|  |  |
| --- | --- |
| Приднестровский государственный университет им. Т.Г. Шевченко  Инженерно-технический институт  Инженерно-технический факультет | |
|  | |
| УДК 681.326 | |
| Кафедра информационных технологий  и автоматизированного управления  производственными процессами | |
|  | |
|  | |
|  | |
|  | |
|  | |
| **Левицкий Егор Андреевич** | |
|  | |
| **РАЗРАБОТКА МЕТОДИКИ ОЦЕНКИ ЭФФЕКТИВНОСТИ РАБОТЫ ИНФОРМАЦИОННОГО ПОРТАЛА** | |
|  | |
| магистерская диссертация  по направлению 552800 «Информатика и вычислительная техника»  магистерская программа 552808 «Информационное и программное обеспечение автоматизированных систем» | |
|  | |
|  | |
|  | |
| Заведующий кафедрой,  к.т.н., доцент | Ю.А. Столяренко |
| Научный руководитель,  к.т.н., доцент | С.Г. Федорченко |
| Магистрант | Е.А. Левицкий |
|  | |
|  | |
|  | |
| Тирасполь, 2014 | |

Аннотация к магистерской диссертации Левицкого Е.А. на тему

«Разработка методики оценки эффективности работы   
информационного портала»

Руководитель: к.т.н, доцент Федорченко С.Г.

Работа содержит 80 страниц, 8 использованных литературных источников, 1 приложение. Работа выполнена на кафедре ИТиАУПП ИТИ.

Настоящая диссертационная работа посвящена разработке методики оценки эффективности работы информационного портала. Проанализирован спектр характеристик, влияющих на эффективность работы сайта. Реализована система автоматизации проведения анализа эффективности работы сайта на примере *web*-сервиса «Информационный портал».

По результатам работы опубликованы три статьи:

1. Левицкий Е.А. Информационный портал // Сб. докл. ИНТК ППС ИТИ ПГУ за 2012 год. — Тирасполь: Тип. ООО “ТесЛайн”, 2013. — С. 65.
2. Левицкий Е.А. Обзор методов и средств анализа статистики сайта // Сб. докл. ИНТК ППС ИТИ ПГУ за 2013 год». — Тирасполь: Тип. ООО “ТесЛайн”, 2014 — С. 158.

Работа прошла апробацию Итоговой научной конференции профессорско-преподавательского состава ИТИ — 2013, на Итоговой научной конференции профессорско-преподавательского состава ИТИ — 2014, на студенческой конференции ИТИ — 2013.

Магистрант Е.А. Левицкий

Руководитель С.Г. Федорченко

### **СОДЕРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
| ВВЕДЕНИЕ | | 4 |
| 1 АНАЛИЗ И ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ | | 5 |
|  | 1.1 Вводные замечания | 5 |
|  | 1.2 Виды задач, решаемые сайтом | 11 |
|  | 1.3 Требования, предъявляемые к сайту | 12 |
|  | 1.4 Технические характеристики сайтов | 15 |
|  | 1.5 Подходы к оценке эффективности работы сайта | 15 |
|  | 1.6 Постановка задачи | 24 |
| 2 РАЗРАБОТКА WEB-РЕСУРСА «ИНФОРМАЦИОННЫЙ ПОРТАЛ» | | 26 |
|  | 2.1 Структура системы «Информационный портал» | 26 |
|  | 2.2 Структура базы данных системы «Информационный портал» | 29 |
|  | 2.3 Структура классов системы «Информационный портал» | 33 |
|  | 2.4 Описание основных принципов обработки запросов | 38 |
|  | 2.5 Выбор методов и средств разработки | 42 |
| 3 РАЗРАБОТКА МЕТОДИКИ ОЦЕНКИ ЭФФЕКТИФНОСТИ  РАБОТЫ САЙТА «ИНФОРМАЦИОННЫЙ ПОРТАЛ» | | 45 |
|  | 3.1 Определение подходящих критериев оценки эффективности | 47 |
|  | 3.2 Разработка методики оценки эффективности | 52 |
| ЗАКЛЮЧЕНИЕ | | 54 |
| ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ  И ТЕРМИНОВ | | 55 |
| СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ | | 56 |
| ПРИЛОЖЕНИЕ — ЛИСТИНГ ИСХОДНЫХ КОДОВ СИСТЕМЫ «ИНФОРМАЦИОННЫЙ ПОРТАЛ» | | 57 |

### **ВВЕДЕНИЕ**

В настоящее время одним из требований, предъявляемых к высшим учебным заведениям, является наличие информационной системы. Данная система должна решать ряд вопросов с точки зрения обучения, организаторских аспектов работы заведения и в научной сфере его жизни.

Множество высших учебных заведений имеют подобного рода системы. Для того, чтобы система в наибольшей степени удовлетворяла запросам заведения, необходимо как можно более четко определить характер того, как система будет решать возлагаемые на нее задачи, более детально представить сами задачи и способы отслеживания хода их выполнения.

Обычно, через некоторое время после создания информационной системы учебное заведение приходит к выводу, что необходимо внести ряд изменений, дабы лучше соответствовать новым реалиям, в которых существует это заведение. Совершенно очевидно, что необходимо предусмотреть такую возможность для информационной системы еще на этапе разработки.

Необходимо выработать методику оценки информационной системы, которая позволит определить уровень эффективности работы системы и позволит помочь в управлении такой информационной системой.

Возникает задача в разработке соответствующей методики оценки эффективности системы, чему и посвящена данная работа.

### **1 АНАЛИЗ И ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

**1.1 Вводные замечания**

Информационные системы, обычно, строятся в виде клиент-серверных приложений. Поэтому рассмотрим особенности структурного устройства клиент-серверных приложений и выделим основные отличия между ними.

Клиент-серверные приложения могут быть классифицированы по подходу к хранению информации и по подходу к доступу к этой информации.

Существует несколько подходов хранения информации:

1. централизованный подход;
2. децентрализованный подход;
3. распределенный подход.

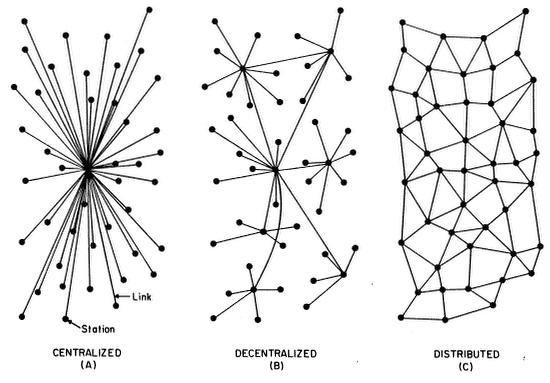


Рисунок 1.1 — Централизованный (A), децентрализованный (B), распределенный (C)

Централизованный подход хранения информации предполагает тот факт, что абсолютно все данные хранятся на главном сервере, а клиентские приложения могут получить доступ различного уровня (чтение, запись) к его информационным ресурсам.

Такой подход удобен в том случае, когда имеется четко выделенный центр, на которого возлагается вся емкостная нагрузка (размер дискового пространства), нагрузка на мощность (вычислительная мощь), нагрузка на трафик (пропускная способность первичной к серверу сети). В таком случае реализация взаимодействия достаточно проста: каждый клиент подключается к серверу, а сервер в свою очередь создает виртуальное пространство для каждого подключенного клиента.

При использовании данного подхода, сервер должен быть достаточно мощным, чтобы быть способным обрабатывать параллельно запросы определенного числа клиентов, к тому же выход из строя сервера означает полное прекращение работы сервиса. Визуальное представление данного подхода можно видеть на рисунке 1.1(A).

Децентрализованный подход предполагает выделение региональных центров соединенных посредством общего центра. В этом случае отказ главного центра повлечет за собой прекращение взаимодействий между региональными центрами, однако остается полная работоспособность внутри регионов. В соответствии с этим снижается нагрузка с главного сервера и перекладывается на региональные. Вообще главный центр может являться лишь маршрутизатором для региональных центров, которые в свою очередь хранят необходимую информацию, причем они могут дублировать данные друг друга, либо хранить независимую отдельную информацию.

Хранение независимой информации в каждом региональном центре подразумевает разбиение общих данных на классы (группы), каждый такой класс данных хранится на отдельном сервере. Совместное хранение общей информации с дублированием предполагает использование региональных серверов в качестве повышения отказоустойчивости.

Обычно, в реальных ситуациях распространенным является смешанный способ хранения данных, когда каждый сервер содержит информацию по определенному классу тематик и содержит некоторую наиболее часто используемую информацию других серверов в качестве кэша, что позволяет снизить нагрузку на сеть и в некоторой степени повысить отказоустойчивость. Данный подход продемонстрирован на рисунке 1.1(B).

Распределенный подход основывается на исключении центров как таковых. В данном случае каждый клиент в свою очередь является и сервером. Данные могут быть уникальными в пределах одного сервера-клиента или же быть сдублированными в большем или меньшем количестве. Это позволяет обеспечить одну из наивысших степеней отказоустойчивости, ведь выход из строя всей системы сводится к минимуму с увеличением количества участников.

Такой подход влечет за собой повышения требований к мощности каждого узла и чрезмерному количеству дубликатов информации. Подобный подход используется в так называемых *torrent* технологиях.

С другой стороны существует несколько основных концепций предоставления доступа к информации:

1. концепция тонкого клиента;
2. концепция толстого клиента;
3. концепция смешанного подхода.

Концепция тонкого клиента подразумевает оснащение клиента элементарными возможностями, такими как передача и прием данных, формирование запроса, распознавание ответа. В таком случае большая часть передаваемых данных по сети содержит описание структуры целевой информации, что требует повышения требований, предъявляемых к самой сети, повышаются требования к серверу.

При использовании концепции тонкого клиента, внесение обновлений является весьма простым делом, достаточно обновить серверную часть системы и все изменения вступят в силу при следующем обращении к нему, что в свою очередь, позволяет добавлять новых клиентов без каких, либо существенных дополнительных затрат.

Основным примером такой концепции являютсятерминальный доступ. Данная концепция продемонстрирована на рисунке 1.2.

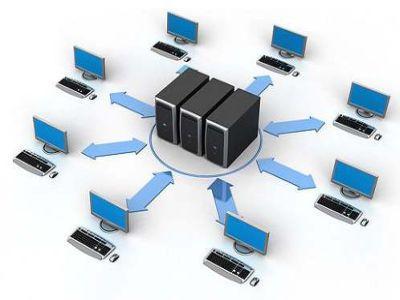


Рисунок 1.2 — Тонкий клиент

Концепция толстого клиента заключается в том, чтобы снизить до минимума количество данных, передаваемых по сети, за счет предварительной рационализации и разбиении передаваемой информации на классы или типы. Достигается это за счет увеличения функциональных возможностей клиентов.

В результате, каждый клиентский узел является почти полностью автономным и требует лишь относительно незначительного потребления трафика. Совершенно очевидно, что подобный выигрыш в снижении требований к сети и серверу достигается за счет значительного усложнения клиентской части приложения, что в сою очередь способствует усложнению процедуры внесения изменений в общую систему и процедуры добавления нового клиента, не говоря уже о значительных объемах информации, передаваемой при каждом обновлении.

Из выше написанного следует, что такой подход следует использовать в том случае, когда частота обращений между клиентами и сервером настолько высока, что необходимо снижать потребляемый трафик в каждом запросе, для того чтобы не превышать пропускную способность используемой сети или значительно уменьшить время получения ответа на запрос.

Простой пример толстого клиента продемонстрирован на рисунке 1.3.

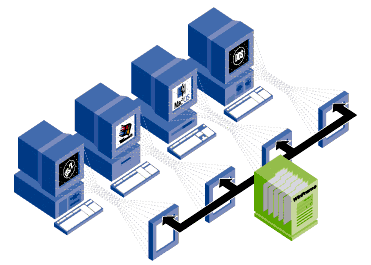


Рисунок 1.3— Толстый клиент

Концепция смешанного подхода предполагает некоторый баланс между той функциональностью, которой должен обладать клиент, и требованиями к пропускной способности сети, по которой передаются данные. В данном случае сохраняется некоторая сложность внесения структурных изменений в организацию работы клиентской части.

Для организации общей схемы связи клиента с конечными данными существует следующие основные подходы:

1. прямой доступ к данным;
2. доступ данных посредствам сервера;
3. доступ к данным посредством разделенного сервера.

Следует учитывать, что пользователи могут использовать совершенно различные типы устройств для обращению к серверу. К многообразию типов устройств добавляется и достаточно существенное многообразие операционных систем, которые могут управлять устройствами.

Общая схема доступа к серверу предоставлена на рисунке 1.4.

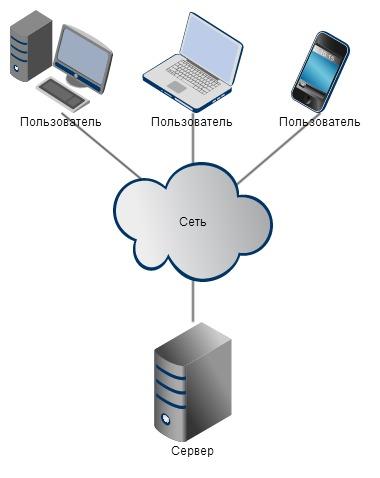


Рисунок 1.4 — Общая схема доступа к серверу

Приведенное здесь разнообразие устройств пользователей является далеко не полным. Но оно все же показывает необходимость использования достаточно универсального способа предоставления пользователям необходимого контента. Разрабатывать отдельный клиент для каждого типа устройств будет чрезмерно затратным, не говоря уже о том что поддержка и организация обновления таких клиентов будет весьма сложным процессом для разработчика.

На данный момент одним из наиболее универсальных является *web* технология на основе *http* протокола, который не зависит от типа устройства и применяется для навигации в сети *Internet*. Одной из особенностей данного подхода является полная передача контента от сервера к пользователю, как данных, так и управляющего кода.

Чаще всего, современные информационные системы строятся именно на базе этой технологии, позволяя делать систему масштабируемой, модульной.

### **1.2 Виды задач, решаемые сайтом**

Существует достаточно существенное множество категорий сайтов, распространенных на просторах всемирной сети: визитные карточки юридических и физических лиц, интернет магазины, новостные ленты, блоги, аудио порталы, видео порталы, игровые порталы, социальные сети и многие, многие другие.

Все многообразие сайтов, на наш взгляд, можно свести к трем группам:

1. Сайты, которые предоставляют информацию о коммерческом объекте. Так называемые финансово ориентированные ресурсы.
2. Сайты, предназначенные для предоставления информации по определенной тематике или группе тематик.
3. Гибридные сайты, представляющие собой смеси в различных пропорциях первых двух групп.

Перед первой группой стоит задача продвинуть «товар» в глазах потребителя, а в лучшем случае продать этот «товар». Товаром может выступать как материальный продукт, так и нематериальный (например продукт интеллектуального труда).

Перед второй группой стоит несколько иная задача, а именно — как можно больше распространить некую информацию. Здесь главным фактором является удобство использование ресурса, то есть его интерфейс взаимодействия с пользователем, насколько быстро пользователь может найти нужную ему информацию и разобраться в данном вопросе. Выходным показателем будет являться отношение полученного знания к потраченному времени, на приобретение этого знания.

Зачастую сайт должен обеспечивать получение некоторой суммы денег, хотя бы для оплаты аренды сервера и домена сайта. В таком случае на сайте размещается реклама, приносящая достаточный для этого доход. Вот типичный пример гибридного сайта. Другим примером гибридного сайта может служить сайт-социальная сеть, которая предоставляет пользователям возможность обмениваться информацией и в тоже время собирать общедоступную информацию о пользователях в маркетинговых целях.

