# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## Наименование

Информационная система «Моя фильмотека».

## Основания для разработки.

Информационная система «Моя фильмотека» это отличный способ всегда быть в курсе своих предпочтений в кинематографе. Имея всегда под рукой возможность, сохранить для будущего просмотра фильм или узнать, просмотрен ли уже какой-либо фильм.

Если обратить внимание и провести опрос часто ли люди запоминают советы друзей, знакомых что же посмотреть, то можно сделать вывод что большинство, прейдя вечером домой уже не помнят что им советовали друзья. «Моя фильмотека» призвана решить данную проблему.

Также возможно будет узнать о уже просмотренных фильмах, о фильмах, которые смотрят, посмотрели, хотят посмотреть ваши друзья зарегистрированные в сервисе. Возможно, почитать отзывы, посмотреть рейтинг, используя при этом простой и дружелюбный интерфейс.

## Исполнитель

Исполнителем проекта является студент Калужского филиала МГТУ им. Н. Э. Баумана, факультета ФНК, группы ИТД – 111, Редькин Степан Дмитриеви.

## Назначение и цель разработки системы

Цель создания системы – разработать серверное приложение в виде REST-сервисов, а также клиентское WEB-приложение и Android приложение.

## Плановые сроки начала и окончания работы по созданию системы

Планируемые сроки начала и окончания работы над проектом: 01.09.2013 – 15.05.2014.

## Требования к программному продукту

### Требование к составу программных компонентов

Для функционирования программ и сервисов, входящих в данный пакет необходимо наличие следующих программных продуктов:

* \*NIX, Microsoft Windows 2K\XP\2003\Vista\Windows 7;
* Web-сервер Apache;
* Контейнер сервлетов, например Apache Tomcat
* Web-браузер или устройство на мобильной ОС Android

### Требования к прикладным программам

Для разработки программы потребуются следующие программные средства:

* *Язык программирования Java* – объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируется в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине (JVM) вне зависимости от компьютерной архитектуры. Дата официального выпуска — 23 мая 1995 года.
* *Язык программирования* [JavaScript](http://www.webmasterwiki.ru/JavaScript) – это созданный фирмой Netscape межплатформенный объектно-ориентированный язык сценариев. [JavaScript](http://www.webmasterwiki.ru/JavaScript" \o "Java Script) это небольшой компактный язык; он не используется для создания самостоятельных приложений, а разработан специально для упрощенного внедрения в другие приложения и продукты, такие как web-браузеры. Находясь внутри рабочей среды-хозяина, [JavaScript](http://www.webmasterwiki.ru/JavaScript" \o "Java Script) может подключаться к объектам этой среды для предоставления программного управления ими. [JavaScript](http://www.webmasterwiki.ru/JavaScript" \o "Java Script) может быть расширено для различных целей путём дополнения его новыми объектами; например: Клиентский [JavaScript](http://www.webmasterwiki.ru/JavaScript" \o "Java Script) расширяет ядро языка, добавляя объекты управления браузером (Navigator или другой аналогичный web-браузер) и **Document Object Model**/**Объектную Модель Документа** (DOM). Например, клиентские расширения позволяют приложению размещать элементы на HTML-форме и реагировать на пользовательские события, такие как щелчок мышью, ввод данных в форму или навигация по страницам. Серверный [JavaScript](http://www.webmasterwiki.ru/JavaScript" \o "Java Script) расширяет ядро языка, добавляя объекты, относящиеся к запуску [JavaScript](http://www.webmasterwiki.ru/JavaScript" \o "Java Script) на сервере. Например, серверные расширения позволяют приложению взаимодействовать с реляционной базой данных, сохраняя информацию между вызовами приложения, или выполнять манипуляции с файлами на сервере. Посредством функциональности LiveConnect, Вы можете осуществлять взаимодействие кодов Java и [JavaScript](http://www.webmasterwiki.ru/JavaScript" \o "Java Script). Из [JavaScript](http://www.webmasterwiki.ru/JavaScript" \o "Java Script) Вы можете инстанциировать Java-объекты и получать доступ к их public-методам и полям. Из Java Вы можете иметь доступ к объектам, свойствам и методам [JavaScript](http://www.webmasterwiki.ru/JavaScript" \o "Java Script).
* *Apache Tomcat*  — контейнер сервлетов с открытым исходным кодом, разрабатываемый Apache Software Foundation. Реализует спецификацию сервлетов и спецификацию JavaServer Pages (JSP) и JavaServer Faces (JSF). Написан на языке Java. Tomcat позволяет запускать веб-приложения, содержит ряд программ для самоконфигурирования.
* *Git* (произн. «гит») — распределённая система управления версиями файлов. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день поддерживается Джунио Хамано.

