные члены и методы, интерфейсы и клонирование объектов. В последующих версиях также были введены пространства имён и замыкания.

Шестая версия PHP находится в стадии разработки с октября 2006 года. В ней уже сделано множество нововведений, как, например, исключение из ядра регулярных выражений POSIX и «длинных» суперглобальных массивов, удаление директив safe\_mode, magic\_quotes\_gpc и register\_globals из конфигурационного файла php.ini. Также много внимания уделено поддержке Юникода.

### Синтаксис

Синтаксис PHP подобен синтаксису языка Си. Некоторые элементы, такие как ассоциативные массивы и цикл foreach, заимствованы из Perl.

Для работы программы не требуется описывать какие-либо переменные, используемые модули и т. п. Любая программа может начинаться непосредственно с оператора PHP.

Простейшая программа Hello world на PHP выглядит следующим образом:

<?php

echo 'Hello, world!';

?>

PHP исполняет код, находящийся внутри ограничителей, таких как <?php ?>. Всё, что находится вне ограничителей, выводится без изменений. В основном это используется для вставки PHP-кода в HTML-документ, например, так:

<html>

<head>

<title>Тестируем PHP</title>

</head>

<body>

<?php echo 'Hello, world!'; ?>

</body>

</html>

Помимо ограничителей <?php ?>, допускается использование дополнительных вариантов, таких как <? ?> и <script language="php"> </script>. Кроме того, до версии 6.0 допускается использование ограничителей языка программирования ASP <% %> (конструкции <? ?> и <% %> могут быть выключены в конфигурационном файле php.ini).

Имена переменных начинаются с символа $, тип переменной объявлять не нужно. В отличие от имён функций и классов, имена переменных чувствительны к регистру. Именованные константы могут быть обявлены как регистрозависимыми, так и регистронезависимыми. Переменные обрабатываются в строках, заключённых в апострофы или двойные кавычки, и heredoc-строках (строках, созданных при помощи оператора <<<).

PHP рассматривает переход на новую строку как пробел, так же как HTML и другие языки со свободным форматом. Инструкции разделяются с помощью точки с запятой (;), за исключением некоторых случаев.

PHP поддерживает три типа комментариев: в стиле языка Си (ограниченные /\* \*/), C++ (начинающиеся с // и идущие до конца строки) и оболочки UNIX (с # до конца строки).

Объектно-ориентированное программирование

PHP поддерживает широкие объектно-ориентированные возможности, полная поддержка которых была введена в пятой версии языка.

Класс в PHP объявляется с помощью ключевого слова class. Методы и поля класса могут быть общедоступными (public, по умолчанию), защищёнными (protected) и скрытыми (private). PHP поддерживает все три основных механизма ООП — инкапсуляцию, полиморфизм и наследование (родительский класс указывается с помощью ключевого слова extends после имени класса). Поддерживаются интерфейсы (ставятся в соответствие с помощью implements). Разрешается объявление финальных, абстрактных методов и классов. Множественное наследование классов не поддерживается, однако класс может реализовывать несколько интерфейсов. Для обращения к методам родительского класса используется ключевое слово parent.

Классы в PHP имеют ряд специальных методов (англ. Magic methods), начинающихся с двух символов подчёркивания. Особо стоит отметить конструктор (\_\_construct(), в версиях до 5.0 конструктором служил метод, одноимённый с классом) и деструктор (\_\_destruct()), а также методы чтения (\_\_get()) и записи (\_\_set()), свёртывания (\_\_sleep()) и развёртывания (\_\_wake()), клонирования (\_\_clone()) и др. Эти методы являются достаточно гибким инструментом: переопределяя их, можно добиться существенного изменения поведения объекта.

Экземпляры класса создаются с помощью ключевого слова new, обращение к полям и методам объекта производится с использованием оператора ->. Для доступа к членам класса из его методов используется переменная $this.

class C1 extends C2 implements I1, I2

{

private $a;

protected $b;

function \_\_construct($a, $b)

{

parent::\_\_construct($a, $b);

$this->a = $a;

$this->b = $b;

}

public function plus()

{

return $a + $b;

}

/\* ............... \*/

}

$d = new C1(1, 2);

echo $d->plus();

Начиная с пятой версии PHP, объекты передаются по ссылке:

class a

{

public $color = 'red';

}

$a = new a();

echo $a -> color; // red

$b = $a;

$b -> color = 'blue';

echo $a -> color; // blue

# Бизнес-процессы

Бизнес-процесс представляет собой систему последовательных, целенаправленных и регламентированных видов деятельности, в которой посредством управляющего воздействия и с помощью ресурсов входы процесса преобразуются в выходы, результаты процесса, представляющие ценность для потребителей.

Ключевыми свойствами бизнес-процесса является то, что это конечная и взаимосвязанная совокупность действий, определяемая отношениями, мотивами, ограничениями и ресурсами внутри конечного множества субъектов и объектов, объединяющихся в систему ради общих интересов с целью получения конкретного результата, отчуждаемого или потребляемого самой системой.

В сфере информационных технологий может рассматриваться в качестве устойчивого информационного процесса (последовательности работ), относящегося к производственно-хозяйственной деятельности компании и обычно ориентированного на создание новой стоимости. Например, компания может сознательно организовать информационный бизнес-процесс своего основного производства. Бизнес-процесс включает в себя иерархию взаимосвязанных функциональных действий, реализующих одну (или несколько) из бизнес целей компании в информационной системе компании, например, управление и анализ выпуска продукции или ресурсное обеспечение выпуска продукции, под продукцией здесь понимают, например, товары, услуги, решения, документы1).[[3]](#footnote-3)

## Просмотр программы

запуск программы

авторизация на сервере СУБД

выбор даты

выбор каналов

выбор типов

выбор времени

просмотр списка

Нужна дополнительная информация?

Просмотр дополнительной информации

нашли требуемое?

завершение программы

нет

нет

Рис. 1. Диаграмма действий бизнес-процесса «Просмотр программы».

Таблица №1. Описание БП «просмотр программы»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Функция** | **Исполнитель** | **Как часто** | **Ресурсы** | |
| **Входящие** | **Исходящие** |
| 1 | Запуск приложения | Пользователь | По мере необходимости |  |  |
| 2 | Авторизация на сервере СУБД | Пользователь | По мере необходимости | Параметры авторизации | Сессия авторизации |
| 3 | Выбор даты | Пользователь | По мере необходимости | Дата | Отфильтрованное по дате расписание |
| 4 | Выбор каналов | Пользователь | По мере необходимости | Список каналов | Отфильтрованное по дате, каналам расписание |
| 5 | Выбор типов | Пользователь | По мере необходимости | Список типов | Отфильтрованное по дате, каналам, типам расписание |
| 6 | Выбор времени | Пользователь | По мере необходимости | Необходимое время | Отфильтрованное по дате, каналам, типам, времени расписание |
| 7 | Просмотр списка | Пользователь | По мере необходимости |  | Ссылка на дополнительную информацию о телепрограмме |
| 8 | Просмотр дополнительной информации | Пользователь | По мере необходимости |  |  |
| 9 | Завершение программы | Пользователь | По мере необходимости |  |  |

Данный бизнес-процесс служит для просмотра расписания, а также дополнительной информации о любом транслируемом фильме, передаче или сериале.