**1.3 Требования, предъявляемые к сайту**

Для решения поставленных задач сайт должен соответствовать определенным требованиям, которые можно разбить на следующие группы:

1. технические требования;
2. информационные требования;
3. эргономические требования.

Рассмотрим данные требования в контексте сайта, предназначенного для предоставления информации по определенной тематике или группе тематик (2 группа сайтов).

Технические требования к такому сайту будут следующими. Сайт должен быть работоспособным, отказоустойчивым, обладать переносимостью, иметь определенную структуру, обладать гибкостью, обладать возможностью модификации, должны быть реализованы средства достаточно простой административной работы с сайтом.

Приветствуется реализация возможности общения между участниками непосредственно через сайт, а так же реализация предоставления возможности пользователям выкладывать свою информацию на сайте, которая будет непосредственно или косвенно касаться тематики ресурса. Благодаря такой возможности в значительной мере снижается нагрузка на администратора сайта с той стороны, что наполнение ресурса информации происходит без его участия, однако с другой стороны, администратору необходимо следить за вносимой пользователями информацией, чтобы она не нарушала установленных правил и цензуры.

Информационные требования для подобного сайта (2-я группа сайтов) могут быть следующими. Сайт должен содержать достаточно полные или законченные блоки информации таким образом, чтобы пользователи, на которых данный ресурс ориентирован, могли понимать и воспринимать данные за удовлетворительный промежуток времени. Информация должна быть актуальной для целевых пользователей, и предоставляться информация должна в соответствии с привычками этих пользователей.

Эргономические требования — интерфейс сайта должен быть удобен в использовании целевым пользователям, быть запоминающимся, приятным и соответствовать основным положениям по относительному расположению и цветовому сопровождению таких элементов, как ссылки, картинки, текстовые блоки.

В связи с перечисленным ранее получается, что критически важным является следующее:

1. определения целевой аудитории;
2. разбиение этой аудитории на типовые группы;
3. определить степень ценности каждой группы;
4. определить относительную или абсолютную численность каждой из типовых групп.

Можно выделить несколько основных классов пользователей:

1. гости;
2. опытные пользователи;
3. эксперты;
4. администраторы;
5. разработчики.

Каждый следующий класс является дополнением (наслоением) над предыдущим и увеличивает уровень доступа к ресурсу.

Гости являются начальным классом, которому доступны лишь открытые сервисы сайта, такие как просмотр новостей, открытых статей, информации о различных разделах и получения ссылок на файлы, которые доступны в открытом виде. Количество пользователей данного класса будет наиболее распространенным. Их целью является получение информации наиболее быстрым путем без необходимости авторизации.

Опытные пользователи в отличие от гостей получают возможность получить доступ к некоторому закрытому контенту, оставлять свои комментарии. Для получения доступа к закрытому контенту пользователю необходимо пройти процесс авторизации. Количество пользователей данного класса будет существенно меньше, нежели гостевых пользователей, главным образом по причине необходимости авторизации.

Эксперты — опытные пользователи, которым предоставляется возможность участвовать в формировании контента сайта. Им также необходимо проходить процесс авторизации. Количество пользователей данного класса будет небольшим. Они должны обладать достаточным уровнем ответственности и доверием со стороны обладателя ресурса.

Администраторы — пользователи, задачей которых является поддержание сайта в единообразном состоянии, исправлять различные мелкие ошибки, такие как опечатки, нарушение в визуальном оформление текстовой информации, отклонять возможность просмотра информации недостаточно подготовленной к публикации. Количество пользователей данного класса будет ограниченно малым.

Разработчики — особый класс пользователей, которые не совсем вписываются в само определение — пользователь. Разработчики не вносят информацию как таковую, они формируют условия для этого. Главным отличием от остальных пользователей является возможность прямого полного доступа к программной части сайта.

**1.4 Технические характеристики сайтов**

Технические характеристики сайта определяются требованиями, предъявляемые к нему. Все сайты характеризуются количественными и качественными параметрами. В зависимости от целей, которые необходимо решать при помощи сайта коэффициенты важности параметров могут меняться в большую или меньшую сторону. Некоторые параметры являются крайне важными при решении всех целей, они определяются необходимости поддержания собственно работы сайта с минимальной мерой надежности.

Количественные:

1. низкая стоимость эксплуатации и обслуживания;
2. достаточная нагрузоспособность сайта;
3. объем публикуемой информации.

Качественные:

1. эргономика;
2. информационная безопасность;
3. надежность и отказоустойчивость работы;
4. независимость от платформы (операционной системы);
5. бесконфликтная смена дизайна с сохранением накопленной информации;
6. возможность накапливать информацию о посещении сайта и поведение во время посещений;
7. требования к квалификации пользователей;
8. богатство и удобство функционала для управления данными;

**1.5 Подходы к оценке эффективности работы сайта**

Разумеется одним из важнейших аспектов является получение обратной связи. Для этого существуют различные метрики — множество методик, преобразующих простые количественные характеристики в выходной показатель того, насколько хорошо сайт выполняет поставленные перед ним задачи. Рассмотрим классификацию имеющихся метрик:

1. бизнес метрики;
2. метрики *usability;*
3. *SEO* метрики;
4. *SMO* метрики.

Бизнес метрики позволяют определить насколько эффективен сайт с точки зрения продаж или стенда рекламных объявлений. Данные метрики вбирают в себя следующие показатели:

* *ROI* (англ. *return on investment* — возврат инвестиций);
* продажи;
* обращения (контакты, регистрации);
* конверсии;
* подписчики.

Если сайт продвигает конкретный набор товаров, то совершенно очевидно что главным показателем эффективности будет являться процент прироста продаж этого товара благодаря сайту. Зачастую используют систему интернет-магазина, в котором пользователь может заказать товар. В результате можно определить как пользователь выбирал товар на сайте, как он его заказывал и что препятствует пользователю на пути к совершению покупки.

Например, полный путь оформления заказа состоит из следующих основных шагов:

1. открытие страницы сайта;
2. выбор товара из каталога, либо поиск интересующего товара;
3. перемещение товара в корзину заказать;
4. оформление формы заказа (как оплатить, куда доставить и т.д.).

На каждом из этих шагов у пользователя могут возникать трудности или психологические конфликты, препятствующие достижению конечной цели — оформление и отправка заказа продавцу, например, пользователю не понравилось, что страница сайта загружалась чрезмерно долго, и он закрыл сайт. При этом на каждом шаге можно определить соотношение успешных переходов и отказов и увидеть какой шаг является «ахиллесовой пятой» во всем маршруте следования пользователя от начала и до конца.

Если заработок на сайте осуществляется за счет размещения рекламы, то в таком случае критически важными будут показатели, зависящие от условий размещения рекламы:

* реклама, оплачиваемая за время показа, требует, чтобы за единицу времени количество уникальных пользователей было выше определенного порога;
* реклама, оплачиваемая за переходы по ссылке требует, чтобы пользователи переходили по рекламной ссылке на сайт заказчика этой рекламы.

Метрики *usability* [5, с. 54] — это показатели удобства использования ресурса. К данным метрикам следует относить следующие показатели:

* время отклика при открытии страницы;
* среднее число просмотров на посетителя;
* время, проведенное на странице;
* время, проведенное на сайте;
* показатель отказов;
* показатель прерываний визита в корзине/форме;
* следующие страницы;
* карты кликов (тепловые карты);
* технология «*eye tracking*» слежения за взглядом пользователя;
* внутренние поиски.

Очевидно, что возвратившимся посетителям понравился указанный сайт. Так что чем больше вернувшихся посетителей, тем более эффективен и успешен ресурс. Одни только посетители из поисковых систем или с социальных ресурсов не могут являться основой, хребтом посещаемости.

Подписчики и возвращающиеся посетители (часто это одни и те же люди) должны составлять существенную аудиторию сайта.

Хотя измерение просмотров иногда бесполезно, так как плохие сайты, где сама структура сайта вынуждает пользователей кликать больше, могут иметь высокое среднее число просмотров на посетителя, тем не менее этот показатель расскажет многое о том, насколько указанный сайт нравится пользователям. Соотношение один просмотр на одного посетителя – это всегда плохо для сайта, за исключением если этот единственный просмотр не заканчивается переходом в корзину.

Время, затрачиваемое и проводимое пользователем на странице, можно истолковывать множеством способов. Но следует четко выделять пользователей, которые лишь бегло просматривают ваш контент, и людей, которые целиком читают статью.

Не всегда длительное время нахождения пользователя на сайте это хорошо, но в большинстве случаев длительность пребывания в 5 минут лучше, чем в 30 секунд, особенно, для контентных проектов или блогов.

Показатель отказов (отказы) — доля визитов, в рамках которых состоялся лишь один просмотр страницы. Данный показатель является одним из самых важных метрик *usability* сайта. Сто тысяч посетителей с показателем отказов в 95% означает, что на самом деле, только пять тысяч пользователей действительно посетили ваш сайт. Так что сайт с гораздо меньшим числом посетителей и хорошим показателем отказов может быть гораздо более эффективным, чем ресурс с огромным, но бестолковым количеством трафика. Целевой трафик является ключевым аспектом эффективности сайта.

Показатель прерываний визита на странице с формой – доля посетителей, перешедших в на страницу с формой, но не отправивших запрос с заполненной формой. С точки зрения бизнеса страницы с формами — это наиболее важные части большинства *web*-сайтов, будь то контактные формы или корзины, которые в большинстве случаев технически представляют собой то же самое.

Например, супермаркет, где половина или больше клиентов отказываются от покупки в середине процесса подсчета ее стоимости, на кассе или после внимательного изучения товара. Необходимо подсчитать этих людей и попробовать «заставить» их остаться.

Простейший путь проверки показателя прерываний в корзине – письмо в службу поддержки каждый раз, когда пользователь, перешедший в корзину, не подтверждает заказ. Иногда, можно заполучить потенциального клиента с неполными данными, которые он ввел.

Для посещения пользователями более одной страницы на сайте существует внутренние ссылки. Некоторые внутренние ссылки – это ссылки, расположенные специальным образом, чтобы пользователи переходили с одной страницы на другую, осуществляя таким образом навигацию по сайту. Проверка «следующих страниц» от конкретной входной страницы позволит определить следуют ли читатели нашему совету по переходу или ожидают чего-то большего. Если для главной страницы вашего сайта «следующей страницей» в большинстве случаев является поиск или карта сайта, то это означает, что пользователь испытывает определенные проблемы при навигации по сайту.

Современные инструменты *web*-аналитики в духе *Web* 2.0 иногда предлагают тепловые карты просмотров страниц сайта или тепловые карты кликов. Таким образом, имеется возможность определить, куда посетители указанного сайта кликают или пытаются кликать (иногда безуспешно в случаях, когда с логотипов или подчеркнутых слов в тексте нет ссылок).

Тепловая карта отслеживания движения взгляда по страницам сайта даже лучше тепловой карты кликов. Для этого понадобится больше чем просто стандартный пакет *web* аналитики. Будет необходимо привлечь к участию в исследовании реальных людей, «похожих» на ваших клиентов. Все это не так просто, но если у вас крупная компания, зависящая от *web* сайта, исследовать его с помощью технологии слежения за взглядом нужно обязательно.

С помощью анализа внутренних поисков на сайте можно определить действительно ли до посетителей указанного сайта не доходит, где искать нужную информацию на сайте (и они идут в поиск), или дизайн страниц просто недостаточно хорошо нацеливает их.

*SEO* метрики [1, с. 219; 7] состоят из следующих показателей:

* количество обратных ссылок;
* качество обратных ссылок;
* дата кэша;
* частота посещений роботов поисковых систем;
* последняя дата посещения роботом поисковой системы;
* количество проиндексированных страниц;
* показатель передачи *page rank*.

Эксперты в *SEO* любят измерять. Данная метрика зародилась на зоре развития интернет бизнеса, когда высокие значения таких показателей как «*Page Rank*», позиции в выдаче и трафик давали серьезное преимущество в борьбе с конкурентами. Но «старая школа» *SEO* до сих пор актуальна и во многих случаях, особенно, когда дело касается обратных ссылок, которые до сих пор определяют успех в ранжировании в выдаче некоторых распространенных поисковых систем. Обратные ссылки следует проверять специальными методами через определенные бесплатные *online* сервисы.

Необходимо отслеживать, сколько людей или, вернее сказать, какое количество страниц ссылается на ваш сайт. Хотя число ссылающихся страниц не столь важно, поскольку ссылки с одного домена для некоторых поисковых систем учитываются, как одна. Так что необходимо сосредоточиться больше на числе ссылающихся сайтов (так называемой популярности домена).

Множество «низкосортных» ссылок на ваш сайт может не иметь практически никакого значения для поисковых систем по сравнению с одной ссылкой с известного и популярного ресурса. Так что нужно определять качество ссылок.

Многие *SEO* специалисты прибегают к проверке даты кэша в поисковых системах (некоторые поисковые системы сохраняют копии большинства web страниц в «кэш») для определения качества сайта и отношения к нему поисковых систем. Если дата кэша старше одного месяца, то сайт либо мертвый (нет свежего контента) либо имеет низкую авторитетность в поисковой системе. Конечно, всегда следует проверять, есть ли вообще у сайта кэш.

Возраст кэша может быть равен одной неделе, но если индексный робот поисковой системы посещает сайт ежедневно, то значит с ним все нормально. Частоту посещений можно отслеживать с помощью аналитических инструментов установленных на сервере.

Последняя дата посещения роботом поисковой системы — этот почти тоже самое, что описанное выше, но только почти. Если вы разместили новую страницу с контентом на сайте, робот поисковой системы заходил на нее вчера, а она до сих пор еще не в индексе этой поисковой системы, то это не очень хорошо. Возможно, она не попадает в индекс в связи с проблемой дублирования контента.

Чем больше проиндексированных страниц, тем лучше. Если на сайте 50 страниц, из которых только 20 проиндексированы, то ваш сайт плохо индексируется. Нужно разбираться почему.

Хотя, по мнению некоторых экспертов, на текущий момент ориентироваться на фактический «*Page Rank*» уже не имеет особого смысла, тем не менее, следует ознакомиться с показателем передачи «*Page Rank*».

К *SMO* метрикам можно отнести следующий набор показателей [8]:

* социальные закладки;
* число упоминаний в социальных ресурсах;
* упоминания на нишевых социальных сайтах;
* число упоминаний в блогосфере;
* количество ссылок по данным поисковых систем по блогам.

В эпоху расцвета социальных ресурсов, пользовательского контента для успешности сайта недопустимо целиком и полностью ориентироваться только на роботов и другие агрегаторы, собирающие данные в автоматическом режиме. Необходимо выяснить, что пользователям нравится, и как они отзываются об указанном сайте или, по крайней мере, как часто.

Сайт или его страница, имеющие сотни или тысячи закладок на страницах специальных ресурсах-закладках нельзя назвать мало успешным. И, если сайт имеет сотни или тысячи социальных закладок на большем количестве популярных сайтах-закладок, то это еще больше означает, что сайт действительно хороший и полезный.

Получение трафика в десятки тысяч пользователей не является беспорным и решающим доказательством популярности сайта. Трафик всего лишь доказывает, что вы находитесь в основной струе, действуете стратегически правильно. Тем не менее число упомнинай на социальных ресурсах позволяет сделать более точный вывод о том, что ресурс хороший, если, конечно, заказчик намеренно не используете услуги по представлению в социальных медиа. Если это не так, то легко можно заметить, что страницы с популярными и хорошими материалами упоминаются почти на всех социальных ресурсах.

Упоминание или рекомендации в социальных сетях – это, действительно, успех, поскольку здесь люди общаются со своими коллегами и почитателями и использует только те ссылки, которым действительно доверяют.