Будут использованы следующие средства разработки базы данных и организации работы с ней:

* *MySQL 5.6.5* – реляционная система управления базами данных (СУБД), а также компактный многопоточный сервер. MySQL характеризуется большой скоростью, устойчивостью и легкостью в использовании. MySQL поддерживает язык запросов SQL в стандарте ANSI 92, и кроме этого имеет множество расширений к этому стандарту, которых нет ни в одной другой СУБД.
* *PHPMyAdmin 4.1.0* – простая и удобная система для работы с СУБД MySQL. Разработанный на языке PHP, PHPMyAdmin легко размещается на Web-сервере, позволяя администрировать базу данных через WEB-браузер без использования дополнительного ПО.

### Требования к архитектуре программы

Система должна быть реализована по средствам клиент-серверной архитектуры.

### Требование к составу технических средств

Так как система является кроссплатформенной и ее компоненты могут функционировать на различных операционных системах, аппаратные требования однозначно установить невозможно. Следовательно, для нормального функционирования всех компонентов разрабатываемой системы необходимо иметь аппаратное обеспечение удовлетворяющее требованиям используемой ОС.

### Требования к надежности

Серверная часть приложения может быть доступна только авторизированным пользователям. Сервер также изолирует базу данных от клиентов, не позволяя работать с данными напрямую.

### Требования к входным – выходным данным

В процессе работы компоненты системы будут активно обмениваться данными по сети.

Пользователь обращается к системе посредством графического интерфейса Web-браузера или клиентского приложения для ОС Android.

Клиент запрашивает и получает данные от сервера посредством web-сервисов, основанных на протоколе HTTP.

Обращения сервера к базе данных должны быть организованы в виде SQL-запросов.

## Стадии и этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| **Стадии разработки** | **Этапы**  **работ** | **Содержание работ** | **Сроки** |
| Техническое задание | Обоснование необходимости разработки программы | Постановка задачи.  Сбор исходных материалов. Выбор и обоснование критериев эффективности и качества разрабатываемой программы. | 01.09.2013-14.09.2013 |
| Разработка и утверждение технического задания | Определение требований к программе. Разработка технико-экономического обоснования разработки программы. Определение стадий, этапов и сроков разработки программы и документации на нее. Выбор языков программирования. Согласование и утверждение технического задания. | 14.09.2013- 18.09.2013 |
| Эскизный проект | Разработка эскизного проекта | Предварительная разработка структуры входных и выходных данных. Проектирование модели базы данных. Разработка общего описания алгоритма решения задачи.  Уточнение методов решения задачи. | 18.09.2013-06.10.2013 |
| Технический проект | Разработка технического проекта | Уточнение структуры входных и выходных данных.  Реализация разработанной модели БД. Разработка алгоритма решения задачи. Определение формы представления входных и выходных данных. Разработка структуры программы. Разработка серверного приложения. Установка протоколов взаимодействия БД и серверной части.  Разработка интерфейса для работы пользователя с приложением. Окончательное определение конфигурации технических средств. | 06.10.2013-25.11.2013 |
| Рабочий проект | Разработка программы | Программирование и отладка программы | 25.11.2013-10.04.2014 |
| Разработка программной документации | Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77. | 10.04.2014- 01.05.2014 |
| Испытания программы | Разработка, согласование и утверждение программы и методики испытаний. Корректировка программы и программной документации по результатам испытаний. | 01.05.2014- 15.05.2014 |

## Техническая документация, предъявляемая по окончанию работы

Вместе с программным обеспечением поставляется пояснительная записка объемом 50-70 стр:

* Титульный лист;
* Оглавление;
* Техническое задание на проект;
* Исследовательская часть (обзор литературы и способов решения, обоснование выбора способа решения, аппаратной и программной платформ, среды разработки)
* Конструкторская часть (разработка алгоритма работы приложения, разработка алгоритмов модулей, реализация их в выбранной среде разработки);
* Технологическая часть (формулирование требований к аппаратной и программной платформам, руководство по установке и администрированию приложения, руководство пользователя).
* Организационно-экономическая часть (экономическое обоснование дипломного проекта, вычисление себестоимости готовой системы и ее окупаемости).
* Охрана труда и защита окружающей среды (Основные критерии обеспечения труда при работе с компьютером)
* Заключение (выводы)

Также предоставляются 10 листов формата А1, содержащих структурные схемы, основные алгоритмы, этапы проектирования системы и демонстрационные чертежи.

# ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

## Обоснование выбора темы проекта

В современном мире количество информации огромно. Человеческий мозг не справляется и не может помнить все. Когда друзья или знакомые советует вам посмотреть какой-нибудь фильм, а ты уже не можешь вспомнить смотрел ты его или нет, про что он. Было бы полезно иметь быстрый доступ к информации о просмотре данного фильма, кратком описании, трейлере. Также когда друзья советуют вам посмотреть какой-либо фильм, как часто вы его вспоминали дома? Опять же было бы полезно иметь под рукой способ, для быстрого сохранения данной информации. Так как в современном мире доступ в интернет существует практически всегда и везде, всегда под рукой имеется гаджет, имеющий доступ в интернет, то было бы замечательно иметь такой сервис, позволяющий хранить информацию о фильмах для просмотра, фильмах, которые уже просмотрены. Именно такой сервис и был выбран в качестве дипломного проекта.

## Сравнение с существующими решениями

В данный момент существуют 2-3 подобных решений, но они либо имеют неудобный и не совсем понятный интерфейс, либо не имеют мобильного клиента. Есть один сервис с удобным интерфейсом и мобильными клиентами под все ныне существующие платформы, но он только для сериалов.

## Обоснование выбора способа решения

### Обоснование выбора архитектуры системы

В основу разрабатываемого программного комплекса положена клиент-серверная архитектура (client-server architecture). Такая архитектура характеризуется наличием двух взаимодействующих самостоятельных процессов – клиента и сервера, которые, в общем случае, могут выполняться на разных компьютерах, обмениваясь данными по сети.

Под клиентом понимается программа, использующая ресурсы, а под сервером – программа, обслуживающая запросы клиентов на получение ресурсов определенного вида.

Понятие архитектуры клиент-сервер связано с делением любой прикладной программы на три основных компонента или слоя. Этими тремя компонентами являются:

* компонент представления (визуализации) данных;
* компонент прикладной логики;
* компонент управления базой данных.

Критерием, позволяющим отнести прикладную программу к клиент-серверной архитектуре является то, что хотя бы один из трех ее компонентов полностью выполняется на другом компьютере, и взаимодействие между компонентами на разных компьютерах осуществляется через ту или иную сетевую среду посредством передачи запросов на получение того или иного ресурса.

Поскольку архитектура клиент-сервер является частным случаем технологии клиент-сервер, в ней обязательно есть клиент и сервер. Соответственно, выделяют клиентскую и серверную стороны приложения. Клиентская сторона приложения функционирует на рабочем месте пользователя, в роли которого в подавляющем числе случаев выступает персональный компьютер. Серверная сторона функционирует на специализированном комплексе, включающем в себя мощные аппаратные средства, требуемый набор стандартного программного обеспечения, систему управления базами данных и собственно структуры данных.

Взаимодействие клиентской и серверной частей приложения осуществляется через сеть - локальную или глобальную. При этом с точки зрения клиента и сервера взаимодействие осуществляется прозрачно, соответственно сетевой компонент здесь включает в себя совокупность необходимого сетевого оборудования, набор программных технологий, обеспечивающих передачу данных между узлами сети, а также собственно протокол или протоколы для обмена запросами и результатами их выполнения.

Компонент визуализации прикладной задачи осуществляет ввод информации пользователем с помощью тех или иных средств, а также вывод информации на экран. Компонент визуализации для клиент-серверной архитектуры всегда исполняется на рабочем месте пользователя.

Компонент прикладной логики решает собственно ту или иную задачу, связанную с обработкой данных в той или иной предметной области. Этот компонент может быть распределен между клиентской и серверной частью различным образом в зависимости от применяемой модели.

Компонент хранения базы данных осуществляет физические операции, связанные с хранением данных, чтением информации из базы данных и записью в нее. В архитектуре клиент-сервер этот компонент всегда выполняется на сервере.

Сети на базе серверов имеют лучшие характеристики и повышенную надежность и безопасность. Сервер владеет главными ресурсами сети, к которым обращаются остальные рабочие станции.

В современной клиент–серверной архитектуре выделяется четыре группы объектов: клиенты, серверы, данные и сетевые службы. Клиенты располагаются в системах на рабочих местах пользователей. Данные в основном хранятся в серверах. Сетевые службы являются совместно используемыми серверами и данными. Кроме того службы управляют процедурами обработки данных.

Сети клиент–серверной архитектуры имеют следующие преимущества:

* позволяют организовывать сети с большим количеством рабочих станций;
* обеспечивают централизованное управление учетными записями пользователей, безопасностью и доступом, что упрощает сетевое администрирование;
* эффективный доступ к сетевым ресурсам;
* пользователю нужен один пароль для входа в сеть и для получения доступа ко всем ресурсам, на которые распространяются права пользователя.

