

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

02.03.03 Математическое обеспечение и администрирование
информационных систем

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

БАКАЛАВРСКАЯ РАБОТА

на тему Разработка веб-сервисов для проведения онлайн конференций и
повышения цитируемости статей

Студент	_____	_____
Руководитель	_____	_____

Допустить к защите
Заведующий кафедрой, к.тех.н, доцент, А.В. Очеповский _____

« _____ » _____ 2016 г.

Тольятти 2016

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

УТВЕРЖДАЮ
Зав.кафедрой «Прикладная
математика и информатика»
А.В.Очеповский

«___» _____ 2016 г.

ЗАДАНИЕ

на выполнение бакалаврской работы

Студент Калмыков Андрей Юрьевич

1. Тема Разработка веб-сервисов для проведения онлайн конференций и повышения цитируемости статей
2. Срок сдачи студентом законченной выпускной квалификационной работы 25 мая 2016 г.
3. Исходные данные к выпускной квалификационной работе: скриптовый язык общего назначения PHP, корпоративная информационная система Битрикс, система управления базами данных MySQL
4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов): анализ состояния вопроса; разработка веб-сервиса для проведения онлайн конференции; разработка веб-сервиса повышения цитируемости статей; апробация предложенных решений; выводы по работе.
5. Ориентировочный перечень графического и иллюстративного материала Блок-схемы поясняющие, работу веб-сервисов; изображения с демонстрацией работы веб-сервисов; графики и диаграммы, поясняющие результат работы.
6. Дата выдачи задания « 11 » января 2016 г.

Заказчик, Проректор по
научно-инновационной
деятельности, к.тех.н.

С.Х. Петерайтис

Руководитель выпускной
квалификационной работы

В.С. Климов

Задание принял к исполнению

А.Ю. Калмыков

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

УТВЕРЖДАЮ
Зав.кафедрой «Прикладная
математика и информатика»
_____ А.В.Очеповский

« ____ » _____ 2016 г.

КАЛЕНДАРНЫЙ ПЛАН
выполнения бакалаврской работы

Студента Калмыков Андрей Юрьевич
по теме Разработка веб-сервисов для проведения онлайн конференций и
повышения цитируемости статей.

Наименование раздела работы	Плановый срок выполнения раздела	Фактический срок выполнения раздела	Отметка о выполнении	Подпись руководителя
1. Анализ состояния вопроса	1.04.2016	1.04.2016	Выполнено	
2. Разработка веб-сервиса для проведения онлайн конференции	11.04.2016	11.04.2016	Выполнено	
3. Разработка веб-сервиса повышения цитирования статей	25.04.2016	25.04.2016	Выполнено	
4. Апробация предложенных решений	9.05.2016	9.05.2016	Выполнено	
Оформление пояснительной записки	12.05.2016	12.05.2016	Выполнено	
Подготовка доклада и графического	20.05.2016 – 20.06.2016	20.05.2016 – 20.06.2016	Выполнено	

материала к защите				
Проверка ВКР в системе «Антиплагиат.ВУЗ»	27.05.2016	27.05.2016	Выполнено	
Сдача пояснительной записки ВКР	20.06.2016	20.06.2016	Выполнено	

Руководитель выпускной
квалификационной работы

В.С. Климов

Задание принял к исполнению

А.Ю. Калмыков

Аннотация

Тема данной выпускной квалификационной работы: Разработка веб-сервисов для проведения онлайн конференций и повышения цитируемости статей.

Целью данной ВКР является разработка сервисов по организации и проведению веб-конференций, а так же сервиса для увеличения индекса Хирша для научных сотрудников ТГУ с помощью языка программирования PHP. Для достижения цели необходимо выполнить следующие задачи:

- Провести анализ состояния вопроса по организации научных конференций и размещения научных работ в сети Интернет для увеличения индекса Хирша.
- Разработать алгоритм по организации научных конференций и размещения научных работ в сети Интернет для увеличения индекса Хирша.
- Спроектировать, разработать и протестировать программную реализацию предложенных решений. И интегрировать их в существующий веб-сайт ТГУ на 1С “Bitrix”.

В ходе выполнения ВКР создан информационный ресурс в виде сайта, который позволяет:

- Вести регистрацию участников и организаторов конференций;
- Формировать отчеты по посещаемости сайта и конференций;
- Обеспечить легкий поиск по научным статьям и конференциям.
- Размещать текста научных работ в сети Интернет для повышения индекса Хирша.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка литературы и приложений.

Во введении описывается актуальность проводимого исследования, формируется цель, и ставятся задачи, которые необходимо решить для достижения цели.

В первой главе описывается анализ сферы деятельности для определения ключевых компонентов проведения видеоконференций и увеличение индекса Хирша.

Во второй главе проводится анализ и выбор программных средств для реализации продукта.

В третьей главе описываются этапы разработки и тестирования программного продукта.

В заключении приводятся основные выводы по работе, описываются достигнутые задачи в ходе выполнения ВКР.

Выпускная квалификационная работа содержит пояснительную записку объемом 44 страницы, включая 15 рисунков, 2 таблицы, список использованной литературы из 20 наименований и одного приложения.

Оглавление

Введение.....	3
1 Анализ предметной области	6
1.1 Общая характеристика процесса проведения научных конференций	6
1.2 Формирование требований к новой информационной системе.....	9
1.3 Порядок контроля и приёмки информационного ресурса.....	18
2 Моделирование и проектирование информационной системы	21
2.1 Макетирование web-сервисов.....	21
2.2 Создание дизайн-прототипа веб-сайта	23
2.3 Изучение технологий разработки web-сервисов	25
2.4 Разработка базы данных	29
3 Программная реализация веб-сервиса для проведения научных конференций и размещения научных работ.....	34
3.1 Выбор средств реализации.....	34
3.2 Базовые классы компонентов	36
Заключение	42
Список используемой литературы	43
Приложение А Результаты выполненной работы.....	45

Введение

По различным данным, около 80-85% информации человек воспринимает визуально, таким образом видеоконференцсвязь является одним из важнейших методов передачи информации человеку. Поэтому с этим применение видеоконференции в медицине, бизнесе, дистанционном обучении, системах безопасности и других областях приносит большую пользу.

Естественно, видеоконференции не заменяют вербального общения, но они помогают добиться нового уровня взаимодействия между людьми, разделёнными сотнями километров. Если верить огромному количеству исследований, на слух человек воспринимает только лишь одну десятую часть передаваемой информации. Но в случае, возможности следить за движениями и мимикой собеседника, процент восприятия информации достигает показателя в 80-85%. Компании, которые используют видеоконференции в обычной жизни, подчеркивают, что системы видеоконференций резко сокращают затраты на время и финансовые затраты компании на семинары, совещания, командировки их сотрудников и консультации.

Видеоконференцсвязь - это та самая технология, которая позволяет людям видеть и слышать друг друга, обмениваться информацией и совместно использовать её в интерактивном режиме, используя возможности привычного всем персонального компьютера, максимально приближая общение на расстоянии к живому общению.

Так же актуальной проблемой для научных работников является сложность в размещении собственных научных публикаций в сети Интернет. По той простой причине, что многие сервисы подобной тематики взимают плату за размещение научных работ на своих сайтах.

Тем не менее на сегодняшний день в наукометрии используется три главных показателя:

1. Индекс Хирша.
2. Индекс цитирования.
3. Импакт-фактор.

Объектом исследования являются сервисы для проведения веб-конференций включающие разнообразные возможности и инструменты, а так же сервисы по увеличению индекса Хирша.

Целью данной ВКР является разработка сервисов по организации и проведению веб-конференций, а так же сервиса для увеличения индекса Хирша для научных сотрудников ТГУ с помощью языка программирования PHP. Для достижения цели необходимо выполнить следующие задачи:

- провести анализ состояния вопроса по организации научных конференций и размещения научных работ в сети Интернет для увеличения индекса Хирша;
- разработать алгоритм по организации научных конференций и размещения научных работ в сети Интернет для увеличения индекса Хирша;
- спроектировать, разработать и протестировать программную реализацию предложенных решений. И интегрировать их в существующий веб-сайт ТГУ на 1С “Bitrix”.

В ходе выполнения ВКР будет создан информационный ресурс в виде сайта, который позволит:

- вести регистрацию участников и организаторов конференций;
- формировать отчеты по посещаемости сайта и конференций;
- размещать материалы научных статей на сайте;
- обеспечить легкий поиск по научным статьям и конференциям;
- размещать текста научных работ в сети Интернет для повышения индекса Хирша.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка литературы и приложений.

Во введении описывается актуальность проводимого исследования, формируется цель, и ставятся задачи, которые необходимо решить для достижения цели.

В первой главе описывается анализ сферы деятельности для определения ключевых компонентов проведения видеоконференций и увеличение индекса Хирша.

Во второй главе проводится анализ и выбор программных средств для реализации продукта.

В третьей главе описываются этапы разработки и тестирования программного продукта.

В заключении приводятся основные выводы по работе, описываются достигнутые задачи в ходе выполнения ВКР.

1 Анализ предметной области

1.1 Общая характеристика процесса проведения научных конференций

Успех проведения конференции в большой мере зависит от технической поддержки мероприятия. Технические проблемы обычно приводят к срыву мероприятия. При этом любое мероприятие имеют особые требования, отклонение от которых в некоторых случаях допустимо, но нежелательно. Таким образом, если в каком-либо мероприятии всё рассчитано почти по минутам, а при проведении возникли неожиданные проблемы, то временные рамки придется все же немного нарушить.

Каждое мероприятие имеет свой план проведения, который подразумевает логичность и порядок действий, направленных на построение мероприятия или конференции в целом. Любой план мероприятия как правило, состоит из пунктов, каждый из которых требует выполнения определенных обязанностей. Чтобы планируемая конференция прошла на должном уровне, необходимо проводить соответствующую подготовку. Вся работа в данном направлении должна быть выполнена четко и проконтролирована. Главным условием является ясное распределение функций и обязанностей между участниками этого процесса. Главное лицо или группа лиц должны относиться к своим функциям с особым вниманием, ответственностью и с пониманием дела.

Можно выделить следующие этапы при организации проведения мероприятия как вида деятельности:

Список участников, как правило, осуществляется либо сотрудником, либо группой лиц ответственными за проведение мероприятия, по согласованию с организатором мероприятия.

Приглашение и оповещение будущих участников мероприятия зависят от вида мероприятия и его масштабности. В одном случае это могут быть адресные приглашения в форме письма-приглашения или пригласительного

билета, в других – рекламные, коммерческие или информационные письма, текст которых строится по схеме: кто приглашает – куда приглашает – когда и где состоится – порядок участия – приглашение к участию.

Регистрация участников проводится непосредственно перед началом содержательной части мероприятия, а её форма зависит от вида мероприятия и его масштабности. На многих событиях, особенно массовых, регистрация участников не предусматривается.

Для регистрации участников события регулярного типа используются так называемые явочные (регистрационные) листы, на которых указывается порядковый номер, фамилия и инициалы участника и оставляется место для подписи. Цель проведения регистрации участников регулярных событий – зафиксировать наличие кворума, то есть необходимого по регламенту количества участников.

При проведении единичных событий также используется регистрация, но цель её – в том, что бы создать возможность в дальнейшем вести справочно-информационную работу. Таким образом для регистрации участников используются формы для регистрации, в которых, отражается следующая информация:

- номер участника;
- ФИО;
- должность и место работы;
- контакты;
- примечание.

Регистрация, как правило, проводится одним лицом (преимущественно во время нерегулярных событий) или группой лиц. Во втором случае используются разные параметры для регистрации, например:

- в алфавитном порядке по фамилиям;
- по группам учреждений-участников;
- по регионам и т.д.