Новостные сайты можно разложить по определенным нишам. Каждая ниша имеет свой популярный сайт социальных новостей, представляющий ее, будь то ниши для поднятия начинающих проектов или хакеров, *web* дизайна или или *web* разработки. На подобных ресурсах можно получить пользователей и читателей, которых реально заботит ваша тематика и их мнение, действительно, важно.

Часто упоминающиеся в социальных сетях страницы несомненно популярны в блогосфере. Сайт является частью обсуждения, если на него часто ссылаются другие блоги. Причем, чем более известен, престижен и пользуется успехом ссылающийся блог, тем лучше.

Основной поиск во многих поисковых системах не показывает большинство ссылок на указанный сайт, в то время как поиск по блогам гораздо более полезен в этом плане. Используя его, можно обнаружить ссылки с других блогов на ваш сайт.

Для подсчета качественных и количественных характеристик существует множество бесплатных сервисов, таких как Яндекс Метрика [7] или *Google Analytics* [6], которые скрывают от пользователя данных сервисов всю сложность сбора и обработки статистических данных и позволяют сосредоточится на анализе результатов их расчетов, предоставляемых пользователю как в числовом, так и в графическом виде (графики, диаграммы и так далее).

Однако, для локального использования, набор средств для сбора статистики является куда менее богатым. В результате подобная задача не имеет готового решения, которое можно было бы получить бесплатно, поэтому необходимо в некоем роде «изобретать велосипед».

Исходя из все сказанного о показателях эффективности становиться совершенно ясно, что применение тех или иных метрик становиться оправданным и актуальным в зависимости от целей, которые должен достигать *web*-ресурс. Так, например, узкоспециализированный сайт, направленный на обеспечение общение и получение информации среди людей определенной профессии, не имеет смысла тратить ресурсы на продвижение и отслеживание данного сайта в рейтинге среди остальных сайтов, предоставляющих краткую информацию на обширное количество тематик. И действительно, ведь достаточно небольшая группа людей, объединенная общей профессиональной деятельностью чаще всего общается по соответствующей тематики в своем узком кругу.

С другой стороны, если сайт является площадкой для размещения статей с совершенно различными тематическими направлениями, от бытовых, до развлекательных и научно-популярных, то привлечение новых авторов, будет одной из основных задач подобного *web*-ресурса, потому что именно авторы будут наполнять его столь нужной и полезной информацией. Для привлечения новых автором хорошо подходит реклама на распространенных медиа ресурсах и социальных сетях.

**1.6 Постановка задачи**

Необходимо разработать специальную методику оценки эффективности информационной системы, а также произвести апробацию данной методики на примере информационного портала Инженерно-технического института Приднестровского Государственного Университета имени Т. Г. Шевченко Приднестровской Молдавской Республики.

Разрабатываемый нами автоматизированный информационный ресурс в виде web-сайта, должен позволить решить следующие задачи:

1. Предоставлять доступ учащимся инженерно-технического института к просмотру информации, необходимой для улучшенного восприятия материала предоставляемого преподавателями.
2. Предоставлять возможность преподавателям инженерно-технического института публиковать информационный материал.
3. Предоставлять возможность публиковать информацию, необходимую для улучшенной организации учебного процесса, такую как расписание звонков, расписание занятий, важные новости и объявления, информация о кафедрах, преподавателях, учебных дисциплин.
4. Обеспечивать разделение доступа к открытой и закрытой частям контента сайта, таким образом, чтобы пользователи не прошедшие процесс авторизации не могли получать закрытые данные.
5. Обладать системой авторизации пользователей на сайте для получения доступа соответствующего уровня.
6. Обладать системой администрирования и управления сайтом.
7. Обладать системой по сбору информации о том, как часто пользователи обращаются к тем или иным разделам сайта, как ведут себя пользователи при просмотре различных страниц сайта.
8. Вычислять показатели эффективности работы информационного портала по требованию администратора.
9. На основании собранной информации о действии пользователей на сайте иметь возможность последующего анализа подобной информации.

Одним из основных ограничений является условие физического расположение сервера ресурса в пределах Инженерно-технического института с подключением этого сервера в общую сеть учреждения.

Ресурс должен достаточно свободно справляться с нагрузкой соответствующей числу одновременных обращений, которое может достигать максимального количества учащихся, имеющим одновременный доступ к сети, с утроенной надежностью.

**2 РАЗРАБОТКА WEB-РЕСУРСА   
«ИНФОРМАЦИОННЫЙ ПОРТАЛ»**

**2.1 Структура системы «Информационный портал»**

Из постановки задач можно сделать вывод, что *web*-ресурс «Информационный портал» должен предоставлять различного рода информацию, относящуюся к учебному процессу и другой деятельности Инженерно-технического института, с разделением уровня доступа к различным этой информации.

Ресурс должен быть доступен в пределах Инженерно-технического института, и сервер необходимо расположить в одной из специализированных аудиторий Инженерно-технического института. При этом на сайт не возлагаются требования по привлечению финансовой прибыли. Таким образом сайт можно отнести ко второй группе, выбирая среди групп сайтов, рассмотренных в разделе 1.2 — «Виды задач, решаемые сайтом».

Для выполнения поставленных задач необходимо разработать структуру системы, определить платформу, на которой будет основываться данная система, и подобрать подходящие инструменты реализации проекта.

Общая структура системы должна быть разбита на составные части — модули. Данные модули можно условно разбить на 2 группы:

1. базовые модули, которые будут являться фундаментом или строительными блоками для модулей следующей группы;
2. композиционные модули, которые состоят из более мелких блоков и также дополняют их функционал своими личными свойствами.

С другой стороны модули можно сгруппировать по семантическим признакам, то есть по тому, за какой слой системы они отвечают:

1. модули для работы с базой данных;
2. модули для работы с пользовательским интерфейсом;
3. модули для обработки внутренней логики.

Такое разбиение называется *MVC* (*Model-View-Controller*) — схема использования нескольких шаблонов проектирования, с помощью которых модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области. Схема предоставлена на рисунке 2.1.

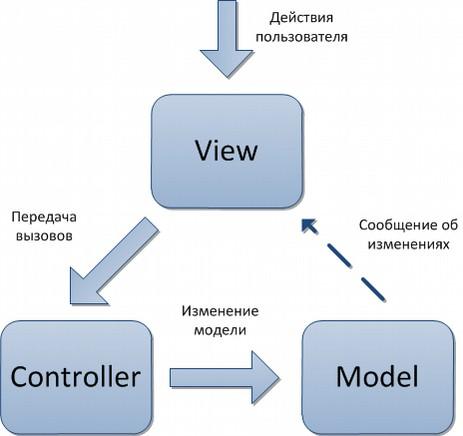


Рисунок 2.1 — Схема *MVC*

Основная идея заключается в том, что и контроллер и представление зависят от модели, но модель никак не зависит от этих двух компонент. Это как раз и позволяет разрабатывать и тестировать модель, ничего не зная о представлении и контроллерах. В идеале контроллер так же ничего не должен знать о представлении (хотя на практике это не всегда так), и в идеале для одного представления можно переключать контроллеры, а так же один и тот же контроллер можно использовать для разных представлений (так, например, контроллер может зависеть от пользователя, который вошел в систему).

Пользователь видит представление, на нем же производит какие-то действия, эти действия представление перенаправляет контроллеру и подписывается на изменение данных модели, контроллер в свою очередь производит определенные действия над моделью данных, представление получает последнее состояние модели и отображает ее пользователю.

Для поддержания качественной реализации *MVC* рекомендуется использовать концепцию *ORM* при организации работы с базой данных.

*ORM* (*Object-relational mapping*) — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных». Схема данной технологии предоставлена на рисунке 2.2.

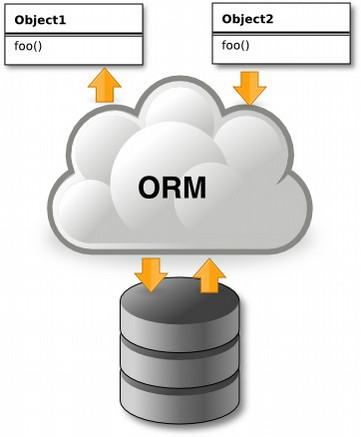


Рисунок 2.2 — Схема ORM

Данный подход в период реализации позволяет решить проблему довольно частого изменения именования таблиц базы данных, изменением полей в этих таблицах, связей между таблицами.

В дальнейшем этот подход позволит решить многие проблемы изменения базы данных в процессе дополнения системы новыми компонентами и сервисами в будущем.

**2.2 Структура базы данных системы «Информационный портал»**

Структура базы данных является одной из важнейших составляющих системы, от нее в существенной мере зависит возможность масштабирования и быстродействие при поиске необходимой информации.

На этапе проектирования база данных представляется на языке инфологического уровня, в котором таблицы представлены в виде сущностей с набором атрибутов. Между сущностями могут быть организованы специальные связи, которые представляют собой бинарные ассоциации, показывающие, каким образом сущности соотносятся или взаимодействуют между собой. Связи подразделяют на следующие типы:

1. один к одному;
2. один ко многим;
3. многие ко многим.

Основная структура базы данных проекта «Информационный портал» должна состоять из следующих элементов — сущностей:

* Сущность «Группа студентов». Ей принадлежит следующий набор атрибутов: идентификатор, наименование, код, описание.
* Сущность «Дисциплина». Ей принадлежит следующий набор атрибутов: идентификатор, сокращенное наименование, наименование, описание.
* Сущность «Направление». Ей принадлежит следующий набор атрибутов: идентификатор, сокращенное наименование, наименование, кафедра, вид обучения, , краткое описание, подробное форматированное описание.
* Сущность «Кафедра». Ей принадлежит следующий набор атрибутов: идентификатор, сокращенное наименование, наименование, факультет, краткое описание, подробное форматированное описание.
* Сущность «Факультет». Ей принадлежит следующий набор атрибутов: идентификатор, сокращенное наименование, наименование, описание, университет.
* Сущность «Университет». Ей принадлежит следующий набор атрибутов: идентификатор, сокращенное наименование, наименование, описание.
* Сущность «Пользователь». Ей принадлежит следующий набор атрибутов: идентификатор, пользовательский псевдоним, пароль, фамилия, имя, электронный адрес, дата последнего входа, флаг активации.
* Сущность «Права доступа». Ей принадлежит следующий набор атрибутов: идентификатор, кодовое название, название, объект доступа.
* Сущность «Группа пользователей». Ей принадлежит следующий набор атрибутов: идентификатор, наименование
* Сущность «Пользовательская статья». Ей принадлежит следующий набор атрибутов: идентификатор, название, форматированное содержимое, автор, дата создания, требуемый уровень доступа.
* Сущность «Комментарий». Ей принадлежит следующий набор атрибутов: идентификатор, содержимое, пользовательская статья, автор, дата создания.
* Сущность «Файл». Ей принадлежит следующий набор атрибутов: идентификатор, ссылка на расположение, наименование, дата загрузки на сервер, владелец файла, требуемый уровень доступа.
* Сущность «Тег». Ей принадлежит следующий набор атрибутов: идентификатор, наименование.
* Сущность «Новость». Ей принадлежит следующий набор атрибутов: идентификатор, заголовок, содержимое, дата создания, дата окончания актуальности, уровень важности.
* Сущность «Расписание звонков». Ей принадлежит следующий набор атрибутов: идентификатор, номер пары, время начала, время окончания.

Далее следует указать необходимые связи между перечисленными сущностями.

Между сущностью «Университет» и сущностью «Факультет» нужно определить тип связи один ко многим.

Между сущностью «Факультет» и сущностью «Кафедра» нужно определить тип связи один ко многим.

Между сущностью «Кафедра» и сущностью «Направление» нужно определить тип связи один ко многим.

Между сущностью «Направление» и сущностью «Дисциплина» нужно определить тип связи м ко многим.

Между сущностью «Пользователь» и сущностью «Группа пользователей» нужно определить тип связи многие ко многим.

Между сущностью «Группа» и сущностью «Права доступа» нужно определить тип связи многие ко многим.

Между сущностью «Пользователь» и сущностью «Файл» нужно определить тип связи один ко многим.

Между сущностью «Пользователь» и сущностью «Пользовательская статья» нужно определить тип связи один ко многим.

Между сущностью «Пользователь» и сущностью «Комментарий» нужно определить тип связи один ко многим.

Между сущностью «Пользовательская статья» и сущностью «Комментарий» нужно определить тип связи один ко многим.

Между сущностью «Тег» и сущностью «Пользовательская статья» нужно определить тип связи многие ко многим.

Между сущностью «Тег» и сущностью «Файл» нужно определить тип связи многие ко многим.

Для большей наглядности далее будет отображена общая схема разрабатываемой базы данных на рисунке 2.3.

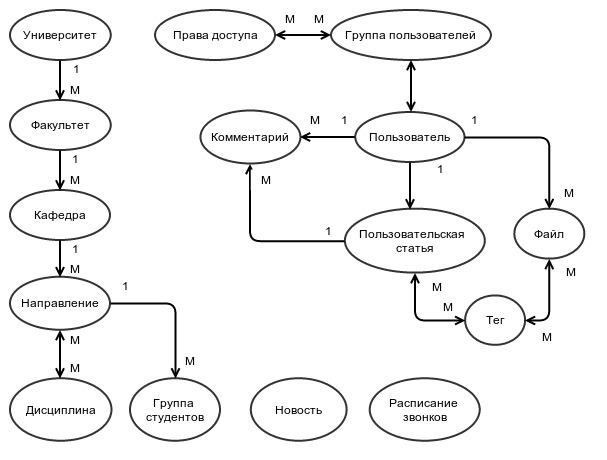


Рисунок 2.3 — Общая схема базы данных

На представленном рисунке связи отображены в виде линий, условное обозначение которых следующее:

1. сущность, в которую входит линия без стрелки и рядом имеется символ «1», характеризуется такой связью «один»;
2. сущность, в которую линия входит со стрелкой и рядом имеется символ «М», характеризуется как «многие».

В том случае, когда две сущности связаны линией без стрелок, означает что между ними связь «один к одному». Если линия с одного конца без стрелки, а с другого со стрелкой, значит между ними связь «один ко многим». Если же линия с обоих концов завершается стрелками, значит между сущностями связь «многие ко многим». Как уже упоминалось ранее, для большей информативности возле линий проставлены специальные символы «1» и «М».

**2.3 Структура классов системы «Информационный портал»**

В соответствии с определенными выше сущностями следует разработать классы. Классы условно можно разбить на две основные группы:

1. группа классов, предназначенная для обеспечения работы таких пользователей сайта как «гость», «продвинутый пользователь» и «эксперт»;
2. группа классов, предназначенная для обеспечения наполнения основного, базового контента ресурса.

Все классы наследуют абстрактный класс «Базовый класс», в котором определены следующие методы:

* создать объект — метод для создания объекта класса и инициализации начальных значений полей;
* изменить объект — метод, позволяющий сохранить и зафиксировать измененное состояние объекта в базе данных;
* удалить объект — метод, позволяющий удалить объект из базы данных;
* получить объект — метод, позволяющий получить один объект из базы данных по совокупному уникальному показателю;
* фильтровать объекты — метод, позволяющий получить множество объектов из базы данных, удовлетворяющим условиям фильтрации.