Наряду с преимуществами сети клиент–серверной архитектуры имеются и некоторые недостатки:

* неполадки сервера могут сделать сеть неработоспособной, что повлечет за собой как минимум потерю сетевых ресурсов;
* требуются квалифицированный персонал для администрирования;
* дополнительные денежные затраты на сеть и сетевое оборудование.

Для реализации системы можно выделить два подхода разработки распределенного клиент-серверного приложения.

* Трехзвенное приложение с «толстым» клиентом;
* Web-ориентированное приложение.

Из недостатков первого подхода ключевыми являются: необходимость разработки отдельного приложения (приложений, т.к. планируются несколько АРМов) под все платформы (Windows, Android и/или iOS), сложность поддержки.

Напротив, web-сервис освобожден от таких недостатков, т.к. для доступа к функционалу и контенту приложения используется веб-браузер, который присутствует на любой современной платформе. Среди недостатков, можно выделить: необходимость верстки страниц под разное разрешение экранов.

Таким образом, принято решение использовать клиент-серверную архитектуру для разработки системы.

### Обоснование выбора СУБД

Перед выбором СУБД для дипломного проекта были определены следующие требования:

* наличие бесплатной версии системы (freeware) или версии под лицензиями «GNU/GPL»;
* поддержка транзакций при высокой производительности;
* доступность документации и информации о СУБД в Интернете;
* наличие дистрибутива под ОС «Windows XP» и под UNIX-подобную систему;
* возможность доступа к данным через интерфейс DBI.

Исходя из требований, выбор СУБД для использования в рамках данного проекта велся среди различного программного обеспечения. Были рассмотрены следующие СУБД:

* «MySQL»;
* «PostgreSQL»;

Все системы являются достойными и надежными СУБД. Взглянем на них немного поближе:

***MySQL*** это популярная бесплатная СУБД уровня малого бизнеса. К ее достоинствам можно отнести простоту и высокую производительность. Система поддерживает несколько типов таблиц и до последних версий системы ни один вид таблиц не поддерживал транзакций, за счет чего достигается высокая производительность. Новые же типы таблиц, с поддержкой транзакций, отличаются низкой производительность. Эта система в основном используется в Интернет проектах, где цена потери данных при одновременном доступе невысока.

***PostgreSQL***, как и *MySQL*, является бесплатной и доступна в исходных кодах. Это система уровня малого и среднего бизнеса. Судя по описанию в Интернете, система имеет множество положительных черт, однако есть очевидная проблема – отсутствие качественной документации. Разработчик периодически размещает на своем сайте новые версии системы и любой желающий может изучить систему по скудным комментариям в исходных кодах.

В таблице 2.1 подведен итог характеристик рассматриваемых СУБД и их соответствие указанным критериям. Степень соответствия каждому критерию указана оценкой в пятибалльной шкале.

*Таблица 2.1 «Характеристики СУБД»*

|  |  |  |
| --- | --- | --- |
| **Критерий** | **СУБД** | |
| **MySQL** | **PostgreSQL** |
| Поддержка транзакций и высокая производительность | 4 | 4 |
| Наличие кэша | 5 | 5 |
| Наличие документации и справочной информации | 5 | 2 |
| Дистрибутив под ОС «Windows» | 5 | 4 |
| Дистрибутив под UNIX-подобную систему | 5 | 5 |

После простого подсчета баллов, верх одержала СУБД «MySQL» с 24 из 25 возможных баллов. «PostgreSQL» набрала 20 баллов. Таким образом, для дипломного проекта будет использована СУБД «MySQL».

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Основными преимуществами СУБД MySQL следует считать:

* MySQL обладает развитыми возможностями масштабирования и адми­нистрирования.
* MySQL поддерживает API к целому ряду языков программирования (вклю­чая Perl, C, PHP и т. д.).
* MySQL может использоваться в среде многих различных систем UNIX, а также в среде Microsoft Windows.
* многопоточность, поддержка нескольких одновременных запросов;
* оптимизация связей с присоединением многих данных за один проход;
* записи фиксированной и переменной длины;
* ODBC драйвер;
* гибкая система привилегий и паролей;
* гибкая поддержка форматов чисел, строк переменной длины и меток времени;
* быстрая работа, масштабируемость;
* совместимость с ANSI SQL;
* бесплатна в большинстве случаев;
* хорошая поддержка со стороны провайдеров услуг хостинга;
* быстрая поддержка транзакций через механизм InnoDB.

