

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 информационной системы музея ТГУ

Студент _____ Ш.Т Мухтар _____
Руководитель _____ А.В Очеповский _____

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 Мухтар Шамиль Тимурович

1. Тема Разработка информационной системы музея ТГУ
2. Срок сдачи студентом законченной выпускной квалификационной работы 19.06.2016
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): Введение. 1. Анализ технологии размещения информации об экспонатах музея ТГУ в сети интернет. 2. Проектирование сайта музея ТГУ 3. Реализация, тестирование и развертывание сайта с полным функционалом, внутри, работающей системы музея ТГУ. Заключение
5. Ориентировочный перечень графического и иллюстративного материала блок-схемы работы приложения, графики, диаграммы, экранные формы, демонстрирующие работоспособность программного продукта; презентация
6. Дата выдачи задания «11» января 2016 г.

Заказчик

Заведующая музеем ТГУ

Т.А. Якимова

Руководитель выпускной
квалификационной работы

А.В. Очеповский

Задание принял к исполнению

Ш.Т. Мухтар

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 Мухтар Шамиля Тимуровича
по теме Разработка информационной системы музея ТГУ

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. Аprobация предложенных решений	9.05.2016	9.05.2016	Выполнено	
5. Оформление	12.05.2016	12.05.2016	Выполнено	

пояснительной записки				
Подготовка доклада и графического материала к защите	20.05.2016 – 20.06.2016	20.06.2016	Выполнено	
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

_____ А.В. Очеповский

Задание принял к исполнению

_____ Ш.Т. Мухтар

Аннотация

Тема: «Разработка информационной системы музея ТГУ»

Актуальность темы данной ВКР обусловлена необходимостью автоматизации и облегчения работы сотрудников музея ТГУ.

Целью данной выпускной квалификационной работы является разработка программы учета материалов для музея ТГУ.

Для достижения поставленной цели в работе решались следующие задачи:

- была проанализирована литература по проблеме автоматизации основных бизнес-процессов деятельности музея;
- была проанализирована деятельность музея в профильной области для определения основных процессов, подлежащих автоматизации;
- был разработан пользовательский интерфейс информационного ресурса на основе базы данных для взаимодействия пользователя на основе концептуальной модели;
- проведено внедрение информационного ресурса на систему управления сайта ТГУ «Bitrix».

Работа состоит из введения, трех глав (аналитической, проектной и части реализации), заключения, списка использованной литературы и приложения.

В работе использованы современные системы и технологии проектирования информационных систем, PHP, MySQL и др.

В заключении сформулированы основные выводы, которые были сделаны в процессе исследования и описаны результаты практической реализации работы.

В приложении предоставлены фрагменты программного кода и другие дополнительные материалы. В работе представлено- 12 рисунков, 4 таблицы. Всего в работе 56 страниц.

Оглавление

Введение	4
Глава 1. Анализ деятельности музея ТГУ	9
1.1 Содержание и структура технологии сбора экспонатов и выставочной деятельности музея	9
1.2. Построение модели как есть	13
1.3 Формирование требований к новой системе	14
1.4 Выбор метода для создания программы учета материалов для музея «ТГУ»....	16
Глава 2. Анализ и выбор средств программной реализации сайта музея ТГУ.....	20
2.1 Анализ средств проектирования базы данных.....	20
2.2 Выбор архитектуры приложения.....	21
2.3 Выбор системы управления базой данных	24
2.4 Проектирование приложений системы	28
2.5 Этапы разработки web-сайта.....	34
2.6 Разработка интерфейса	36
2.7 Техническое обеспечение.....	36
Глава 3. Программная реализация информационной системы музея ТГУ	40
3.1 Анализ принципов построения Web-сайта.....	40
3.2 Обоснование и выбор программного обеспечения.....	40
3.3 Разработка базы данных	41
3.3.1 Логическое моделирование базы данных	43
3.3.2 Физическое моделирование	46
3.3.3 Концептуальная модель.....	47
3.4. Базовые классы компонентов.....	48
3.4.1 Класс Main.....	48
3.4.2. Класс Eksp.....	49
3.4.2. Класс Auth.....	52

Заключение.....	56
Список используемой литературы	58
Приложение Образцы интерфейсных форм.....	60

Введение

Массовые коммуникации и цифровые технологии в информационном пространстве развиваются в современном мире очень быстро, поэтому во время выбора данной темы актуальность этой тенденции была взята за основу. Данное явление воплощает собой симбиоз, который удивительным образом служит во благо культуры, искусства, эстетике и духовном развитии, при этом общение между объектом, имеющим культурное наследие и человеком, как конечным его потребителем значительно облегчается. Сегодня человечество живет в эпохе Интернета.

Культурную жизнь нельзя представить без всемирной паутины и учреждения культуры, также начали активно подхватывать данный "тренд", при этом медленно, но уверенно создаются онлайн ресурсы музеев по всему миру. Данная тенденция является общемировой и нельзя сказать, что она замечена только на Западе, ведь все больше российских музеев, не только в Центральном регионе, но еще в глубинке начинают создавать свои онлайн ресурсы, тем самым популяризируя себя, а компьютер превратился в неотъемлемый инструмент музейного работника. В наши дни, музеи могут похвастаться наличием параллельной жизни во всемирной паутине и эта жизнь, все четче и четче приобретает самостоятельный характер. В интернете, не только делятся информацией, но еще проводят и научные конференции, различного рода дискуссии, создаются и координируются проекты и т.д.

Исходя из этого, музейное представительство в интернете также меняется. Музеи не только используют интернет в качестве мощного PR подспорья, но еще и усложняют в организационном плане свои ресурсы. Интерактивные формы все больше ими осваиваются. Музеи стараются представить себя потенциальному посетителю в лучшем виде с трехмерной графикой, интерактивными планами этажей и т.д. Сегодня, более полу тысячи отечественных музеев имеют свой электронный ресурс и их адреса можно

найти на сайте, где собрана общая информация о российских музеях «<http://www.museum.ru>».

Информационная революция разворачивается на наших глазах, и мы являемся непосредственными ее участниками, а роль информации и информатизация общества значительно увеличились. Объемы предлагаемой информации сегодня растут очень быстро, а с этим показателем увеличивается и компьютеризация общества, почти-то во всех сегментах социума. Для многих культурологов, философов и социологов спор об информационной эпохе, которая является последней в сфере развития человечества и в которой, оно находится сейчас, не заканчивается, ведь характерной особенностью данной эпохи является многообразность информационно-коммуникативных сетей.

Интенсивная информатизация подстегнула в развитии такие сферы как наука, образование, искусство, культура. Информационное общество способствует тому, что культура и ее специфические сегменты начали очень активно развиваться. Ее глобальное изменение, также обуславливается еще и тем фактом, что появились новые аудиовизуальные электронные средства, а также способы обработки информации. Именно данный нюанс активно отражается на развитии таких культурных учреждений как: музеи, библиотеки, архивы. Современные учреждения культуры в своем распоряжении имеют следующий перечень информационных продуктов: информационные киоски, мультимедийные диски, цифровые презентации, виртуальные экскурсии и экспозиции, виртуальные филиалы и это еще не все.

Что касается России, то говоря о плюсах и минусах некой задержки в развитии музейной коммуникации в стране, нужно учитывать объективность точки зрения. Да, российское музейное общество вступило в освоение электронных систем коммуникаций с неким опозданием, но данная особенность все-таки позитивно отразилась на отечественных музеях в целом, ведь они используют современные наработки в данной сфере и благодаря этому факту, не тратят время на модернизацию уже введенных технологий.

Объектом данной выпускной квалификационной работы является процесс сбора материалов сотрудниками «ТГУ»

В качестве **предмета** выпускной квалификационной выступает информационная система учета материалов музейного фонда, которая будет разработана в ходе работы.

Целью данной выпускной квалификационной работы является реализация программного обеспечения для учета материалов музея ТГУ, выявлению актуальных проблем, возникающих при организации системного подхода для учета материалов и обеспечение защиты уникальных данных материалов, хранящихся в музее. Для достижения указанной цели поставлены **следующие задачи:**

- проанализировать литературу по проблеме автоматизации основных бизнес-процессов деятельности музея;
- проанализировать деятельность музея в профильной области для определения основных процессов, подлежащих автоматизации; уделить особое внимание процессу автоматизации системы учета экспонатов и занесения их в базу данных;
- разработать пользовательский интерфейс информационного ресурса на основе базы данных для взаимодействия пользователя на основе концептуальной модели;
- провести тестирование информационного ресурса на локальном сервере для выявления основных ошибок в работе информационного ресурса;
- провести внедрение информационного ресурса на систему управления сайта ТГУ «Bitrix».

В ходе выполнения выпускной квалификационной работы будет создан информационный ресурс в виде сайта, который позволит:

- размещать информацию по экспонатам музея;
- выполнять быстрый поиск по экспонатам;

- вести виртуальные выставки работниками музея;
- вести раздел новостей и размещать ближайшие события музея.

В ходе выполнения выпускной квалификационной работы планируется выполнять анализ литературы и систематического подхода работы музеев с целью нахождения наиболее оптимального решения для разработки программного обеспечения способного выполнять поставленные задачи. Выполнить систематизацию информации относительно систем шифрования данных с целью сравнить и выявить наиболее подходящий алгоритм шифрования для выполнения поставленной задачи в выпускной квалификационной работе. Произвести теоретический анализ существующих способов передачи данных на сервер [3,5].

Практическая значимость данной выпускной квалификационной работы заключается в том, чтобы создать программы учета материалов для музея ТГУ и внедрить её на платформу работающего сайта , позволяющего включать систему с многими пользователями или устройствами и реализовать защищённую передачу данных на сервер.

Структура выпускной квалификационной работы обусловлена предметом, целью и задачами исследования. Работа состоит из введения, трёх глав, заключения и приложения.

Введение раскрывает актуальность, определяет степень научной разработки темы, объект, предмет, цель, задачи и методы исследования, раскрывает теоретическую и практическую значимость работы.

В первой главе будет произведён анализ предметной области, будет выбрана для рассмотрения типовая организация, которая будет использовать создаваемую информационную систему учета материалов для музея ТГУ

Во второй главе будет выполнено моделирование и проектирование информационной системы учета материалов для музея ТГУ

В третьей главе описывается механизм реализации задумки по созданию электронного сайта для музея ТГУ.

В заключении подводятся итоги исследования, формулируются выводы по, рассматриваемой в ходе выполнения работы, теме.