Первая группа классов представлена на рисунке 2.4 и состоит из следующего набора классов:

* права доступа;
* группа пользователей;
* пользователь;
* пользовательская статья;
* файл;
* комментарий;
* тег;

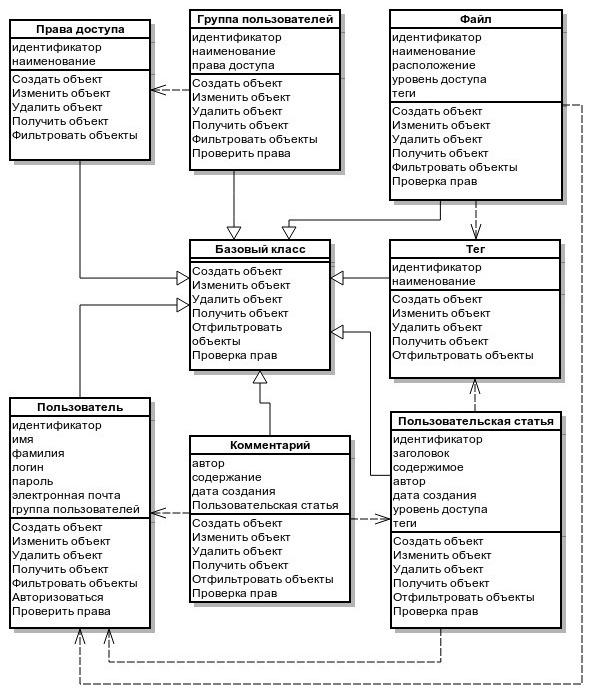


Рисунок 2.4 — Диаграмма классов первой группы

Класс «Права доступа» наследует «Базовый класс» и определяет права доступа к тому или иному объекту базы данных и содержит поля идентификатор, наименование. Для управления объектами имеются следующие методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты.

Класс «Группа пользователей» наследует «Базовый класс» и определяет то, какими правами доступа обладает все те, кто принадлежит к определенной группе. Данный класс обладает полями и методами для управления. Поля: идентификатор, наименование, права доступа. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты, проверить права.

Класс «Пользователь» наследует «Базовый класс» и описывает основные атрибуты пользователей сайта и их принадлежность к группам. Данный класс обладает полями и методами, управляющими объектами. Поля: идентификатор, имя, фамилия, логин, пароль, электронная почта, группы пользователей. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты, проверить права, авторизоваться.

Класс «Пользовательская статья» наследует «Базовый класс» и описывает атрибуты статьи. Данный класс обладает полями и методами, управляющими объектами. Поля: идентификатор, заголовок, содержимое, автор, дата создания, уровень доступа, теги. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты, проверить права.

Класс «Комментарий» наследует «Базовый класс» и описывает атрибуты комментариев пользователей на пользовательские статьи. Данный класс обладает полями и методами, управляющими объектами. Поля: идентификатор, заголовок, автор, дата создания, пользовательская статья. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты, проверить права.

Класс «Файл» наследует «Базовый класс» и описывает атрибуты файлов, которые загружены на сервер сайта и выдаются пользователям в соответствии с их уровнем доступа. Данный класс обладает полями и методами, управляющими объектами. Поля: идентификатор, наименование, расположение, владелец, дата загрузки на сервер, уровень доступа, теги. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты, проверить права.

Класс «Тег» наследует «Базовый класс» и описывает атрибуты тегов, которые определяют основные критерии поиска для файлов и пользовательских статей. Данный класс обладает полями и методами, управляющими объектами. Поля: идентификатор, наименование. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты, проверить права.

Вторая группа представлена на рисунке 2.5 и состоит из следующего набора классов:

* университет;
* факультет;
* кафедра;
* направление;
* дисциплина;
* группа студентов.

Класс «Университет» наследует «Базовый класс» и описывает поля и методы соответствующих объектов. Поля: идентификатор, сокращенное наименование, наименование, описание. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты.

Класс «Факультет» наследует «Базовый класс» и описывает поля и методы факультетов, относящихся к указанному университету. Поля: идентификатор, сокращенное наименование, наименование, описание, университет. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты.

Класс «Кафедра» наследует «Базовый класс» и описывает поля и методы кафедр, относящихся к указанному факультету. Поля: идентификатор, сокращенное наименование, наименование, описание, факультет. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты.

Класс «Направление» наследует «Базовый класс» и описывает поля и методы направлений, выпускаемых указанной кафедрой . Поля: идентификатор, сокращенное наименование, наименование, описание, кафедра, дисциплины. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты.

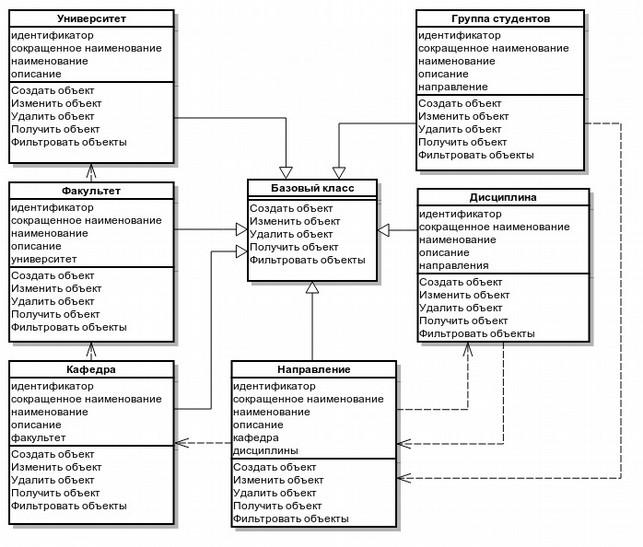


Рисунок 2.5 — Диаграмма второй группы классов

Класс «Дисциплина» наследует «Базовый класс» и описывает поля и методы соответствующих объектов. Поля: идентификатор, сокращенное наименование, наименование, описание, направления. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты.

Класс «Группа студентов» наследует «Базовый класс» и описывает поля и методы групп студентов, относящихся к указанным направлениям. Поля: идентификатор, сокращенное наименование, наименование, описание, направление. Методы: создать объект, изменить объект, удалить объект, получить объект, фильтровать объекты.

**2.4 Описание основных принципов обработки запросов**

Взаимодействие клиента с сервером происходит при помощи запросов. Запрос состоит из заголовка и тела. При каждом запросе на стороне сервера необходимо проверять от кого пришел запрос, тип запроса, проверка прав клиента на получение результатов посланного запроса. Для описания процесса обработки пользовательского запроса можно использовать диаграмму вариантов использования, которая предоставлена на рисунке 2.6.

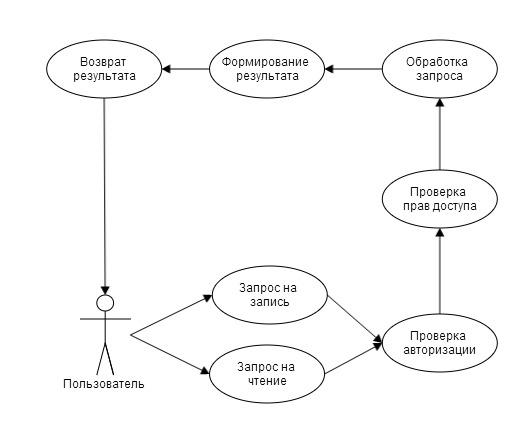


Рисунок 2.6 — Диаграмма вариантов использования обработки запроса пользователя

На данной диаграмме показано, что процесс проходит с выполнением следующего алгоритма:

1. пользователь посылает запрос серверу, который по существу может быть двух видов: на запись или на чтение.
2. сервер проверяется не авторизован ли пользователь.
3. сервер проверяет доступна ли информация или операция, которую запросил клиент.
4. сервер производит обработку непосредственно запроса обработка запроса, в соответствии с результатами проверки прав доступа.
5. сервер производит формирование результата — ответа на запрос клиента;
6. сервер посылает ответ обратно пользователю.

Сам процесс авторизации продемонстрирован на рисунке 2.7 и проходит в два этапа:

1. запрос на получение формы для авторизации;
2. отправка формы с заполненными полями для авторизации.

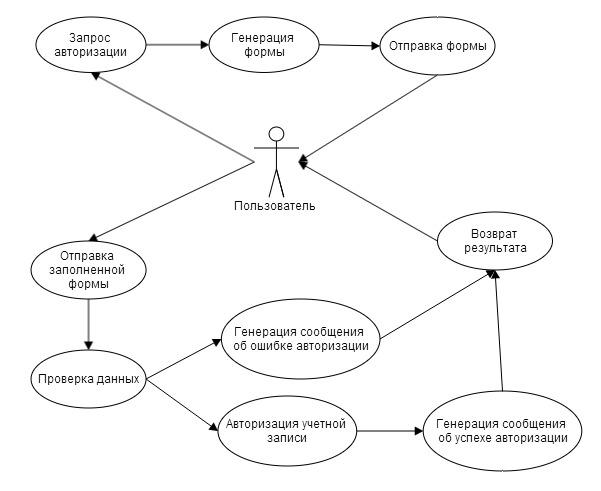


Рисунок 2.7 — Диаграмма вариантов использования авторизации пользователя

Этап запроса формы происходит в следующем порядке:

1. пользователь посылает запрос авторизации;
2. сервер генерирует форму авторизации;
3. сервер отправляет сгенерированную форму обратно пользователю.

Этап авторизации происходит в следующем порядке:

1. пользователь отправляет на сервер форму авторизации заполненную всеми необходимыми данными;
2. сервер производит проверку данных формы;
3. если данные формы некорректны, то происходит генерация сообщения об ошибке авторизации и переход к пункту 5;
4. если данные формы корректны, то фиксируется авторизация пользователя в системе.
5. сервер генерирует сообщение об успехе авторизации.
6. сервер возвращает результат обработки запроса обратно пользователю.

Авторизация необходима для того, чтобы на стороне сервера можно было определить права доступа пользователя, не заставляя его каждый раз подтверждать свои права, вводя свои логин и пароль.

Рассмотрим процесс редактирования статьи пользователем. Данный пример позволит нам полее подробно изучить процесс обработки сложного запроса. Здесь будут показаны все проверки, которые необходимо проводить для обеспечения разделения доступа к информации. Так же будут показаны этапы обработки, которые необходимо пройти в случае отрицательного результата той или иной проверки.

Весь процесс состоит из двух этапов:

1. этап получения формы для редактирования статьи;
2. этап отправки заполненной данными формы.

Фактически каждый этап представляет собой самостоятельный запрос, а значит многие шаги этих этапов будут совпадать по причине их однородности.

Различием между ними будет являться характер запроса. В одном случае производится запрос на получение данных с соответствующим уровнем доступа, а в другом случае производится запрос на применение и внос изменений в базу данных.

Процесс редактирования статьи продемонстрирован на рисунке 2.8.

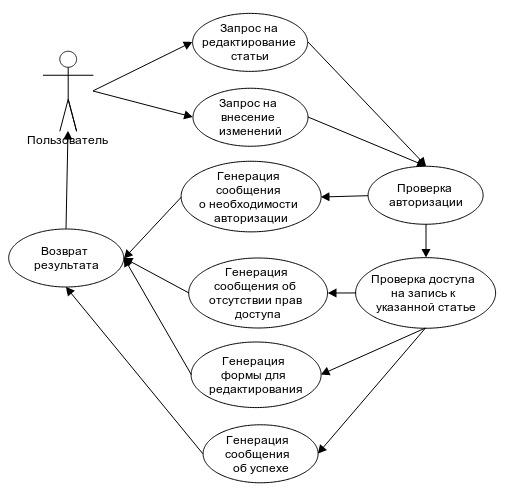


Рисунок 2.8 — Диаграмма вариантов использования   
редактирования пользовательской статьи

Этап получения формы состоит из следующего набора шагов:

1. пользователь отправляет запрос формы для редактирование указанной статьи;
2. сервер проверяет пользователя на авторизацию, если пользователь авторизован, осуществляется переход к шагу 4, иначе к шагу 3;
3. сервер генерирует сообщение о необходимости авторизации и переходит к шагу 7;
4. сервер проверяет доступ на запись пользователя к указанной статье, если у пользователя есть права на редактирование, осуществляется переход к шагу 6, иначе переход к шагу 5;
5. сервер генерирует сообщение о отсутствие прав на редактирование статьи и переходит к шагу 7;
6. сервер генерирует форму для внесения изменения в указанную статью с пред заполненными полями данными самой статьи;
7. сервер возвращает сгенерированный результат работы пользователю.

Этап отправки заполненной формы для изменения состоит из следующего набора шагов:

1. пользователь отправляет форму с запросом на внесение изменений указанной статьи;
2. сервер проверяет пользователя на авторизацию, если пользователь авторизован, осуществляется переход к шагу 4, иначе к шагу 3;
3. сервер генерирует сообщение о необходимости авторизации и переходит к шагу 8;
4. сервер проверяет доступ на запись пользователя к указанной статье, если у пользователя есть права на редактирование, осуществляется переход к шагу 6, иначе переход к шагу 5;
5. сервер генерирует сообщение о отсутствие прав на редактирование статьи и переходит к шагу 8;
6. сервер генерирует сообщение об успехе внесения изменений в указанную статью;
7. сервер возвращает сгенерированный ответ пользователю.

**2.5 Выбор методов и средств разработки**

Для реализации проекта «Информационный портал» необходимо произвести анализ и выбор средств реализации, среди множества имеющихся. В настоящее время выбор средств разработки и реализации программных приложений весьма разнообразен и отличается типами лицензий и функциональными возможностями.

Среди типов лицензий можно выделить два основных: коммерческие и свободные. Коммерческие лицензии подразумевают оплату за использование продукта и ограничение на его эксплуатацию и модификацию. Свободные лицензии предоставляют абсолютно безграничный доступ к продуктам, как с точки зрения цены, которой попросту нет, так и с точки зрения модификаций.

При реализации любого проекта используется определенный язык программирования. Каждый язык имеет свои особенности и реализации. В общем необходимо сказать, что, как правило, приходится балансировать между скоростью реализации проекта и скоростью работы реализованного проекта.

В настоящее время в разработке *web* приложений преобладают следующие языки программирования:

* *C++*;
* *C#*;
* *Java*;
* *JavaScript*;
* *PHP*;
* *Python*;
* *Ruby*.

В виду того, что сайта информационного портала кроме своего основного предназначения по предоставлению информации должен производит математический анализ и расчет сложных математических формул, то следует выбрать язык, для которого имеются соответствующие библиотеки. Таким является язык *Python*.

Используем для разработки информационного портала язык программирования *Python*. Этот язык программирования распространяется под свободной лицензией и разрабатывается соответствующим сообществом. К основным преимуществам следует отнести простоту синтаксиса, которая позволяет кратко описывать сложные алгоритмы, а также обладает великим множеством библиотек, направленных на решения тех или иных типов задач.

Одним из примеров библиотек данного языка является *Django Framework*, предназначенный для успешной и быстрой реализации *web* ресурсов. Эта библиотека в полной мере поддерживает концепции MVC и ORM, содержит богатый набор средств для организация разделения доступа к информации, создания и обращения к различным базам данных.

*Django Framework* позволяет сосредоточиться на реализации проекта и отложить отложить выбор конкретной системы управления базой данных на завершающий этап, так как рассчитан на взаимодействие с большинством из современных соответствующих систем.

Рассмотрим некоторые из часто используемые системы управления базами данных в *web* разработке:

* *SQLite*;
* *My SQL*;
* *MS SQL*;
* *ORACLE SQL*;
* *POSTGRE SQL*.

Главным образом следует отметить, что наиболее используемыми являются *SQLite* и *My SQL* по причине их простоты, малому размеру и общедоступности. Однако предпочтение следует отдать именно *My SQL*, так как эта система управления базами данных выполняет свои функции более эффективно, чем *SQLite*.