## Добавление фильма

запуск программы

авторизация на сервере субд

аутентификация по паролю

поиск информации в интернете

фильтрация необходимой информации

внесение информации

завершение программы

Рис. 2. Диаграмма действий бизнесс-процесса «Добавление фильма».

Таблица №2. Описание БП «добавление фильма»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Функция** | **Исполнитель** | **Как часто** | **Ресурсы** | |
| **Входящие** | **Исходящие** |
| 1 | Запуск приложения | Редактор | По мере необходимости |  |  |
| 2 | Авторизация на сервере СУБД | Редактор | По мере необходимости | Параметры авторизации | Сессия авторизации |
| 3 | Аутентификация по паролю | Редактор | По мере необходимости | Параметры авторизации | Новые права доступа в системе |
| 4 | Поиск информации | Редактор | По мере необходимости | Текст запроса, адреса поисковых систем, время | Нефильтрованная информация |
| 5 | Фильтрация ифнормации | Редактор | По мере необходимости | Нефильтрованная информация | Отфильтрованная информация |
| 6 | Внесение информации | Редактор | По мере необходимости | Отфильтрованная информация | Запись в БД |
| 7 | Завершение программы | Редактор | По мере необходимости |  |  |

Данный бизнес-процесс служит для добавления нового фильма в базу данных нашей ИС.

## Создание нового пользователя

создание группы

назначение прав доступа

запуск программы

авторизация на сервере субд

аутентификация по паролю

определение уровня доступа

группа пользователей с такими правами существует?

выбор группы

заполнение персональной информации

нет

завершение работы

Рис. 3. Диаграмма действия бизнес-процесса «Создание нового пользователя»

Таблица №3. Описание БП «создание нового пользователя»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Функция** | **Исполнитель** | **Как часто** | **Ресурсы** | |
| **Входящие** | **Исходящие** |
| 1 | Запуск приложения | Администратор | По мере необходимости |  |  |
| 2 | Авторизация на сервере СУБД | Администратор | По мере необходимости | Параметры авторизации | Сессия авторизации |
| 3 | Аутентификация по паролю | Администратор | По мере необходимости | Параметры авторизации | Новые права доступа в системе |
| 4 | Определение уровня доступа | Администратор | По мере необходимости | Список необходимых для выполнения пользователем функций | Набор прав пользователя |
| 5 | Создание группы пользователей | Администратор | По мере необходимости | Набор прав пользователя | Новая группа |
| 6 | Занесение персональной информации | Администратор | По мере необходимости | Информация о пользователе | Новый пользователь ИС |
| 7 | Завершение программы | Администратор | По мере необходимости |  |  |

Данный бизнес-процесс служит для создания новых пользователей ИС.

Техническое задание

## Общие сведения

**Полное наименование системы**:

Автоматизированная информационная система «ТВ Программа».

**Документы, на основании которых ведется разработка**:

Задание на курсовое проектирование, выданное кафедрой Информационных систем Сыктывкарского Лесного Института.

**Область применения и краткая характеристика объекта, на котором планируется использование системы:**

ИС «ТВ Программа» предназначена для управления базой данных телевизионной программы.

## Назначение и цели создания системы

### Назначение

Информационная система «ТВ программа» предназначена для реализации следующих бизнес-процессов и возможностей:

* просмотр телепрограммы
* фильтрация телепрограммы
* экспорт телепрограммы
* разграничение прав доступа
  + по типу доступа
    - без авторизации
    - с авторизацией
      * нет доступа
      * только чтение
      * полный доступ
* хранение и управление следующими объектами
  + расписание
  + каналы
  + жанры
  + типы
  + передачи
    - персоны
  + фильмы
    - персоны
    - жанры
  + сериалы
    - персоны
    - жанры
    - эпизоды
  + персоны
  + роли
  + страны
  + пользователи
  + группы
  + права
  + таблицы

### Создать электронную версию телевизионной программы

## Характеристика объектов автоматизации

* группы пользователей
  + название;
* жанры фильмов, сериалов
  + Название;
* каналы
  + название;
  + начало вещания (дата начала вещания);
  + сайт;
  + порядок (дополнительное поле для сортировки);
  + логотип;
  + описание.;
* передачи
  + название;
  + продолжительность;
  + описание;
* пользователи
  + имя;
  + группа;
  + email.
* права групп пользователей
  + название;
* расписание(программа)
  + Канал;
  + Категория объекта (фильм, сериал, передача);
  + Объект;
  + Дата;
  + Время начала;
  + Время конца.
* Сериалы
  + название;
  + продолжительность;
  + страна;
  + статус;
  + описание;
* страны
  + название
  + код;
* типы (для расписания)
  + название;
* персоны
  + фамилия имя отчество;
  + пол;
  + рост;
  + дата рождения;
  + дата смерти;
  + биография.
* фильмы
  + название;
  + продолжительность;
  + описание;
  + страна;
  + бюджет;
  + сборы;
  + описание.

## Требования к системе

### Общие требования

Разрабатываемая система должна представлять собой законченный программный продукт, реализующий автоматизацию основных бизнес-процессов. Для обеспечения эффективной работы система должна иметь удобный графический интерфейс для пользователя, понятное назначение функций и наглядный результат обработки информации, обладать наличием возможности по добавлению информации в БД, средствами защиты от сбоев и восстановления. Необходимым и обязательным является наличие встроенной интерактивной справочной информации по работе с системой и программной документации на систему; техническое задание, описание программы, исходный текст программы, руководства администратора и пользователя.

### Требования к характеристикам взаимосвязей создаваемой системы со смежными системами, требования к ее совместимости и способы обмена информации

Программное обеспечение системы должно обеспечивать интеграцию и совместимость на информационном уровне с другими системами.

### Требования к численности персонала, его квалификации и режиму его работы:

Программа разработана для домашнего использования обыденными пользователями. Особых навыков и знаний при работе с системой не требуется.

### Требования к надежности

При размещении на технической площадке, удовлетворяющей требованиям эксплуатационной документации, система должна обеспечивать необслуживаемое функционирование в круглосуточном режиме с допустимыми перерывами на профилактику и перенастройку и простоями в связи с неисправностью не более 48 часов в год, при среднем времени устранения неисправности, вызвавшей простой, не более 3 часов. Система должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях система должна выдавать пользователю соответствующие аварийные сообщения, после чего возвращаться в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

### Требования к безопасности

Все технические решения, использованные при создании программного изделия, а также при определении требований к аппаратному обеспечению, должны соответствовать действующим нормам и правилам техники безопасности, пожаробезопасности и взрывобезопасности, а также охраны окружающей среды при эксплуатации.

### Требования к эргономике и технической эстетике

Работа с системой должна осуществляться через приложение обладающее графическим, пользовательским интерфейсом.

Основные требования к интерфейсу:

* Интерфейс программы должен быть интуитивно понятным.
* Удобная навигация.
* Присутствие Главного меню.
* Удобное добавление и редактирование данных.
* Все формы должны быть открываться в удобном для пользователя месте и должны вмещаться в наиболее часто используемые разрешения экрана.
* Формы должны быть или single или могли раскрываться на весь экран с растяжением всех объектов находящихся в них.
* Заголовок окна должен соответствовать содержимому.
* Интерфейс должен быть многопользовательским.