Перед тем как началась регистрация заранее устанавливаются столы с табличками, на которых написаны признаки группировки: буквы алфавита, наименования учреждений, регионы и т.д.

В 2005 году американский физик Хорхе Хирш из университета Сан-Диего, Калифорния, предложил наукометрический показатель, который впоследствии назвали его именем – индексом Хирша. Эта количественная характеристика продуктивности ученого основана на количестве его публикаций и количестве цитирований этих публикаций. Ученый имеет индекс h , если он опубликовал h статей, на каждую из которых сослались как минимум h раз. То есть, если ученый имеет индекс Хирша равный 12 (что считается неплохим результатом), то у него есть как минимум 12 статей, каждая из которых имеет цитируемость 12 и выше. Индекс показывает, насколько деятельность данного исследователя заметна другим ученым в данной научной области, какое влияние она оказывает на развитие направления.

Индекс цитирования – это принятый в научном мире показатель «значимости» трудов какого-либо ученого. Представляет собой число ссылок на публикации ученого в реферируемых научных периодических изданиях. Наличие в научно-образовательных организациях ученых, обладающих высоким индексом, говорит о высокой эффективности и результативности деятельности организации в целом.

Импакт-фактор журнала — широко используемый формальный численный показатель важности научного журнала, разработанный еще в 60-х годах Институтом научной информации (ISI), и публикующийся в журнале «Journal Citation Report». Импакт-фактор показывает, сколько раз в среднем цитируется каждая опубликованная в журнале статья в течение двух последующих лет после выхода. Например, чтобы рассчитать импакт-фактор журнала «J» за 2010 год, нужно число цитирований в течение 2010 года статей, опубликованных в данном журнале в 2008-2009 гг., разделить на количество статей, опубликованных в данном журнале за тот же период (в 2008-2009 гг.).

1.2 Формирование требований к новой информационной системе

Формирование требований к разрабатываемым веб-сервисам являются одной из важных задач на начальных этапах разработки веб-сервиса. Требования – это возможности или условия, которым должен соответствовать сервис или проект. Главная задача этапа определения требований заключается в определении, обсуждении того, что действительно нужно в форме, понятной и посетителю веб-сервиса и членам команды разработчиков [1-5].

Требования к техническому обеспечению часто разделяют на функциональные и нефункциональные. К функциональным требованиям относят перечень функций, которые обязана выполнять система, причем должно быть указано, как система реагирует на те или иные входные данные, как она ведет в себя в определенных ситуациях и т.д. По функциональному назначению сайт подразделяется на подсистему, обеспечивающую хранение информации в базе данных, а также набор пользовательских и администраторских приложений.

Подсистема хранения данных должна обеспечивать надежное хранение всей необходимой для функционирования системы информации. Данная подсистема должны реализоваться средствами СУБД MYSQL и состоять из следующих компонентов:

- структура для хранения информации в виде реляционной базы данных (БД);
- стандартные средства СУБД контроля целостности данных, встроенные в систему PHPmyadmin.

Пользовательское приложение должно предоставлять пользователям информационного ресурса удобный интерфейс для работы в сайте. Приложение должно осуществлять проверку вводимых данных на корректность, обеспечивать логику и функциональность.

Администраторское приложение (код программы) предназначено для выполнения разработчиком служебных функций по сопровождению

информационного ресурса, таких как добавление новых модулей или изменение текущих к имеющимся в сайте функциям. Приложения должны функционировать на любой программно-аппаратной платформе с установленным веб-сервером Apache.

Нефункциональные требования отображает характеристики системы и ее окружения. Кроме того к ним относятся перечень ограничений, накладываемых на действия и процессы, выполняемые системой. Многие нефункциональные требования применяются к системе в целом, а не к отдельным ее компонентам. Это означает, что они более значимы и критичны, чем отдельные функциональные требования. Ошибка, допущенная в функциональном требовании, может ухудшить качество системы, ошибка в нефункциональных требованиях может сделать систему неработоспособной.

Модель требований FURPS+ [1] делит предъявляемые требования на несколько категорий:

1. Функциональные требования – возможности, свойства, безопасность.
2. Нефункциональные требования:
 - удобство – справочная система, человеческий фактор, документация;
 - надежность – возможность восстановления и предсказуемость поведения, частота сбоев;
 - производительность – точность, доступность, время отклика, использование ресурсов;
 - возможность поддержки – возможность поддержки соответствие международным стандартам, адаптивность, возможность конфигурирования;
 - реализация – языки и средства, требования к ресурсам, аппаратное обеспечение;
 - интерфейс – ограничения накладываемые необходимостью взаимодействия с внешними системами;
 - операции – управление системой и ее параметры.

Категории FURPS+ нужно использовать при описании требований, чтобы не забыть важные аспекты жизнедеятельности системы. Некоторые из подобных требований (удобство, надежность, производительность и возможность поддержки) называются атрибутами качества (quality attributes).

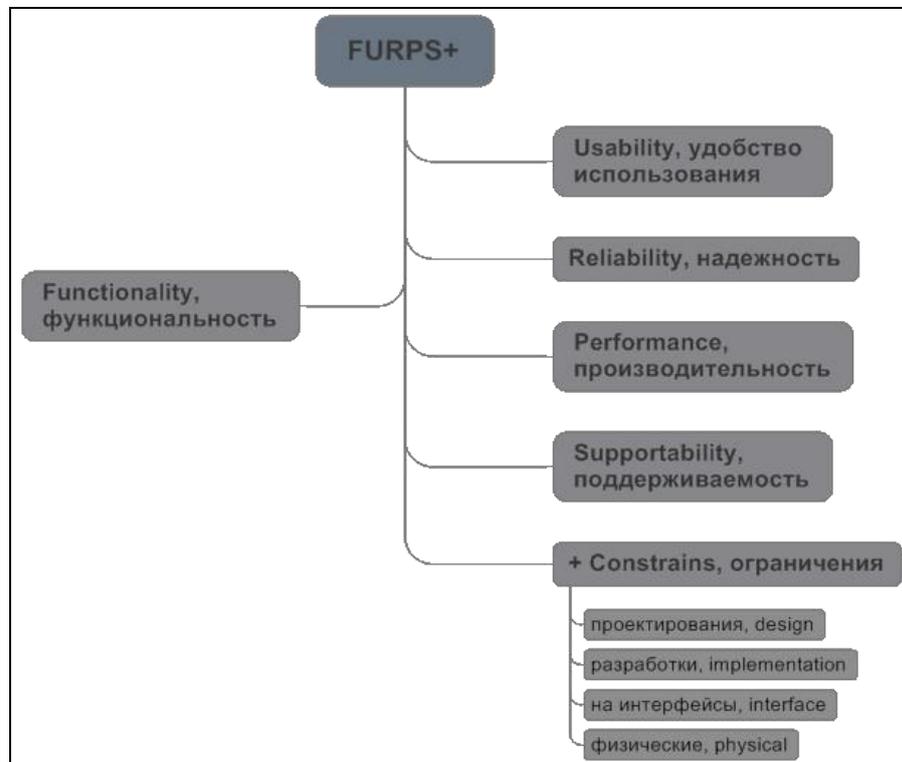


Рисунок 1.1 – Классификация требований FURPS+

Для выявления требований существуют следующие методы:

- определение требований на основе точек зрения (Viewpoint-Oriented Requirements Definition (VORD));
- этнографический подход;
- варианты использования.

В унифицированном процессе функциональные требования представляются с помощью вариантов использования – текстового описания взаимодействия пользователя с системой и диаграмм использования, позволяющие кратко выразить основные взаимодействия в графическом виде [6-10].

Сформулируем основные функциональные и нефункциональные требования к разрабатываемой информационной системе по проведению научных конференций на базе сайта ТГУ.

Наш анализ показывает, что весь набор критериев оценки может быть разделен на четыре основные группы:

- критерии, относящиеся к вводу и валидации данных;
- критерии, относящиеся к обработке и хранению данных;
- критерии, относящиеся к администрированию системы.

К функциональным требованиям информационной системы на этапе ввода и валидации данных можно отнести:

- возможность автоматической проверки введенных данных на предмет их корректности, например, у создаваемой научной конференции обязательно должно быть название и дата проведения;
- возможность автоматической валидации данных.

К функциональным требованиям информационной системы на этапе обработки и хранения данных можно отнести:

- накопление статистических данных по проведенным научным конференциям;
- поддержка целостности данных;
- применение многомерной модели хранения данных;
- возможность статистической обработки данных.

К функциональным требованиям информационной системы на этапе администрирования можно отнести:

- использование одной из моделей разграничения прав доступа пользователей к информационной системе;
- шифрование данных;
- защита данных от несанкционированного доступа;
- защита персональных данных;

- создание и редактирование информации о пользователях информационной системы;
- возможность регулярного создания резервных копий данных.

К нефункциональным требованиям информационной системы можно отнести:

- возможность удалённого доступа к системе через Internet;
- использование единой технологической платформы;
- построение информационной системы на базе свободного программного обеспечения;
- построение информационной системы на принципах открытой системы;
- построение информационной системы на базе сервис-ориентированной архитектуры;
- возможность использование технологии хранилищ данных;
- возможность беспроводного мобильного доступа к информационной системе;
- возможность доступа к данным через веб-интерфейс.

Ускорение научно-технического прогресса предъявляет все более высокие требования к принятию решений достаточно быстро.

Условиями успешного овладения техническими знаниями является умение работать с различными пользовательскими компьютерными программами и различным техническим периферийными устройствами.

Поэтому разрабатываемый сайт дает большие преимущества при его использовании.

Сайт выполняет следующие функции:

- предоставление информации по проводимым конференциям;
- регистрация пользователей для участия в конференциях;
- регистрация организаторов конференций.

Целями разработки и внедрения информационного ресурса являются:

- наглядность и простота;

- простой и удобный интерфейс пользователя;
- возможность вносить изменения в отдельные модули.

По функциональному назначению сайт подразделяется на подсистему, обеспечивающую хранение информации в базе данных, а также набор пользовательских и администраторских приложений.

Подсистема хранения данных должна обеспечивать надежное хранение всей необходимой для функционирования системы информации. Данная подсистема должны реализоваться средствами СУБД MYSQL и состоять из следующих компонентов:

- структура для хранения информации в виде реляционной базы данных (БД);
- стандартные средства СУБД контроля целостности данных, встроенные в систему PHPmyadmin.

Пользовательское приложение должно предоставлять пользователям информационного ресурса удобный интерфейс для работы в сайте. Приложение должно осуществлять проверку вводимых данных на корректность, обеспечивать логику и функциональность.

Администраторское приложение (код программы) предназначено для выполнения разработчиком служебных функций по сопровождению информационного ресурса, таких как добавление новых модулей или изменение текущих к имеющимся в сайте функциям. Приложения должны функционировать на любой программно-аппаратной платформе с установленным веб-сервером Apache.

Штатный режим функционирования определяется в соответствии с правилами внутреннего трудового распорядка, принятого в организации, с учётом возможности предоставления сотрудникам доступа к работе в системе во внеурочное время, праздничные и выходные дни по производственной необходимости, за исключением часов, отведенных для технического обслуживания, с обязательным предварительным оповещением пользователей.

Техническое обслуживание программно-аппаратных средств информационного ресурса должно осуществляться:

в рабочие дни: с 07:00 до 08:00, с 12:00 до 13:00 и с 16:30 до 17:30;

в выходные дни: не регламентировано.