**3 РАЗРАБОТКА МЕТОДИКИ ОЦЕНКИ ЭФФЕКТИВНОСТИ   
РАБОТЫ САЙТА «ИНФОРМАЦИОННЫЙ ПОРТАЛ»**

**3.1 Определение подходящих критериев оценки эффективности**

Как было показано в главе 1, все показатели эффективности работы можно представить в виде следующих групп:

1. бизнес метрики;
2. метрики *usability;*
3. *SEO* метрики;
4. *SMO* метрики.

Необходимо определить те показатели, которые можно использовать для *web*-ресурса «Информационный портал». Для этого рассмотрим каждую группу метрик в контексте данного сайта.

Бизнес метрики предназначены для определения эффективности сайта с точки зрения генерации дополнительной финансовой прибыли. И так как сайт «Информационный портал» не ставит перед собой цели получения финансовой выгоды, то применение данной группы метрик не имеет практического смысла.

Рассмотрим метрики *usability*. Данная группа метрик предназначена для определения эффективности сайта в контексте его удобства использования пользователями. Так как сайт «Информационный портал» предоставляет информацию пользователям и предоставляет авторам статей размещать свою информацию, позволяющую улучшить степень обучаемости студентов, то предоставление авторам удобных инструментов для написания текстов и достаточно эффективная организация навигации по сайту являются первоочередными задачами, которые необходимо решить.

Можно с уверенностью сказать, что применение метрик *usability* позволят оценить успешность реализации тех или иных подходов в организации пользовательского интерфейса.

*SEO* метрики предназначены для определения «известности» *web*-ресурса в поисковых системах. В данном случае разработка сайта ведется таким образом, чтобы поисковые роботы повышали рейтинг сайта в определенной поисковой системе, причем, различные поисковые системы применяют свои критерии и подходы в определении ранга сайта среди остальных сайтов.

Такие метрики подразумевают, что *web*-ресурс доступен этим поисковым системам. Это возможно только в том случае, если сайт транслируется во всемирную сеть *Internet*. В данном случае сайт «Информационный портал» доступен исключительно во внутренней сети ИТИ, что означает его недосягаемость для поисковых роботов, а, следовательно, применение *SEO* метрик является невозможным.

*SMO* метрики позволяют определить эффективность сайта в контексте его упоминания в социальных сетях, и уровень одобрения пользователей социальных сетей касательно указанного сайта. Из определения следует, что для использования данной группы метрик необходимо иметь доступ к социальным сетям, в которых и будет производиться поиск упоминаний и анализироваться характер и тональность этих упоминаний.

Все социальные сети доступны через *Internet*. В результате этого *SMO* метрики не могут быть использованы в виду того, что web-ресурс «Информационный портал» не подразумевает доступа к глобальной сети.

В результате, для расчета эффективности данного web-ресурса, следует выделить следующие показатели:

* число уникальных посетителей;
* среднее число просмотров на посетителя;
* время, проведенное на странице;
* время, проведенное на сайте;
* показатель отказов;
* показатель прерываний визита в различных формах;
* следующие страницы;
* внутренние поиски.

**3.2 Разработка методики оценки эффективности**

Методика оценки эффективности основана на функции комплексной оценки качества Харрингтона-Менчера. Основная идея заключается в использовании различных комбинаций определенных критериев эффективности работы сайта для использования в качестве параметров *D*-функции [3, с. 296].

Условные обозначения факторов будут следующими:

1. X1*i*– число посетителей *i*-й группы за указанный период;
2. X1*i max*– наибольшее число посетителей *i*-й группы, выбранное из всех указанных периодов;
3. X2*i* – число просмотров страниц сайта пользователями *i*-й группы за указанный период;
4. X2*i max* – наибольшее число просмотров страниц сайта пользователями *i-*й группы из всех указанных периодов;
5. X3*i* – число просмотров с длительностью больше *ti* страниц пользователями i-й группы за указанный период;
6. X3*i max* – наибольшее число просмотров с длительностью больше *ti* страниц пользователями i-й группы из всех указанных периодов;
7. X4*i* – число скачиваний различных файлов пользователями *i*-й группы за указанный период;
8. X4*i max* – наибольшее число скачиваний файлов пользователями *i*-й группы из всех указанных периодов;
9. X5*i* – среднее время проведенное на странице сайта пользователями *i*-й группы за указанный период;
10. X5*i max* – наибольшее среднее время проведенное на странице сайта пользователями *i*-й группы из всех указанных периодов;
11. X6*i* – среднее время проведенное на сайте пользователями *i*-й группы за указанный период;
12. X6*i max* – наибольшее среднее время проведенное на сайте пользователями *i*-й группы из всех указанных периодов;
13. X7*i* – число поисковых запросов, сделанных пользователями i-й группы за указанный промежуток времени;
14. X7*i max* – наибольшее число поисковых запросов, сделанных пользователями i-й группы из всех указанных промежутков времени;

Указанные выше факторы используются для определения следующих показателей эффективности сайта:

1. показатель посещаемости сайта;
2. показатель интенсивности просмотра страниц сайта;
3. показатель скачиваний;
4. показатель длительности прибывания на страницах;
5. показатель длительности прибывания в общем на сайте;
6. показатель активности использования поисковых запросов на сайте;

Показатель посещаемости сайта позволяет рассмотреть изменение количества уникальных посещений сайта с течением времени. Показатель посещаемости рассчитывающийся по формуле 3.1.

, (3.1)

где *i* — номер группы пользователей;

*n* — количество групп пользователей;

*α*1*i* — степень важности для *i*-й группы.

Показатель интенсивности просмотра страниц сайта предназначен для определения изменения активности пользователей. Данный показатель является комплексным и рассматривает несколько слагаемых. Первые два слагаемых определяют все просмотры страниц. Вторые два слагаемых учитывают более тщательные просмотры страниц, когда пользователь той или иной группы тратит достаточное время для прочтения существенной части информации, которая расположена на странице.

Данный показатель рассчитывается по формуле 3.2;

,(3.2)

где *i* — номер группы пользователей;

*n* — количество групп пользователей;

*α*2*i*, *α*3*i* — степени важности для *i*-й группы;

*β*21,  *β*22,  *β*23, *β*24 — степени важности соответствующих слагаемых.

Показатель скачиваний определяет изменение интенсивности скачивания контента в зависимости от времени. Контент может быть представлен различными файлами, которые могут отличаться как по типу файла, так и по семантической нагрузке.

Этот показатель рассчитывается по формуле 3.3;

, (3.3)

где *i* — номер группы пользователей;

*n* — количество групп пользователей;

*α*4*i* — степень важности для *i*-й группы;

*β*31,  *β*32 — степени важности соответствующих слагаемых.

Показатель длительности прибывания на страницах позволяет увидеть изменения в том, на сколько много времени в среднем тратит пользователь той или иной группы на просмотр страницы с течением времени.

Можно предположить, что чем дольше пользователь тратит времени на просмотр страницы, тем интереснее для него является информация, расположенная на страницах сайта.

Данный показатель рассчитывается по формуле 3.4.

, (3.4)

где *i* — номер группы пользователей;

*n* — количество групп пользователей;

*α*5*i* — степень важности для *i*-й группы.

Показатель длительности прибывания в общем на сайте позволяет определить изменение длины интервала времени, в течение которого пользователь ведет активную деятельность на сайте и пользуется данными, которые выложены на сайте. Таким образом в дальнейшем можно определить, какую часть времени из всего учебного процесса пользователи различных групп проводят на сайте, и узнать, какую роль занимает сайт в учебном процессе и организационном процессе учреждения.

Данный показатель рассчитывается по формуле 3.5;

, (3.5)

где *i* — номер группы пользователей;

*n* — количество групп пользователей;

*α*6*i* — степень важности для *i*-й группы.

Показатель активности использования поисковых запросов на определенных страницах сайта позволяет определить, на сколько активно пользователи используют подсистему внутреннего поиска, насколько активность использования в один интервал времени, отличается от другого аналогичного по продолжительности интервала времени.

Показатель активности использования поисковых запросов рассчитывается по формуле 3.6;

, (3.6)

где *i* — номер группы пользователей;

*n* — количество групп пользователей;

*α*5*i* — степень важности для *i*-й группы.

В свою очередь эти элементарные показатели группируются в более общий показатель эффективности работы сайта, как это показано в формуле 3.7.

, (3.7)

Весь алгоритм расчета эффективности работы сайта состоит из следующей последовательности этапов:

1. сбор данных о поведении пользователей;
2. агрегация данных для получения входных параметров методов расчета;
3. непосредственное применение методов расчета.

Сбор данных осуществляется в то время, когда пользователи проводят активные действия с сайтов информационного портала. Деятельность пользователя в определенной мере отслеживается и записывается в специальную базу данных.

Перед использованием методов расчета определяются необходимые входные параметры, которые можно получить из базы данных, путем агрегации информации, которая в ней хранится. Далее определяются степени важности всех необходимых элементов формулы расчета.

На завершающем этапе все необходимые параметры подставляются в указанную результирующую формулу, и высчитывается показатель эффективности работы сайта.

**3.3 Разработка системы автоматизации оценки эффективности сайта**

Система автоматизированной оценки эффективности сайта должна представлять собой отдельный *web*-сервер, который будет способен обрабатывать достаточно быстро огромное число простых запросов.

Главной задачей такого сервера будет получение запросов от клиентских страниц информационного портала с данными о поведении пользователей и сохранять их в специальную базу данных. Другой задачей будет предоставление доступа к сохраненным данным в агрегированном виде.

Схема доступа к серверу будет аналогична схеме доступа к информационному порталу — с разделением прав доступа. Однако количество типов пользователей, которым будет разрешено получать данные сводится только к типу «Администратор».

Структура базы данных весьма проста и состоит всего из трех сущностей:

1. справочник причин ухода;
2. справочник типов запросов;
3. накопитель информации.

Сущность «Справочник причин ухода» содержит следующие атрибуты:

* идентификатор;
* наименование.

Сущность «Справочник типов запросов» содержит следующие атрибуты:

* идентификатор;
* наименование.

Сущность «Накопитель информации» предназначен для хранения определенной информации о поведении пользователей на указанных страницах. Сущность содержит следующие атрибуты:

* идентификатор;
* пользователь;
* URL источника;
* URL назначения;
* время проведенное на странице;
* причина ухода со страницы;
* тип запроса.

Между сущностями «Справочник причин ухода» и «Накопитель информации» следует установить связь типа «один ко многим».

Между сущностями «Справочник типов запросов» и «Накопитель информации» следует установить связь типа «один ко многим».

Рассмотреть описанные сущности можно на рисунке 3.1.



Рисунок 3.1 — Схема базы данных накопления информации   
о поведении пользователей

**ЗАКЛЮЧЕНИЕ**

В результате проделанной работы нами была разработана методика оценки эффективности информационной системы.

Данная методика была внедрена для оценки эффективности информационного портала Инженерно-технического института Приднестровского Государственного Университета имени Т. Г. Шевченко Приднестровской Молдавской Республики. Данный информационный ресурс был разработан для осуществления следующих основных задач:

1. Предоставлять доступ учащимся инженерно-технического института к просмотру информации, необходимой для улучшенного восприятия материала предоставляемого преподавателями.
2. Предоставлять возможность преподавателям инженерно-технического института публиковать информационный материал.
3. Предоставлять возможность публиковать информацию, необходимую для улучшенной организации учебного процесса, такую как расписание звонков, расписание занятий, важные новости и объявления, информация о кафедрах, преподавателях, учебных дисциплин.

Одним из основных ограничений являлось условие физического расположение сервера *web*-ресурса в пределах Инженерно-технического института с подключением этого сервера в общую сеть учреждения.

Ресурс должен достаточно хорошо справился с начальными нагрузками, которые были проведены на этапе заполнения сайта необходимой информацией. Серьезные нагрузки применены не были. Тестирование сервера при высоких нагрузках будет возможным при введении в полную эксплуатацию информационного портала Инженерно-техническим институтом.

**ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ,   
СИМВОЛОВ, ЕДИНИЦ И ТЕРМИНОВ**

ИТИ — Инженерно-технический институт.

ПГУ — Приднестровский Государственный Университет.

*SEO* — *Search engine optimization*.

*SMO* — *Social media optimization*.

*ROI* — *Return on investment*.

*MVC* — *Model-View-Controller*.

*ORM* — *Object-relational mapping.*

т. д. — так далее.

т. п. — тому подобное.

**СПИСОК ЛИТЕРАТУРЫ**

1. Авинаш Кошик. Веб-аналитика: анализ информации о посетителях веб-сайтов. Web-аналитика. Маркетинг в Интернет — М. Диалектика. 2008. — 464 с.
2. Ашманов Игорь Станиславович, Иванов Андрей Александрович. Продвижение сайта в поисковых системах — М. Вильямс. 2011. — 304 с.
3. Долгов Юрий Александрович. Статистическое моделирование. Учебник для вузов. — 2-е. изд., доп. — Тирасполь: Полиграфист, 2011. — 352 с.
4. Евдокимов Николай Владимирович, Лебединский Игорь Владимирович. Раскрутка веб-сайта: практическое руководство по *SEO* 3.0 — М. Диалектика-Вильямс. 2010. — 288 с.
5. Якоб Нильсен, Кара Перниче. Веб-дизайн: анализ удобства использования веб-сайтов по движению глаз — М. Вильямс. 2010. — 480 с.
6. Сервис *Google analytics* для обеспечения возможности проведения аналитики работы сайта от компании *Google*. [Электронный ресурс]. — Режим доступа: [http://www.google.com/analytics/]
7. Сервис Яндекс-Метрика для обеспечения возможности проведения аналитики работы сайта от компании Яндекс. [Электронный ресурс]. — Режим доступа: [https://www.metrika.yandex.ru/]
8. Метрики оценки эффективности продвижения сайта. [Электронный ресурс]. — Режим доступа: [http://www.mexboy.ru/?p=1620]

**ПРИЛОЖЕНИЕ**

**ЛИСТИНГ ИСХОДНЫХ КОДОВ СИСТЕМЫ   
«ИНФОРМАЦИОННЫЙ ПОРТАЛ»**

Листинг файла *settings.py*. Данный файл содержит код инициализации всей системы, определяет используемую систему управления базой данных, определяет используемые дополнительные модули.