### Требования к эксплуатации, техническому обслуживанию, ремонту и хранению комплектов системы

Для бесперебойной работы при регулярной, круглосуточной эксплуатации с еженедельными техническими осмотрами индивидуальным пользователем.

### Требования к защите информации от несанкционированного доступа

Разрабатываемая система должна обладать защитой от несанкционированного доступа. Обязательно наличие подсистемы отвечающей за регистрацию и авторизацию пользователей. Права доступа к информации хранящейся в базе данных назначаются системным администратором в соответствии с полномочиями и обязанностями каждого отдельного сотрудника.

### Требования к средствам защиты от внешних воздействий

Физическая защита аппаратных компонентов системы, носителей данных, резервирование ресурсов и текущее обслуживание реализуется при помощи технических и организационных средств.

### Дополнительные требования

Программное обеспечение системы должно автоматически восстанавливать свое функционирование при корректном перезапуске аппаратных средств. Система должна предусматривать возможность организации автоматического или ручного резервного копирования с использованием стандартных программных и аппаратных средств.

## Требования к функциям системы

Первоначальный ввод информации в базу данных с целью длительного хранения на диске, организация поступающих заказов от клиентов, формирование отчетов, обеспечение надежного функционирования.

## Требования к видам обеспечения:

### Требования к информационному обеспечению

Уровень хранения данных в системе должен быть построен на платформе объектно-реляционной СУБД в соответствии с техническими требованиями к комплексу средств автоматизации правительственных органов Российской Федерации. Для обеспечения целостности данных могут использоваться как встроенные механизмы СУБД, так и хранимые процедуры.

### Требования к программному обеспечению для сервера

Системное: операционная система Microsoft Windows Server 2003 и выше, драйвера для встроенных и внешних устройств. Прикладное: программы диагностики программных средств и антивирусной защиты.

### Требования к программному обеспечению для клиентских машин

Операционная система Microsoft Windows ХР, драйвера для встроенных и внешних устройств, MySQL. Прикладное: программы диагностики программных средств и антивирусной защиты.

### Требования к техническому обеспечению для сервера

* Компьютер с процессором с частотой не менее 133 МГц; рекомендуется 550 МГц или больше; поддержка до четырех процессоров на одном сервере
* ОЗУ не менее 128 МБ; рекомендуется 256 МБ и более; максимально 4 ГБ
* Необходим монитор VGA или оборудование с поддержкой перенаправления консоли; рекомендуется монитор Super VGA с разрешением 800 x 600 или более высоким.

### Требования к техническому обеспечению для клиентских машин

* Процессор Pentium с частотой 233 МГц или более быстрый (рекомендуется не менее 300 МГц)
* Не менее 64 МБ оперативной памяти (рекомендуется не менее 128 МБ)
* Клавиатура, мышь Microsoft Mouse или совместимое указывающее устройство
* Видеокарта и монитор, поддерживающие режим Super VGA с разрешением не менее чем 800x600 точек.

### Требования к составу и содержанию работ по созданию и развитию системы

По итогам испытаний по мере необходимости Исполнителем проводится корректировка проектной и эксплуатационной документации и устранение выявленных недостатков программного обеспечения. Указанные работы могут производиться на стадии опытной эксплуатации, если недостатки не носят характера и приемо-сдаточная комиссия сочтет возможным принять систему в эксплуатацию с замечаниями.

## Порядок контроля и приемки системы

Сдача – приемка работ осуществляет комиссия из трех человек: Кирпичев А.Н., Чадин И.Ф., Лавреш И.И.

## Требования к документированию

Руководство пользователя, Руководство администратора, описание программы.

# Инфологическая модель

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком (последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка). Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

Сущность – любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в базе данных. Сущностями могут быть люди, места, самолеты, рейсы, вкус, цвет и т.д. Необходимо различать такие понятия, как тип сущности и экземпляр сущности. Понятие тип сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое. Экземпляр сущности относится к конкретной вещи в наборе. Например, типом сущности может быть ЖАНР, а экземпляром – комедия, драма и т.д.

Атрибут – поименованная характеристика сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различного типа сущностей (например, ЦВЕТ может быть определен для многих сущностей: СОБАКА, АВТОМОБИЛЬ, ДЫМ и т.д.). Атрибуты используются для определения того, какая информация должна быть собрана о сущности. Примерами атрибутов для сущности АВТОМОБИЛЬ являются ТИП, МАРКА, НОМЕРНОЙ ЗНАК, ЦВЕТ и т.д. Здесь также существует различие между типом и экземпляром. Тип атрибута ЦВЕТ имеет много экземпляров или значений: Красный, Синий, Банановый, Белая ночь и т.д., однако каждому экземпляру сущности присваивается только одно значение атрибута.

Абсолютное различие между типами сущностей и атрибутами отсутствует. Атрибут является таковым только в связи с типом сущности. В другом контексте атрибут может выступать как самостоятельная сущность. Например, для автомобильного завода цвет – это только атрибут продукта производства, а для лакокрасочной фабрики цвет – тип сущности.

Ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся.

Связь – ассоциирование двух или более сущностей. Если бы назначением базы данных было только хранение отдельных, не связанных между собой данных, то ее структура могла бы быть очень простой. Однако одно из основных требований к организации базы данных – это обеспечение возможности отыскания одних сущностей по значениям других, для чего необходимо установить между ними определенные связи. А так как в реальных базах данных нередко содержатся сотни или даже тысячи сущностей, то теоретически между ними может быть установлено более миллиона связей. Наличие такого множества связей и определяет сложность инфологических моделей.

Стержень

Ассоциация

Ассоциация

Обозначение

Рис. 4. Обозначения инфологической модели.