К сайту предъявляются следующие требования к обеспечению надёжности:

Сайт должен отслеживать исключительные ситуации всех входящих в состав подсистем, возникающие в процессе выполнения программного кода и нарушающие его работу.

Надёжность работы информационного ресурса в процессе его эксплуатации должен быть обеспечен пользователем, который его эксплуатирует на основе принятых правил и норм по обслуживанию программного обеспечения и компьютерной техники.

Информационно-программное обеспечение должно быть унифицировано по структуре, алгоритмам, средствам взаимодействия с пользователем, технологии обработки информации, принятым у пользователя.

При разработке интерфейса информационного ресурса необходимо соблюдать следующие требования:

- оптимальным разрешением экрана монитора для отображения форм системы является разрешение 1280x1024 и более;
- интерфейс должен быть рассчитан на преимущественное использование манипулятора типа «мышь», то есть управление системой должно осуществляться с помощью набора экранных меню, кнопок, значков и т. п. элементов. Клавиатурный режим ввода должен использоваться главным образом при заполнении и/или редактировании текстовых и числовых полей экранных форм;
- все надписи экранных форм, а также сообщения и подсказки, выдаваемые пользователю в процессе нормального функционирования системы должны быть на русском языке;

- использование ярких цветов допускается только для выделения элементов, требующих привлечения внимания пользователя (предупреждения, сообщения, критически важная информация и т.п.);
- элементы ввода данных, выбора и управления должны располагаться последовательно в порядке их заполнения слева направо и/или сверху вниз;
- не допускается сочетание цветов текста и фона, плохо различимых друг на друге.

Требования к защите информации от несанкционированного доступа следующие. Аутентификация пользователей при входе в подсистему требуется, а аудит системных событий не требуется.

Требования к квалификации пользователей информационного ресурса следующие. Пользователи информационного ресурса должны иметь базовые навыки работы с компьютером. Сотрудники, являющиеся пользователями информационного ресурса, должны быть ознакомлены с эксплуатационной документацией на сайт (руководством пользователя) и руководящими документами, регламентирующими работу пользователей в локально-вычислительной сети организации.

Требования к функциям информационного ресурса:

Применение создаваемого информационного ресурса должно улучшить и упростить процесс работы пользователя за счёт реализации следующих функций:

- добавление, редактирование информации о проводящихся конференциях;
- регистрация новых пользователей;
- участие в видеоконференциях.

Названные функции должны быть реализованы набором web-страниц, реализующее пользовательский интерфейс, функциональность. Сайт в качестве

хранилища информации должен использовать БД со своей подсистемой хранения.

Возможности ролей системы:

- Участник – возможность просмотра различной информации.
- Докладчик – возможность просмотра различной информации, регистрироваться для выступления на конференции.
- Организатор – просмотр, редактирование, поиск, удаление, добавление информации. Отметка и удаление информации.
- Администратор – полный доступ ко всем частям сайта.

Требования к информационному обеспечению информационного ресурса:

Общие принципы, используемые при построении информационного ресурса:

- структуризация данных;
- комплексное представление информации.

Первый принцип должен быть обеспечен за счёт использования реляционной БД для хранения информации.

Второй принцип должен быть обеспечен за счет реализации пользовательского интерфейса на основе среды программирования PHP, HTML, CSS.

Требования к лингвистическому обеспечению:

Сайт должен быть разработан с использованием технологий построения клиентских систем. В качестве основного языка программирования должен быть использован язык PHP.

Требования к архитектуре информационного ресурса:

Основой архитектуры системы должно быть трехуровневое клиентское приложение – администратор, обычный пользователь, зарегистрированный пользователь.

Требования к программно-аппаратному обеспечению информационного ресурса:

Требования к программному обеспечению пользовательской рабочей станции:

- операционная система Microsoft Windows 7 и выше;
- браузер с поддержкой HTML5 и CSS3.

Обеспечения устойчивого функционирования информационного ресурса из-за отсутствия каких либо файлов, производится при помощи проверки в теле программы на отсутствие в директории с программой. Если какой-то файл отсутствует, то появляется сообщение об ошибке.

Контроль входной информации сведен к минимуму информации, передаваемой в программу, что уменьшает степень возникновения некорректной работы информационного ресурса.

Контроль выходной информации осуществляется путем отслеживания за содержимым файлов.

Минимальные системные требования, необходимые для работы информационного ресурса:

- объем ОЗУ не менее 256 Мб;
- объем жесткого диска не менее 900 Мб;
- микропроцессор с тактовой частотой не менее 1 ГГц.

Рекомендуемые системные требования, необходимые для работы информационного ресурса:

- объем ОЗУ не менее 512 Мб;
- объем жесткого диска не менее 20 Гб;
- микропроцессор с тактовой частотой не менее 1.5 ГГц.

1.3 Порядок контроля и приёмки информационного ресурса

Контроль и приёмка информационного ресурса осуществляются в соответствии с этапами разработки информационного ресурса и планом внедрения информационного ресурса.

Программный продукт, полученный в ходе процесса кодирования, отладки и тестирования должен соответствовать настоящему ТЗ. С целью

проверки соответствия информационного ресурса требованиям ТЗ проводятся приемочные испытания в порядке, определенном требованиями подраздела 4 ГОСТ 34.603 – 92 [11-15].

Перечень основных событий, которые следует выполнить при подготовке к вводу информационного ресурса в действие:

1) Подготовка аппаратно-программного комплекса для работы информационного ресурса, соответствующего требованиям, указанным в разделе 2.11.3.4 настоящего документа.

2) Генерация объектов БД.

3) Публикация системы на сервере, проверка работоспособности.

4) Ознакомление пользователей с эксплуатационной документацией.

5) Тестовая эксплуатация информационного.

6) Промышленная эксплуатация информационного ресурса.

Источники разработки информационного ресурса и нормативные документы представлены в таблице 1.

Таблица 1 – Перечень нормативных документов

Наименование документа	Год
ГОСТ 34.601 – 90. Автоматизированные системы. Стадии создания	1990
ГОСТ 34.602 – 89. Техническое задание на создание автоматизированной системы	1989
ГОСТ 34.201 – 89. Виды, комплектность и обозначения документов при создании автоматизированных систем	1989
ГОСТ 24.104 – 85. Автоматизированные системы управления. Общие требования	1985
ГОСТ Р 51275-99. Защита информации. Объект информации. Факторы, воздействующие на информацию. Общие положения	1999
ГОСТ Р ИСО/МЭК 12207-99. Информационная технология. Процессы жизненного цикла программных средств.	1999

Объектом автоматизации является процесс по организации проведения научных конференций на основе портала ТГУ.

Человек, который захотел провести свою конференцию с помощью компьютера, на сегодняшний день может выбрать любую площадку по проведению конференций. Только для начала стоит определиться, для каких конкретно задач и как часто будет использоваться эта площадка. Из различных вариантов подобных сервисов, действительно, не так легко найти оптимальный и наилучший вариант.

В результате анализа предметной области по организации научных конференций, выявления основных недостатков и нюансов при проведении научных конференций было принято решение, что ТГУ в настоящее время необходима автоматизация процесса по проведению научных конференций.

Проведено исследование и обоснование выбора средств разработки информационного ресурса. Определены основные требования к информационному ресурсу.

2 Моделирование и проектирование информационной системы

2.1 Макетирование web-сервисов

Схематичная модель сайта является одним из основных эскизных проектов, которые создаются перед тем как изготовить оригинал-макет сайта, и используется при разработке дизайна сайтов, чтобы показать структуру сайта, настраиваемые пути пользователя, наиболее важные элементы интерфейса пользователя, их положение и взаимосвязь между страницами сайта. Каркасы страниц сайта отображают в черно-белом цвете наиболее важные элементы интерфейса, такие как заголовок и нижний колонтитул сайта, форма контактов, навигация.

Каркасное моделирование — это процесс, который может значительно сократить время, необходимое для проектирования и разработки, устраняя потенциальные визуальные отвлекающие факторы и фокусируя внимание разработчиков проекта на базовой функциональности и стратегических факторах маркетинга [16-18].

В процессе анализа существующих систем для прототипирования, был выбран онлайн-сервис MockFlow. MockFlow — это проектные, объединенные (в реальном времени) модели интерфейсов пользователя для вашего программного обеспечения и веб-сайтов.

Перед тем как разрабатывать любую систему необходимо иметь план, для веб-сайтов этим планом является структура сайта. Структурой сайта является древовидное иерархическое представление всех элементов сайта, включая информационные разделы и их подразделы. Другими словами – это заготовка веб-сайта, которая впоследствии наполняется текстовой и графической информацией. Структура веб-сайта условно можно разделить на внутреннюю и внешнюю структуру.

Внутренняя структура включает в себя расположение страниц сайта относительно перехода с главной страницы. При разработке внутренней

структуры сайта можно провести аналогию с оглавлением книги с различными повестями, в которой отображаются все главы и разделы, по которым пользователь может переходить к тому или иному произведению. Внутренняя структура сайта разрабатывается на этапе проектирования сайта.

Внешняя структура представляет собой уже более подробную блок-схему сайта с отображением отдельных элементов сайта на каждой его странице. Разработка внешней структуры веб-сайта чаще всего происходит на этапе создания макета сайта.

При создании веб-сайта для агентства недвижимости было решено выбрать стандартную схему размещения элементов на странице. В эту схему входят:

- шапка – представляет собой верхушку сайта, чаще всего содержащую в себе навигационную панель, форму авторизации и приветствие. Шапка веб-сайта может быть как статической относительно содержания страницы, так и фиксированной – всегда находится вверху экрана, независимо от прокрутки страницы;
- содержание – отображает в себе основную информацию, представленную на данной странице, обычно представлен материал, идущий сплошным текстом с графическими элементами или списком материалов. Находится в центральной части страницы;
- боковая панель – может содержать в себе какие-либо элементы управления, навигации или же просто информационные или рекламные блоки. Может находиться как с одной стороны веб-страницы, так и по обеим его сторонам;
- подвал сайта – в нем обычно отображают создателей сайта, небольшую информацию, например, адрес агентства. Так же как и шапка сайта может быть зафиксированной внизу экрана.

2.2 Создание дизайн-прототипа веб-сайта

Разработка дизайна сайта имеет большую значимость, так как дизайн первое, что замечает пользователь.

Существует такая фраза: «Экономить на дизайне не стоит». Это правда, так как сайт «встречают по одежке» и если дизайн сайта будет раздражающим, небрежным или невразумительным, то такой сайт часть посетителей просто не захочет смотреть и сразу закроет.

На данном этапе определяется внешний вид сайта, какая структура будет использована, и какие элементы будут на страницах сайта. А так же цветовая гамма, для разработки сайта.

Для сайта лучше всего подходит инфо-дизайн. Этот термин обозначает максимальную информативность элементов сайта, привязанную к тематике, основной идеи сайта. А так же удобство представления и восприятия основной информации.

После постановки задач для разработки будущего веб-сайта, наступает следующий этап — разработка дизайн-макета. Ориентируясь по пунктам ТЗ разрабатывается будущий шаблон, или шаблоны, из которых можно выбрать нужный вариант. На этом этапе будет определено, то как будет дизайн веб-сайта выглядеть в целом, то какие элементы будут использованы в виде графики, а так же то какая структура у страниц будет в целом и то какие цветовые решения будут использоваться на сайте для пример смотреть рисунок 2.1.