*# coding: utf-8*

*import os*

*BASE\_DIR = os.path.dirname(os.path.dirname(\_\_file\_\_))*

*DEBUG = True*

*TEMPLATE\_DEBUG = DEBUG*

*ADMINS = (*

*# ('Your Name', 'your\_email@example.com'),*

*)*

*MANAGERS = ADMINS*

*DATABASES = {*

*'default': {*

*'ENGINE': 'django.db.backends.sqlite3',*

*'NAME': os.path.join(BASE\_DIR, 'university.db'),*

*'USER': '',*

*'PASSWORD': '',*

*'HOST': '',*

*'PORT': '',*

*}*

*}*

*ALLOWED\_HOSTS = []*

*TIME\_ZONE = 'Europe/Tiraspol'*

*LANGUAGE\_CODE = 'ru-RU'*

*SITE\_ID = 1*

*USE\_I18N = True*

*USE\_L10N = True*

*USE\_TZ = True*

*MEDIA\_ROOT = '/home/fallgamlet/sites\_html/UPortal/media/'*

*MEDIA\_URL = '/media/'*

*STATIC\_ROOT = ''*

*STATIC\_URL = '/static/'*

*ADMIN\_MEDIA\_PREFIX = '/static/admin/'*

*STATICFILES\_DIRS = (*

*'/home/fallgamlet/sites\_html/UPortal/sitestatic',*

*)*

*STATICFILES\_FINDERS = (*

*'django.contrib.staticfiles.finders.FileSystemFinder',*

*'django.contrib.staticfiles.finders.AppDirectoriesFinder',*

*)*

*SECRET\_KEY = '@fzcr$7\*#&13-8!&2(+t!nw#ayg-h&(1wr&(n\*&vc\_^ztg^e59'*

*TEMPLATE\_LOADERS = (*

*'django.template.loaders.filesystem.Loader',*

*'django.template.loaders.app\_directories.Loader',*

*)*

*MIDDLEWARE\_CLASSES = (*

*'django.middleware.common.CommonMiddleware',*

*'django.contrib.sessions.middleware.SessionMiddleware',*

*'django.middleware.csrf.CsrfViewMiddleware',*

*'django.contrib.auth.middleware.AuthenticationMiddleware',*

*'django.contrib.messages.middleware.MessageMiddleware',*

*)*

*ROOT\_URLCONF = 'UPortal.urls'*

*WSGI\_APPLICATION = 'UPortal.wsgi.application'*

*TEMPLATE\_DIRS = (*

*'/home/fallgamlet/sites\_html/UPortal/templates',*

*'/home/fallgamlet/sites\_html/UPortal/ckeditor/templates',*

*)*

*INSTALLED\_APPS = (*

*'django.contrib.auth',*

*'django.contrib.contenttypes',*

*'django.contrib.sessions',*

*'django.contrib.sites',*

*'django.contrib.messages',*

*'django.contrib.staticfiles',*

*'django.contrib.admin',*

*'university',*

*'ckeditor',*

*'blog',*

*)*

*CKEDITOR\_UPLOAD\_PATH = MEDIA\_ROOT + 'uploads/'*

*CKEDITOR\_RESTRICT\_BY\_USER = True*

*CKEDITOR\_UPLOAD\_PREFIX = '/media/uploads/'*

*CKEDITOR\_CONFIGS = {*

*'default': {*

*'toolbar': 'toolbar\_Full',*

*'height': 300,*

*'width': 800,*

*'toolbarCanCollapse': True,*

*'linkShowTargetTab': True,*

*'linkShowAdvancedTab': True,*

*},*

*'awesome\_ckeditor': {*

*'toolbar': 'Basic',*

*},*

*}*

*LOGGING = {*

*'version': 1,*

*'disable\_existing\_loggers': False,*

*'filters': {*

*'require\_debug\_false': {*

*'()': 'django.utils.log.RequireDebugFalse'*

*}*

*},*

*'handlers': {*

*'mail\_admins': {*

*'level': 'ERROR',*

*'filters': ['require\_debug\_false'],*

*'class': 'django.utils.log.AdminEmailHandler'*

*}*

*},*

*'loggers': {*

*'django.request': {*

*'handlers': ['mail\_admins'],*

*'level': 'ERROR',*

*'propagate': True,*

*},*

*} }*

Листинг файла url*.py*. Файл содержит описание псевдонимов субадресов, по которым будут обращаться пользователи с определенными запросами, а также содержит определение соответствия запросов и обработчиков запросов .

*# coding: utf-8*

*from django.conf.urls import patterns, url, include*

*from UPortal import views*

*from django.conf import settings*

*from django.contrib import admin*

*admin.autodiscover()*

*urlpatterns = patterns('',*

*(r'^admin/', include(admin.site.urls)),*

*(r'^ckeditor/', include('ckeditor.urls')),*

*(r'^blog/', include('blog.urls')),*

*(r'^$', views.main),*

*(r'^help/$', views.helps),*

*(r'^private/$', views.private\_cabinet),*

*(r'^news/$', views.news),*

*(r'^cathedra/$', views.cathedra\_info),*

*(r'^cathedra/(\d+)/$', views.cathedra\_info),*

*(r'^cathedra/direction/$', views.direction\_info),*

*(r'^cathedra/direction/(\d+)/$', views.direction\_info),*

*(r'^normdocument/$', views.normative\_document),*

*(r'^normdocument/(\d+)/$', views.normative\_document),*

*(r'^articles/$', views.articles),*

*(r'^schedule/$', views.schedule),*

*(r'^content/$', views.content),*

*(r'^content/search/$', views.search),*

*(r'^curdate/$', views.getcurrentdate),*

*(r'^media/(?P<path>.\*)$', 'django.views.static.serve' {'document\_root': settings.MEDIA\_ROOT}),*

*)*

Листинг файла *models.py*. Файл содержит описание классов, которые являются объектно-ориентированными представлениями таблиц базы данных. Здесь определяется функционал взаимодействия с базой данных.

*#! /usr/bin/python*

*# coding: utf-8*

*from django.db import models*

*from django.contrib import admin*

*from ckeditor.fields import RichTextField*

*import datetime*

*def book\_upload\_path(instance, filename):*

*"""Generates upload path for FileField for Books"""*

*res = u'books/%s/%s' % (instance.booktype.sname, filename)*

*return res;*

*def ducument\_upload\_path(instance, filename):*

*res = u'documents/%s/%s' %(instance.doctype.pk, filename)*

*return res;*

*class MyModel():*

*def getModel(self):*

*return self.\_\_class\_\_.\_\_name\_\_*

*class Article(models.Model, MyModel):*

*author = models.CharField(max\_length = 90, verbose\_name = 'Автор')*

*title = models.CharField(max\_length = 120, verbose\_name = 'Заголовок')*

*body = RichTextField(verbose\_name = 'Седержание', config\_name='default')*

*date = models.DateField(auto\_now = True, verbose\_name = 'Дата', default = datetime.datetime.now())*

*def \_\_unicode\_\_(self):*

*return u'%s %s' %(self.author, self.title)*

*class Meta:*

*verbose\_name = 'Пост'*

*verbose\_name\_plural = 'Посты'*

*class University(models.Model, MyModel):*

*sname = models.CharField(max\_length = 50, verbose\_name = 'Сокращение')*

*name = models.CharField(max\_length = 200, verbose\_name = 'Название')*

*description = models.TextField(max\_length = 1000, verbose\_name ='Описание', blank=True)*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.sname*

*class Meta:*

*verbose\_name = 'Университет'*

*verbose\_name\_plural = 'Университеты'*

*class Faculty(models.Model, MyModel):*

*sname = models.CharField(max\_length = 50, verbose\_name ='Сокращение')*

*name = models.CharField(max\_length = 200, verbose\_name ='Название')*

*description = models.TextField(max\_length = 1000, verbose\_name ='Описание', blank = True)*

*university = models.ForeignKey(University, verbose\_name ='Университет')*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.sname*

*class Meta:*

*verbose\_name = 'Факультет'*

*verbose\_name\_plural = 'Факультеты'*

*class Cathedra(models.Model, MyModel):*

*sname = models.CharField(max\_length = 50, verbose\_name ='Сокращение')*

*name = models.CharField(max\_length = 200, verbose\_name ='Название')*

*description = models.TextField(max\_length = 2048, verbose\_name = 'Описание', blank = True)*

*faculty = models.ForeignKey(Faculty, verbose\_name ='Факультет')*

*post = RichTextField(verbose\_name = 'Пост', null=True)*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.sname*

*class Meta:*

*verbose\_name = 'Кафедра'*

*verbose\_name\_plural = 'Кафедры'*

*class Subject(models.Model, MyModel):*

*sname = models.CharField(max\_length = 40, verbose\_name ='Сокращение')*

*name = models.CharField(max\_length = 200, verbose\_name ='Название')*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.name*

*class Meta:*

*ordering = ['name']*

*verbose\_name = 'Дисциплина'*

*verbose\_name\_plural = 'Дисциплины'*

*class EducatingType(models.Model, MyModel):*

*name = models.CharField(max\_length = 50, verbose\_name ='Название')*

*description = models.TextField(max\_length = 1000, verbose\_name = 'Описание', blank = True)*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.name*

*class Meta:*

*ordering = ['name']*

*verbose\_name = 'Вид обучения'*

*verbose\_name\_plural = 'Виды обучения'*

*class Direction(models.Model, MyModel):*

*sname = models.CharField(max\_length = 50, verbose\_name ='Сокращение')*

*name = models.CharField(max\_length = 200, verbose\_name ='Название')*

*description = models.TextField(max\_length = 2048, verbose\_name = 'Описание', blank = True)*

*cathedra = models.ForeignKey(Cathedra, verbose\_name ='Кафедра')*

*educationtype = models.ForeignKey(EducatingType, verbose\_name='Вид обучения', null=True)*

*subjects = models.ManyToManyField(Subject, verbose\_name ='Дисциплины')*

*post = RichTextField(verbose\_name = 'Пост', null=True)*

*def \_\_unicode\_\_(self):*

*return u'%s %s' %(self.educationtype.name, self.sname)*

*class Meta:*

*#ordering = ['educationtype', 'cathedra', 'sname']*

*verbose\_name = 'Направление'*

*verbose\_name\_plural = 'Направления'*

*class WorkerType(models.Model, MyModel):*

*name = models.CharField(max\_length = 200, verbose\_name ='Название')*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.name*

*class Meta:*

*verbose\_name = 'Тип работника'*

*verbose\_name\_plural = 'Типы работников'*

*class Worker(models.Model, MyModel):*

*last\_name = models.CharField(max\_length = 60, verbose\_name ='Фамилия')*

*first\_name = models.CharField(max\_length = 60, verbose\_name ='Имя')*

*middle\_name = models.CharField(max\_length = 60, verbose\_name ='Отчество')*

*workertype = models.ForeignKey(WorkerType, verbose\_name ='Должность')*

*cathedra = models.ForeignKey(Cathedra, verbose\_name = 'Кафедра')*

*def \_\_unicode\_\_(self):*

*return u'%s %s %s' %(self.last\_name, self.first\_name, self.middle\_name)*

*class Meta:*

*ordering = ['last\_name']*

*verbose\_name = 'Работник'*

*verbose\_name\_plural = 'Работники'*

*class StudentGroup(models.Model, MyModel):*

*sname = models.CharField(max\_length = 40, verbose\_name ='Сокращение')*

*name = models.CharField(max\_length = 200, verbose\_name ='Название')*

*direction = models.ForeignKey(Direction, verbose\_name ='Направление')*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.sname*

*class Meta:*

*verbose\_name = 'Группа студентов'*

*verbose\_name\_plural = 'Группы студентов'*

*class Student(models.Model, MyModel):*

*last\_name = models.CharField(max\_length = 60, verbose\_name ='Фамилия')*

*first\_name = models.CharField(max\_length = 60, verbose\_name ='Имя')*

*middle\_name = models.CharField(max\_length = 60, verbose\_name ='Отчество')*

*studentgroup = models.ForeignKey(StudentGroup, verbose\_name ='Группа')*

*def \_\_unicode\_\_(self):*

*return u'%s %s %s' %(self.first\_name, self.middle\_name, self.last\_name)*

*class Meta:*

*verbose\_name = 'Студент'*

*verbose\_name\_plural = 'Студенты'*

*class BookType(models.Model, MyModel):*

*sname = models.CharField(max\_length = 60, verbose\_name = 'Сокращение')*

*name = models.CharField(max\_length = 200, verbose\_name = 'Название')*

*description = models.TextField(blank = True, verbose\_name = 'Описание')*

*image = models.FileField(upload\_to = 'preview', blank = True, verbose\_name = 'Изображение', default="preview/document.png")*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.name*

*class Meta:*

*ordering = ['name']*

*verbose\_name = 'Тип книги'*

*verbose\_name\_plural = 'Типы книг'*

*class Book(models.Model, MyModel):*

*title = models.CharField(max\_length = 200, verbose\_name = 'Название')*

*authors = models.CharField(blank = True, max\_length = 200, verbose\_name = 'Авторы')*

*year = models.IntegerField(null = True, blank = True, verbose\_name = 'Год')*

*publisher = models.CharField(max\_length = 200, blank = True, verbose\_name = 'Издатель')*

*place = models.CharField(max\_length = 60, blank = True, verbose\_name = 'Место')*

*page = models.IntegerField(null = True, blank = True, verbose\_name = 'Страниц')*

*booktype = models.ForeignKey(BookType, verbose\_name = 'Тип')*

*subjects = models.ManyToManyField(Subject, verbose\_name = 'Дисциплины', null = True, blank = True)*

*bookfile = models.FileField(max\_length=512, upload\_to = book\_upload\_path, blank = True, verbose\_name = 'Файл')*

*description = models.TextField(blank = True, verbose\_name = 'Описание')*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.title*

*class Meta:*

*verbose\_name = 'Книга'*

*verbose\_name\_plural = 'Книги'*

*class NormativeDocumentType(models.Model, MyModel):*

*sname = models.CharField(max\_length = 60, verbose\_name = 'Сокращение')*

*name = models.CharField(max\_length = 200, verbose\_name = 'Название')*

*description = models.TextField(blank = True, verbose\_name = 'Описание')*

*image = models.FileField(upload\_to = 'preview', blank = True, verbose\_name = 'Изображение', default="preview/document.png")*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.name*

*class Meta:*

*ordering = ['name']*

*verbose\_name = 'Тип нормативного документа'*

*verbose\_name\_plural = 'Типы нормативных документов'*

*class NormativeDocument(models.Model, MyModel):*

*title = models.CharField(max\_length = 200, verbose\_name = 'Название')*

*doctype = models.ForeignKey(NormativeDocumentType, verbose\_name = 'Тип')*

*direction = models.ForeignKey(Direction, blank=True, null=True, verbose\_name ='Направление')*

*docfile = models.FileField(max\_length=512, upload\_to = ducument\_upload\_path, blank = True, verbose\_name = 'Файл')*

*description = models.TextField(blank = True, verbose\_name = 'Описание')*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.title*

*class Meta:*

*verbose\_name = 'Нормативный документ'*

*verbose\_name\_plural = 'Нормативные документы'*

*class News(models.Model, MyModel):*

*title = models.CharField(max\_length = 120, verbose\_name = 'Заголовок')*

*body = models.TextField(verbose\_name = 'Седержание')*

*startdate = models.DateField(verbose\_name = 'Дата создания', auto\_now=True, default = datetime.datetime.now())*

*enddate = models.DateField(verbose\_name = 'Дата актуальности', default = datetime.datetime.now())*

*weight = models.IntegerField(verbose\_name = 'Важность', default = 5, choices = [(i+1,i+1) for i in range(10)])*

*def \_\_unicode\_\_(self):*

*return u'%s' %(self.title)*

*class Meta:*

*verbose\_name = 'Новость'*

*verbose\_name\_plural = 'Новости'*

*class BellSchedule(models.Model, MyModel):*

*num = models.IntegerField(verbose\_name='Номер')*

*timestart = models.TimeField(verbose\_name='Начало')*

*timeend = models.TimeField(verbose\_name='Конец')*

*def \_\_unicode\_\_(self):*

*return u'%i: %s - %s' %(self.num, str(self.timestart), str(self.timeend))*

*class Meta:*

*ordering = ['num']*

*verbose\_name = 'Расписание звонков'*

*verbose\_name\_plural = 'Расписания звонков'*

*class NumeratorWeek(models.Model, MyModel):*

*date = models.DateField(verbose\_name='Дата числителя')*

*def \_\_unicode\_\_(self):*

*return u'%i: %s' %(self.pk, str(self.date))*

*class Meta:*

*ordering = ['-date']*

*verbose\_name = 'Расписание числителя и знаменателя'*

*verbose\_name\_plural = 'Расписания числителя и знаменателя'*

*class WelcomeMessage(models.Model, MyModel):*

*title = models.CharField(max\_length = 100, verbose\_name ='Заголовок')*

*author = models.CharField(max\_length = 50, verbose\_name ='Автор')*

*date = models.DateField(verbose\_name = 'Дата', default = datetime.datetime.now())*

*post = RichTextField(verbose\_name = 'Пост', null=True)*

*def \_\_unicode\_\_(self):*

*return u'%s' %self.title*

*class Meta:*

*verbose\_name = 'Приветственное сообщение'*

*verbose\_name\_plural = 'Приветственные сообщения'*

Листинг файла *forms.py*. Данный файл содержит описание формы ввода, которые используются для предоставления пользователю, а именно, формы поиска учебной литературы.