### Обоснование выбора WEB-сервера Apache

Перед выбором Web-сервера для дипломного проекта были определены следующие требования:

* наличие бесплатной версии системы (freeware) или версии под лицензиями «GNU/GPL»;
* распространенность;
* надежность;
* гибкость;
* доступность документации и информации в Интернете;

Исходя из требований, выбор сервера для использования в рамках данного проекта велся среди различного программного обеспечения. Были рассмотрены следующие WEB-сервера:

* «Apache HTTP Server»;
* «nginx»;
* «Microsoft IIS 7.0»

Рассмотрим каждый из этих серверов немного подробнее.

***Apache HTTP-сервер.*** С апреля 1996 это самый популярный HTTP-сервер в Интернете; в августе 2007 года он работал на 51% всех веб-серверов, а в мае 2010 года – на 48%. По данным *Netcraft (www.netcraft.com)* на январь 2011 года, более 160 млн. сайтов обслуживаются веб-сервером Apache, что составляет 59% от общего числа веб-сайтов. Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает IPv6.

***nginx*** – это очень быстрый HTTP сервер, который часто используют вместо или совместно с Apache, чтобы снизить нагрузку на сервер и увеличить скорость обработки запросов посетителей. Модульная архитектура дает Apache огромные возможности, которые, однако, большинство из нас никогда не будет использовать. Ценой за эту функциональность является большой расход системных ресурсов. Обычный сайт, как правило, содержит больше статичных файлов (изображения, файлы стилей и JavaScript), чем скриптов. Для того чтобы передать эти файлы посетителю, не требуется никакой специальный функционал, задача предельно проста. Следовательно, и обрабатывать такие запросы нужно простым, легковесным веб-сервером, таким, как nginx.

***Internet Information Service*** – Информационные службы Интернета (IIS) вместе с операционной системой Microsoft® Windows Server 2008™ обеспечивают комплексные, надежные, масштабируемые, безопасные и регулируемые возможности веб-сервера при работе с внутренними и внешними сетями, а также с Интернетом. IIS является инструментом для создания мощных коммуникационных платформ динамических сетевых приложений. Различные организации используют IIS для поддержки и управления веб-страниц в Интернете или во внутренней сети, для поддержки и управления FTP-узлами, для маршрутизации новостей и почты, которые используют протоколы NNTP и SMTP. IIS 6.0 поддерживает последние веб-стандарты, такие как Microsoft ASP.NET, XML и протокол SOAP, для разработки, реализации и управления веб-приложениями. IIS 6.0 включает новые возможности, созданные в помощь организациям, техническим специалистам и веб-администраторам для достижения желаемой производительности, надежности, масштабируемости и безопасности тысяч веб-узлов на одном или нескольких серверах IIS.

В таблице 2.2 подведен итог характеристик рассматриваемых Web-серверов и их соответствие указанным критериям. Степень соответствия каждому критерию указанна оценкой в пятибалльной шкале.

*Таблица 2.2 «Характеристики WEB-серверов»*

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **Сервер** | | |
| **Apache** | **ngnix** | **IIS** |
| распространенность | 5 | 4 | 3 |
| надежность | 5 | 5 | 5 |
| гибкость | 5 | 5 | 5 |
| наличие документации и справочной информации | 5 | 5 | 5 |
| Стоимость | бесплатная | бесплатная | бесплатная |

По данным сайта *www.netcraft.com* (определение и сбор статистики по используемым в мире веб-серверам) WEB-сервер **«Apache»** занимаетлидирующие позиции.

Таким образом, в качестве WEB-сервера был выбран сервер «Apache».

Преимущества Web-сервера Apache:

* Бесплатность.
* Многопоточность в UNIX.
* Новая система сборки.
* Поддержка различных протоколов.
* Улучшенная поддержка отличных от UNIX платформ.
* Новый API для Apache.
* Поддержка протокола IPv6.
* Использование фильтров.
* Сообщения об ошибках на разных языках.
* Упрощенная конфигурация.
* Поддержка юникода Windows NT.
* Новая библиотека для работы с регулярными выражениями.
* Безопастность.

## Выбор средств разработки

### PHPStorm– среда разработки и отладки веб-проектов

Перед выбором средств разработки для дипломного проекта были определены следующие требования:

* высокоэффективная среда разработки;
* поддержка UTF-8;
* подсветка кода;
* простота использования;
* отладка PHP - скриптов
* быстрая вставка всех функций PHP с подсказкой параметров;

Исходя из требований, были рассмотрены следующие средств разработки:

* «EditPlus»;
* «PHP Expert Editor»;
* «Macromedia Home Site».