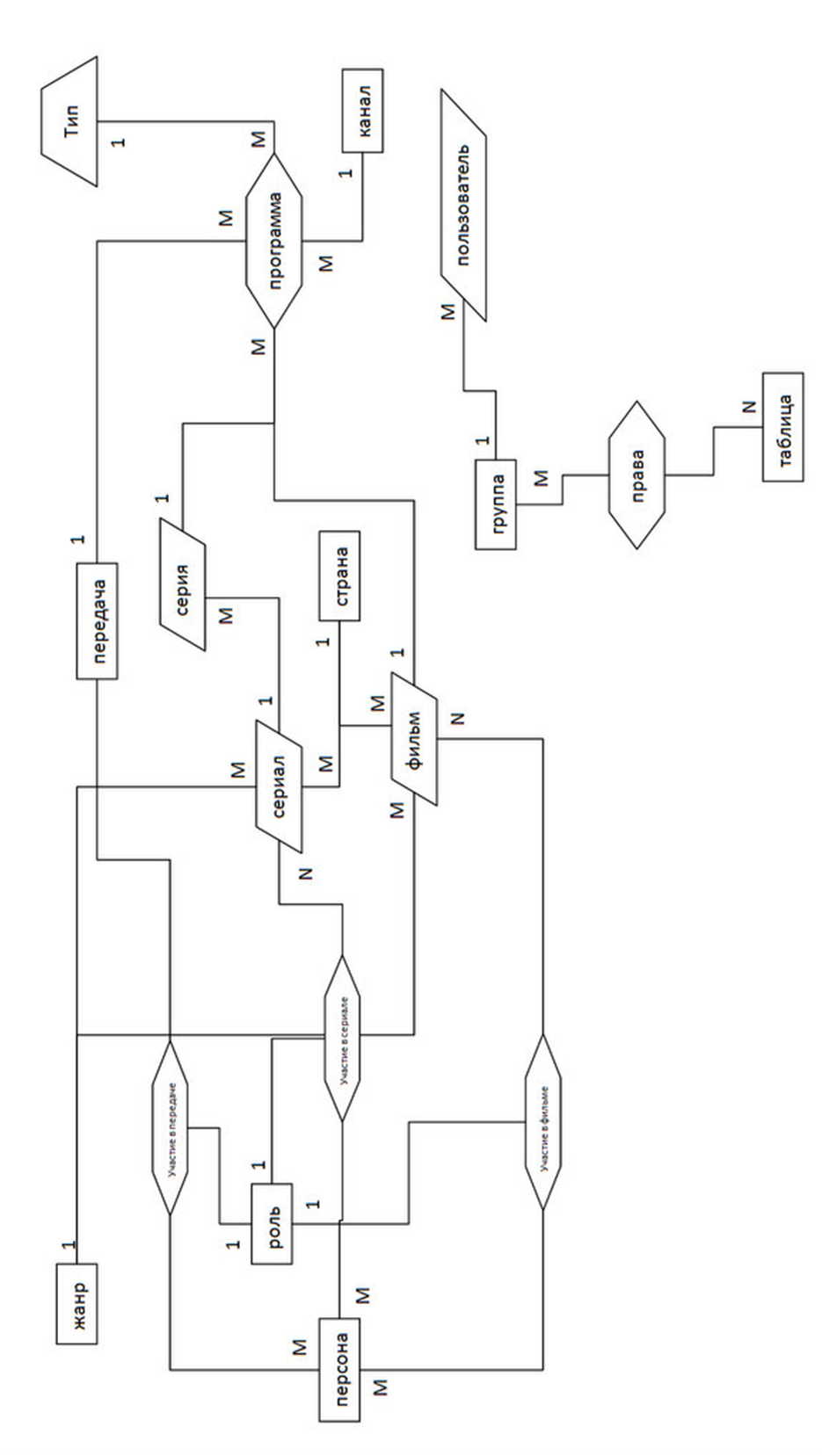


Рис. 5. Инфологическая модель без указания свойств.

# Даталогическая модель

Даталогическая модель данных – это совокупность отношений, содержащих всю информацию, которая должна храниться в БД. Однако пользователи могут воспринимать такую базу данных как совокупность таблиц.

Каждая таблица состоит из однотипных строк и имеет уникальное имя.

Строки имеют фиксированное число полей (столбцов) и значений (множественные поля и повторяющиеся группы недопустимы). Иначе говоря, в каждой позиции таблицы на пересечении строки и столбца всегда имеется в точности одно значение или ничего.

Строки таблицы обязательно отличаются друг от друга хотя бы единственным значением, что позволяет однозначно идентифицировать любую строку такой таблицы.

Столбцам таблицы однозначно присваиваются имена, и в каждом из них размещаются однородные значения данных (даты, фамилии, целые числа или денежные суммы).

Полное информационное содержание базы данных представляется в виде явных значений данных, и такой метод представления является единственным. В частности, не существует каких-либо специальных "связей" или указателей, соединяющих одну таблицу с другой.

При выполнении операций с таблицей ее строки и столбцы можно обрабатывать в любом порядке безотносительно к их информационному содержанию. Этому способствует наличие имен таблиц и их столбцов, а также возможность выделения любой их строки или любого набора строк с указанными признаками

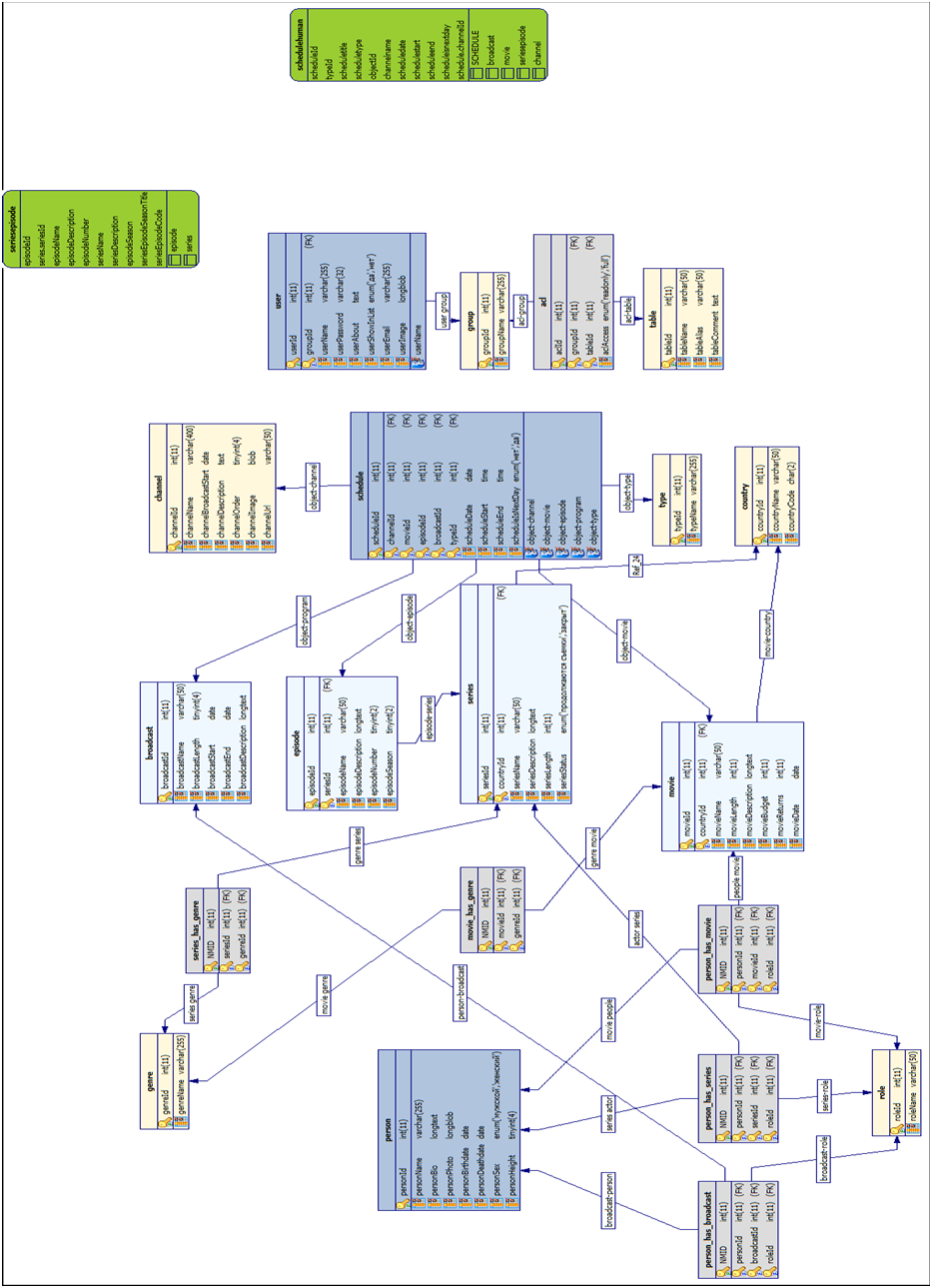


Рис. 6. Даталогическая модель.