Глава 1. Анализ деятельности музея ТГУ

1.1 Содержание и структура технологии сбора экспонатов и выставочной деятельности музея

На протяжении многих лет, начиная с конца сороковых годов прошлого столетия, появилось словосочетание «собираТЕЛЬская работа» или по-другому, «комплектование музейного фонда». По мнениям научных сотрудников Ю.П. Пищулина и Н.П. Финягиной (<http://allrefs.net/c12/46908/p9/>), два данных понятия должны толковаться по-разному. Авторы считали, что комплектование фонда музея является непосредственной деятельностью организации. Цель музейных работников – поиск и приобретение объектов, которые по тематике подходят для музейных экспонатов. Само понятие «комплектование музейных фондов» приняло со временем совершенно другое значение – идеологическое.

Собрание материальных памятников культуры – это процесс состоящий из нескольких пунктов: практическое и теоретическое нахождение экспонатов для музея, их организацию и сбор. Но однако, это не только работа, направленная на поиск материальных памятников. Само изучение позволяет узнать больше о духовном аспекте жизни прошлых эпох, узнать об обычаях и традициях, о быте тех времен, религиозных и не религиозных обрядов, отношения между людьми. Благодаря понятию устройства того мира, гораздо легче осуществлять поиск материальных экспонатов и изучать жизнь общества в целом.

Самое главное в процессе комплектование музейных фондов – это предельный учет всего происходящего и написание документов, подтверждающих историческую ценность находок, как материальных, так и духовных. Все исторические явления, процессы и события обязательно должны были происходить на самом деле и запись должна быть сделана без малейших

искажений. Благодаря этому, создается полноценная картина жизни людей минувших лет.

Комплектование музейных фондов – это прежде всего исследовательская работа. Данная работа не проста в выполнении, для ее осуществления нужно проделать не мало изучений и исследований, работа должна быть насыщена и практической, и теоретической частью. Этапы такого исследовательского проекта включают в себя поиск информации и документов, проведение исследования на историческую ценность, анализ собранного материала.

Процесс по комплектованию фонда музеев есть двух видов. Первый из них – тематический. В таком случае музейные работники осуществляют сбор экспонатов, схожих по тематике. Это может быть музей, например, посвященный истории рыцарского ордена Тамплиеров. Данные музейные выставки обычно являются полными, несмотря на типологическую однообразность. Для поиск такой однотипной информации могут быть задействованы вещественные материалы, видео и аудио материалы, письменные документы. Другой метод, по которому производится комплектация музеев – систематический. Систематический способ еще именуют типологическим. С помощью него осуществляется сбор музейных предметов, относящихся к одной эпохе.

Есть еще вид комплектации, объединяющий два предыдущих вида – систематический и тематический. Он называется комплексным и действительно помогает собрать полный комплект и документов, и экспонатов по направленности музея, что однозначно понравится и посетителям-туристам, и для тех, кто посещает музей с образовательной целью.

В России 90 процентов всех музеев осуществляют работу по комплектованию по одному способу. Это процесс связан, прежде всего, с созданием выставочных экспозиций, которые собираются в ходе исследований, проводимых историками. Таким образом, именуемый проект, который

закрепляется в конкретной очередности с компанией поступлений стабильных ключей, при возможности осуществлении взаимосвязей с создателями и собственниками объектов увлекательных с целью музея и изучения географии музейных исследований. Тут так же учитываются и важные пошлины согласно тарифам нашего времени. Создание многообещающего проекта предполагает собою, как координационную, так и экспериментальную музееведческую проблему. Перспективный проект считается базой формирования годичных проектов комплектования фондов. Они включают в себя определенное исследование задач проекта, в то время как, согласование с существующими в директиве музея денег, требует присутствия квалифицированных профессионалов. Годичные проекты смотрятся коллективом и берутся в составе всеобщего годичного проекта деятельность музея.

Таким способом, академическая теория комплектования музейных фондов, представляет собой академическую концепцию музея, включает общее понимание о направленностях и способах комплектования в согласовании с профилем музея и его участком в музейной концепции. В ней обуславливаются аспекты отбора используемых материалов, с учетом вопросов музея.

Формы комплектования предполагают собою конкретные воздействия, в процессе каковых в музейный актив приобретаются объекты музейной значимости: данные экспедиции, академические командировки, бартер коллекциями среди музеев, получение объектов в ценных салонах, напрямую с учреждений и индивидуальных персон и т.п.

Данная модель комплектования музейного фонда дает возможность дополнять ранее существующие типологические собрания, формировать новейшие типологические выставки, отображающие многознаменательные действия, обихода и т.п. Объекты имеют все шансы действовать в галереях, согласно документации..

Приток объектов с учреждений и индивидуальных персон в значительном находится в зависимости от ресурсов и многообразия конфигураций

собираетельской деятельности музея, что обретает отображение в его проектах академического комплектования, взаимосвязей музея с созданием, учреждениями культуры, управленческими органами города и страны

Значительную роль играют шансы исполнения маркетинговых промоакций музея, равно как и выставки новейших поступлений, с предписанием фамилий спонсоров, выставки, приуроченные к значимым датам в городе и стране. Представления работников музея согласно радио и телевизору с программами, которые рекламируют коллекции музея, а кроме того распространение среди числа жителей флаеров с описаниями выставок и экспозиций музеев.

Выявление объектов музейной значимости полученных подобным способом, основывается на многолетних исследованиях над установленными областями существования: над группами жителей, функционерами культуры, над формированием моды и т.д.

Предлагаемые музеем использованные материалы проходят нелегкий путь, представляющий собой научно аргументированный подбор объектов музейной значимости. В музеях практикуется подготовительный способ объектов в скоротечное сохранение, данное непременно закрепляется особым действием способа из-за подписью собственника и работника музея. В то же время вносится предание, включающая данные о объектах, популярные передающему личности: о происхождении объектов, события бытования, методах использования. Данный акт расписывается собственником объектов. Предание может помочь обнаружить музейную роль объектов, однако имеет необходимость в контроле в процессе их последующего исследования.

Далее объекты подвергаются экспертизе профессионалов музея, а в случае если имеется потребность – в таком случае и иных органов, предоставляющих писчие решения.

На основе действия способ, решений профессионалов и своих исследований конечности фондово-покупочной комиссии музея разрешают проблема о способе либо возврате объектов, о введении принимаемых объектов в актив музейных объектов либо в актив учено-добавочных использованных материалов. Разрешают проблема о цены закупаемых объектов, и все без исключения данное закрепляется в особом протоколе.

После извлечения музеем единства собственника в передачу объектов в предлагаемых обстоятельствах, из-за конкретную цена оформляется документ о способе объектов в непрерывное сохранение. Этот документ тарифицируется начальником, о чем говорит его роспись и печать музея. Затем объекты записываются в поступлении фонда музейных объектов, либо в фонд учебно-добавочных использованных материалов и переходят надлежащим хранителям фондов согласно особым сдаточным бумагам.

Итак, были проанализированы проблемы, способы и методы комплектования, концепция учета и документирования материалов музея, был изучен способ и вид сбора материалов. Способ занесения, экспонатов в хранилище. Были изучены методы хранения и учета информации об экспонатах.

1.2. Построение модели как есть

Сотрудники музея ТГУ всегда заняты тем, что сверяют движение экспонатов, или заполняют книгу поступления новых экспонатов. Описание музейного экспоната — длительная и трудоемкая процедура, она нужна для того, чтобы в случае пропажи, а затем находки можно было опознать предмет. Представьте, как описать грампластинку прошлых лет и не перепутать её с другой? Или шариковую ручку? Лазерный диск?

У сотрудников музейного фонда всегда есть план работы на день и на год. Они должны сверять наличие материала с тем, что есть в книгах учета.

До настоящего момента музейные работники музея ТГУ ведут учет экспонатов по «амбарным книгам» — каждое произведение искусства заносится в книгу учета вручную.

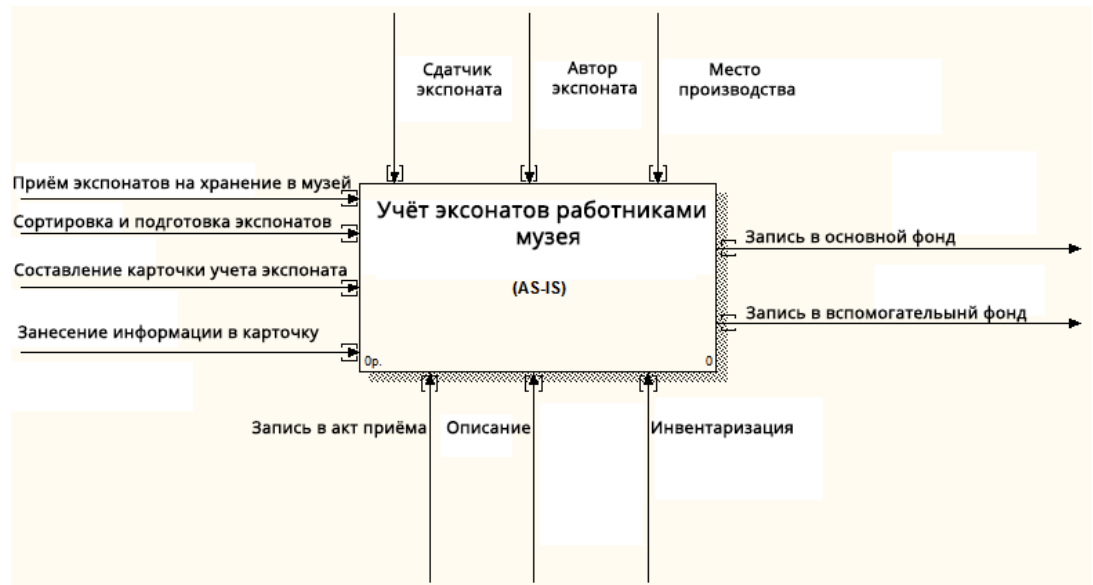


Рисунок 1.1- Построение модели как есть музея ТГУ

Экспонаты часто перемещаются: из фондов на экспозицию, из зала в зал, они «гастролируют» по музеям других городов и возвращаются назад, поэтому очень легко запутаться, сделать ошибку при заполнении карточки экспоната или вообще потерять экспонат.

Таким образом, в результате построения модели AS-IS, были выявлены следующие недостатки:

- долгий и трудоёмкий сбор информации о материале;
- неудобное хранение информации;
- громоздкое хранение информации;
- долгий поиск по карточкам экспонатов;
- невозможность демонстрации экспонатов более широкой публике.

1.3 Формирование требований к новой системе

Если сделать грамотное представление музея в интернете, то можно выделить следующие пункты, которые позитивно повлияют на него целом:

- сайт музея позволяет ему глубже интегрироваться в систему отечественных и иностранных организаций подобного профиля;
- появиться еще одна возможность для презентации своих коллекций;
- у музея появляется также возможность привлечь недостающие ему ресурсы как бы извне;
- электронный ресурс для музея может выступить в роли экспериментальной площадки для проектирования и разработки новых музейных моделей;
- музей по средствам своего электронного ресурса в значительной мере сможет расширить число своих как виртуальных, так и реальных посетителей, при этом процесс приобщения к культуре происходит в очень удобном формате.