Рассмотрим данные среды для разработки автоматизированной системы немного поближе:

**PHPStorm** – PhpStorm представляет собой интеллектуальный редактор для [PHP](http://ru.wikipedia.org/wiki/PHP), [HTML](http://ru.wikipedia.org/wiki/HTML) и [JavaScript](http://ru.wikipedia.org/wiki/JavaScript" \o "JavaScript) с возможностями анализа кода на лету, предотвращения ошибок в коде и автоматизированными средствами [рефакторинга](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3" \o "Рефакторинг) для PHP и JavaScript. [Автодополнение](http://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%B4%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5" \o "Автодополнение) кода в PhpStorm поддерживает спецификацию PHP 5.3 и 5.4 (современные и традиционные проекты), включая пространства имен, замыкания, типажи и синтаксис коротких массивов. Имеется полноценный [SQL](http://ru.wikipedia.org/wiki/SQL)-редактор с возможностью редактирования полученных результатов запросов.

PhpStorm разработан на основе платформы [IntelliJ IDEA](http://ru.wikipedia.org/wiki/IntelliJ_IDEA" \o "IntelliJ IDEA), написанной на [Java](http://ru.wikipedia.org/wiki/Java" \o "Java). Пользователи могут расширить функциональность среды разработки за счет установки плагинов, разработанных для платформы IntelliJ, или написав собственные плагины.

***EditPlus*** – 32-разрядный текстовый редактор веб-страниц. Редактор имеет множество мощных и удобных возможностей для создателей веб-страниц.

Основными возможностями редактора являются: подсветка синтаксиса для различных типов файлов, панель инструментов HTML, пользовательские панели инструментов, номера строк, линейку, подсветку адресов интернета и электронной почты, авто-завершение, встроенные текстовые клипы, выделение столбцов, мощные поиск и замена и многое другое.

***PHP Expert Editor*** – удобный редактор для PHP, Perl, Python, HTML, Java Script и других файлов с поддержкой UTF-8. Программа разработана специально для PHP мастеров и имеет встроенный отладчик PHP. PHP Expert Editor имеет встроенный HTTP сервер и позволяет запускать PHP, Perl, Python, скрипты. Также можно использовать любой внешний HTTP сервер. Проверка синтаксиса PHP, встроенный браузер, FTP клиент с поддержкой SFTP, Кодовый и файл эксплорер, поддержка проектов, настраиваемые шаблоны кода, настраиваемая подсветка кода, и многие другие функции значительно облегчают работу разработчика.

В таблице 2.3 представлен итог характеристик рассматриваемых средств разработки и их соответствие указанным критериям. Степень соответствия каждому критерию указана оценкой в пятибалльной шкале.

*Таблица 2.3 «Характеристики средств разработки»*

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **Средства разработки** | | |
| PHPStorm | EditPlus | PHP Expert Editor |
| высокоэффективная среда разработки | 5 | 5 | 3 |
| поддержка UTF-8 | 5 | 5 | 1 |
| подсветка кода | 5 | 5 | 5 |
| простота использования | 5 | 5 | 4 |
| отладка PHP - скриптов | 5 | 4 | 3 |
| быстрая вставка всех функций PHP с подсказкой параметров | 5 | 4 | 3 |

Получив данные из таблицы, и проанализировав сильные и слабые стороны вышеперечисленных средств разработки, можно сделать вывод, что для данного дипломного проекта будет использована среда PHPStorm.

Основными возможностями редактора PHPStorm являются: подсветка синтаксиса для различных типов файлов, панель инструментов HTML, пользовательские панели инструментов, номера строк, линейку, подсветку адресов интернета и электронной почты, авто-завершение, встроенные текстовые клипы, выделение столбцов, мощные поиск и замена, в том числе поиск определённой фразы в группе файлов, множественная отмена и восстановление, модуль проверки орфографии (загружается отдельно), настраиваемые горячие клавиши и многое другое.