Таблица №4. Таблица прав – «acl».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| aclId | Int(11) | PK, auto\_increment, unsigned, not null |
| groupId | Int(11) | FK, unsigned, , not null |
| tableId | Int(11) | FK, unsigned, , not null |
| aclAccess | enum('нет доступа', 'только чтение', 'полный доступ' | Default ‘только чтение’ |

В таблице «acl» хранятся права доступа пользователей ко всем важным таблицам базы данных.

Таблица №5 Таблица передач – «broadcast».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| broadcastId | Int(11) | PK, auto\_increment, unsigned, not null |
| broadcastName | Varchar(400) l | Not null |
| broadcastLength | Int(11) | Unsigned, default null |
| broadcastDescription | Longtext |  |

В таблице «broadcast» хранятся все передачи, которые выходят в эфир, в которые включают себя название передачи, её длину и описание.

Таблица №6. Таблица каналов – «channel».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| channelId | Int(11) | PK, auto\_increment, unsigned, not null |
| channelName | Varchar(400) | Not null |
| channelBroadcastStart | Date | Not null |
| channelDescription | Longtext |  |
| channelOrder | Tinyint(4) | Not null defaul ‘0’ |
| channelImage | Blob | Not null |
| channelUrl | Varchar(50) | Not null |

В таблице «channel» хранятся все каналы, которые показывают фильмы, передачи, сериалы. Включает в себя название канала, дату начала вещания, описание, порядок канала в сетке, логотип канала и ссылку на официальный сайт.

Таблица №7. Таблица стран– «country».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| countryId | Int(11) | PK, auto\_increment, unsigned, not null |
| countryName | Varchar(50) | Not null |
| countryCode | Char(2) | Not null |

В таблице «country» хранятся страны. Таблица содержит столбцы название и код.

Таблица №8. Таблица эпизодов – «episode».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| episodeId | Int(11) | PK, auto\_increment, unsigned, not null |
| seriesId | Int(11) | FK, unsigned, not null |
| episodeName | Varchar(50) | Not null |
| episodeDescription | Longtext |  |
| episodeNumber | Tinyint(2) | Unsigned zerofill, not null |
| episodeSeason | Tinyint(2) | Unsigned zerofill, not null |

В таблице «episode» хранятся эпизоды сериалов, которые включают в себя название, описание, порядковый номер в сезоне и сам сезон.

Таблица №9. Таблица жанров – «genre».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| genreId | Int(11) | PK, auto\_increment, unsigned, not null |
| genreName | Varchar(255) | Default null |

В таблице «genre» хранятся жанры фильмов, сериалов. Включает в себя название.

Таблица №10. Таблица групп – «group».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| groupId | Int(11) | PK, auto\_increment, unsigned |
| groupName | Varchar(255) | Not null |

В таблице «group» хранятся группы пользователей. Содержит столбец с названием группы.

Таблица №11. Таблица фильмов – «movie».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| movieId | Int(11) | PK, auto\_increment, unsigned, not null |
| countryId | Int(11) | FK, unsigned, not null |
| movieName | Varchar(50) | Not null |
| movieLength | Int(11) | Unsigned default null |
| movieDescription | Longtext |  |
| movieBudget | Int(11) | Unsigned not null |
| movieDate | date | Not null |

В таблице «фильмы» хранятся фильмы. Содержит в себе страну производства, название, длительность, описание, бюджет и дату премьерного показа.

Таблица №12. Таблица жанров фильмов – «movie\_has\_genre».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| NMID | Int(11) | PK, auto\_increment, not null |
| movieId | Int(11) | FK, unsigned, not null |
| genreId | Int(11) | FK, unsigned, not null |

В таблице «movie\_has\_genre» хранятся ссылки на фильм и жанр. Таблица показывает к каким жанрам принадлежит каждый фильм.

Таблица №13. Таблица персон – «person».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| personId | Int(11) | PK, auto\_increment, unsigned, not null |
| personName | Varchar(255) | Not null |
| personBio | Longtext |  |
| personBirthdate | Date | Default null |
| personDeathdate | Date | Default null |
| personSex | Enum(‘мужской’,’женский’) | Default null |
| personHeight | Tinyint(4) | Unsigned defaul null |

В таблице «person» хранятся персоны, которые снимаются в фильмах и сериах, а также участвуют в передачах. Содержит столбцы: имя, биография, дата рождения, дата смерти, пол, рост.

Таблица №14. Таблица персон передач – «person\_has\_broadcast».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| NMID | Int(11) | PK, auto\_increment, not null |
| personId | Int(11) | FK, unsigned, not null |
| broadcastId | Int(11) | FK, unsigned, not null |
| roleId | Int(11) | FK, unsigned, not null |

В таблице «person\_has\_broadcast» хранятся ссылки на персону, передачу и роль. Таблица показывает в каких передачах участвовала персона и её роль в данной передаче.

Таблица №15. Таблица персон фильмов – «person\_has\_movie».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| NMID | Int(11) | PK, auto\_increment, not null |
| personId | Int(11) | FK, unsigned, not null |
| movieId | Int(11) | FK, unsigned, not null |
| roleId | Int(11) | FK, unsigned, not null |

В таблице «person\_has\_move» хранятся ссылки на персону, фильм и роль. Таблица показывается в каких фильмах участвовала персона и её роль в данном фильме.

Таблица №16. Таблица персон сериалов – «person\_has\_series».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| NMID | Int(11) | PK, auto\_increment, not null |
| personId | Int(11) | FK, unsigned, not null |
| seriesId | Int(11) | FK, unsigned, not null |
| roleId | Int(11) | FK, unsigned, not null |

В таблице «person\_has\_series» хранятся ссылки на персону, сериал и роль. Таблица показывается в каких сериалах участвовала персона и её роль.

Таблица №17. Таблица расписания – «schedule».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| scheduleId | Int(11) | PK, auto\_increment, unsigned, not null |
| channelId | Int(11) | FK, unsigned |
| movieId | Int(11) | FK, unsigned |
| episodeId | Int(11) | FK, unsigned |
| broadcastId | Int(11) | FK, unsigned |
| typeId | Int(11) | FK, unsigned, not null |
| scheduleDate | Date | Not null |
| scheduleStart | Time | Not null |
| scheduleEnd | Time | Not null |
| scheduleIsNextDay | Enum(‘да’,’нет’) | Not null default ‘нет’ |

В таблице «schedule» хранится расписание трансляций фильмов, передач, сериалов. Содержит ссылки на канал, передачу, эпизод, сериал, тип, а также дату, время начала и конца, поле «следующего дня».

Таблица №18. Таблица сериалов – «series».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| seriesId | Int(11) | PK, auto\_increment, unsigned, not null |
| countryId | Int(11) | FK, unsigned, not null |
| seriesName | Varchar(50) | Not null |
| seriesStatus | Enum(‘закрыт’, ‘продолжаются съемки’) |  |
| seriesDescription | Longtext | Not null |
| seriesLength | Int(11) | Unsigned default null |

В таблице «series» хранятся сериалы. Таблица содержит столбцы: страна, название, статус, описание, длительность.