С точки зрения объективности стоит подметить, что подавляющее большинство отечественных музеев, имеющих свой интернет ресурс не могут похвастаться хорошо разработанным сайтом, который активно используют выше представленные пункты, которые положены в основу деятельности любого полностью, удовлетворяющего всем требованиям разработанного музейного сайта и это является для них перспективой будущего. Опыт отечественных и зарубежных музеев говорит о том, что в современном мире, каждый музей должен иметь свой собственный информационный ресурс. На основании проведенного выше анализа представляется возможным выдвинуть функциональные требования разрабатываемой информационной системе:

- информационная система должна быть размещена на существующем сайте ТГУ <http://www.tltsu.ru>;
- информационная система должна обладать функционалом для записи данных об экспонатах;

- информационная система должна иметь достаточно гибкий функционал, для осуществления информативной работы с экспонатами музея;
- информационная система должна иметь достаточно вычислительных мощностей для обеспечения шифрования данных;
- информационная система информационная система должна обладать подключением к внешней сети [1].

В результате выдвинутых функциональных требований следует сделать выбор о том, каким образом лучше осуществить подход для разработки информационной системы музея Тольяттинского государственного университета.

1.4 Выбор метода для создания программы учета материалов для музея «ТГУ»

В итоге следует сделать выбор о том, как разработать программы учета материалов для музея «ТГУ»

Следует ли совершенствовать уже полностью готовую информационную систему или же разрабатывать свою. Для более наглядного выбора средства разработки следует рассмотреть таблицу 1.1, в которой производится сравнение методов улучшения информационной системы музея ТГУ в зависимости от требований к функциональной части.

Таблица 1.1 – Сравнение достоинств ИС музея и традиционного музея

	Сравнение	ИС музея	Классический музей
1	Оценка стоимости создания	Небольшие затраты	Огромные затраты
2	Время, потраченное на создание	Несколько месяцев	Несколько лет
3	Посещений гостей в день	Не ограничено	Ограничено
4	Территориальный охват	Из любой точки планеты	Только из населенного пункта
5	Площадь музея	Неограниченна	Ограничена
6	Рекламная мощность музея	Оперативное обновление в любое время	Обновление рекламы требует времени
7	Месторасположение музея	-	Экономически целесообразно только в больших городах
8	Выбор языка для гостей музея	Выбор языка не ограничен	Есть ограничения
9	Число экспонатов музея	Не ограничено	Ограничено

10	Ознакомление с новыми экспонатами	В любое время	Имеются ограничения
11	Обновление экспозиции музея	В любое время	Как правило, по заранее составленному плану
12	График работы музея	Круглосуточно	Определенное время

Но, разумеется нужно всегда помнить о таких замечательных плюсах традиционного музея, как общение напрямую с экскурсоводами в режиме реального времени, атмосфера вдохновения, историческая подпитка, возможность увидеть множество уникальных экспонатов.

Виртуальный музей и сама его идеология во многом похожа на идеологию базы данных, но конечно она должна отличаться по нескольким важным показателям: Идеология виртуального музея может быть похожа на идеологию создания базы данных, но должна иметь ряд принципиальных отличий:

- нет обязательного условия показа всех экспонатов виртуального музея, необходимым и достаточным будет рассказ о наиболее известных людям экспонатах, которые пользуются спросом, рассказы должны быть очень подробными и наглядными, можно их иллюстрировать;
- количество экспонатов не имеет принципиального значения;
- ради возможности представить людям экспонаты, которые хранит музей, собственно и создается его виртуальная версия.

Рассмотрим структуру четырех музеев в таблице № 1.2. А именно : Эрмитаж, музей Кремль, музей Радио и телевидения, музей русского примитива. Необходимо сделать сравнение по наличию данных о музеях на сайтах, информации о музее (контактной), выставках и т.д.; по наличию цифровых коллекций; размещение на сайте различных научных материалов; информации о постоянной экспозиции музея; по отсутствию или наличию экскурсии в видеоформате; по возможности нахождения каких-либо экспонатов в поиске. Такой анализ был проведен с виртуальным Эрмитажем, как с наиболее удачным примером web-сайта из данной категории.

Таблица 1.2 Сравнение информационных систем музеев

Параметр	ИС Эрмитажа (http://www.hermitagemuseum.org/wps/portal/hermitage)	Виртуальный музей Кремля (tours.kremlin.ru/#/ru&1_5)	Виртуальный музей «Радио и телевидения»	ИС музей русского примитива www.museum.ru/museum/primitiv/
1 История музея	+	-	-	-
2 Различная информация о музее	+	+	-	-
3 Коллекции (цифровые, отдельные предметы)	+	-	+	+
4 Научные материалы (статьи)	+	+	+	+
5 Постоянная экспозиция музея	+	-	+	-
6 Видеозаписи по музею	+	+	-	-
7 Система поиска экспонатов	+	+	-	-

После анализа, проведенного с виртуальными музеями, мы можем определить следующие возможности, которые выделяются как главные и функциональные, таковые представлены в таблице 1.3

Таблица 1.3 Наиболее приоритетные функциональные возможности виртуальных музеев

Для посетителей сайта	Для организаторов сайта
<ol style="list-style-type: none"> 1. удобства навигации по ИС; 2. быстрый поиск необходимой информации; 3. посещение выставок, размещенных на ИС. 	<ol style="list-style-type: none"> 1. размещение информации о своей компании на виртуальном стенде; 2. представление информации о достижениях и предложениях; 3. получение статистических сведений о посещаемости и рейтинге виртуального музея.
<ol style="list-style-type: none"> 1. получение полной информации о музее; 2. возможность оставить отзыв о ресурсе в целом или отдельном компоненте, установить рабочие контакты; 3. возможность ознакомиться с новостными сообщениями информационной системы. 	<ol style="list-style-type: none"> 1. внесение изменений в информацию, размещенную на виртуальном стенде; 2. размещение на ресурсе виртуального варианта реальных выставок как их логическое продолжение под тем же названием и с информацией о состоявшихся на ней событиях; 3. возможность разместить специализированную виртуальную выставку, посвященную деятельности какой-либо ассоциации, гильдии, общества или союза предпринимателей.

В результате полного анализа о деятельности музеев, способа сбора информации, представления экспонатов посетителям музея, следует сделать вывод о необходимости создания своего программного продукта для учета материалов музея ТГУ. Создание комплексной информационной системы, обеспечивающей реализацию основных операций на единой музейной базе данных, организующей и упорядочивающей все процессы подготовки документов, способно дать реальную экономию трудозатрат и привести к созданию сводного каталога музейного собрания. Современный интерактивный музей должен научиться управлять этой информацией [3].

Глава 2. Анализ и выбор средств программной реализации сайта музея ТГУ

2.1 Анализ средств проектирования базы данных

Огромное число проектов направлено именно на претворение в жизнь ИС, при создании которых происходит обработка различной сложности данных.

Задача, которая ставится при этом – проектирование базы данных конкретного типа. Разработанная в итоге информационная система должна удовлетворять требованиям, которые заданы изначально. Это определенные функции и содержание. При условии заранее заданных пределов и ограничений [8,9].

Функциональные требования, которые могут быть представлены.:

- выдача отчетов о поступивших экспонатах в музей за последние 6 месяцев;
- выдача отчетов по кварталам;
- сортировка экспонатов по определенным группам;
- быстрый поиск по базе данных.

Примеры ограничений:

- максимальное время, отпущенное на проект;
- технические ограничения;
- количество денежных средств, которое можно на него потратить.

Существуют и другие ограничивающие факторы, в зависимости от объекта реализации, сложности и специфики проекта. При использовании информационных технологий расчет всегда ведется на онлайн работу.

Также необходимо принимать во внимание технологические средства, которые могут быть доступны для выполнения проекта.

Приведем пример. Требуется реализация базы данных в архитектуре "файл-сервер". Эта база должна подходить под набор определенных требований по определенным параметрам:

- адаптируемость программы
- функциональность при работе
- быстрая обработка и реагирование;
- хорошая пропускная способность и многозадачность;
- минимальное время на реакцию, сопровождаемое точностью;
- безопасность процессов

2.2 Выбор архитектуры приложения

Базы данных содержат различного рода архитектуры. Это такие данные, которые хранятся в базах и описываются моделями представления этих данных. Для создания базы данных больше подойдет реляционная модель базы данных. Архитектурой базы данных будет система клиент-сервер [5, 6].

В нашем случае используется двухзвенная архитектура. В любой сети, Архитектура по системе «клиент-сервер» создает принципы по организации взаимодействия в сети, которая имеет сервера, узлы-поставщики специфических функций и потребителей этих самых функций. Мы используем двухзвенную архитектуру. В каждой сети, которая построена в условиях современных технологий, есть элементы взаимодействия.

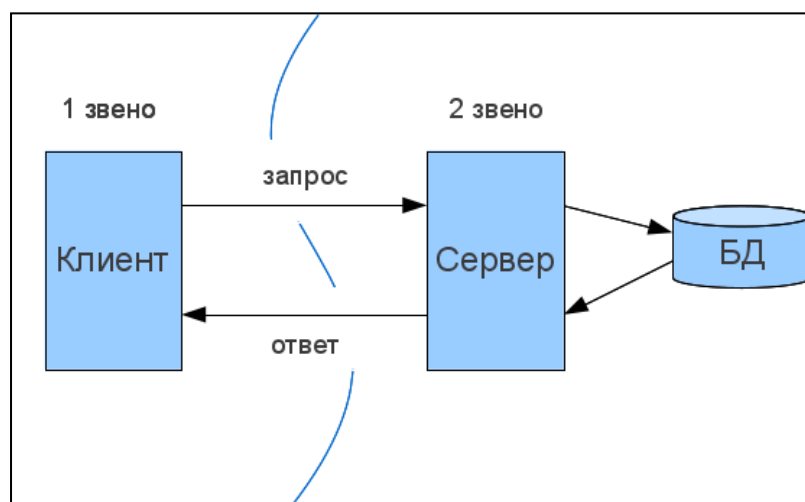


Рисунок 2. 2 - Двухзвенная клиент-серверная архитектура.

Архитектура называется 2-х звенной из-за того, что необходимо производить распределение 3-х компонентов между клиентом и сервером. Они базовые.. Т.е. сервер не вызывает сторонние сетевые приложения и не обращается к сторонним ресурсам для выполнения какой-либо части запроса.