*#! /usr/bin/python*

*#-\*- coding: utf-8 -\*-*

*from django import forms*

*import django.contrib.admin.widgets as admin\_widjets*

*import models*

*class SearchBook(forms.Form):*

*title = forms.CharField(label = u'Название книги', max\_length = 90, required = False)*

*authors = forms.CharField(label = u'Авторы', max\_length = 90, required = False)*

*btype = forms.ModelMultipleChoiceField(*

*label = u'Типы',*

*queryset = models.BookType.objects.all(),*

*required = False*

*)*

*subjects = forms.ModelMultipleChoiceField(*

*label = u'Дисциплины',*

*queryset = models.Subject.objects.all(),*

*required = False*

*)*

*def is\_empty(self):*

*result = True*

*cd = self.cleaned\_data*

*for key in cd:*

*result &= not bool(cd[key])*

*return result*

Листинг файла *admin.py*. Данный файл содержит описание моделей базы данных, используемых в административном интерфейсе. Описывается то, как они будут представлены администратору сайта, какие действия с этими моделями допускается производить.

*#! /usr/bin/python*

*# coding: utf-8*

*from django.contrib import admin*

*from university.models import \**

*class WorkerInlineAdmin(admin.TabularInline):*

*model = Worker*

*extra = 1*

*class CathedraAdmin(admin.ModelAdmin):*

*inlines = (WorkerInlineAdmin,)*

*fieldsets = (*

*(None, {*

*'classes': ['extrapretty'],*

*'fields': ('sname', 'name', 'faculty', 'description')*

*}),*

*(u'Пост', {*

*'classes': ('collapse',),*

*'fields': ('post',)*

*}),*

*)*

*class BookTypeAdmin(admin.ModelAdmin):*

*list\_filter = ('sname',)*

*class SubjectAdmin(admin.ModelAdmin):*

*pass*

*#list\_filter = ('sname',)*

*class DirectionAdmin(admin.ModelAdmin):*

*list\_filter = ('educationtype', 'cathedra')*

*formfield\_overrides = {*

*models.ManyToManyField: {*

*'widget': admin.widgets.FilteredSelectMultiple(u'дисциплины', is\_stacked=False)*

*},*

*}*

*class BookAdmin(admin.ModelAdmin):*

*list\_filter = ('booktype', 'subjects')*

*#raw\_id\_fields = ('subject',)*

*formfield\_overrides = {*

*models.ManyToManyField: {*

*'widget': admin.widgets.FilteredSelectMultiple(u'дисциплины', is\_stacked=False)*

*},*

*}*

*class NormativeDocumentAdmin(admin.ModelAdmin):*

*list\_filter = ('doctype',)*

*class BellScheduleAdmin(admin.ModelAdmin):*

*list\_filter = ('num',)*

*admin.site.register(University)*

*admin.site.register(Faculty)*

*admin.site.register(Cathedra, CathedraAdmin)*

*admin.site.register(EducatingType)*

*admin.site.register(Direction, DirectionAdmin)*

*admin.site.register(Subject, SubjectAdmin)*

*admin.site.register(BookType, BookTypeAdmin)*

*admin.site.register(Book, BookAdmin)*

*admin.site.register(Worker)*

*admin.site.register(WorkerType)*

*admin.site.register(Student)*

*admin.site.register(StudentGroup)*

*admin.site.register(Article)*

*admin.site.register(News)*

*admin.site.register(BellSchedule, BellScheduleAdmin)*

*admin.site.register(NormativeDocumentType)*

*admin.site.register(NormativeDocument, NormativeDocumentAdmin)*

*admin.site.register(NumeratorWeek)*

*admin.site.register(WelcomeMessage)*

Листинг файла *views.py*. Данный файл содержит описание методов обработки запросов, поступающий на сервер.

*#! /usr/bin/python*

*# -\*- coding: utf-8 -\*-*

*from django.http import HttpResponse*

*from django.shortcuts import render\_to\_response*

*from django.db.models import Q*

*import sys*

*import datetime*

*import copy*

*from university import models*

*from university import forms*

*urlvoc = {*

*"": u"Главная",*

*"news": u"Новости",*

*"help": u"Помощь",*

*"search": u"Поиск",*

*"articles": u"Статьи",*

*"content": u"Учебный материал",*

*"cathedra": u"Информация о кафедрах",*

*"direction": u"Направления",*

*"normdocument": u"Нормативные документы",*

*"schedule": u"Расписание"*

*}*

*menulinks = (*

*("/", u"Главная"),*

*("/cathedra", u"Информация о кафедрах"),*

*("/cathedra/direction/", u"Информация о направлениях"),*

*("/news/", u"Новости и объявления"),*

*("/schedule/", u"Расписание"),*

*("/normdocument", u"Нормативные документы"),*

*("/content/", u"Учебный материал (поиск)"),*

*)*

*def main(response):*

*params = dict()*

*params['menulinks'] = menulinks*

*params['links'] = getlinks(response)*

*params['welcomemessage'] = getLastWelcomeMessage()*

*params['news'] = getNews(5)*

*params['bellschedule'] = models.BellSchedule.objects.order\_by('num')*

*params['numeratorweek'] = getLastNumeratorWeek()*

*#*

*return render\_to\_response('main.djhtml', params)*

*def helps(response):*

*links = getlinks(response)*

*return render\_to\_response('help.djhtml', {'menulinks': menulinks, 'links': links})*

*def search(response):*

*links = getlinks(response)*

*if response.method == 'GET':*

*sform = forms.SearchBook(response.GET)*

*if sform.is\_valid() and not sform.is\_empty():*

*cd = sform.cleaned\_data*

*books = []*

*qtitle = Q(title\_\_icontains = cd['title'])*

*qauthor = Q()*

*if cd['authors']:*

*authors = cd['authors'].split(' ')*

*for author in authors:*

*qauthor |= Q(authors\_\_icontains = author)*

*if cd['btype']:*

*qbtype = Q(booktype\_\_pk\_\_in = cd['btype'])*

*else:*

*qbtype = Q()*

*if cd['subjects']:*

*qsubject = Q(subjects\_\_pk\_\_in = cd['subjects'])*

*else:*

*qsubject = Q()*

*books += models.Book.objects.filter(qtitle, qauthor, qbtype, qsubject).distinct()*

*return render\_to\_response('search.djhtml', {'menulinks': menulinks, 'links': links, 'sform': sform, 'books': books})*

*else:*

*sform = forms.SearchBook()*

*return render\_to\_response('search.djhtml', {'menulinks': menulinks, 'links': links, 'sform': sform})*

*def content(response):*

*links = getlinks(response)*

*return render\_to\_response('content.djhtml', {'menulinks': menulinks, 'links': links})*

*def private\_cabinet(response):*

*links = getlinks(response)*

*return render\_to\_response('private.djhtml', {'menulinks': menulinks, 'links': links})*

*def news(response):*

*links = getlinks(response)*

*newslist = []*

*today = datetime.date.today()*

*newslist = models.News.objects.filter(enddate\_\_gt = today).order\_by('-startdate', '-weight')*

*return render\_to\_response('news.djhtml', {'menulinks': menulinks, 'links': links, 'news': newslist})*

*def articles(response):*

*links = getlinks(response)*

*articles = models.Article.objects.all()*

*return render\_to\_response('articles.djhtml', {'menulinks': menulinks, 'links': links, 'articles': articles})*

*def schedule(response):*

*links = getlinks(response)*

*return render\_to\_response('schedule.djhtml', {'menulinks': menulinks, 'links': links})*

*def cathedra\_info(response, cath\_pk=None):*

*links = getlinks(response)*

*if cath\_pk:*

*departments = models.Cathedra.objects.filter(pk = cath\_pk)*

*if departments:*

*links[-1] = [links[-1][0], departments[0].sname]*

*else:*

*links.pop()*

*else:*

*departments = models.Cathedra.objects.order\_by('name')*

*educationtypes = list(models.EducatingType.objects.order\_by('name'))*

*directions = list(models.Direction.objects.order\_by('name'))*

*for dep in departments:*

*dep.edtypes = copy.deepcopy(educationtypes)*

*for edtype in dep.edtypes:*

*edtype.directions = list()*

*for direction in directions:*

*if direction.cathedra.pk == dep.pk and direction.educationtype.pk == edtype.pk:*

*edtype.directions.append(direction)*

*return render\_to\_response('cathedra\_info.djhtml', {'menulinks': menulinks, 'links': links, 'departments':departments})*

*def direction\_info(response, dir\_pk=None):*

*links = getlinks(response)*

*if dir\_pk:*

*directions = models.Direction.objects.filter(pk = dir\_pk)*

*if directions: links[-1] = [links[-1][0], directions[0].sname]*

*else: links.pop()*

*edtypes = None*

*else:*

*edtypes = models.EducatingType.objects.all()*

*directions = models.Direction.objects.order\_by('educationtype', 'name')*

*return render\_to\_response('direction\_info.djhtml', {'menulinks': menulinks, 'links': links, 'directions':directions, 'edtypes': edtypes})*

*def normative\_document(response, dir\_pk=None):*

*links = getlinks(response)*

*if dir\_pk:*

*directions = models.Direction.objects.filter(pk = dir\_pk)*

*normdocs = models.NormativeDocument.objects.filter(direction = dir\_pk).order\_by('doctype')*

*if directions:*

*links[-1] = [links[-1][0], directions[0].sname]*

*else:*

*links.pop()*

*edtypes = None*

*else:*

*directions = models.Direction.objects.order\_by('educationtype', 'name')*

*normdocs = models.NormativeDocument.objects.all()*

*edtypes = models.EducatingType.objects.all()*

*doctypes = models.NormativeDocumentType.objects.order\_by('name')*

*for direct in directions:*

*direct.doctypes = dict([(t.pk, [t.pk, t.name]) for t in doctypes])*

*for ndoc in normdocs:*

*if ndoc.direction.pk == direct.pk:*

*direct.doctypes[ndoc.doctype.pk].append(ndoc)*

*direct.doctypes = direct.doctypes.values()*

*return render\_to\_response('normative\_document.djhtml', {'menulinks': menulinks, 'links': links, 'directions':directions, 'edtypes': edtypes})*

*def getNews(count=None):*

*today = datetime.date.today()*

*news = models.News.objects.filter(enddate\_\_gt = today).order\_by('-startdate')*

*if count:*

*news = news[:5]*

*#*

*return news*

*def getLastWelcomeMessage():*

*welcmess = models.WelcomeMessage.objects.order\_by('-date')[:1]*

*if len(welcmess) > 0:*

*welcmess = welcmess[0]*

*else:*

*welcmess = None*

*#*

*return welcmess*

*def getLastNumeratorWeek():*

*numeratorweek = models.NumeratorWeek.objects.order\_by('-date')[:1]*

*if len(numeratorweek) > 0:*

*numeratorweek = numeratorweek[0]*

*else:*

*numeratorweek = None*

*return numeratorweek*

*def getlinks(response):*

*href = response.path*

*links = href.split("/")*

*links.pop()*

*links[0] = ["/", urlvoc[links[0]]]*

*for i in range(1, len(links)):*

*if links[i] in urlvoc:*

*links[i] = [links[i-1][0] + links[i]+"/", urlvoc[links[i]]]*

*else:*

*links[i] = [links[i-1][0] + links[i]+"/", links[i]]*

*#*

*return links*

*def getcurrentdate(response):*

*curdate = datetime.date.today()*

*print "method in, date: "+str(curdate)*

*import json*

*print "imported json lib"*

*res = json.dumps({"curdate": str(curdate)})*

*print res*

*print "method out"*

*#*

*return HttpResponse(res)*

Листинг файла *base.djhtml*. Данный файл содержит описание базового шаблон основных страниц пользовательского интерфейса.

*<!DOCTYPE HTML>*

*<html>*

*<head>*

*{% block meta %}*

*<meta http-equiv="content-type" content="text/html; charset=utf-8" />*

*<meta http-equiv="content-language" content="ru" />*

*{% endblock %}*

*{% block links %}*

*<link rel="stylesheet" type="text/css" href="/static/css/main\_style.css">*

*{% endblock %}*

*<title>{% block title %}ИНФОРМАЦИОННЫЙ ПОРТАЛ ИТИ {% endblock %}</title>*

*{% block scripts %}*

*<script type="text/javascript" src="/static/js/jquery.js"></script>*

*<script type="text/javascript" src="/static/js/my\_script.js"></script>*

*{% endblock %}*

*</head>*

*<body>*

*<div id="background">*

*<a href="/admin/"><div class='to\_admin'>Администрирование</div></a>*

*<div id="top">*

*<b>*

*<h1 style="font-size: 32pt;">ИНФОРМАЦИОННЫЙ ПОРТАЛ</h1>*

*<h1>Инженерно-технический институт </h1>*

*<br/>*

*<h3 align='right'>Мы готовы делиться своими знаниями со всеми, кто ставит перед собой новые цели, бросает вызов сложностям и стремится расти </h3>*

*</b>*

*</div>*

*<div id="links">*

*{% for link, verbose\_name in links %}*

*<a href="{{ link }}"> {{ verbose\_name }}</a>*

*{% endfor %}*

*</div>*

*<div id="menu">*

*<ul>*

*{% for item in menulinks %}*

*<li><a href="{{ item.0 }}">{{ item.1 }}<span></span></a></li>*

*{% empty %}*

*Нет ни одного элемента меню*

*{% endfor %}*

*</ul>*

*{% include 'bell\_schedule.djhtml' %}*

*</div>*

*<div id="content">*

*{% block workspace %} {% endblock %}*

*</div>*

*{% comment %}*

*{% endcomment %}*

*</div>*

*<div id="foot">*

*автор: Левицкий Егор Андреевич*

*</div>*

*</body>*

*</html>*

Листинг файла *main.djhtml*. Даннй файл содержит описание шаблона «главной» страницы сайта. Наследует шаблон *base.djhtml*.

{% extends "base.djhtml" %}

{% block scripts %}

{{ block.super }}

<script type="text/javascript" src="/static/js/calendar-widget.js"></script>

<script type="text/javascript">

$(document).ready(function(){

var today = new Date()

$("#calendar").calendarWidget();

$.ajax({

url : '/curdate/',

type: 'get',

dataType: 'json',

success: function(jsonRes){

if(jsonRes){

console.log(jsonRes);

}

},

error: function(){

console.log('something error')

}

});

});

</script>

{% endblock %}

{% block workspace %}

<div style="width: 300px; float:right; background:transparent;">

<div id="calendar" class="calendar\_table"></div>

<div class="print\_news">

{% for item in news %}

<p>

<b>{{ item.title|upper }}</b>

<br/>

{{ item.body|linebreaksbr|capfirst }}

</p>

{% empty %}

<p> Новостей нет. 8)</p>

{% endfor %}

</div>

</div>

<div class="main\_about">

<div class='welcome\_message' style='background:#fff'>

{{ welcomemessage.post|safe }}

</div>

{#% include 'bell\_schedule.djhtml' %#}

<!--div id="calendar" class="calendar\_table"></div-->

</div>

{% endblock %}

Листинг файла *direction\_info.djhtml*. Данный файл содержит описание пользовательского интерфейса страницы с отображением информации о направлениях, выпускаемых кафедрами факультета. Наследует шаблон *base.djhtml*.