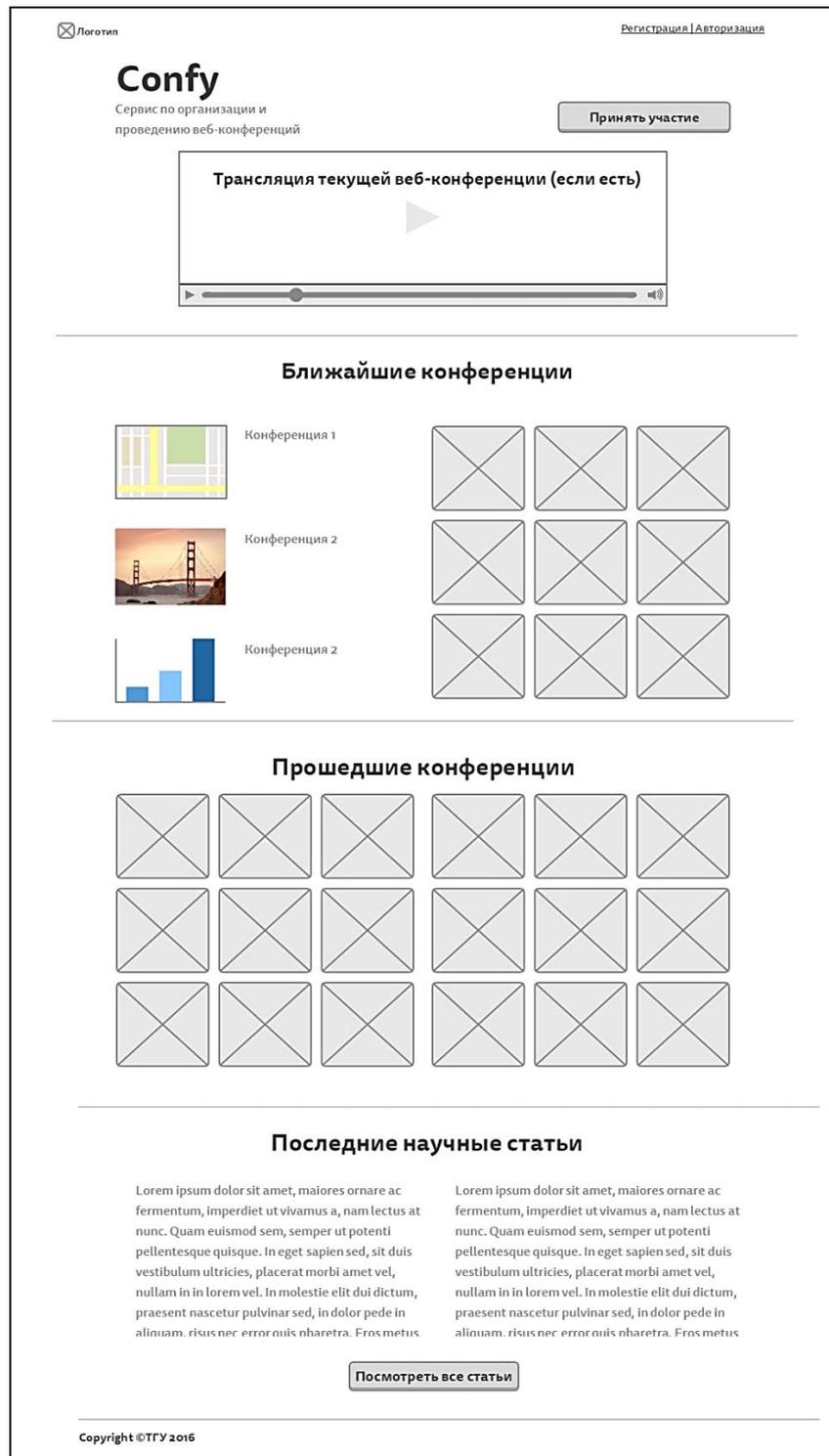


Рисунок 2.1 – Первая версия макета главной страницы

После чего предоставляется в формате PSD, программы Adobe Photoshop, в виде картинке. Разработка дизайна веб-сайта как правило занимает от нескольких дней, до нескольких недель, очень трудоёмкие работы могут проводиться даже месяц.

2.3 Изучение технологий разработки Web-сервисов

Разработку любого веб-сервиса или веб-сайта можно разделить на несколько этапов:

Первый этап – верстка сайта. После этапа создания дизайн-прототипа, и после того как Заказчик утвердил дизайн-прототип (в письменном виде или по электронной почте) за работу принимается верстальщик — это такой человек, который переводит дизайн-прототип на язык, понятный браузеру с использованием языка разметки HTML.

Верстка — это процесс написания CSS и HTML кода для веб-страниц. Каждый код отвечает за то, чтобы все располагаемые элементы на странице размещались там, где нужно. Верстальщик обязан знать все особенности браузеров, так как для каждого браузера необходим свой подход верстки.

Существуют всего два основных подхода к построению структуры HTML документа — в основе первого лежат таблицы, при помощи которых происходит позиционирование всех элементов страницы, в основе второго — дивы и стили, которые их упорядочивают.

Изначально, все сайты оформлялись при помощи таблиц. Таким образом, данный подход более старый и устоявшийся. Он намного понятнее и легче читается, благодаря чему всё ещё используется на большинстве. Однако, он гораздо сложнее в корректировании и дополнении. Если необходимо что-то добавить на страницу, то придётся тщательно разбираться во всей её структуре.

Вторым, а быть может и первым по важности, недостатком табличной вёрстки является большой объём ненужной информации, которая увеличивает вес загружаемой страницы. Это конечно не слишком важно в условиях скоростного интернета, который становится стандартом на сегодняшний день, однако, оптимизированные по объёму сайты, даже поисковыми системами ранжируются немного выше конкурентов.

Ну и в завершение последний недостаток этого типа разметки — «подтормаживайте» во время загрузки. Всё дело в том, что пока не загрузятся

все вложенные в таблицу элементы, таблица не прорисовывается в браузере. Это не может вызывать удивление у посетителей страницы.

Блочная верстка лишена перечисленных выше недостатков табличного подхода. Но в сравнении с ним, блочная верстка более сложная в создании. Гораздо проще разместить на странице таблицу и разместить в её ячейках нужные элементы, чем позиционировать при помощи CSS. Основные стили описаны в таблице 2.

Таблица 2 – Таблица основных стилей CSS

Описание	Стили
Ширина	width
Высота	height
Обтекаемость	float
Цвет текста	color
Выравнивание текста внутри блока	text-align
Отступ внутри блока	padding
Отступы от границ блока	margin
Цвет фона	background-color

В данной таблице, Таблица 2, описаны далеко не все стили, а только самые распространенные при работе с CSS. Названия стилей просты для понимания и применения.

Второй этап – программирование сайта. Довольно часто этап программирования и этап создания верстки объединяют в один. На небольших проектах оба действия в состоянии выполнить один и тот человек. На крупных проектах в силу сложности работ эти этапы разделяют.

На этапе программирования происходит создание всех страниц сайта, определяется порядок работы меню, расставляются гипер-ссылки, создается динамика на сайте, программируются такие составляющие, как гостевая книга, форум, новостная лента и проч.

Если сайт должен иметь администраторский интерфейс, то он создается именно на этапе программирования.

Очень важно на этапе программирования определить, на какой системе администрирования будет работать сайт. На сегодня на рынке множество систем администрирования.

В результате работы верстальщика и программиста получается сайт без информационного наполнения. Физически сайт в таком виде представляет из себя набор файлов.

Сроки выполнения работ по верстке и программированию зависят от сложности проекта и могут составлять от недели до 2-х месяцев.

На этом этапе идет разработка механизмов системы администрирования сайта, которая в будущем позволит сотрудникам компании менять, обновлять информацию на сайте. То есть создается рабочая версия сайта, готовая к наполнению текстов и графических материалов.

Язык программирования представляет собой знаковую систему, которая предназначена для описания алгоритмов. Одни из самых популярных языков программирования: Python, C++, Java, PHP, Perl и др.

Третий этап – наполнение сайта информацией. На данном этапе информация, которая была предоставлена заказчиком, размещается на сайте, т.е. путем перевода в специальный формат текст и графика располагаются на сайте на определенных страницах, и эта информация становится доступной для просмотра.

Срок исполнения так же зависит от трудности проекта, объема информации, который нужно будет расположить на сайте, и того вида, в котором ее представил Заказчик. Если информация была представлена в электронном виде, а графика не требует дополнительной обработки, то наполнение сайта происходит достаточно быстро. Если необходима дополнительная работа, как например, поиск или набор текста, сканирование фотографий или создание рисунков, то этап наполнения сайта может стать одним из самых длительных.

Эффективность работы сайта связана с понятием контента, то есть текстовой и графической информации. Иногда материалы для заполнения есть у заказчика, если нет, то он обращается к фрилансерам, для того, чтобы они написали необходимые статьи.

Четвертый этап – расположение сайта в сети Интернет. Данный этап обязательно выполняется после проведения всех вышеперечисленных работ. Он может проводиться параллельно с любым из этапов. Он заключается в том, что файлы сайта располагаются на хостинге.

Пятый этап – тестирование сайта. Данный этап можно осуществить как до, так и после размещения сайта по его основному адресу. На данном этапе выявляются все недочеты и ошибки в программировании и контенте. Срок тестирования зависит от трудности проекта, но, как правило, не превышает 1 месяца.

Шестой этап – раскрутка сайта. Полностью готовый и наполненный информацией сайт не будет гарантировать приток пользователей. Для того, чтобы на ваш сайт заходили посетители, о нем в первую очередь надо заявить. Большинство посетителей заходят на сайты через поисковые системы, отсюда можно предположить, что популярность веб-сайта будет зависеть от его позиции в результатах поиска.

Чтобы обеспечить высокую посещаемость сайта, его в первую очередь нужно зарегистрировать в популярных поисковых системах, так же можно сделать обмен ссылками, или баннерами. Чтобы не заниматься этим самому, как правило заказывают раскрутку у фрилансеров, которые как профи своего дела, они уже регистрируют сайт по базам каталогов и сделают все необходимое для раскрутки веб-ресурса.

Седьмой этап – поддержка сайта. Следует учитывать, каким образом будут добавляться или редактироваться материалы и разделы сайта. Возможно, потребуется сделать архив статей, куда будут попадать не актуальные новости. Для таких целей выделяется сотрудник, который будет следить за частотой обновления, предоставлять свежую информацию, убирать старую.

2.4 Разработка базы данных

Инфологическая модель баз данных применяется на этапе проектирования базы данных, то есть после письменного описания предметной области. Сам процесс проектирования длительный, он требует долгих и трудоемких обсуждений с заказчиком, со специалистами в данной предметной области. Инфологическая модель базы данных должна включать в себя такое формализованное описание предметной области, что бы оно легко «читалось» не только специалистами по базам данных, но и простыми людьми. И такое описание должно быть настолько ёмким и понятным, чтобы можно было как можно лучше оценить глубину и корректность проработки проекта базы данных, и конечно же, как говорилось выше, оно не должно быть привязано к конкретной СУБД.

Инфологическое проектирование, прежде всего, связано с попыткой представления семантики предметной области в модели БД. Реляционная модель данных в силу своей простоты и лаконичности не позволяет отобразить смысл предметной области.

Рассмотрим проектирование базы данных на примере разрабатываемого веб-сервиса по проведению научных конференций.

Пусть необходимо построить базу данных, содержащую информацию о проводимых конференциях и их участниках.

На основании внимательного анализа предметной области можно выделить следующие сущности модели «сущность-связь»: «Пользователи», «Группы-пользователи», «Группы пользователей» и «Конференции» (рисунок 2.2).