Расположение компонентов на стороне клиента или сервера определяет следующие основные модели их взаимодействия в рамках двухзвенной архитектуры:

- сервер терминалов — распределенное представление данных;
- файл-сервер — доступ к удаленной базе данных и файловым ресурсам;
- сервер БД — удаленное представление данных;
- сервер приложений — удаленное приложение.

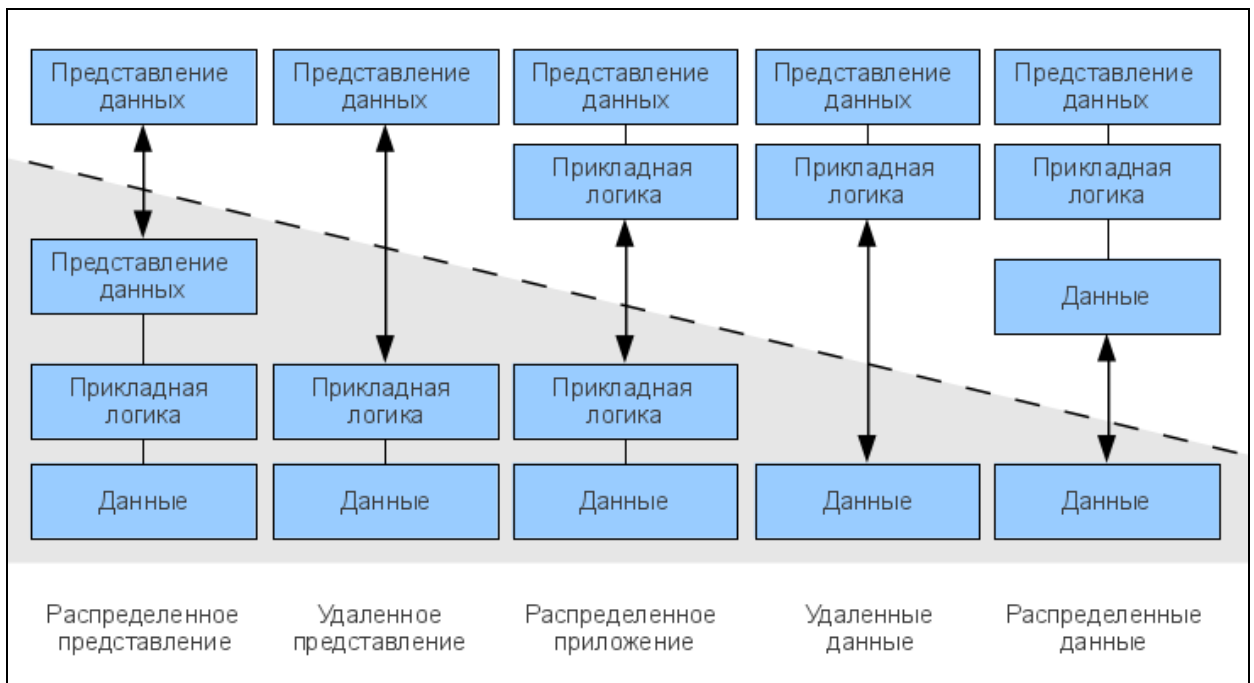


Рисунок 2.3 - Модели клиент-серверного взаимодействия

Сервер базы данных - это программа, которая запускается на машине-сервере и обслуживает доступ клиентов к базе данных. Сервер обслуживает запросы клиентов по доступу к базе данных, обновлению данных, защите

данных, резервному копированию и т.д., сервер обслуживает сразу несколько клиентов.

Положительным моментом организации ИС по архитектуре клиент-сервер является удачное сочетание централизованного хранения, обслуживания и коллективного доступа к общей корпоративной информации и индивидуальной работой пользователей над персональной информацией.

При такой архитектуре сервер БД обеспечивает выполнение основного объема обработки данных. Формируемые пользователем или приложением запросы поступают к серверу БД в виде инструкций языка SQL. Сервер БД выполняет поиск и извлечение нужных данных, которые затем передаются на компьютер пользователя [11].

Важнейшим достоинством применения БД в ИС является обеспечение независимости данных от прикладных программ.

В нашем случае удобно использовать архитектуру клиент-сервер. Преимущества архитектуры клиент-сервер:

- делает возможным, в большинстве случаев, распределение функций вычислительной системы между несколькими независимыми компьютерами в сети. Это позволяет упростить обслуживание вычислительной системы. В частности, замена, ремонт, модернизация или перемещение сервера не затрагивают клиентов; все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа;
- позволяет объединить различных клиентов. Использовать ресурсы одного сервера часто могут клиенты с разными аппаратными платформами, операционными системами и т. п.;

К недостаткам архитектуры можно отнести:

- неработоспособность сервера может сделать неработоспособной всю вычислительную сеть;
- поддержка работы данной системы требует отдельного специалиста - системного администратора;

Архитектура клиент-сервер позволит получать информацию о расписании по любому пользователю в любое время и с любого рабочего места, которое подключено к локальной сети. При этом сами данные будут храниться на отдельном сервере [13].

2.3 Выбор системы управления базой данных

Важным этапом разработки приложения базы данных служит именно этап выбора БД: какая система управления наиболее подходит. В конечном счете она должна быть удовлетворяющей потребности и текущих, и будущих веб-расписаний. Нужно обязательно брать в расчет денежные затраты, которые будут происходить для приобретения оборудования, системы как таковой, на создание ПО на основе системы, на процесс обучения сотрудников. СУБД должна быть соответствующей необходимым требованиям, среди которых ее функциональные возможности возможности:

- функция мобильности, а именно это независимость системы от той области, внутри которой протекает ее работа. Каждый процесс должен быть автоматизирован и выполнять свою собственную функцию;
- функция распределенности, причина использования информационной системы на основе базы данных – это желание соединить взгляды на всю информационную область организации целиком. Объединение обработки и хранения всех данных на уровне одного сервера является очень подходящим и простым способом. Но не всегда можно это осуществить,

поэтому люди и применяют базу данных. Разные системы по разному управляют данными;

- масштабируемость, обязательно нужно учитывать, выбирая СУБД, имеет ли эта система возможность быть соответствующей при росте систем информации. Такой рост выражается как в увеличении количества пользователей, так и объема данных в хранилище и объеме информации, находящейся в стадии обработки;
- сетевые возможности, часто системы разрешают использование широкого диапазона протоколов сети и служб по работе и администрированию.

Производители СУБД производят средства создания и выпуска специальных приложений для их систем. Они помогают, в частности, пользоваться возможностями сервера наилучшим образом, стоит рассмотреть также и возможности средств разработки приложений [17, 19].

Рассмотрим эти средства:

- проектирование, многие системы обладают автоматическим проектированием. Это базы данных и прикладные программы. Они очень различаются у разных производителей;
- перевод на многие языки, это в наше время очень важная функция, которая увеличивает области применения приложения;
- WEB- приложения, использование интернета является неотъемлемой частью жизни. Многие средства имеют инструменты для онлайн-версий программ;
- разный язык программирования, это сильно ускоряет действие и функционал приложения, а также делает систему доступной разработчикам;
- оптимизация запросов, то есть выбор способа выполнения, который вырабатывает оптимальный план, используя семантику преобразований и

синтаксические нормы. Поскольку часто использование непроцедурного языка делает выполнение запроса бесполезным;

- надежность системы представляет собой бережное сохранение информации, которое будет производиться в любом случае (даже при сбоях), а также это работа в безотказном режиме и защита от чужого вмешательства;
- возможность восстановления после сбоя, работа всей системы может нарушиться при программных и аппаратных нарушениях. Поэтому от механизма восстановления зависит вся дальнейшая эффективность работы системы;
- возврат (откат) всех изменений, транзакция должна выполняться либо целиком, либо не происходить вообще. Это означает, что все результаты будут аннулированы в случае если транзакция не выполнена;
- резервное копирование, носители информации могут быть повреждены и выведены из строя, если произошел аппаратный сбой. В данном случае невозможно восстановить потерянные данные если не было произведено резервное копирование (хотя бы частично). Это крайне важная функция и ее нельзя недооценивать;
- многоуровневая защита, система информации содержит в себе секретные данные, следовательно для защиты от взлома используют идентификацию пользователя в системе.

Требования, предъявляемые к рабочей среде:

- поддерживаемые паратные платформы;
- самые минимальные требования касательно оборудования;
- максимальный размер внутренней памяти;
- информационные системы, под управлением которых способна работать СУБД;
- скорость работы;
- производительность.

Из множества числа современных СУБД, используемые в сфере Internet, были выбраны следующие, наиболее подходящие варианты: MySQL, PostgreSQL, Microsoft SQL Server. Приведем сравнительный анализ СУБД для выбора оптимальной на основе основных критерий к СУБД, требований к проектируемой БД, а также возможности реализации (таблица 1) [7].

Таблица 2.1 – Сравнительный анализ СУБД

Критерии оценки	Microsoft SQL Server	PostgreSQL	MySQL
Размер базы данных более 1 гигабайта	+	+	+
Доступность для большого количества пользователей	+	+	+
Защищенность web-сервера	+	+	+
Защита данных	+	+	+
Языковая мощность SQL	+	+	-
Минимальные требования к ЭВМ	-	+	+
Легкость настройки	-	+	+

По завершении проведенного анализа, нами выбрана MySQL. Самая быстрая и небольшая СУБД. Помимо прочего она бесплатная. Это самый оптимальный вариант по доступу пользователей, производительности и по финансовым затратам, а точнее их отсутствию. Множество пользователей сможет моментально получать ответы на их запросы согласно нашему проекту, что и было изначальной целью. Таким образом мы получаем решение поставленной задачи. Теперь давайте рассмотрим как происходит проектирование приложений системы.

2.4 Проектирование приложений системы

PHP (Hypertext Preprocessor - Препроцессор Гипертекста) – это распространенный язык для сценариев общего назначения и с открытым исходным кодом.

Чтобы реализовать данный проект мы выбрали язык PHP. Здесь основополагающим фактором остается практичность. То есть работа должна быть быстрой, эффективной, решение вопросов и задач должно протекать максимально четко и конверсивно, PHP имеет свои достоинства:

- эффективная работа;
- простота;
- традиционность;
- гибкость системы;
- безопасность.

PHP - язык программирования, специально разработанный для написания Web-приложений (скриптов, сценариев), исполняющихся на Web-сервере. Языковой синтаксис часто базируется на синтаксисе Java, C и Perl. (они похожи с PHP) Следовательно, профессиональному программисту будет очень просто

работать с PHP. С другой стороны, он проще, нежели C, поэтому и обычный веб-мастер поймет его принцип работы.