Таблица №19. Таблица жанров сериалов – «series\_has\_genre».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| NMID | Int(11) | PK, auto\_increment, not null |
| seriesId | Int(11) | FK, unsigned, not null |
| genreId | Int(11) | FK, unsigned, not null |

В таблице «series\_has\_genre» хранятся ссылки на сериал и жанр. Таблица показывает все жанры конкретного сериала.

Таблица №20. Таблица таблиц – «table».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| tableId | Int(11) | PK, auto\_increment, unsigned, not null |
| tableName | Varchar(50) | Not null |
| tableAlias | Varchar(50) | Not null |
| tableComment | Text |  |

В таблице «table» хранятся таблицы. Содержит название и комментарий. Таблица используется для контроля доступа.

Таблица №21 Таблица типов – «type».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| typeId | Int(11) | PK, auto\_increment, unsigned, not null |
| typeName | Varchar(255) | Not null |

В таблице «type» хранятся типы транслируемого контента. Содержит в себе название.

Таблица №22. Таблица пользователей – «user».

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Комментарий |
| userId | Int(11) | PK, auto\_increment, unsigned, not null |
| groupId | Int(11) | FK, unsigned, not null |
| userName | Varchar(255) | Not null |
| userPassword | Varchar(32) | Not null |
| userAbout | Text |  |
| userShowInList | Enum(‘да’,’нет’) | Not null default ‘да’ |
| userEmail | Varchar(255) | Default null |
| userImage | longblog |  |

В таблице «user» хранятся пользователи систиемы. Содержит в себе ссылку на группу, пароль, описание, служебное поле «показывать в списке», адрес электронной почты, фотографию.

# Описание Web-приложения

Приложение построено с учетом самых современных технологий и стандартов, используемых во всемирной паутине.

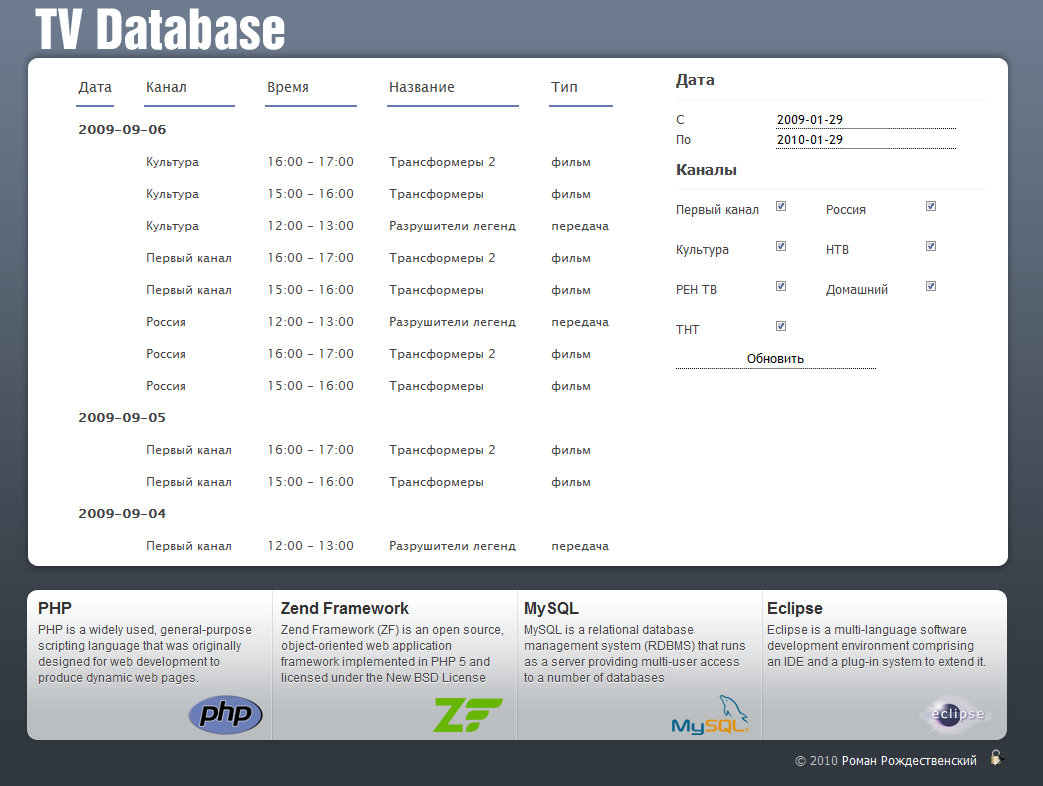


Рис. 7. Главная страница

Для обновления информации о телепрограмме используется технология AJAX, что позволяет снизить нагрузку на сервер, так как необходимо не полностью генерировать страницу, а загружать только необходимую информацию



Рис. 8 AJAX-запрос к серверу

Приложение построено с использованием паттернов проектирования: MVC, singleton, data table gateway, frontController и других. Использование MVC позволяет отделить бизнес-логику приложения от оформления и вести разработку параллельно.