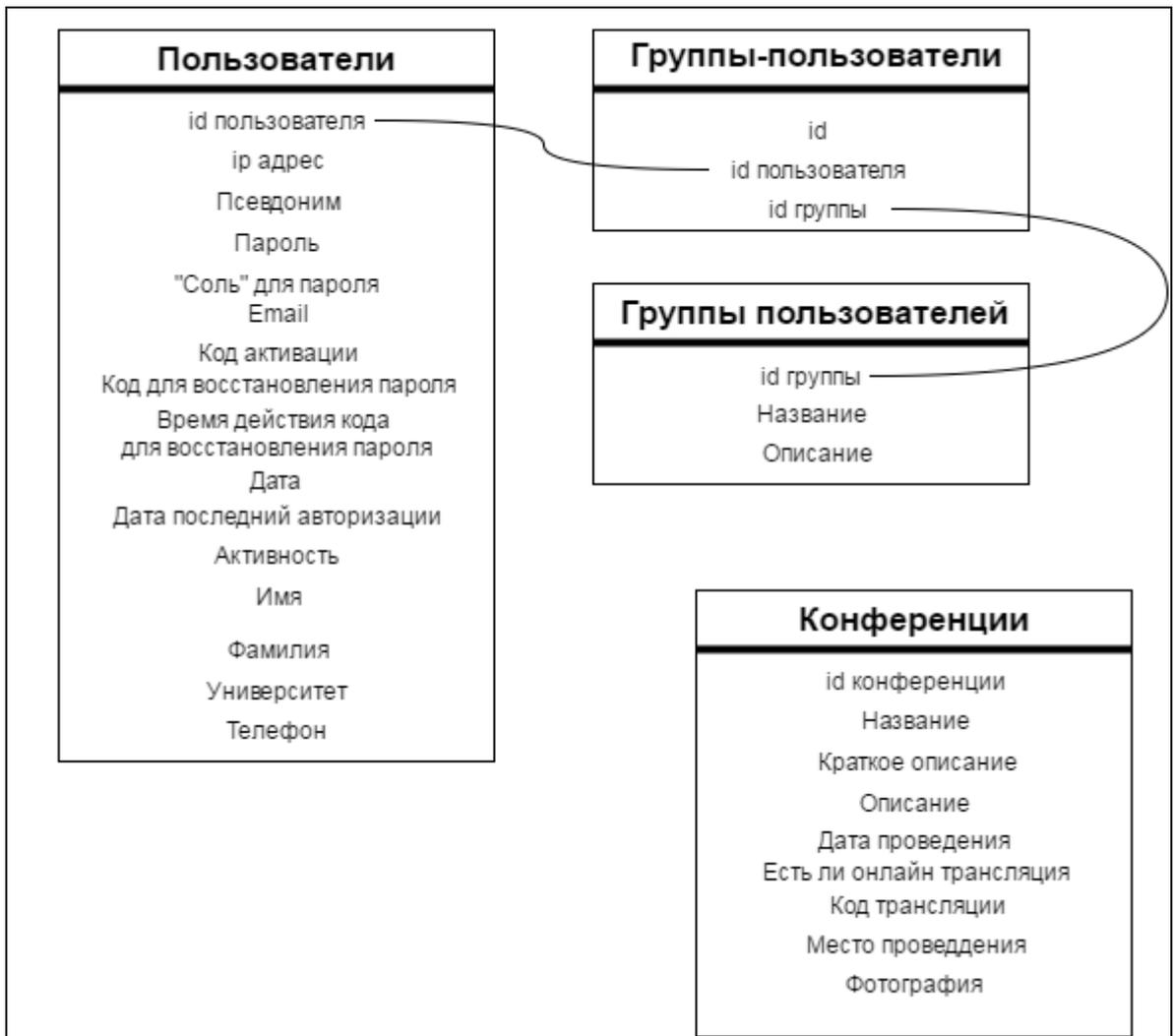


Рисунок 2.2 – Сущности модели для базы данных веб-сервиса по проведению научных конференций

Теперь рассмотрим проектирование базы данных на примере разрабатываемого веб-сервиса по размещению научных работ.

Пусть необходимо построить базу данных, содержащую информацию о размещенных на сайте научных работах.

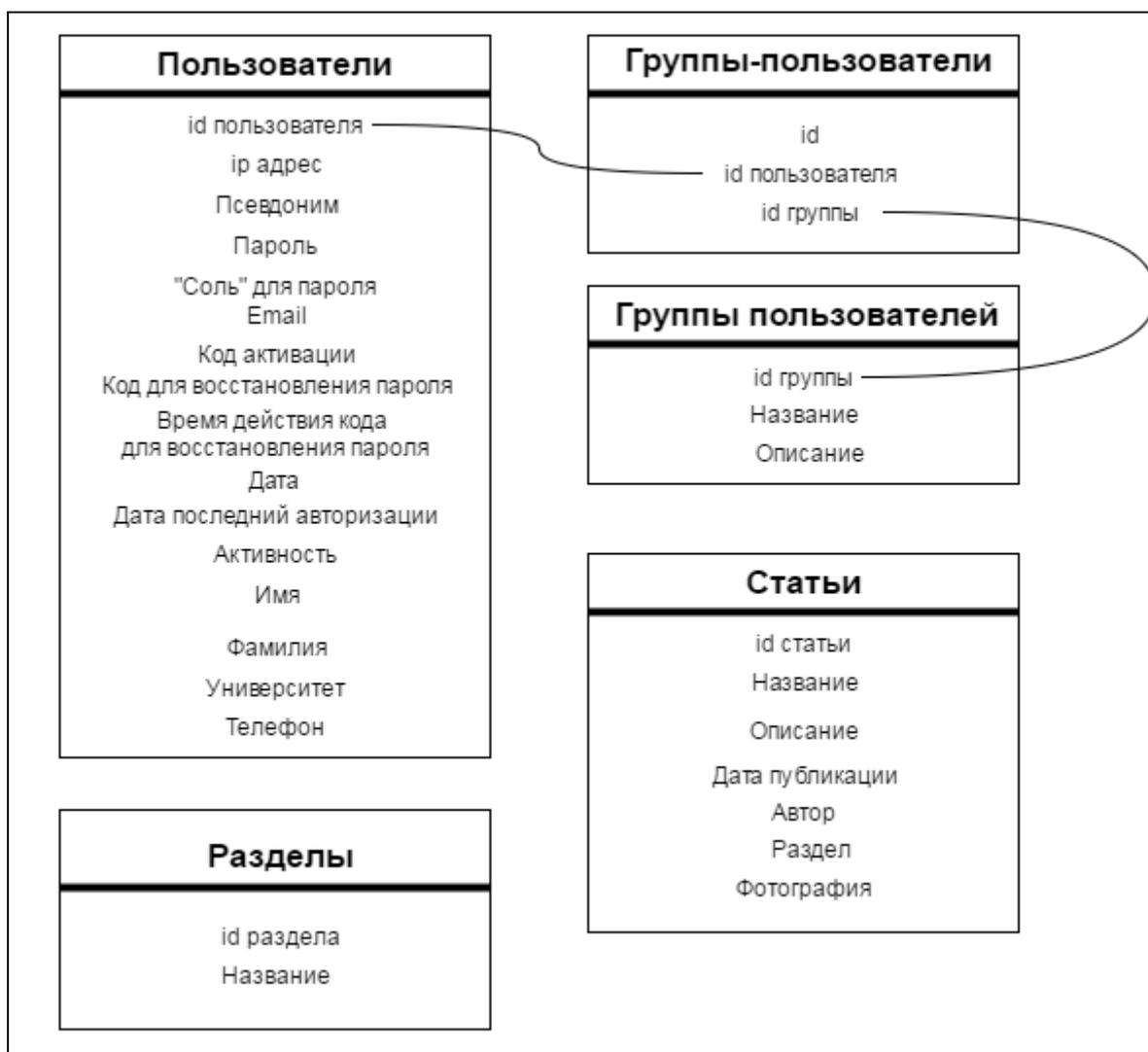


Рисунок 2.3 – Сущности модели для базы данных веб-сервиса по размещению научных работ

Если понимать язык условных обозначений, которые соответствуют категориям ER-модели, то ее можно легко «читать», следовательно, она доступна для анализа программистам-разработчикам, которые будут разрабатывать отдельные приложения. Она имеет однозначную интерпретацию, в отличие от некоторых предположений естественного языка, и поэтому здесь не может быть никакого недопонимания со стороны разработчиков. Общий подход к построению базы данных с использованием ER-метода состоит, прежде всего, в построении инфологической модели, включающей в себя все важные сущности и связи. Следующим шагом в процессе проектирования базы данных является построение набора предварительных таблиц и указание

предполагаемого первичного ключа для каждой таблицы. В реляционных БД даталогическое или логическое проектирование приводит к разработке схемы БД, т.е. совокупности схем отношений, которые адекватно моделируют абстрактные объекты предметной области и семантические связи между этими объектами. Основой анализ корректности схемы являются так называемый функциональные зависимости между атрибутами БД. Некоторые зависимости между атрибутами отношений являются нежелательными из-за побочных эффектов и аномалий, которые они вызывают при модификации БД. При этом под процессом модификации БД понимается внесение новых данных в БД или удаление некоторых данных из БД, а также обновление значений некоторых атрибутов.

На основе даталогических моделей были построены следующие физические модели баз данных для разрабатываемых веб-сервисов.

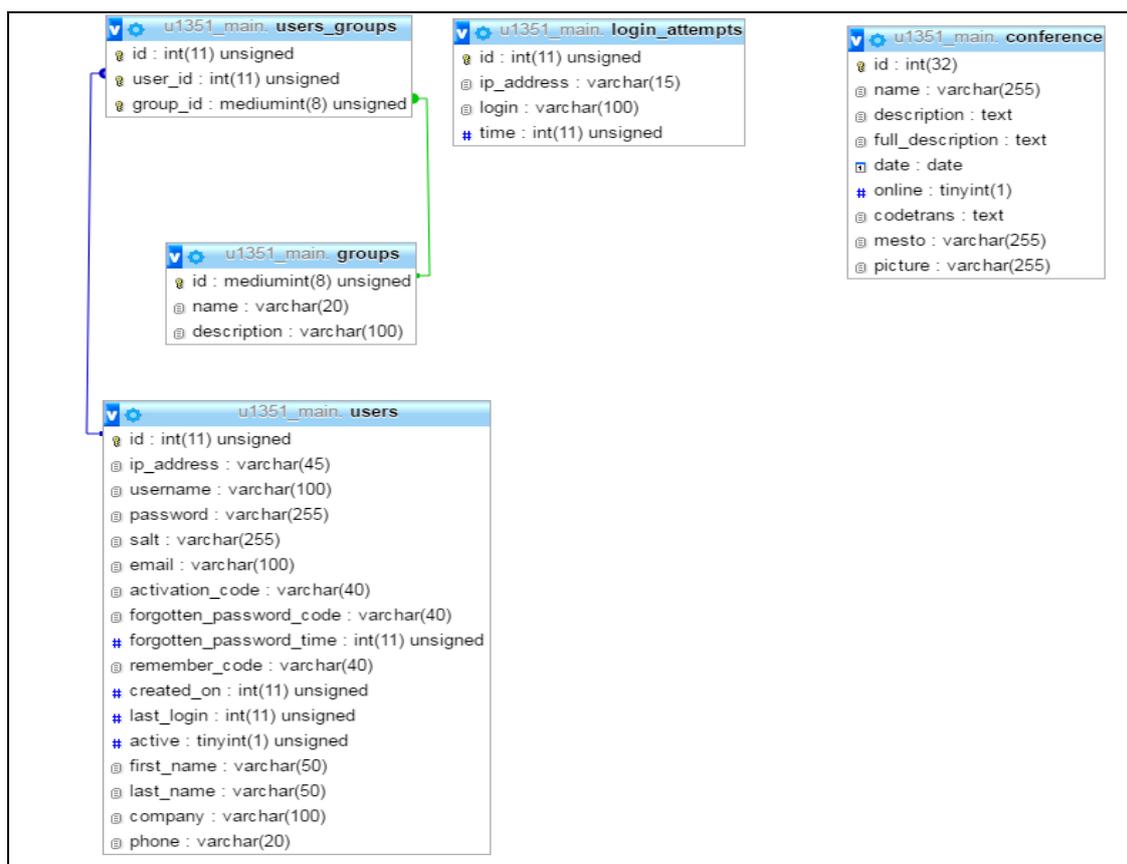


Рисунок 2.4 – Физическая модель базы данных веб-сервиса по проведению научных конференций