PHP может состоять как из одной строки, так и из десяти тысяч строк. Это будет зависеть от специфики работы в конкретном случае. Механизмы PHP работают сами, выполняют код после первой экранирующей последовательности и продолжают выполнять, пока он не найдет парную экранирующую последовательность.

Данный язык имеет свойство такое, как возможность включения в html - код страницы, которая будет обработана PHP- интерпретатором. Это очень удобно. То есть мы способны использовать систему для того, чтобы сформировать HTML-документ и обезопасить себя от многочисленных вызовов от внешних сценариев.

Большинство PHP-сценариев, небольших в особенности, обрабатываются с большей скоростью, нежели программы-аналоги. Но, откомпилированные исполняемые файлы будут выполнять работу значительно быстрее. В любом случае, производительности PHP более, чем достаточно для создания серьезных и качественных приложений.

В PHP сочетаются механизмы безопасности, которые должны находиться под контролем. PHP при выполнении и настройке в правильном режиме обеспечивает максимальную безопасность и увеличивает возможности.

PHP способна работать в безопасном режиме. Он ограничивает возможности применения PHP пользователями по некоторым показателям. К примеру, время на выполнение запроса и размер памяти.

Администратор также ставит ограничения на каталог. То есть пользователь сможет смотреть и выполнять сценарий PHP не в том объеме, в каком обычно [18].

В стандартный набор функций PHP входит ряд надежных механизмов шифрования. Другое преимущество заключается в том, что исходный текст

сценариев PHP нельзя просмотреть в браузере, поскольку сценарий компилируется до его отправки по запросу пользователя.

К стандартным функциям относится ряд механизмов для шифра.

Еще одним достоинством является невозможность просмотра исходного текста сценария через браузер, так как он компилируется перед его отправкой по запросу.

Данный язык программирования встраиваемый и имеет такую отличительную особенность, как гибкость к разным потребностям, о которой говорилось выше.

Обычно его рекомендуют использовать совместно с HTML, но он также интегрируется и в JavaScript, WML, XML и др. Помимо всего прочего, профессионально структурированные приложения PHP отлично расширяются при необходимости.

Сценарии передаются любым устройствам, имеющим браузеры. А значит и сотовым телефонам, электронным книгам, планшетам, и конечно же ноутбукам и ПК.

У пользователей нет ограничений по выбору серверов. Это связано с тем, что PHP не имеет кода под конкретный web сервер. Это язык, который не зависит от платформы и может использоваться и в Unix, и в Windows.

Система «1С-Битрикс: Управление сайтом» продаётся в одной из семи возможных редакций. А конкретно: «1С-Битрикс: Управление сайтом» продаётся в одной из семи составленных фирмой-разработчиком редакций (Старт, Стандарт, Эксперт, Малый бизнес, Бизнес, Портал, Большой бизнес), определяющих набор модулей и функциональность системы. Они определяют функциональность и набор необходимых модулей. Сегодня доступными являются 26 таковых: главный, Управление структурой, Информационные блоки, Поиск, Соц сети, защита проактивная, Компрессия, Веб-формы, рассылки, подписки, опросы, форумы, блоги, Фотогалерея, аналитика, Техподдержка, реклама, Почта, Обучение, Тестирование, Переводы, Валюта ,

Веб-сервисы, каталог, онлайн-магазин, Документооборот, AD/LDAP, Мониторинг производительности.

Model-view-controller (MVC, «модель-представление-поведение», «модель-представление-контроллер», «модель-вид-контроллер») — Это схема нескольких шаблонов проектирования. С их помощью данные приложения, интерфейс и контакт с пользователями делятся на 3 составляющие части, чтобы создание и модификации одной из составляющих не влекла за собой сильные изменения других.

Шаблон (MVC) имеет простой путь для структурирования приложения. Основной целью ставится деление. Пользовательский интерфейс и бизнес-логика должны быть отдельными структурами. В конечном итоге, легче выполнять масштабирование, тестирование, сопровождение и реализацию, что изображено на рисунке 2.3

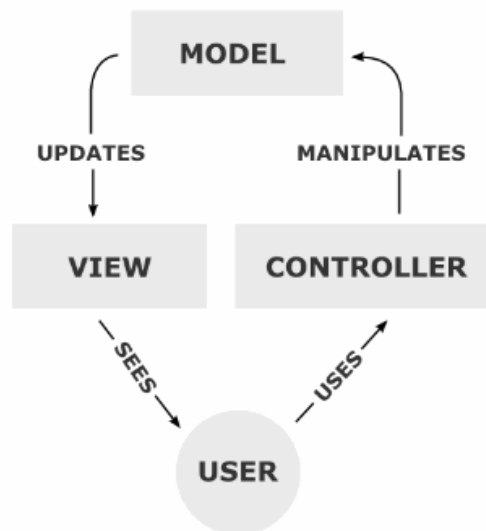


Рисунок 2.3- Концептуальная схема шаблона MVC

В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением.

Последовательность работы приложения MVC можно расписать так:

- когда человек заходит в ресурс, скрипт инициализации делает экземпляр приложения, после чего он его запускает. Видна главная страница сайта или другое изображение;
- приложение видит запрос пользователя и находит контроллер и действие по данному запросу. В случае главной страницы, происходит действие по умолчанию, называемое `index`;
- создается экземпляр контроллера и запускается процесс, где находятся, например, вызовы модели. Они считывают информационные данные.

Следующим этапом формируется представление с данными, которые получены из модели и выводятся результаты.

Для чего нужна модель и что это такое? В ней находится бизнес-логика, а также методы ORM и обработки. Таким образом модель становится довольно толстой. Модель — содержит бизнес-логику приложения и включает методы выборки (это могут быть методы ORM), обработки (например, правила валидации) и предоставления конкретных данных, что зачастую делает ее очень толстой, что вполне нормально.

Модель не должна напрямую взаимодействовать с пользователем. Все переменные, относящиеся к запросу пользователя должны обрабатываться в контроллере.

Модель не должна генерировать HTML или другой код отображения, который может изменяться в зависимости от нужд пользователя. Такой код должен обрабатываться в видах.

Одна и та же модель, например: модель аутентификации пользователей может использоваться как в пользовательской, так и в административной части приложения. В таком случае можно вынести общий код в отдельный класс и наследоваться от него, определяя в наследниках специфичные для подприложений методы.

Вид — используется для задания внешнего отображения данных, полученных из контроллера и модели.

Виды содержат HTML-разметку и небольшие вставки PHP-кода для обхода, форматирования и отображения данных.

Не должны напрямую обращаться к базе данных. Этим должны заниматься модели.

Не должны работать с данными, полученными из запроса пользователя. Эту задачу должен выполнять контроллер.

Может напрямую обращаться к свойствам и методам контроллера или моделей, для получения готовых к выводу данных.

Виды обычно разделяют на общий шаблон, содержащий разметку, общую для всех страниц (например, шапку и подвал) и части шаблона, которые используют для отображения данных выводимых из модели или отображения форм ввода данных.

Контроллер — связующее звено, соединяющее модели, виды и другие компоненты в рабочее приложение. Контроллер отвечает за обработку запросов пользователя. Контроллер не должен содержать SQL-запросов. Их лучше держать в моделях. Контроллер не должен содержать HTML и другой разметки. Её стоит выносить в виды.

В хорошо спроектированном MVC-приложении контроллеры обычно очень тонкие и содержат только несколько десятков строк кода. Чего, не скажешь о Stupid Fat Controllers (SFC) в CMS Joomla. Логика контроллера довольно типична и большая ее часть выносится в базовые классы.

Модели, наоборот, очень толстые и содержат большую часть кода, связанную с обработкой данных, т.к. структура данных и бизнес-логика, содержащаяся в них, обычно довольно специфична для конкретного приложения [20].

В нашем приложении, будет использоваться данный шаблон (паттерн) проектирования. Контроллер является важным звеном, соединяющим модели, виды и другие части системы в одно приложение. Он обрабатывает запросы пользователя. Контроллер не должен иметь SQL-запросов. Их лучше содержать

в моделях. Контроллер не должен содержать HTML и другой разметки. Её лучше включать в видах.

Лучше, если контроллер тонкий и имеет всего несколько десятков строк. Такое приложение считается удачно спроектированным. Например, SFC в CMS Joomla обладает очень типичной логикой контроллера и ее большая часть уходит в базовые классы [20]. В нашем приложении мы будем использовать следующий шаблон проектирования или паттерн

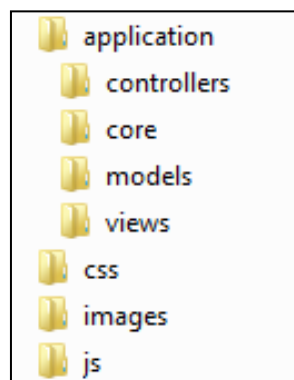


Рисунок 2.4 - Структура файлов и папок в приложении

В папке core (ядро) находятся основные классы для работы. Это:

- Controller – класс в котором реализована, функция «контроллера», и служит родительским классом для «контроллеров»;
- Database – реализует функцию работы с базой данной;
- Model – класс в котором реализована, функция «модели», и служит родительским классом для «моделей»;
- Route - класс, который будет запускать методы контроллеров, которые в свою очередь будут генерировать вид страниц;
- View – класс, который реализует функцию «представления», и служит родительским классом для «представлений»;

2.5 Этапы разработки web-сайта

Была поставлена первая задача: разработать структурную схему проекта web– сайта для использования в среде Internet и локальной сети Тольяттинского

Государственного Университета. По мнению руководителей выпускной квалификационной работы, разрабатываемый web-сайт должен обладать следующими особенностями:

- гибкостью,
- удобной для администраторов системой управления структурой;
- web-сайт должен поддерживать использование звука, графических вставок, анимации, которые должны усиливать эмоционально-ценностный компонент содержания, формировать мотивацию;
- для пользователей должна быть также реализована возможность распечатать любую страницу web-сайта;
- для посетителей сайта должен быть создана виртуальная выставка, в которой пользователи могли бы ознакомиться с музейными экспонатами и задать интересующие их вопросы и получать на них ответы в кратчайшие сроки.

Однако главной задачей проектирования было создание системы управления содержимым, которая бы позволяла вносить изменения на web-сайт с возможностью разграничения прав доступа к содержимому и независимостью от технических специалистов.

Рисунок 2.5- Первая версия страницы редактирования карточки экспоната администратором

2.6 Разработка интерфейса

Титульная страница (главная) любого сайта должна максимально информативно и в сжатом объёме отображать необходимую пользователю информацию о сайте. На главной странице необходимо поместить логотип ТГУ, основное меню сайта (для навигации по его структуре), форму аутентификации (входа зарегистрированных пользователей), регистрационную ссылку (регистрация новых клиентов), ленту новостей, ссылки на информацию предоставляемой сотрудниками музея.