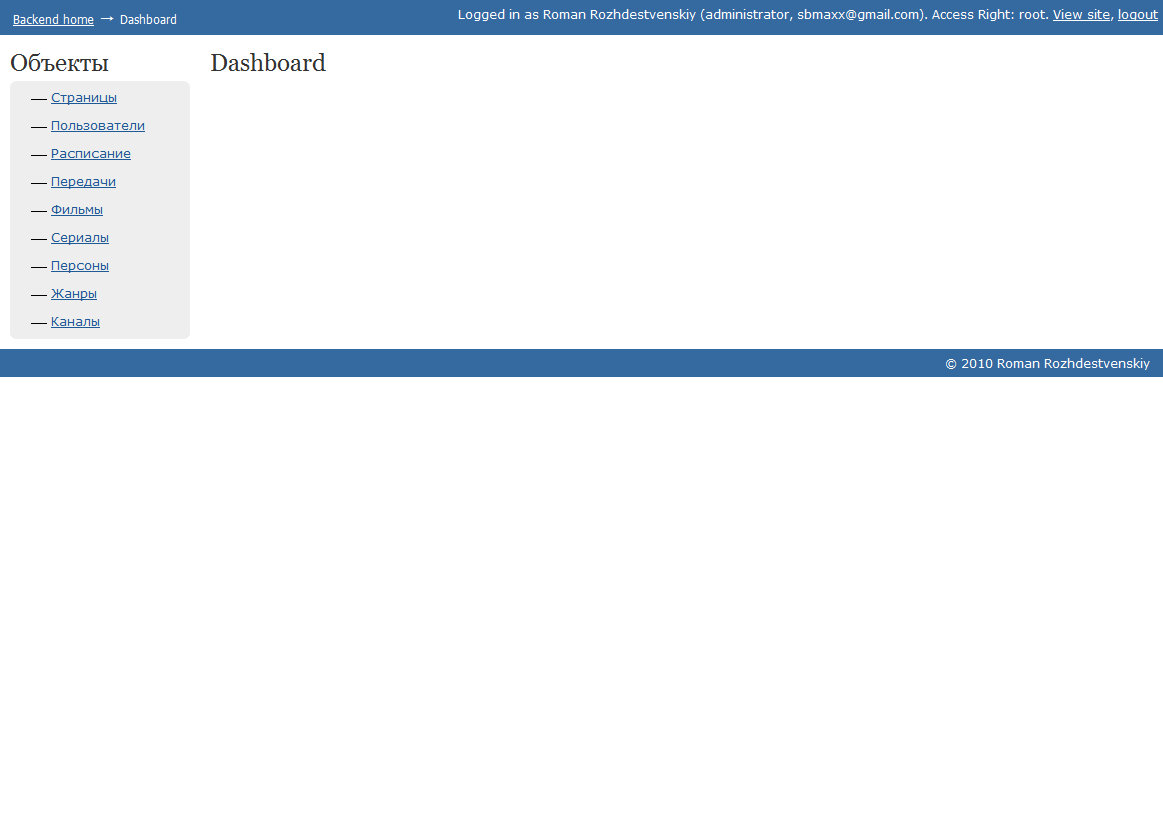


Рис. 9. Начальная страница панели администратора

Описание форм

## Форма редактирования канала

С помощью данной формы можно изменять всю необходимую информацию о телеканале:

* Название
* Дату начала вещания
* Ссылку на интернет-сайт
* Описание
* Логотип

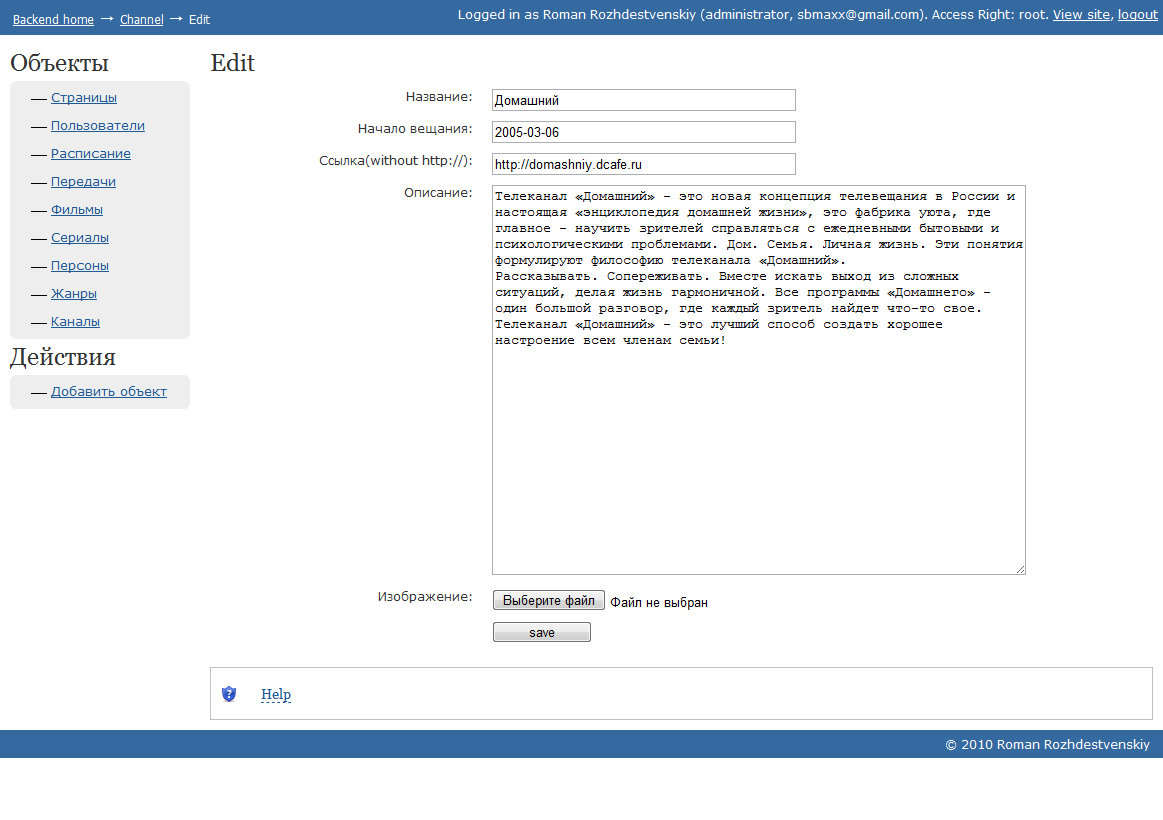


Рис. 10. Форма редактирования канала.

Все поля формы проходят валидацию, а данные перед отправкой на сервер СУБД экранируются в автоматическом режиме.

## Форма авторизации

Форма авторизации предназначена для аутентификации пользователя в системе с целью предоставления ему доступа в раздел администратора.

Логика авторизации защищена от SQL-инъекций. После авторизации данные сохраняются в сессии, а при выборе галочки «запомнить» сохраняются в cookies на две недели.

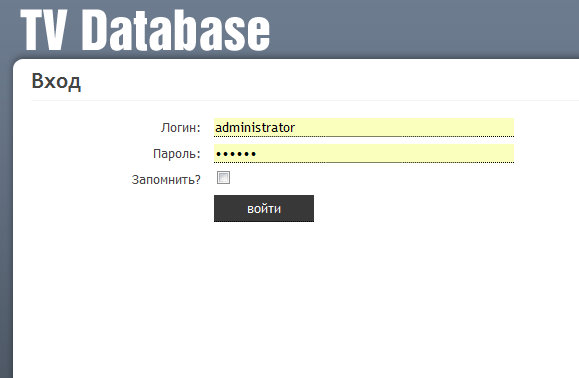


Рис. 11. Форма авторизации

# Описание системы отчетности

Для того, чтобы посмотреть отчет по любой из сущностей системы необходимо авторизоваться и пройти в панель администратора, после этого в левой части необходимо выбрать объект

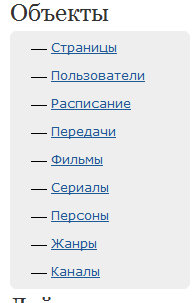


Рис. 12 Список отчетов

## Отчет о пользователях

Данный отчет показывает краткую информацию о пользователях:

* идентификатор
* логин
* email
* фио

Подробные данные отображаются для каждого из пользователей по отдельности

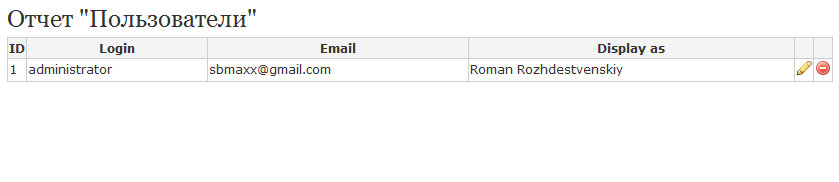


Рис. 13 Отчет «пользователи»

## Отчет о каналах

Данный отчет предназначен для отображения информации о списке каналов, которые присутствуют в системе:

* Идентификатор
* Название
* Дата начала вещания
* Изображения
* Ссылка на официальный сайт

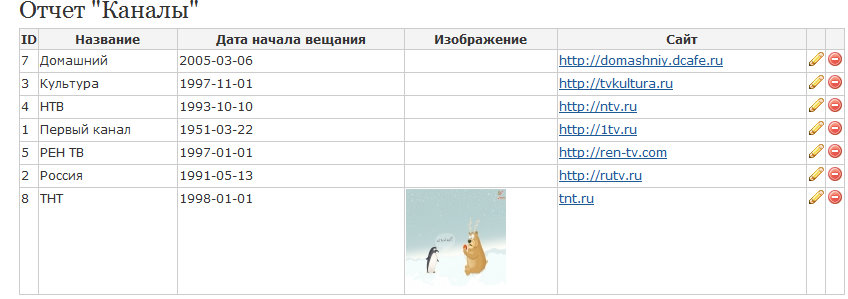


Рис. 14 Отчет «каналы»

# Описание RSS-каналов

В системе используется RSS-поток для оперативного оповещения пользователя об обновлениях телепрограммы.