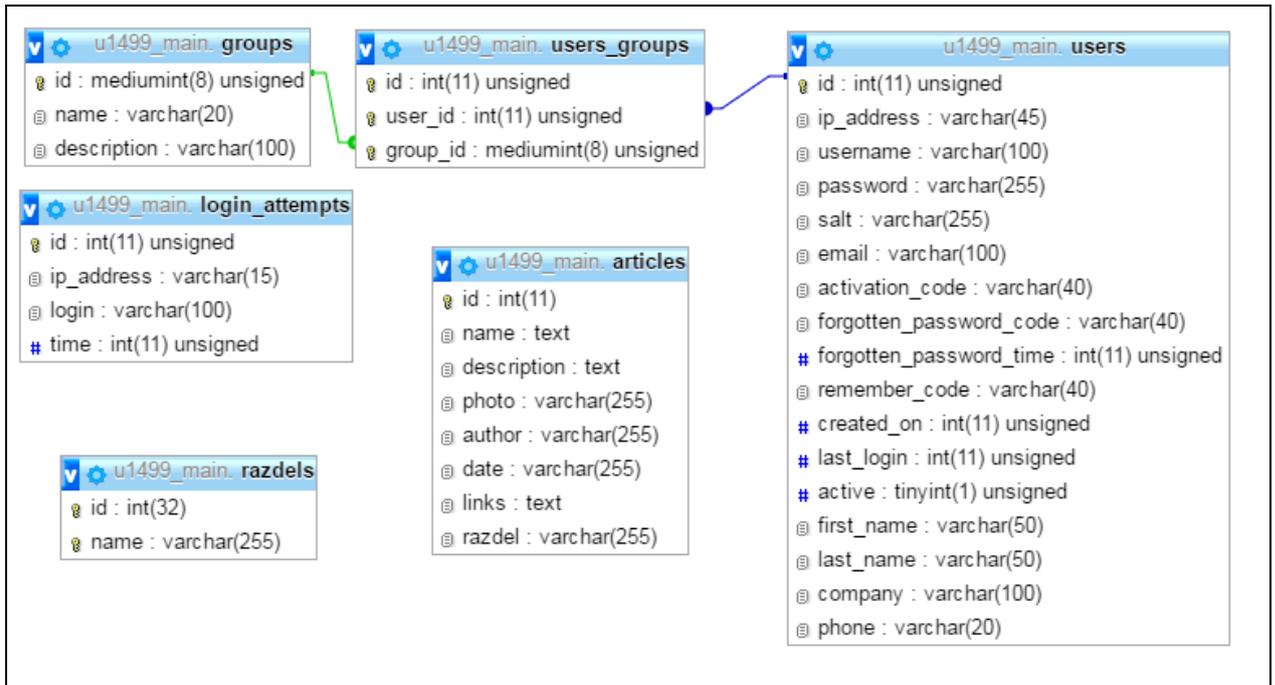


Рисунок 2.5 – Физическая модель базы данных веб-сервиса по размещению научных работ

На основе анализа предметной области были разработаны инфологическая, физическая (даталогическая) модели. Построены различные схемы. Проведено исследование и обоснование выбора средств разработки информационного ресурса. Определены основные требования к информационному ресурсу.

3 Программная реализация веб-сервиса для проведения научных конференций и размещения научных работ

3.1 Выбор средств реализации

Удачный Web-сайт - это в высшей степени эффективный инструмент - он способен захватывать внимание аудитории. Сделав всего несколько изменений, простой Web-сайт может превратиться в более надежный и эффективный инструмент. Web-сайт, способный привлечь внимание и вызвать любопытство, побудит посетителей не только просмотреть оставшиеся страницы и совершить какие-либо действия, но и снова посетить его через некоторое время, а также рекомендовать своим друзьям и знакомым.

Для разработки веб-сервисов был выбран фреймворк написанный на языке программирования PHP под названием CodeIgniter, т.к. его гибкие функциональные возможности позволяют создавать более совершенные Web-узлы, включающие средства для профессионального проектирования, разработки, работы с данными и публикации, необходимые для создания динамических и более сложных Web-узлов. CodeIgniter позволяет усовершенствовать процесс Web-разработки в следующих трех ключевых областях.

Усовершенствованные средства проектирования позволяют улучшить оформление Web-узлов. Новые средства разметки и работы с графикой упрощают процесс создания Web-узлов, полностью отвечающих замыслу пользователя.

Средства проектирования позволяют повысить качество создаваемого кода и усовершенствовать навыки в области программирования. Встроенные средства разработки сценариев обеспечат поддержку интерактивности в создаваемых продуктах. Благодаря профессиональным средствам написания кода можно работать быстрее, эффективнее и точнее.

Можно организовать общение и обмениваться данными по-новому, создавая управляемые данными Web-узлы с широкими интерактивными возможностями.

CodeIgniter – построен по архитектуре MVC (Model-View-Controller), и его работу можно наглядно увидеть на рисунке 3.1.

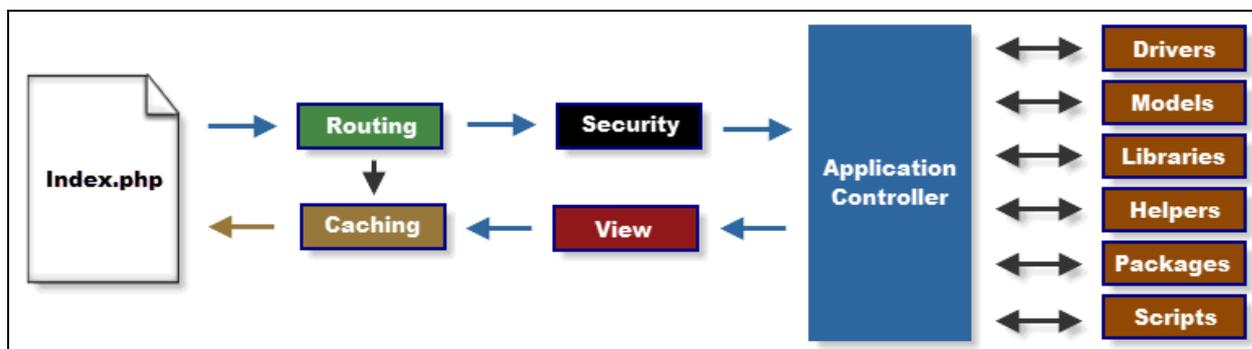


Рисунок 3.1 – Схема работ фреймворка

В качестве СУБД была выбрана MySQL основанием для этого служит ряд немаловажных причин, и одной из них является то, что MySQL является продуктом класса Open Source (открытые исходные тексты), который можно получить. Как правило, сервер и клиент MySQL входят в любой дистрибутив операционных систем семейства BSD (FreeBSD, NetBSD, OpenBSD) и Linux, которые используются на большинстве веб-серверов, но, при необходимости, последнюю версию MySQL для всех поддерживаемых систем также можно найти на сайте компании MySQL AB — разработчиков MySQL.

Другая немаловажная причина выбора MySQL заключается в том, что ее создатели с самого начала разработки этой СУБД поставили во главу угла ее быстродействие, пожертвовав при этом некоторыми удобствами для разработчиков. Связка PHP + MySQL обеспечивают очень высокое быстродействие, которого очень трудно достичь другими средствами. Очень хорошая связь MySQL с PHP стала еще одной причиной выбора этой СУБД. Поддержка MySQL входит в стандартную сборку PHP, и можно быть уверенным, что проблем обращения к серверу MySQL из PHP-скриптов не будет. Для обеспечения взаимодействия PHP с другими СУБД приходится компилировать его самостоятельно из исходных кодов с дополнительными

опциями. Таким образом, можно считать, что дешевизна, легкодоступность, производительность и тесная взаимосвязь с РНР это лучший вариант для реализации данного программного продукта.

После того как были выбраны все программные инструменты, я приступил к разработке иерархии базы данных.

3.2 Базовые классы компонентов

В настоящем разделе будут описаны классы, и функции которые были написаны в ходе разработки сервиса для проведения конференций.

Иерархия базовых классов компонентов берёт своё начало с класса Main. Он реализует основную функциональность по выводу конференций в браузер пользователя.

`public function index()` – из модели (класса) Conferene запрашивает список текущих конференций, и затем выводит их на главную страницу.

```
public function index()
{
    //print_r($this->conference->listconf());
    $this->data['content'] = $this->conference->listconf(4);
    $this->data['title'] = "Научные конференции";
    $this->parser->parse('conference/index', $this->data);
}
```

Рисунок 4.2 – Функция index

`public function conference($id)` – данная функция на вход получает ID конференции, после чего происходит вызов функции `fullconf($id)` из класса Conference которая возвращает массив с данным о за прощенной конференции, и возвращает обработанные данные в браузер пользователя.

```

public function conference($id){
    $conf = $this->conference->fullconf($id);
    $this->data['title'] = $conf[0]['name'];
    if($conf[0]['codetrans']=="){
        $this->data['content']='<h2>'. $conf[0]['name']. '</h2><div style="width: 800px;
border: 1px solid grey;
margin: 0 auto;
margin-top: 20px;
margin-bottom: 20px;
padding: 10px;
border-radius: 5px;
box-shadow: 0px 0px 20px 4px; text-align:center;">
<br>
<span><strong>Место проведения: </strong>'. $conf[0]['mesto']. '</span><br>
<span><strong>Дата проведения: </strong>'. $conf[0]['date']. '</span><br>
<p>'. $conf[0]['full_description']. '</p><br>
<!-- LikeBtn.com BEGIN -->
<span class="likebtn-wrapper" data-lang="ru" data-i18n_like="Буду участвовать!"></span>
<script>(function(d,e,s){if(d.getElementById("likebtn_wjs"))return;a=d.createElement(e);m=d.getElementsByTagName(e)[
0];a.async=1;a.id="likebtn_wjs";a.src=s;m.parentNode.insertBefore(a, m)})(document,"script","/w.likebtn.com/js/w/widget.js");</script>
<!-- LikeBtn.com END -->
</div>';
    }else{
        $this->data['content']='<h2>'. $conf[0]['name']. '</h2><div style="width: 800px;
border: 1px solid grey;
margin: 0 auto;
margin-top: 20px;
margin-bottom: 20px;
padding: 10px;
border-radius: 5px;
box-shadow: 0px 0px 20px 4px; text-align:center;">
'. $conf[0]['codetrans']. '<br>
<span><strong>Место проведения: </strong>'. $conf[0]['mesto']. '</span><br>
<span><strong>Дата проведения: </strong>'. $conf[0]['date']. '</span><br>
<p>'. $conf[0]['full_description']. '</p><br>
<!-- LikeBtn.com BEGIN -->
<span class="likebtn-wrapper" data-lang="ru" data-i18n_like="Буду участвовать!"></span>
<script>(function(d,e,s){if(d.getElementById("likebtn_wjs"))return;a=d.createElement(e);m=d.getElementsByTagName(e)[
0];a.async=1;a.id="likebtn_wjs";a.src=s;m.parentNode.insertBefore(a, m)})(document,"script","/w.likebtn.com/js/w/widget.js");</script>
<!-- LikeBtn.com END -->
</div>';
    }
    $this->parser->parse('conference/index', $this->data);
}