2.7 Техническое обеспечение

Техническое обеспечение - это комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

Важнейший фактор повышения эффективности производства в любой отрасли является улучшение управления. Совершенствование форм и методов управления происходит на основе достижений научно-технического прогресса, дальнейшего развития информатики, занимающейся изучением законов,

методов и способов накопления, обработки и передачи информации с помощью различных технических средств.

Современные технические средства обеспечения управления информационными ресурсами по своему составу и функциональным возможностям весьма разнообразны. К ним относятся средства вычислительной техники, средства коммуникационной техники, средства организационной техники.

Компьютерная техника предназначена, в основном, для реализации комплексных технологий обработки и хранения информации и является базой интеграции всех современных технических средств обеспечения управления информационными ресурсами.

Коммуникационная техника предназначена, в основном, для реализации технологий передачи информации и предполагает как автономное функционирование, так и функционирование в комплексе со средствами компьютерной техники.

Организационная техника предназначена для реализации технологий хранения, представления и использования информации, а также для выполнения различных вспомогательных операций в рамках тех или иных технологий информационной поддержки управленческой деятельности.

Оснащение различной техникой, позволяет экономить управленческие и накладные расходы, проводить более эффективное внутрифирменное планирование, управление и контроль. Обеспечить руководителя как можно быстрее необходимой ему достоверной информацией для принятия оперативных решений возможно с помощью новейших технических средств. Руководитель в своей деятельности использует различные технические средства для принятия управленческих решений, такие как:

- телекоммуникационная связь (телефон, факс, электронная почта, локальная сеть, глобальная сеть Интернет;

- вычислительная техника (персональный компьютер, калькулятор);
- копировально-множительная техника (принтер, сканер, ксерокс);
- информационные системы (телевидение, средства массовой информации - газеты, журналы).

Посредством этих технических средств руководитель принимает в короткое время обоснованные и правильные решения. Поэтому важно не только определить наличие этих средств, но и проанализировать их качественный уровень. Немаловажным здесь является уровень подготовленности персонала, способность его использовать все возможности современных технических средств.

Сейчас, когда появляется новое программное обеспечение, у которого требования к аппаратной части возрастает, возникает потребность в очередной модернизации персональных компьютеров.

Минимальные технические требования:

На стороне сервера:

- IBM-совместимый компьютер с МП Intel Pentium III с минимальной частотой 1ГГц;
- 512 Мб ОЗУ;

На стороне клиента:

- IBM-совместимый компьютер с МП Intel Pentium II;
- 512 Мб ОЗУ.

Требования к персоналу

Требования к персоналу, осуществляющему работу с информационным ресурсом, определяются спецификой работы с персональным компьютером. Работники должны:

- обладать по крайней мере начальными навыками работы с вычислительной техникой и периферийными устройствами, с ПК в частности;

иметь достаточные навыки четкой работы с большими объемами данных, исключая появление ошибок при формировании информационных массивов;

– знать и соблюдать правила техники безопасности при работе с электротехническим оборудованием; иметь допуск к работе с компьютером по состоянию здоровья.

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

Глава 3. Программная реализация информационной системы музея ТГУ

В данной главе описывается самое главное, а именно то, как разрабатывался программный продукт, какие технологии были использованы и как проект работает в целом.

3.1 Анализ принципов построения Web-сайта

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.

3.2 Обоснование и выбор программного обеспечения

Для создания Web-сайта ВКР 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, которые полностью воплощают замысел 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;
- расширение возможностей. 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

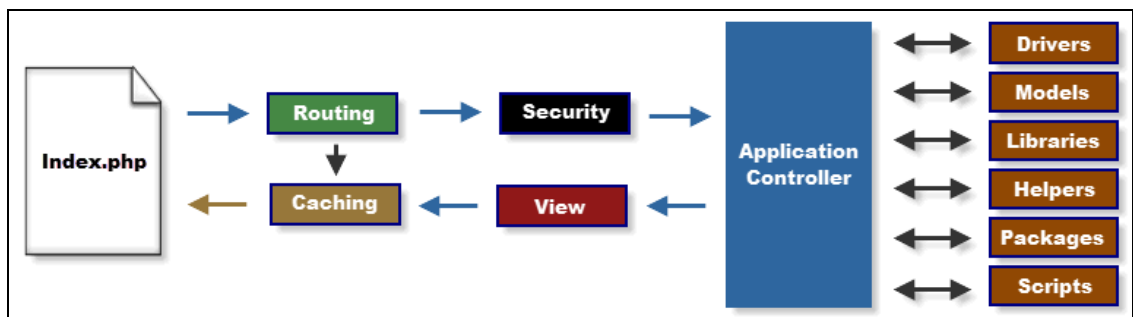


Рисунок 3.1- схема работы CodeIgniter

3.3 Разработка базы данных

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 PHP это лучший вариант для реализации данного программного продукта.

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

В ходе анализа существующих систем, был выбрана следующая иерархия базы данных

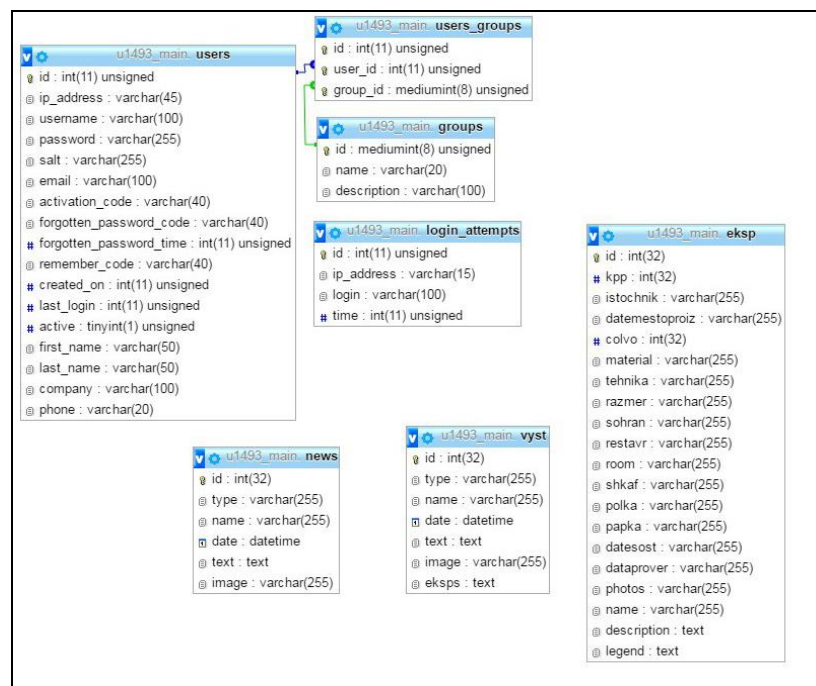


Рисунок 3.2- Иерархия базы данных

- в таблице users хранятся данные о всех пользователях системы;
- таблица groups служит для расположения в ней списка существующих групп пользователей;
- в таблице users_groups происходит сопоставления ID пользователя и групп в которых он находится;
- таблица login_attempts служит для записи фактов не правильной авторизации пользователей, дабы защититься от атак вида брутфорс (перебор паролей);
- в таблице exrs – будут располагаться все данные об экспонатах.

3.3.1 Логическое моделирование базы данных

Логическая модель данных описывает факты и объекты, подлежащие регистрации в будущей базе данных. Основными компонентами такой модели являются сущности, их атрибуты и связи между ними. Как правило, физическим аналогом сущности в будущей базе данных является таблица, а физическим аналогом атрибута — поле этой таблицы. С логической точки зрения сущность представляет собой совокупность однотипных объектов или фактов, называемых экземплярами этой сущности. Физическим аналогом экземпляра обычно является запись в таблице базы данных. Как и записи в таблице реляционной СУБД, экземпляры сущности должны быть уникальными, то есть полный набор значений их атрибутов не должен дублироваться. И так же, как и поля в таблице, атрибуты могут быть ключевыми и не ключевыми. На этапе логического проектирования для каждого атрибута обычно определяется примерный тип данных. Конкретизация происходит на этапе физического проектирования, так как различные СУБД поддерживают разные типы данных и ограничения на их длину или точность.

Одной из основных частей информационного обеспечения является информационная база, которая представляет собой совокупность данных, с помощью которых удовлетворяются информационные потребности

управленческих процессов и решаемых задач. Разработка БД выполняется с помощью моделирования данных. Цель моделирования данных состоит в обеспечении разработчика ИС концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных. Наиболее распространенным средством моделирования данных являются диаграммы "сущность-связь" (ERD).

Сущность (Entity) — множество экземпляров реальных или абстрактных объектов (людей, событий, состояний, идей, предметов и др.), обладающих общими атрибутами или характеристиками. Любой объект системы может быть представлен только одной сущностью, которая должна быть уникально идентифицирована.

Атрибут (Attribute) — любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Атрибут представляет тип характеристик или свойств, ассоциированных с множеством реальных или абстрактных объектов (людей, мест, событий, состояний, идей, предметов и т.д.).

Связь (Relationship) — поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Связь — это ассоциация между сущностями, при которой каждый экземпляр одной сущности ассоциирован с произвольным (в том числе нулевым) количеством экземпляров второй сущности, и наоборот.

Существуют несколько типов связей:

- один-к-одному - означает, что один экземпляр первой сущности (левой) связан с одним экземпляром второй сущности (правой). Связь один-к-одному чаще всего свидетельствует о том, что на самом деле мы имеем всего одну сущность, неправильно разделенную на две;

- один-ко-многим - означает, что один экземпляр первой сущности связан с несколькими экземплярами второй сущности. Это наиболее часто используемый тип связи. Первая сущность называется родительской, вторая – дочерней;
- много-ко-многим - означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Тип связи много-ко-многим является временным типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи должен быть заменен двумя связями типа один-ко-многим путем создания промежуточной сущности.

В процессе моделирования логической ER-модели были выделены следующие сущности и атрибуты:

- администраторы (ID, Логин, E-mail, Пароль);
- экспонаты (id, № КПП, источник, дата и место производства, количество, материал, техника, размер, сохранность, комната, шкаф, полка, папка, дата составления, дата проверки, фото, имя, описание, легенда);
- события (Id, название, текст, фото);
- новости (Id, название, текст, фото);
- летопись (Id, название, текст, фото).

В ходе построения логической модели данных были выделены таблицы и соответствующие им атрибуты. Созданная модель в нормализации не нуждается.

Решения, принятые на этом уровне, при разработке модели предметной области, определяют некоторые границы, в пределах которых можно развивать созданную модель данных, в пределах же этих границ можно принимать различные решения.

Следующим этапом будет построение физической модели данных на основе логической.