*{% extends "base.djhtml" %}*

*{% block scripts %}*

*{{ block.super }}*

*<script type="text/javascript">*

*$(document).ready(function(){*

*$(".accordion h4:first").addClass("active");*

*$(".accordion .data").hide();*

*$(".accordion h4").click(function(){*

*$(this).next(".data").slideToggle("slow").siblings(".data:visible").slideUp("slow");*

*$(this).toggleClass("active");*

*$(this).siblings("h4").removeClass("active");*

*});*

*});*

*</script>*

*{% endblock %}*

*{% block workspace %}*

*{% if departments|length == 1%}*

*<div class="main\_about" style='width:96%; padding: 10px 1em;'>*

*<h4>Кафедра: {{ departments.0.name }}</h4>*

*<h4>Сокращение: {{ departments.0.sname }}</h4>*

*<br />*

*<div class='directions'>*

*Направления:*

*<ul>*

*{% for edtype in departments.0.edtypes %}*

*{% for direction in edtype.directions %}*

*{% if forloop.first %}*

*<li>*

*{{ edtype.name }}:*

*{% endif %}*

*<br/> - <a href='/cathedra/direction/{{ direction.pk }}'>{{ direction.name }}</a>*

*{% endfor %}*

*{% if forloop.last %}*

*</li>*

*{% endif %}*

*{% endfor %}*

*</ul>*

*</div>*

*{% if departments.0.post %}*

*{{ departments.0.post.body|safe }} <br />*

*{% endif %}*

*</div>*

*{% else %}*

*<div class='accordion'>*

*{% for item in departments %}*

*<h4>{{ item.name }}</h4>*

*<div class='data'>*

*<a class="botton" href='/cathedra/{{ item.pk }}'>Подробнее о кафедре</a>*

*Сокращеине: <b>{{ item.sname }}</b>*

*<div class='directions'>*

*Направления:*

*<ul>*

*{% for edtype in item.edtypes %}*

*{% for direction in edtype.directions %}*

*{% if forloop.first %}*

*<li>*

*{{ edtype.name }}:*

*{% endif %}*

*{{ edtype.name }}:*

*<br /> - <a href='/cathedra/direction/{{ direction.pk }}'>{{ direction.name }}</a>*

*{% if forloop.last %}*

*</li>*

*{% endif %}*

*{% endfor %}*

*{% endfor %}*

*</ul>*

*</div>*

*</div>*

*{% empty %}*

*Не указаны кафедры.*

*{% endfor %}*

*</div>*

*{% endif %}*

*{% endblock %}*

Листинг файла *cathedra\_info.djhtml*. Данный файл содержит описание пользовательского интерфейса страницы с отображением информации о кафедрах факультета. Наследует шаблон *base.djhtml*.

{% extends "base.djhtml" %}

{% block scripts %}

{{ block.super }}

<script type="text/javascript">

$(document).ready(function(){

//$(".accordion h4:first").addClass("active");

$(".accordion div.data").hide();

$(".accordion h4").click(function(){

$(this).next("div.data").slideToggle()

.siblings("div.data:visible").slideUp();

$(this).toggleClass("active");

$(this).siblings("h4").removeClass("active");

});

$(".educationtype").click(function(){

var id = this.id;

$("div.accordion").fadeOut("fast", function(){

$("div.accordion h4, div.data").hide();

$("div.accordion h4.class\_"+id).show();

//$("div.accordion div.class\_"+id+":first").show();

}).fadeIn();

});

$("#edtype\_all").click(function(){

$("div.accordion").fadeOut("fast", function(){

$("div.accordion h4").show();

$("div.accordion div.data:not(first)").hide();

//$("div.accordion div.data:first").show();

}).fadeIn();

});

});

</script>

{% endblock %}

{% block workspace %}

{% if directions|length == 1 %}

<div style="text-align: left; margin:10px; padding:10px;">

<b>Направление: {{ directions.0.name|capfirst }}</b>

<br />

<b>Сокращение: {{ directions.0.sname}}</b>

{% if directions.0.educationtype %}

<br>

<b>Вид обучения: {{ directions.0.educationtype.name }}</b>

{% endif %}

{% if directions.0.cathedra %}

<br>

<b>Кафедра: {{ directions.0.cathedra.name }} ({{directions.0.cathedra.sname}})</b>

{% endif %}

{% if directions.0.post %}

<br /> {{ directions.0.post|safe }}

{% endif %}

{% if directions.0.subjects %}

<p>

Дисциплины профессионального блока:

{% for subject in directions.0.subjects.all %}

<br /> - {{ subject.name|capfirst }}

{% endfor %}

</p>

{% endif %}

</div>

{% else %}

<div style="background: none; margin:10px; padding:10px; display:block;">

{% for edtype in edtypes %}

<div id="edtype\_{{ edtype.pk }}" class="educationtype clickable" style="display: inline; margin:5px; padding:10px; position: relative;">

{{ edtype.name }}

</div>

{% endfor %}

<div id="edtype\_all" class="clickable" style="display: inline; margin:5px; padding:10px; position: relative;">

Все

</div>

</div>

<div class='accordion'>

{% for item in directions %}

<h4 class="class\_edtype\_{{ item.educationtype.pk }}">{{ item.name|capfirst }}</h4>

<div class="data class\_edtype\_{{ item.educationtype.pk }}">

<a class="botton" href='/cathedra/direction/{{ item.pk }}'>Подробнее о направлении</a>

<b> Сокращение: {{ item.sname }}</b>

{% if item.educationtype %}

<br><b>Вид обучения: {{ item.educationtype.name }}</b>

{% endif %}

{% if item.cathedra %}

<br><b>Кафедра: {{ item.cathedra.name }}</b>

{% endif %}

{% if item.description %}

<br>{{ item.description|linebreaksbr }} <br>

{% endif %}

{% if item.subjects %}

<p>

Дисциплины профессионального блока:

{% for subject in item.subjects.all %}

<br> - {{ subject.name|capfirst }}

{% endfor %}

</p>

{% endif %}

</div>

{% empty %}

Не указаны направления.

{% endfor %}

</div>

{% endif %}

{% endblock %}

Листинг файла *normative\_document.djhtml*. Данный файл содержит описание шаблона пользовательского интерфейса страницы с отображением списка нормативных документов, относящихся к различным кафедрам факультета. Шаблон наследует *base.djhtml*.

{% extends "base.djhtml" %}

{% block scripts %}

{{ block.super }}

<script type="text/javascript">

$(document).ready(function(){

$(".accordion h4:first").addClass("active");

$(".accordion div.data").hide();

$(".accordion h4").click(function(){

$(this).next("div.data").slideToggle()

.siblings("div.data:visible").slideUp();

$(this).toggleClass("active");

$(this).siblings("h4").removeClass("active");

});

$(".educationtype").click(function(){

var id = this.id;

$("div.accordion").fadeOut("fast", function(){

$("div.accordion h4, div.data").hide();

$("div.accordion h4.class\_"+id).show();

$("div.accordion div.class\_"+id+":first").show();

}).fadeIn();

});

$("#edtype\_all").click(function(){

$("div.accordion").fadeOut("fast", function(){

$("div.accordion h4").show();

$("div.accordion div.data:not(first)").hide();

$("div.accordion div.data:first").show();

}).fadeIn();

});

});

</script>

{% endblock %}

{% block workspace %}

{% if directions|length == 1 %}

<div style="text-align: left; margin:10px; padding:10px;">

<b>Направление: {{ directions.0.name|capfirst }}</b>

<br />

<b>Сокращение: {{ directions.0.sname}}</b>

{% if directions.0.educationtype %}

<br>

<b>Вид обучения: {{ directions.0.educationtype.name }}</b>

{% endif %}

{% if directions.0.cathedra %}

<br>

<b>Кафедра: {{ directions.0.cathedra.name }} ({{directions.0.cathedra.sname}})</b>

{% endif %}

{% if directions.0.subjects %}

<br />Дисциплины:

{% for subject in directions.0.subjects.all %}

<br /> - {{ subject.name|capfirst }}

{% endfor %}

{% endif %}

{% if directions.0.post.body %}

<br /> {{ directions.0.post.body }}

{% endif %}

</div>

{% else %}

<div style="background: none; margin:10px; padding:10px; display:block;">

{% for edtype in edtypes %}

<div id="edtype\_{{ edtype.pk }}" class="educationtype clickable" style="display: inline; margin:5px; padding:10px; position: relative;">

{{ edtype.name }}

</div>

{% endfor %}

<div id="edtype\_all" class="clickable" style="display: inline; margin:5px; padding:10px; position: relative;">

Все

</div>

</div>

<div class='accordion'>

{% for item in directions %}

<h4 class="class\_edtype\_{{ item.educationtype.pk }}">{{ item.name|capfirst }}</h4>

<div class="data class\_edtype\_{{ item.educationtype.pk }}">

<a class="botton" href='/cathedra/direction/{{ item.pk }}'>Подробнее о направлении</a>

<b> Сокращение: {{ item.sname }}</b>

{% if item.educationtype %}

<br><b>Вид обучения: {{ item.educationtype.name }}</b>

{% endif %}

{% if item.cathedra %}

<br><b>Кафедра: {{ item.cathedra.name }}</b>

{% endif %}

{% if item.doctypes %}

<p>

<br /> <b>Нормативные документы:</b>

{% for ndoclist in item.doctypes %}

{% if ndoclist|length > 2%}

<br />{{ndoclist.1}}:

{% for ndoc in ndoclist%}

{% if forloop.counter0 >= 2 %}

{% if ndoc.docfile %}

<br /> - <a href="{{ ndoc.docfile.url }}">{{ndoc.title}}</a>

{% else %}

<br /> - {{ndoc.title}}

{% endif%}

{% endif %}

{% endfor %}

<br />

{% endif %}

{% endfor %}

</p>

{% endif %}

</div>

{% empty %}

Не указаны направления.

{% endfor %}

</div>

{% endif %}

{% endblock %}

Листинг файла *search.djhtml*. Данный файл содержит описание шаблона пользовательского интерфейса страницы поиска. Наследует шаблон *base.djhtml*.

{% extends 'base.djhtml' %}

{% block links %}

{{ block.super }}

<link rel="stylesheet" type="text/css" href="/static/css/widgets.css">

{% endblock %}

{% block scripts %}

{{ block.super }}

<!--

<script type="text/javascript" src="/static/cp\_adm/js/jsi18n.js"></script>

<script type="text/javascript" src="/static/cp\_adm/js/core.js"></script>

<script type="text/javascript" src="/static/cp\_adm/js/jquery.js"></script>

<script type="text/javascript" src="/static/cp\_adm/js/jquery.init.js"></script>

<script type="text/javascript" src="/static/cp\_adm/js/SelectBox.js"></script>

<script type="text/javascript" src="/static/cp\_adm/js/SelectFilter2.js"></script>

-->

<script ype="text/javascript">

$(document).ready(function(){

//$("select").css({"min-width":"100%", "max-width":"100%","min-height":"50px"});

$("select").each(function(){

var divtmp = $("<div class='div\_select'></div>");

var ultmp = $("<ul class='ul\_select'></ul>");

divtmp.css({"width":"100%", "height":"100px", "background":"#ffffff", "border":"1px solid black"});

$(this).after(divtmp);

$(divtmp).append(ultmp);

$(this).children().each(function(){

var curOption = $(this);

var cur = this;

var v = curOption.val();

var t = curOption.text();

var litmp = $("<li value='"+v+"'>"+t+"</li>");

if(cur.selected)

$(litmp).addClass("li\_select\_checked");

else

$(litmp).addClass("li\_select");

$(ultmp).append(litmp);

litmp.click(function(){

cur.selected = !cur.selected;

var curli = $(this);

curli.removeClass();

if(cur.selected)

curli.addClass("li\_select\_checked");

else

curli.addClass("li\_select");

});

})

}).hide();

$(".accordion h4:first").addClass("active");

$(".accordion p").hide();

$(".accordion h4").click(function(){

$(this).next("p").slideToggle("slow").siblings("p:visible").slideUp("slow");

$(this).toggleClass("active");

$(this).siblings("h4").removeClass("active");

});

});

</script>

{% endblock %}

{% block workspace %}

<center>

<form id='search\_book' action='' method='get'>

{% include "search\_book.djhtml"%}

</form>

</center>

<br>

{% if books %}

{% include "book\_preview.djhtml"%}

{% endif %}

{% endblock %}

Листинг файла *search\_book.djhtml*. Данный файл содержит описание визуального оформления формы поиска учебной литературы.

<table class='search\_book'>

<tr>

<th><label>{{sform.title.label}}:</label></th> <td>{{ sform.title}}</td>

<th width='20px'></th>

<th><label>{{sform.authors.label}}:</label></th> <td>{{ sform.authors}}</td>

</tr>

<tr height='10px'></tr>

<tr>

<th><label>{{sform.btype.label}}: </label></th> <td>{{ sform.btype }}</td>

<th width='20px'></th>

<th><label>{{sform.subjects.label}}:</label></th> <td>{{ sform.subjects }}</td>

</tr>

<tr height='20px'></tr>

<tr><td colspan='5' align='right'><input id='sub0' class='submit' type='submit' value='Найти'></td></tr>

</table>

Листинг файла *book\_preview.djhtml*. Файл содержит описание визуального представления информации о книге.

<div class='accordion'>

{% for book in books %}

<h4>

{% if book.booktype.image %}

<img src="{{ book.booktype.image.url }}" alt='Нет картинки'>

{% else %}

<img src="/media/preview/document.png" alt='Нет картинки'>

{% endif %}

{% if book.bookfile %}

<a class="botton" href="{{ book.bookfile.url }}" title="{{ book.bookfile.size|filesizeformat }}">Файл</a>

{% endif%}

{{ book.title }}

</h4>

<p>

{% if book.authors %} Авторы: {{ book.authors }} <br>{% endif %}

{% if book.publisher %} Издатель: {{ book.publisher }} {{ book.place }} <br> {% endif %}

{% if book.year %} Год: {{ book.year }} <br> {% endif %}

{% if book.page %} Страниц: {{ book.page }} <br> {% endif %}

Тип: {{ book.booktype.name }} <br>

Дисциплины:

{% for s in book.subjects.all %}

{{ s.name }}{% if forloop.last %}.{% else %},{% endif %}

{% endfor %}

{% if book.description %}<br/>{{ book.description }} {% endif %}

</p>

{% endfor %}

</div>

Листинг файла *bell\_schedule.djhtml*. Данный файл содержит описание визуального представления расписания звонков.

<table class='bell\_schedule'>

<caption><b>Расписание звонков</b></caption>

{% for item in bellschedule %}

<tr>

<td>{{ item.num }}</td> <td>{{ item.timestart|time:'H:i' }}</td><td>{{ item.timeend|time:'H:i' }}</td>

</tr>

{% empty %}

в разработке

{% endfor %}

</table>

Листинг файла *news.djhtml*. Данный файл содержит описание шаблона визуального представления различных новостей. Наследует шаблон *base.djhtml*.

{% extends "base.djhtml" %}

{% block workspace %}

<div class="print\_news" style="width:49%; float:left;">

{% for item in news %}

{% if forloop.counter0|divisibleby:"2" %}

<p>

<b>{{ item.title|upper }}</b>

<br/>

{{ item.body|linebreaksbr|capfirst }}

</p>

{% endif %}

{% empty %}

<p> Новостей нет. 8)</p>

{% endfor %}

</div>

<div class="print\_news" style="width:49%; float:right;">

{% for item in news %}

{% if forloop.counter|divisibleby:"2" %}

<p>

<b>{{ item.title|upper }}</b>

<br/>

{{ item.body|linebreaksbr|capfirst }}

</p>

{% endif %}

{% empty %}

<p> Новостей нет. 8)</p>

{% endfor %}

</div>

{% endblock %}