```

Рисунок 5.3 – Функция conference

Класс Conference один из главных классов во всем проекте, так как именно он отвечает за взаимодействие всей системы непосредственно с конференциями.

public function addconf(\$name, \$description, \$date, \$online=0, \$codetrans, \$mesto, \$picture, \$fulldescription) – данная функция отвечает за добавление новой коференции в базу данных, в нее непосредственно передаются все данные конференции из HTML формы которая располагается в административной части сайта.

```

public function addconf($name, $description, $date, $online=0, $codetrans, $mesto, $picture, $fulldescription){
    $fulldesc = htmlentities($fulldescription, ENT_QUOTES);
    $desc = htmlentities($description, ENT_QUOTES);
    $query = $this->db->query("INSERT INTO `u1351_main`.`conference` (`id`,`name`,`description`,`date`,`online`,`codetrans`,`mesto`,`picture`,`
full_description`) VALUES (NULL, '$name', '$desc', '$date', '$online', '$codetrans', '$mesto', '$picture', '$fulldesc')");
    header("Location:/panel/listconf");
}

```

Рисунок 6.4 – Функция addconf

public function fullconf(\$id) – данная функция на входе получает ID конференции и делает непосредственный запрос в базу данных, после чего возвращает данные о конференции в виде массива.

```

public function fullconf($id){
    $query = $this->db->query("SELECT * FROM `u1351_main`.`conference` WHERE `id`='{$id}'");
    $result = $query->result_array();
    return $result;
}

```

Рисунок 7.5 – Функция fullconf

public function listconf() – в этой функции происходит запрос всех текущих конференции, после чего возвращается двумерный массив с данным о конференциях.

```

public function listconf($count=999999){
    $query = $this->db->query("SELECT * FROM `u1351_main`.`conference` LIMIT 0, {$count}");
    $result = $query->result_array();
    return $result;
}

```

Рисунок 8.6 – Функция listconf

public function uploadphoto(\$tmpfile) – данная функция служит для загрузки фотографий конференций на сервер, на вход она получает временный адрес, после чего данная функция возвращает адрес куда был записан файл.

```

public function uploadphoto($tmpfile){
    $uploaddir = '/home/u1351/domains/u1351.blue.elastictech.org/uploads/';
    //var_dump($tmpfile);
    $name = time().str_replace(".", "", $tmpfile['name']);
    //var_dump($tmpfile['name']);
    $uploadfile = $uploaddir.time().str_replace(".", "", $tmpfile['name']);

    // Копируем файл из каталога для временного хранения файлов:
    if (copy($tmpfile['tmp_name'], $uploadfile))
    {
        return $name;
    }
}

```

Рисунок 9.7 – Функция uploadphoto

public function editconf(\$id, \$name, \$description, \$date, \$online=0, \$codetrans, \$mesto, \$fulldescription) – данная функция отвечает за редактирование конференции, её работа похожа на работу функции addconf() за исключением того что она перезаписывает данные редактируемой конференции.

```

public function editconf($id, $name, $description, $date, $online=0, $codetrans, $mesto, $fulldescription){
    $fulldesc = htmlentities($fulldescription, ENT_QUOTES);
    $desc = htmlentities($description, ENT_QUOTES);
    $query = $this->db->query("UPDATE `u1351_main`.`conference` SET `name`='{$name}', `description`='{$description}', `full_description`='{$fulldescription}'
    date='{$date}', `online`='{$online}', `codetrans`='{$codetrans}', `mesto`='{$mesto}' WHERE `conference`.`id`='{$id}'");
    header("Location: /panel/listconf");
}

```

Рисунок 10.8 – Функция editconf

`public function delconf($id)` – данная функция служит для удаления конференции по её ID.

```
public function delconf($id){
    $query = $this->db->query("DELETE FROM `u1351_main`.`conference` WHERE `conference`.`id` = '$id'");
    header("Location:/panel/listconf");
}
```

Рисунок 11.9 – Функция delconf

Класс Articles один из главных классов отвечающий за работу с научными статьями.

`public function addart()` – данная функция отвечает за добавление новой научной работы в базу данных, в нее непосредственно передаются все данные из HTML формы которая располагается в административной части сайта.

```
public function addart(){
    if(empty($_POST)){
        $this->load->view('addart');
    }else{
        $photo = $this->uploadphoto($_FILES['photo']);
        $query = $this->db->query("INSERT INTO `u1499_main`.`articles` (`name`,`description`,`author`,`date`,`links`,`razdel`,`photo`) VALUES ('$_POST[name]','$_POST[description]','$_POST[author]','$_POST[date]','$_POST[links]','$_POST[razdel]','$_photo');");
        header("Location:/admin/");
    }
}
```

Рисунок 12.10 – Функция addart

`public function editart($id)` – данная функция на входе получает ID научной работы и делает непосредственный запрос в базу данных, после чего возвращает данные о работе в виде массива для последующего редактирование с помощью HTML формы.

```
public function editart($id){
    if(empty($_POST)){
        $query = $this->db->query("SELECT * FROM `articles` WHERE `id` = '$id'");
        $art = $query->result_array();
        $data['art'] = $art;
        //var_dump($data);
        $this->parser->parse('editart',$data);
    }else{
        if($_FILES['photo']['name'] == ""){
            $query = $this->db->query("UPDATE `u1499_main`.`articles` SET `name` = '$_POST[name]', `description` = '$_POST[description]', `author` = '$_POST[author]', `date` = '$_POST[date]', `links` = '$_POST[links]', `razdel` = '$_POST[razdel]' WHERE `articles`.`id` = '$id'");
        }else{
            $photo = $this->uploadphoto($_FILES['photo']);
            $query = $this->db->query("UPDATE `u1499_main`.`articles` SET `photo` = '$photo', `name` = '$_POST[name]', `description` = '$_POST[description]', `author` = '$_POST[author]', `date` = '$_POST[date]', `links` = '$_POST[links]', `razdel` = '$_POST[razdel]' WHERE `articles`.`id` = '$id'");
        }
        header("Location:/admin/");
    }
}
```

Рисунок 13.11 – Функция editart

`public function delart($id)` – эта функция отвечает за удаление научной работы из базы данных по её ID.

```

public function delart($id){
    $query = $this->db->query("DELETE FROM `u1499_main`.`articles` WHERE `articles`.`id` = '$id'");
    header("Location:/admin/");
}

```

Рисунок 14.12 – Функция delart

public function list_raz() – данная функция отвечает за запрос списка научных статей в том или ином разделе сайта.

```

public function list_raz(){
    $query = $this->db->query('SELECT * FROM `razdels`');
    foreach ($query->result() as $row)
    {
        echo '<li><a href="/welcome/razdel/'. $row->name. '/">'. $row->name. '</a></li>';
    }
}

```

Рисунок 15.12 – Функция list_raz

Класс Auth отвечает за систему пользователей в нашем программном продукте. Данный класс уже входит в архив фреймворка CodeIgniter. Этот класс является стандартным.

Общая схема авторизации пользователей в проекте изображена на рисунке 3.13.

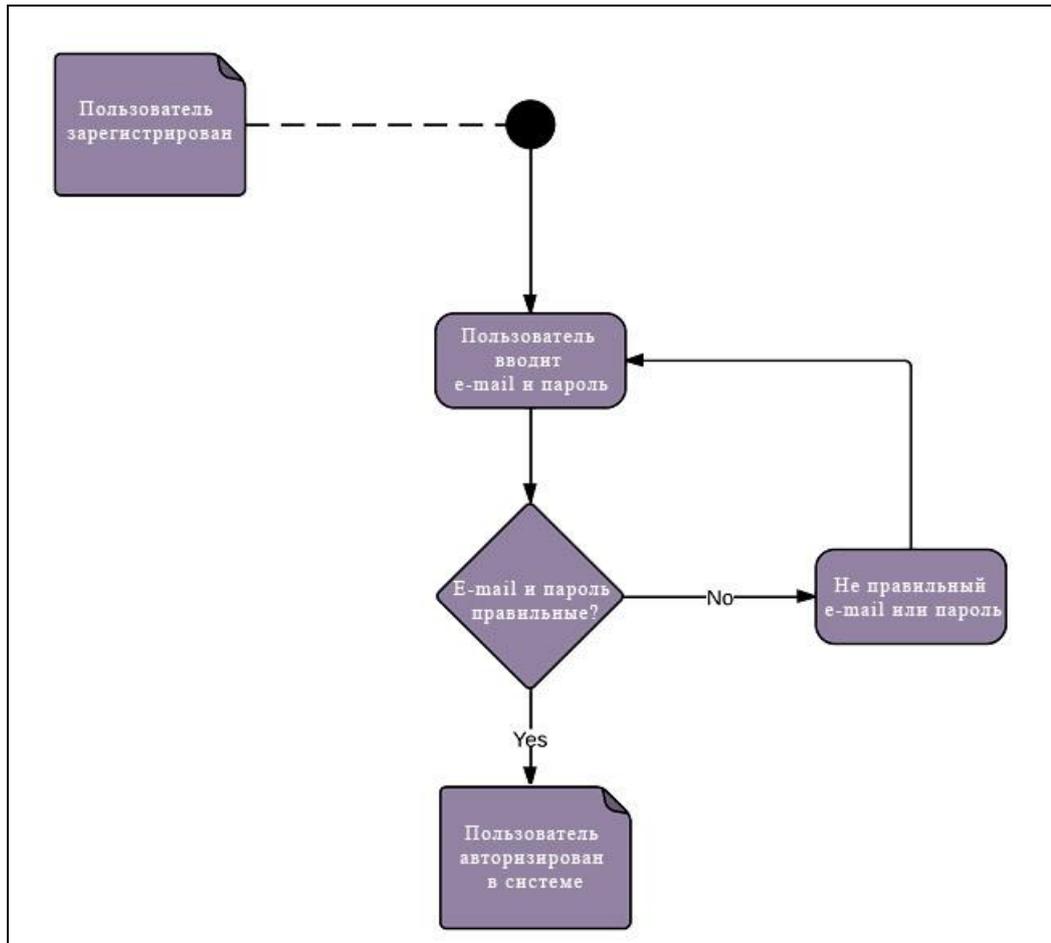


Рисунок 3.13 – Схема авторизации пользователей

В результате выполненных действий получилось создать полностью самостоятельный информационный продукт, который отвечает все поставленным требованиям и целям. Были использованы последние версии средств разработки веб-приложений. Разработана база данных. Реализован интерфейс информационного ресурса.

Заключение

Целью данной выпускной квалификационной работы являлось создание информационного ресурса, предназначенного для проведения научных конференций на базе Тольяттинский государственный университет, а так же для увеличения индекса Хирша для научных сотрудников ТГУ. Данная система состоит из двух основных подсистем:

- подсистема управления научными конференциями;
- подсистема, отвечающая за размещение научных работ для последующего цитирования.

Для достижения поставленной цели были подробно изучены и проанализированы все наиболее популярные системы.

Были поставлены задачи изучения предметной области, разработки интерфейса, позволяющего проводить научные конференции, а также разработки универсального сервиса, предназначенного для размещения научных работ сотрудниками ТГУ.