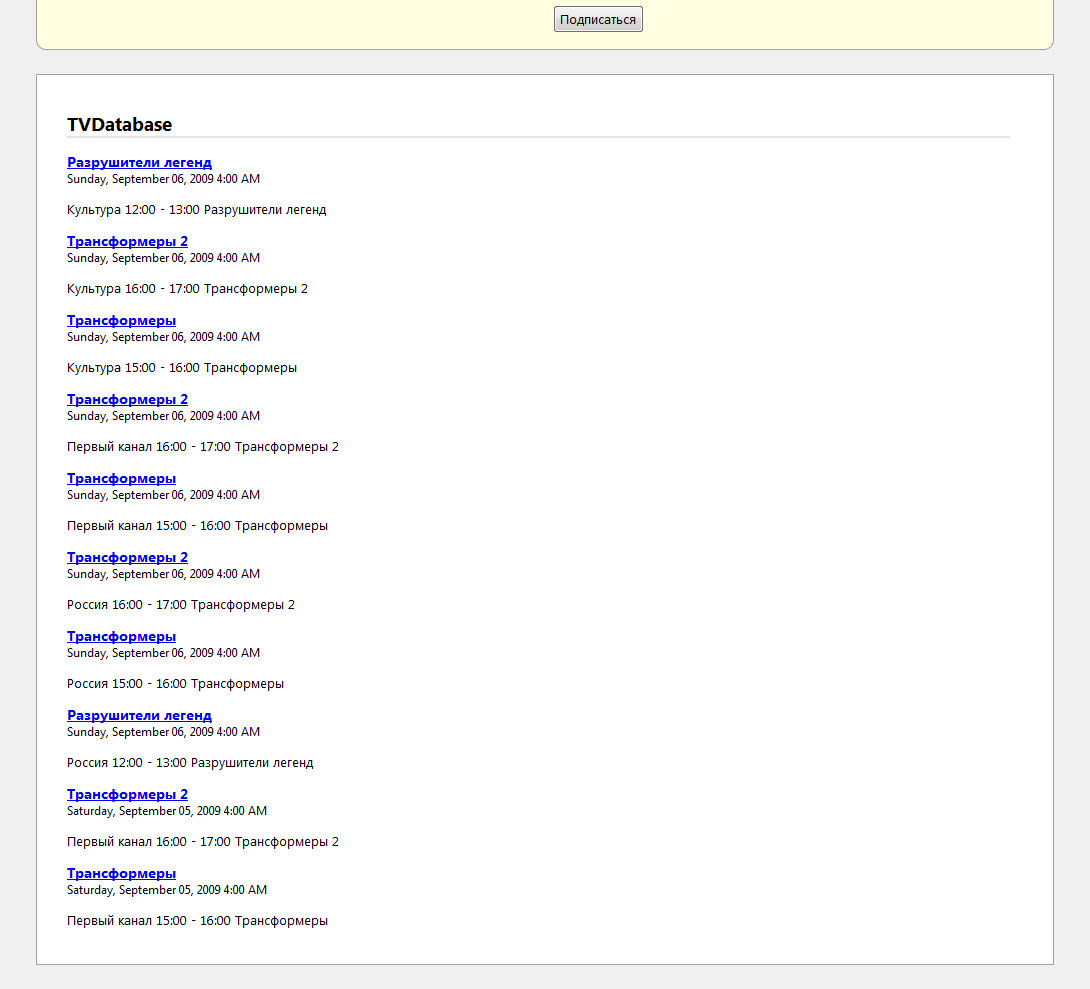


Рис. 15. Отображение RSS-потока в Mozilla Firefox

Для подписки на поток можно воспользоваться любым из бесплатных агрегаторов, например google reader.

# Заключение и выводы

В процессе разработки курсовой работы использовались следующие ПП:

* Eclipse PDT для написание php, css, html, js кода
* Microsoft Windows 7 RC: очень удобная операционная система;
* Firefox: отладка ajax-запросов
* Paint.Net: обработка фотографий, иконок;
* MicroOLAP MySQL Designer: составление схемы БД;
* Microsoft Office: лучший офисный пакет;
* Far: для управления всеми файлами данной программы;
* Picasa: обработка скриншотов;

Во время разработки курсовой работы были подробно изучены все технологии дисциплины «Технологии Интернет», а именно PHP, HTML, CSS, XML, JavaScript.

# Библиографический список

1. **Федорчук А.** Как создаются Web-сайты: краткий курс [текст] / А. Федорчук – СПб: Питер, 2000. – 224 с.
2. **Сергеев, А. П.** HTML и XML. Профессиональная работа [текст] / А.П. Сергеев – М.: Издательский дом «Вильямс», 2004. – 880 с.
3. **Котеров Д.** PHP. В подлиннике [текст] / Д. Котеров — СПб.: «БХВ-Петербург», 2005. —1120 с.
4. **Кириллов, В.В.** Структурированный язык запросов (SQL) [текст] / В.В. Кирилов – СПб.: ИТМО, 1994. – 80 с.
5. **Мейер, М.** Теория реляционных баз данных [текст] / М. Мейер – М.: Мир, 1987. – 608 с.
6. **Хаббардь, Дж**. Автоматизированное проектирование баз данных. [текст] / Дж. Хаббард – М.: Мир, 1984. – 294 с.
7. **Wikipedia.org**: Свободная энциклопедия [Электронный ресурс]. URL: <http://ru.wikipedia.org> (дата обращения: 20.01.2010)

Приложение A. Руководство пользователя

Приложение обладает простым и интуитивно понятным интерфейсом.

Для работы с приложением необходимо никаких специальных знаний, достаточно умения обращения с web-сайтом и браузером.

Приложение B. Руководство администратора

Для установки приложения необходимо настроить виртуальный хост в веб-сервере apache, установить дамп базы mysql и внести необходимые настройки в файл config.ini.

Настройки, необходимые для корректной работы приложения:

* сервер базы данных
* имя пользователя базы данных
* пароль пользователя базы данных
* порт для подключения к базе данных

Кроме этого для корректной загрузки картинок на операционных системах \*NIX необходимо поставить права на запись для директории data и рекурсивно вниз по дереву.

1. URL: http://ru.wikipedia.org/wiki/Информационная\_Система [↑](#footnote-ref-1)
2. URL: http://ru.wikipedia.org/wiki/База\_данных [↑](#footnote-ref-2)
3. URL: http://ru.wikipedia.org/wiki/Бизнес-процесс [↑](#footnote-ref-3)