3.3.2 Физическое моделирование базы данных

Физическая модель БД определяет способ размещения данных на носителях, а также способ и средства организации эффективного доступа к ним. Поскольку СУБД функционирует в составе и под управлением операционной системы, то организация хранения данных и доступа к ним зависит от принципов и методов управления данными операционной системы.

К вопросам организации данных относятся:

- выбор типа записи – единицы обмена в операциях ввода-вывода;
- выбор способа размещения записей в файле и, возможно, метода оптимизации размещения;
- выбор способа адресации и метода доступа к записям.

Стадия физического проектирования БД в общем случае включает:

- выбор способа организации БД;
- разработку спецификации внутренней схемы;
- описание отображения концептуальной схемы во внутреннюю.

В отличие от ранних СУБД, многие современные системы не предоставляют разработчику какого-либо выбора на этой стадии. Реально к вопросам проектирования физической модели можно отнести:

- выбор схемы размещения данных (разделение по файлам или тип RAID-массива);
- определение числа и типа индексов.

Способ хранения БД определяется механизмами СУБД автоматически по умолчанию на основе спецификаций концептуальной схемы БД, и внутренняя схема в явном виде в таких системах не используется. Внешние схемы БД обычно конструируются на стадии разработки приложений.

Физическая модель данных, как правило, создается на основе логической, поэтому каждому объекту логической модели соответствует объект физической

модели (хотя соответствие может быть неоднозначным). В физической модели данных сущности логической модели данных соответствует таблица, экземпляру сущности – строка в таблице, а атрибуту – колонка таблицы. Кроме перечисленных выше объектов, физическая модель может содержать объекты, тип которых зависит от СУБД: индексы, представления, последовательности, триггеры, процедуры и т.п. Если в логической модели данных не имеет большого значения, какой тип данных у атрибута, то в физической важно описать всю информацию о конкретных объектах.

3.3.3 Концептуальная модель базы данных

Концептуальная модель - это отражение предметной области, для которой разрабатывается база данных. Не вдаваясь в теорию, отметим, что это некая диаграмма с принятыми обозначениями элементов. Так, все объекты, обозначающие вещи, обозначаются в виде прямоугольника. Атрибуты, характеризующие объект - в виде овала, а связи между объектами - ромбами. Мощность связи обозначаются стрелками (в направлении, где мощность равна многим - двойная стрелка, а со стороны, где она равна единице - одинарная). На рисунке 3.3 рассмотрим концептуальную модель базы данных ИС музея Тольяттинского государственного университета

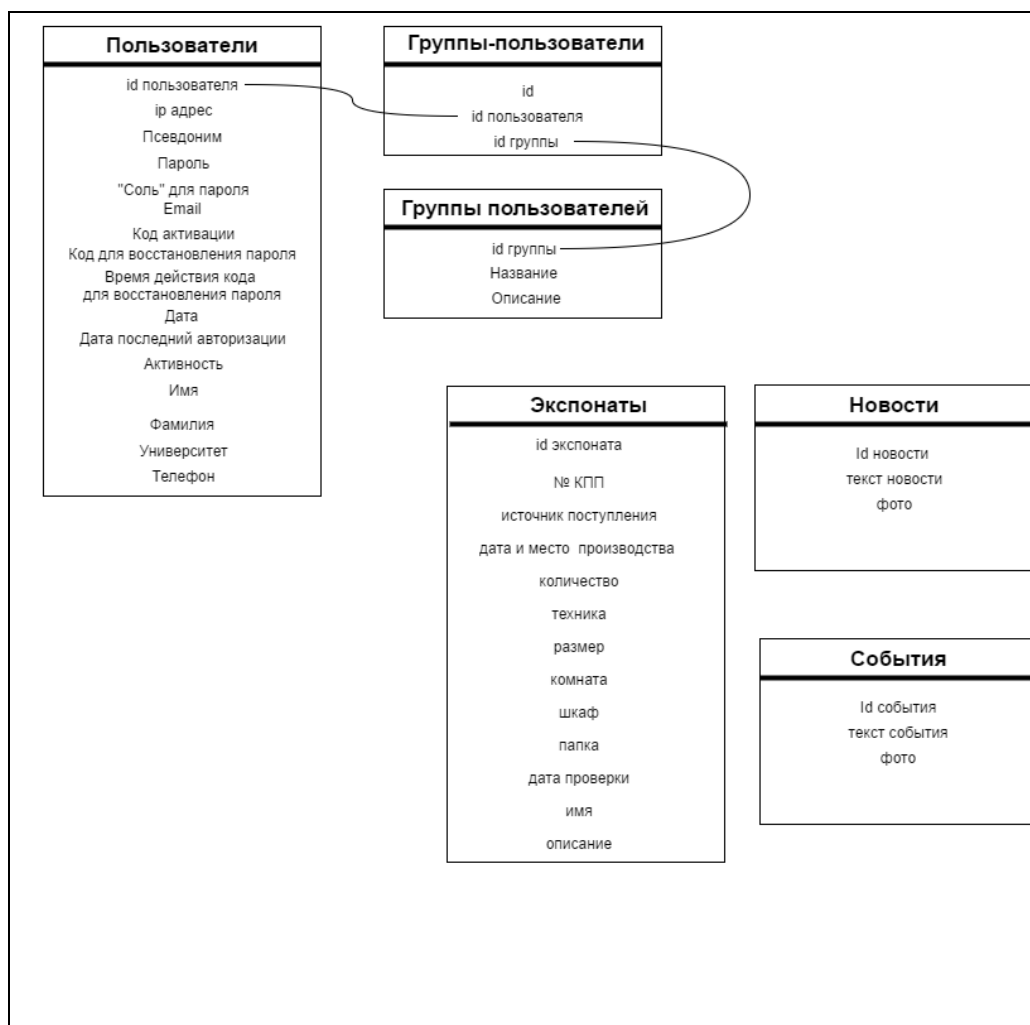


Рисунок 3.3- Концептуальная модель базы данных ИС музея ТГУ

3.4. Базовые классы компонентов

В настоящем разделе будут описаны классы, и функции которые были написаны в ходе разработки программы автоматизированного учета музея ТГУ.

3.4.1 Класс Main

Иерархия базовых классов компонентов берёт своё начало с класса Main. Он реализует основную функциональность по выводу запрашиваемой информации в браузер пользователя. На рисунке 3.4 рассмотрим пример работы класса Main, так же рассмотрим функции данного класса

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Main extends CI_Controller {

    /**
     * Index Page for this controller.
     *
     * http://example.com/index.php/welcome/index
     * - or -
     * Since this controller is set as the default controller in
     * config/routes.php, it's displayed at http://example.com/
     *
     * So any other public methods not prefixed with an underscore will
     * map to /index.php/welcome/<method_name>
     * @see https://codeigniter.com/user_guide/general/urls.html
     */
    public function index()
    {
        $this->load->view('museum/index');
    }

    public function spisokeksp() {
        $data['eksps'] = $this->eksp->spisokeksp();
        //var_dump($this->eksp->spisokeksp());

        foreach($data['eksps'] as $key=>$eksp) {
            $i=1;
            $data['eksps'][$key]['smalldesc'] =
mb_substr($data['eksps'][$key]['description'], 0, 210) . "...";
            foreach(json_decode($eksp['photos']) as $photo){
                $data['eksps'][$key]['photo' . $i] = "/uploads/" . $photo;
                $i++;
            }
        }

        $this->parser->parse('museum/spisokeksp_main', $data);
        //var_dump($data);
    }
}

```

Рисунок 3.4-Фрагмент кода класса Main

- `public function index()` – из модели (класса) `eksp` запрашивает список текущих экспонатов, и затем выводит их на главную страницу;
- `public function eksp($id)` – данная функция на вход получает ID экспоната, после чего происходит вызов функции `fulleksp($id)` из класса `Eksp` которая возвращает массив с данным о запрошенном экспонате, и возвращает обработанные данные в браузер пользователя.

3.4.2. Класс Eksp

Класс `Eksp` один из главных классов во всем проекте, так как именно он отвечает за взаимодействие всей системы непосредственно с экспонатами

,рассмотрим пример работы на рисунке 3.5

```
?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Eksp extends CI_Model
{
    public function addeksp($eksp, $photos){
        $query = "INSERT INTO `u1493_main`.`eksp` (`description`,
`legend`, `kpp`, `istochnik`, `datemestoproiz`, `colvo`, `material`, `tehnika`,
`razmer`, `sohran`, `restavr`, `room`, `shkaf`, `polka`, `papka`, `datesost`,
`datapровер`, `photos`, `name`) VALUES ('$eksp[description]', '$eksp[legend]',
'$eksp[kpp]', '$eksp[istochnik]', '$eksp[datemestoproiz]', '$eksp[colvo]',
'$eksp[material]', '$eksp[tehnika]', '$eksp[razmer]', '$eksp[sohran]',
'$eksp[restavr]', '$eksp[room]', '$eksp[shkaf]', '$eksp[polka]', '$eksp[papka]',
'$eksp[datesost]', '$eksp[datapровер]', '$photos', '$eksp[name]')";
        $this->db->simple_query($query);
    }

    public function uploadphotos($photos, $sid){
        $photki = array();
        $uploaddir = $_SERVER['DOCUMENT_ROOT']."/uploads/".$sid."/";
        @mkdir($uploaddir, 0777);
        for($i=0; $i<=3; $i++){
            if($photos['photo']['tmp_name'][$i]!=""){
                copy($photos['photo']['tmp_name'][$i], $uploaddir.time().basename($photos
['photo']['name'][$i]));
                $photki[]="/".$sid."/".time().basename($photos['photo']['name'][$i]);
            }
        }
        return json_encode($photki);
    }

    public function spisokeksp(){
        $query = $this->db->query('SELECT * FROM `eksp`');
        return $query->result_array();
    }

    public function spisokeksp2($search){
        $query = $this->db->query("SELECT * FROM `eksp` WHERE `name` LIKE ");
    }
}
```

Рисунок 3.5- фрагмент кода класса Экспс

- public function addeksp()– данная функция отвечает за добавление нового экспоната в базу данных, в нее непосредственно передаются все данные экспоната из HTML формы которая располагается в административной части сайта;
- public function fulleksp(\$sid) – 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 listeksps() – в этой функции происходит запрос всех текущих экспонатах, после чего возвращается двумерный массив с

данным о экспонатах;

- `public function uploadphoto($tmpfile)` – данная функция служит для загрузки фотографий экспонатов на сервер, на вход она получает временный адрес, после чего данная функция возвращает адрес куда был записан файл;
- `public function editeksp()` – данная функция отвечает за редактирование конференции, её работа похожа на работу функции `addeksp()` за исключением того что она перезаписывает данные редактируемого экспоната;
- `public function deleksp ($id)` – данная функция служит для удаления экспоната по её ID.

3.4.2. Класс Auth

Класс Auth отвечает за систему пользователей в нашем программном продукте, рассмотрим пример работы на рисунке 3.5