Также можно выделить следующие особенности программы, которые повлияли на ее выбор в качестве основного средства для разработки системы:

* Программа имеет интуитивно понятный MDI интерфейс;
* Мощная и легко настраиваемая подсветка синтаксиса для различных типов файлов (HTML, CSS, PHP, JavaScript, VBScript и т.д.);
* Встроенный в программу полноценный браузер для предварительного просмотра создаваемых веб-страниц, перехода по ссылкам, и для просмотра локальных файлов;
* Встроенный FTP-клиент;
* Панель HTML позволяет легко вставлять общие теги HTML, а также обрамлять тегами выделенный текст;
* Авто-завершение – экономящая время возможность, которая заменяет короткое сокращение в завершённую строку;
* Окно клипов содержит в себе библиотеки текстовых фрагментов, предназначенные для быстрой вставки в документ.
* Режим автоматического переноса слов позволяет удобно редактировать длинные строки;
* Полноценная поддержка «drag and drop»;

### Язык разметки HTML

**HTML** (от [англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *HyperText Markup Language* — «язык [гипертекстовой](http://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82) разметки») — стандартный [язык разметки](http://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) документов во [Всемирной паутине](http://ru.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0). Большинство [веб-страниц](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0" \o "Веб-страница) содержат описание разметки на языке HTML (или [XHTML](http://ru.wikipedia.org/wiki/XHTML)). Язык HTML интерпретируется [браузерами](http://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) и отображается в виде документа в удобной для человека форме.

Язык HTML является приложением («частным случаем») [SGML](http://ru.wikipedia.org/wiki/SGML) (стандартного обобщённого языка разметки) и соответствует международному стандарту [ISO](http://ru.wikipedia.org/wiki/ISO) 8879.

Язык [XHTML](http://ru.wikipedia.org/wiki/XHTML) является более строгим вариантом HTML, он следует всем ограничениям [XML](http://ru.wikipedia.org/wiki/XML) и, фактически, XHTML можно воспринимать как приложение языка XML к области разметки гипертекста.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам [HTTP](http://ru.wikipedia.org/wiki/HTTP) или [HTTPS](http://ru.wikipedia.org/wiki/HTTPS), в виде простого текста или с использованием [сжатия](http://ru.wikipedia.org/wiki/%D0%A1%D0%B6%D0%B0%D1%82%D0%B8%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

На самом деле, если вы откроете файл с расширением htm (или html) в любом текстовом редакторе, вы увидите, что это не что иное, как ... обычный текст, правда "испорченный" какими-то словами в угловых скобках. Эти "ненужные" слова и представляют собой средства разметки текста, которые понимает программа, работающая с html-файлами(такую программу называют браузер).Похожий принцип построения документов не нов, и встречается ,например, в текстовом процессоре TeX.   
То есть html - это обычный текст (информация) плюс управляющие элементы - тэги (удобное представление этой самой информации).   
Отсюда следует, что для того, чтобы создать гипертекст, нужно всего лишь знать как, куда и в каком количестве "вносить" эти управляющие элементы.

**Тэги** - (от англ. tag) это средство, позволяющее браузеру предоставить имеющуюся у вас информацию в виде, удобном для восприятия, поиска, чтения. Это в своем роде язык программирования, определяющий ,как будет выглядеть на экране следующая строка, что нужно выделить, а что - наоборот - столь очевидно, что нет необходимости это подчеркивать.  
Тэгом считается определенное выражение, помещенное в скобки вида <>. Весьма условно тэги можно разделить на граничные и унарные. Унарные тэги представляют собой очевидные операции, предписывающие браузеру совершить определенное действие по постижению данного тэга при обработке html - документа. Пример - наиболее часто употребляемый тэг <BR> - переход на другую строку, весь следующий за ним гипертекст, браузер будет представлять, начиная с новой строки. Унарные тэги не относятся к определенным участкам текста, а ,скорее всего, к всей последующей части документа. С граничными тегами сложнее, но надо учитывать, что их много больше , нежели унарных. Граничные тэги определяют, как будет выглядеть на экране пользователя часть текста, определенная этими элементами, причем в начальном тэге можно указывать различные параметры, определенные данному тэгу, в конечном тэге эти параметры присутствовать не должны (браузером они игнорируются).

Пример граничного тэга: гиперссылка <a href="http://uic.tula.ru/">Сервер ТулГУ</a>

Здесь

* *a* - управляющее слово, показывающие, что текст вплоть до тега </а> ,будет являться гипертекстовой ссылкой (то есть указнием браузеру просмотреть определенный html - документ при совершении управляющего действия ( шелчка мыши) над "ограниченным" текстом)
* *href="..."* - параметр, определяющий ,какой документ будет просмотрен (это может быть как файл на локальной или удаленной машинах, так и просто IP-адрес удаленной(с условием, что эта машина предоставляет http - сервис))

Некоторые параметры определены по умолчанию браузером, некоторые- пользователем браузера, но есть и такие, определять которые необходимо при создании страницы(пример – тот же параметр href тэга <a>).   
  
Из всего вышесказанного следует, что ,зная наиболее употребляемые тэги и их параметры, любой пользователь легко может редактировать и создавать гипертекст. При этом необходимо учитывать, что тэги могут включать в себя другие тэги, а так же необходимо иметь представление о структуре html - документов.