Процесс разработки проходил с применением концепции Continuous Integration (непрерывная интеграция), что позволяло отслеживать ход реализации проекта и выявлять проблемы с помощью интеграционного тестирования и обеспечивать требуемый уровень качества.

Платформа была разработана на языке PHP. Для реализации авторизации и аутентификации, а так же проекта в целом использовался фреймворк CodeIgniter.

Результаты развернутого тестирования разработанного приложения показали, что удалось достичь хорошей производительности системы. За счет проведения нагрузочного тестирования удалось доказать эффективность работы разработанного приложения на относительно слабых машинах.

Список используемой литературы

1. Информационный сайт фирмы IBM [Электронный ресурс]. – Режим доступа: <http://www.ibm.com/developerworks/rational/library/4706.html>
2. Колисниченко Д.Н. PHP и MySQL. Разработка веб-приложений / Д.Н. Колисниченко. – Санкт-Петербург: Изд-во «БХВ-Петербург», 2015. – 592с.
3. Ульман Л. PHP и MySQL. Создание интернет-магазинов, 2-е изд. / Л. Ульман; пер. с англ. А. Сергеев. – М.: «Вильямс», 2015. – 544с.
4. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS-2-е изд. / Р. Никсон – СПб.: Питер, 2013. – 560с.
5. Бейли, Л. Изучаем SQL / Л. Бейли – СПб.: Питер, 2012. – 573с.
6. Зандстра, М. PHP. Объекты, шаблоны и методики программирования, 4-е издание / Издательство "Вильямс", 2015. – 576с.
7. Olsson, M. PHP 7 Quick Scripting Reference / Olsson Mikael. – Apress 2016. – 160 p.
8. Chris, P. CodeIgniter: MVC / Chris Pitt. – Apress, 2012. – 365 p.
9. Charles, B. Expert MySQL / Charles Bell. – Apress 2012. – 611 p.
10. Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд. – СПб.: Изд-во «БХВ-Петербург», 2015. – 766с.
11. Kevin, M. Introduction to Non-functional Requirements / Kevin MacG Adams. – Springer International Publishing 2015. – 72 p.
12. Стивенс Р. Алгоритмы. Теория и практическое применение: учеб. пособие / Р. Стивенс. – Эксмо, 2016. – 544 с.
13. Олейник, П.П. Корпоративные информационные системы: учеб. пособие / П. П. Олейник. – СПб.: Питер, 2012. – 176 с.
14. Mikael O. CSS Quick Syntax Reference Guide / Mikael Olsson. – Apress 2014. – 124с.

15. Бородаев Д. А. WEB-сайт как объект графического дизайна: монография / Д. А. Бородаев./ – Харьков, 2004. – 232 с.
16. Корнеев, И. К. Информационные технологии : учебное пособие / И. К. Корнеев./ - М.: Проспект, 2007.- 224 с.
17. Лабберс П. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений / П. Лабберс, Б. Олаберс./ – М.: Вильямс, 2011. – 272с.
18. Алексеев, А.П. Введение в Web-дизайн: учебное пособие / А. П. Алексеев./ – М. : СОЛОН-ПРЕСС, 2008. – 192 с.
19. Kroenke, David M. Database Processing: Fundamentals, Design, and Implementation / David M. Kroenke, David J. Auer. – 14th edition. – Pearson Education Limited, 2015. – 640 p.
20. Kromann, Frank M. PHP and MySQL Recipes: A Problem-Solution Approach / Frank M. Kromann. – 2nd edition. – Appress, 2015. – 700 p.

Результаты выполненной работы

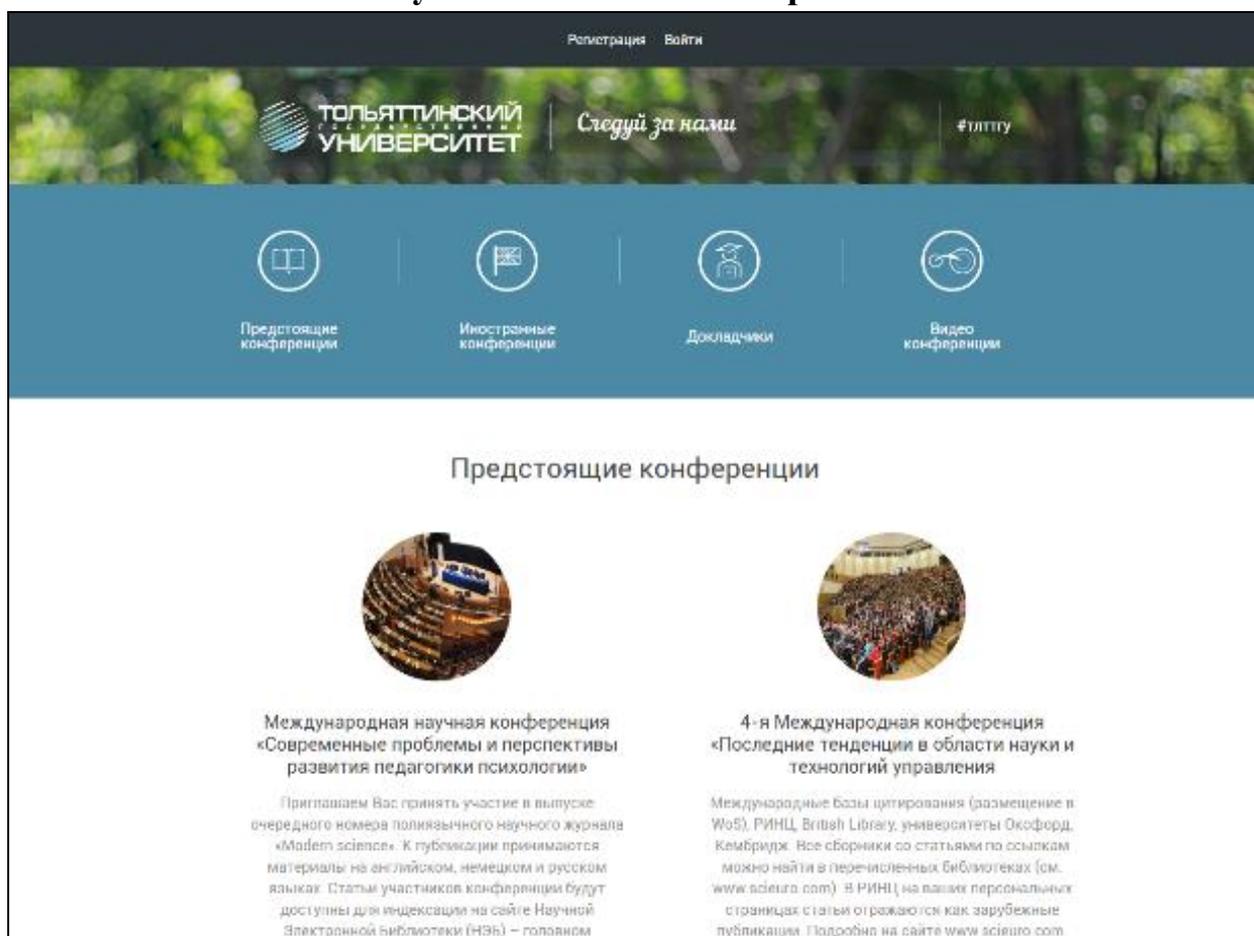


Рисунок А.1 – Главная страница сайта по проведению научных конференций

Регистрация Войти

 ТОЛЬЯТТИНСКИЙ
ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

Следуй за нами

#глтту

Предстоящие конференции

Иностранные конференции

Докладчики

Видео конференции

Регистрация

Имя:	<input type="text"/>
Фамилия:	<input type="text"/>
Учебное заведение:	<input type="text"/>
E-mail:	<input type="text"/>
Телефон:	<input type="text"/>
Статус:	<input type="text" value="Участник"/>
Пароль:	<input type="password"/>
Пароль ещё раз:	<input type="password"/>
<input type="button" value="Регистрация"/>	

Режим работы

приемная комиссия
пн - пт: 9.00 до 17.00.
сб - вс: выходной

Наш адрес:

445020, г. Тольятти,
ул. Белорусская, 14,
главный корпус ТГУ,

Позвонить нам

(8482) 50-11-00

E-mail

Рисунок А.2 – форма регистрации участников на сайте по проведению научных конференций

Управление пользователями | Управление конференциями | Выйти

ТОЛЬЯТТИНСКИЙ
ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

Следуй за нами

#тлгту

Предстоящие конференции

Иностранные конференции

Докладчики

Видео конференции

Создание конференции

Афиша

Название конференции

Краткое писание конференции

Дата проведения

Онлайн трансляция

Место проведения

Полное писание конференции

Рисунок А.3 – Форма создания новой конференции на сайте по проведению научных конференций



Рисунок А.4 – Главная страница сайта по размещению научных работ

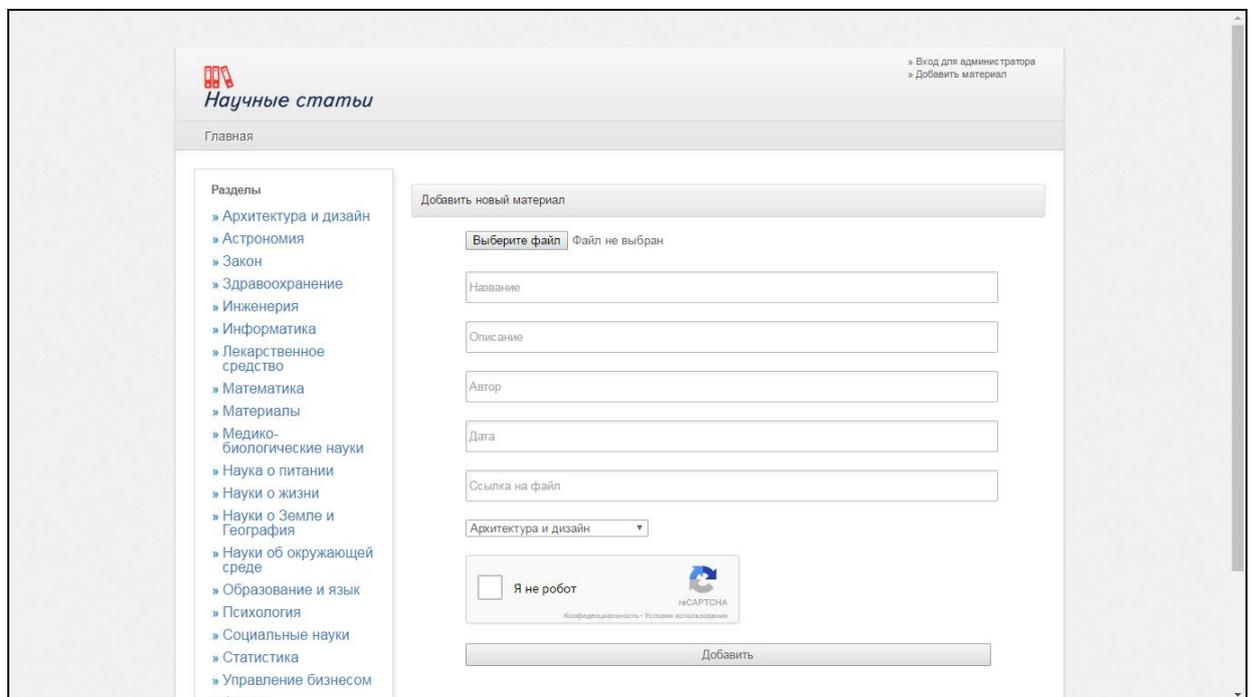


Рисунок А.5 – Форма заявки для размещения научной работы