### Язык создания клиентских сценариев JavaScript

Разработанный компанией Netscape, язык был включён в браузер Netscape Navigator начиная со второй версии и первоначально назывался LiveScript. Синтаксис языка брал начало от языков Си и Java, и, поскольку технология Java была в то время очень модной, LiveScript переименовали в JavaScript, получив соответствующую лицензию у Sun. Компания Microsoft, увидев успех JavaScript, создала свою версию этого языка под названием JScript. Прочие производители браузеров также создали свои версии этого языка, что делает задачу написания сложного универсального (совместимого с любым браузером) скрипта довольно трудной. Для решения проблем совместимости ассоциация ECMA предложила стандарт ECMA-262. По возможностям ECMAScript примерно соответствует JavaScript 1.1.

JavaScript также находит применение в качестве скриптового языка доступа к объектам приложений. Платформа Mozilla (XUL/Gecko) использует JavaScript. Среди сторонних продуктов, например, Java, начиная с версии 6, содержит встроенный интерпретатор JavaScript на базе Rhino. Сценарии JavaScript поддерживаются в таких приложениях Adobe, как Adobe Photoshop, Adobe Dreamweaver или Adobe Illustrator.

JavaScript обладает рядом свойств объектно-ориентированного языка, но благодаря прототипированию поддержка объектов в нём отличается от традиционных ОО языков. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам — функции как объекты первого уровня, объекты как списки, карринг (currying), анонимные функции, замыкания (closures) — что придаёт языку дополнительную гибкость.

JavaScript имеет C-подобный синтаксис, но по сравнению с языком Си имеет следующие коренные отличия:

* объекты, с возможностью интроспекции и динамического изменения типа через механизм прототипов
* функции как объекты первого класса
* обработка исключений
* автоматическое приведение типов
* автоматическая сборка мусора
* анонимные функции

Семантика языка сходна с семантикой языка Self.

### Библиотека jQuery

jQuery – библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API по работе с AJAX.

Основной целью создания jQuery разработчик (Джон Резиг) видел возможность закодировать многоразовые куски кода, которые позволят упростить JavaScript и использовать их так, чтобы не беспокоиться о кросс-браузерных вопросах. Библиотека была представлена общественности на компьютерной конференции «BarCamp» в Нью-Йорке в 2006 году.

Возможности библиотеки:

* Движок кроссбраузерных CSS-селекторов Sizzle, выделившийся в отдельный проект;
* Переход по дереву DOM, включая поддержку XPath как плагина;
* События;
* Визуальные эффекты;
* AJAX-дополнения;
* JavaScript-плагины.

Точно так же, как CSS отделяет визуализацию от структуры HTML, JQuery отделяет поведение от структуры HTML. Например, вместо прямого указания на обработчик события нажатия кнопки управление передаётся JQuery идентифицирующей кнопки и затем преобразовывающей его в обработчик события клика. Такое разделение поведения и структуры также называется «принципом ненавязчивого JavaScript».

Библиотека jQuery содержит функциональность, полезную для максимально широкого круга задач. Тем не менее, разработчиками библиотеки не ставилась задача совмещения в jQuery функций, которые подошли бы всюду, поскольку это привело бы к большому коду, бо́льшая часть которого не востребована. Поэтому была реализована архитектура компактного универсального ядра библиотеки и плагинов. Это позволяет собрать для ресурса именно ту JavaScript-функциональность, которая на нём была бы востребована.

### Язык SQL

SQL (англ. Structured Query Language — язык структурированных запросов) — универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных.

В 1981 году IBM объявила о своём первом основанном на SQL программном продукте — SQL/DS. Чуть позже к ней присоединились Oracle, Relational Technology и другие производители.

Первый стандарт языка SQL был принят ANSI (Американским национальным институтом стандартизации) в 1986 и ISO (Международной организацией по стандартизации) в 1987 (так называемый SQL level 1) и несколько уточнён в 1989 году (SQL level 2). Дальнейшее развитие языка поставщиками СУБД потребовало принятия в 1992 нового расширенного стандарта (ANSI SQL-92 или просто SQL-2). Следующим стандартом стал SQL-99. В настоящее время действует стандарт, принятый в 2003 году (SQL-3). SQL по праву завоевал первенство среди аналогичных языков запросов.

**Преимущества языка:**

* Независимость от конкретной СУБД.
* Наличие стандартов.
* Декларативность.

**Недостатки:**

* Несоответствие реляционной модели данных.
* Отступления от стандартов.

Сложность работы с иерархическими структурами.