```
<?php defined('BASEPATH') OR exit('No direct script access allowed');

class Auth extends CI_Controller {

    function __construct()
    {
        parent::__construct();
        $this->load->database();
        $this->load->library(array('ion_auth', 'form_validation'));
        $this->load->helper(array('url', 'language'));

        $this->form_validation->set_error_delimiters($this->config-
>item('error_start_delimiter', 'ion_auth'), $this->config-
>item('error_end_delimiter', 'ion_auth'));

        $this->lang->load('auth');
    }

    // redirect if needed, otherwise display the user list
    function index()
    {

        if (!$this->ion_auth->logged_in())
        {
            // redirect them to the login page
            redirect('auth/login', 'refresh');
        }
        elseif (!$this->ion_auth->is_admin()) // remove this elseif if
you want to enable this for non-admins
        {
            // redirect them to the home page because they must be an
administrator to view this
            return show_error('You must be an administrator to view
this page.');
```

```
        }
        else
        {
            $this->data['message'] =
(validation_errors()) ? validation_errors() : $this->session-
>flashdata('message');
```

```
            //list the users
            $this->data['users'] = $this->ion_auth->users()-
>result();
            foreach ($this->data['users'] as $k => $user)
            {
                $this->data['users'][$k]->groups = $this-
>ion_auth->get_users_groups($user->id)->result();
            }

            $this->_render_page('auth/index', $this->data);
        }
    }
}
```

Рисунок 3.6- Фрагмент кода класса Auth

Ниже идет перечисление всех функций с описанием параметров и сообщениями, которые при этом возвращаются функциями messages и errors:

- activate(\$id, \$code=false) — активация аккаунта;
- \$id — id из таблицы users;
- \$code — код активации;
- messages: activate_successful;
- errors: activate_unsuccessful;
- deactivate(\$id) — деактивация аккаунта;
- \$id — id из таблицы users;
- messages: deactivate_successful;
- errors: deactivate_unsuccessful;
- change_password(\$identity, \$old, \$new) — сменить пароль;
- \$identity — идентификатор пользователя;
- \$old — старый пароль;
- \$new — новый пароль;
- messages: password_change_successful;
- errors: password_change_unsuccessful;
- forgotten_password(\$email) — выслать письмо для восстановления пароля;
- \$email — адрес электронной почты, на который нужно выслать письмо
- messages: forgot_password_successful;
- errors: forgot_password_unsuccessful;
- forgotten_password_complete(\$code) — проверить код для завершения восстановления пароля;
- \$code — код из письма для восстановления пароля;
- messages: password_change_successful;
- errors: password_change_unsuccessful;
- register(\$username, \$password, \$email, \$additional_data, \$group_name = false) — создать нового пользователя;

- \$username — имя пользователя;
- \$password — его пароль;
- \$email — адрес электронной почты;
- \$additional_data — метаданные;
- \$group_name — имя группы, к которой нужно причислить пользователя;
- messages: account_creation_successful, activation_email_successful;
- errors: account_creation_unsuccessful, deactivate_unsuccessful, activation_email_unsuccessful
- login(\$identity, \$password, \$remember=false) — авторизовать пользователя;
- \$identity — идентификатор пользователя;
- \$password — его пароль;
- \$remember — запоминать ли пользователя для автовхода;
- messages: login_successful;
- errors: login_unsuccessful;
- logout() — завершить текущий сеанс, выйти;
- messages: logout_successful;
- logged_in() — проверить, авторизован ли пользователь;
- is_admin() — проверить, является ли пользователь администратором;

Общая схема авторизации пользователей в проекте изображена на рисунке 3.7

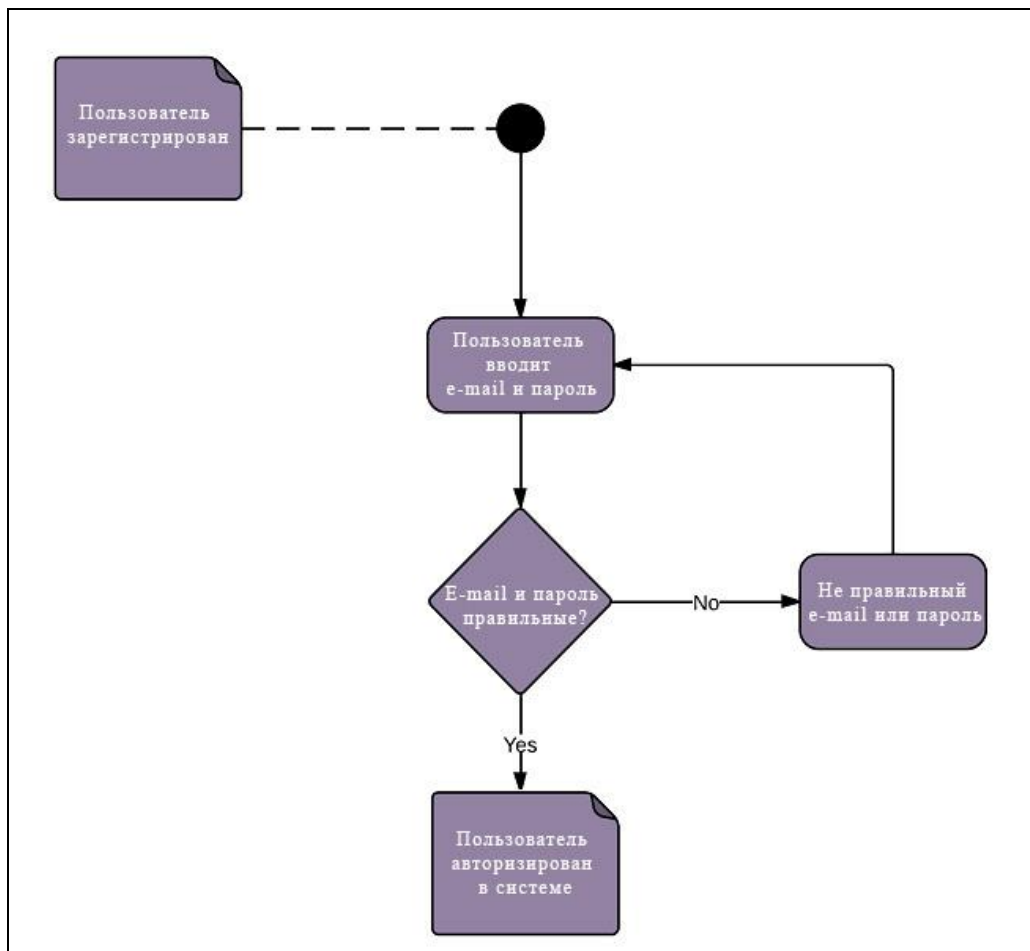


Рисунок 3.7- Общая схема авторизации пользователей в проекте

В результате выполненных действий получилось создать полностью самостоятельный информационный продукт, который отвечает все поставленным требованиям и целям. Разработана база данных. Реализован интерфейс информационного ресурса.

Заключение

В ходе выполнения выпускной квалификационной работы, была создана информационная система для музея ТГУ. Данный программный продукт, предназначен, для организации сбора данных экспонатах с целью занесения этих данных защищённым шифровкой пакетом данных в единую базу данных. Данные предоставляемые созданной ИС, могут позволить улучшить информационные ресурсы музея, облегчить работу работников музея, улучшить бизнес-процессы музея.

Подобная система в дальнейшем может расширять свой функционал, расширяя тем самым актуальность применения данной системы в реальных условиях.

В результате выполнения бакалаврской работы удалось выполнить все поставленные в техническом задании задачи, разработана база данных и подтвердить знания материала преподаваемого в процессе обучения специальности математическое обеспечение и администрирование информационных систем.

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

Учебники и учебные пособия

1. Алексеев, А.П. Введение в Web-дизайн: учебное пособие / А. П. Алексеев./ – М. : СОЛОН-ПРЕСС, 2008. – 192 с.
2. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем / Б. Бейзер./ - СПб.: Питер, 2004. - 318с.: ил.
3. Бородаев Д. А. WEB-сайт как объект графического дизайна: монография / Д. А. Бородаев./ – Харьков, 2004. – 232 с.
4. Грофф Д. SQL: Полное руководство / Д. Грофф, П. Вайнберг./ - К.: ВНУ, 2001. - 816с.: ил.
5. Дари К. AJAX и PHP: разработка динамических веб-приложений / К. Дари, Б. Бринзаре./ – СПб.: Символ-Плюс, 2006. – 336с., ил.
6. Дейт Д. Введение в системы баз данных / Д. Дейт./ — М.: Издательский дом "Вильямс", 2005. — 1328 с.: ил
7. Конноли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика /Т. Конноли, К. Бегг, А. Стратчан./ – М.: Вильямс, 2000. - 1093 с.
8. Корнеев, И. К. Информационные технологии : учебное пособие / И. К. Корнеев./ - М.: Проспект, 2007.- 224 с.
9. Куроуз Д. Компьютерные сети. Многоуровневая архитектура Интернета: 2-е издание / Д. Куроуз, К. Росс./ – СПб.: Питер, 2004. – 765с.
10. Лабберс П. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений / П. Лабберс, Б. Олаберс./ – М.: Вильямс, 2011. – 272с.
11. Лоусон Б. Изучаем HTML5. Библиотека специалиста. / Б. Лоусон, Р. Шарп./ – СПб.: Питер, 2012. – 286с.

12. 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.
13. Руководство по PHP [Электронный ресурс]: PHP: Руководство по PHP – Manual / группа документирования PHP. – Электрон. дан. – [2016]. – Режим доступа: <http://php.net/manual/ru/index.php>
14. Справочное руководство по MySQL [Электронный ресурс]: MySQL Manual. – Электрон. дан. – Режим доступа: <http://www.mysql.ru/docs/man/>
15. Бейли, Л. Изучаем SQL / Л. Бейли – СПб.: Питер, 2012. – 573с.

Литература на иностранном языке

1. Mikael O. CSS Quick Syntax Reference Guide / Mikael Olsson, 2014. – 124с.
2. Kevin, M. Introduction to Non-functional Requirements / Kevin MacG Adams, 2015. – 72с.
3. Chris, P. CodeIgniter: MVC / Chris Pitt, 2012. – 365с.
4. Charles, B. Expert MySQL / Charles Bell, 2012. – 611с.
5. Verona. J. Practical DevOps, Packt Publishing, 2016.

Образцы интерфейсных форм

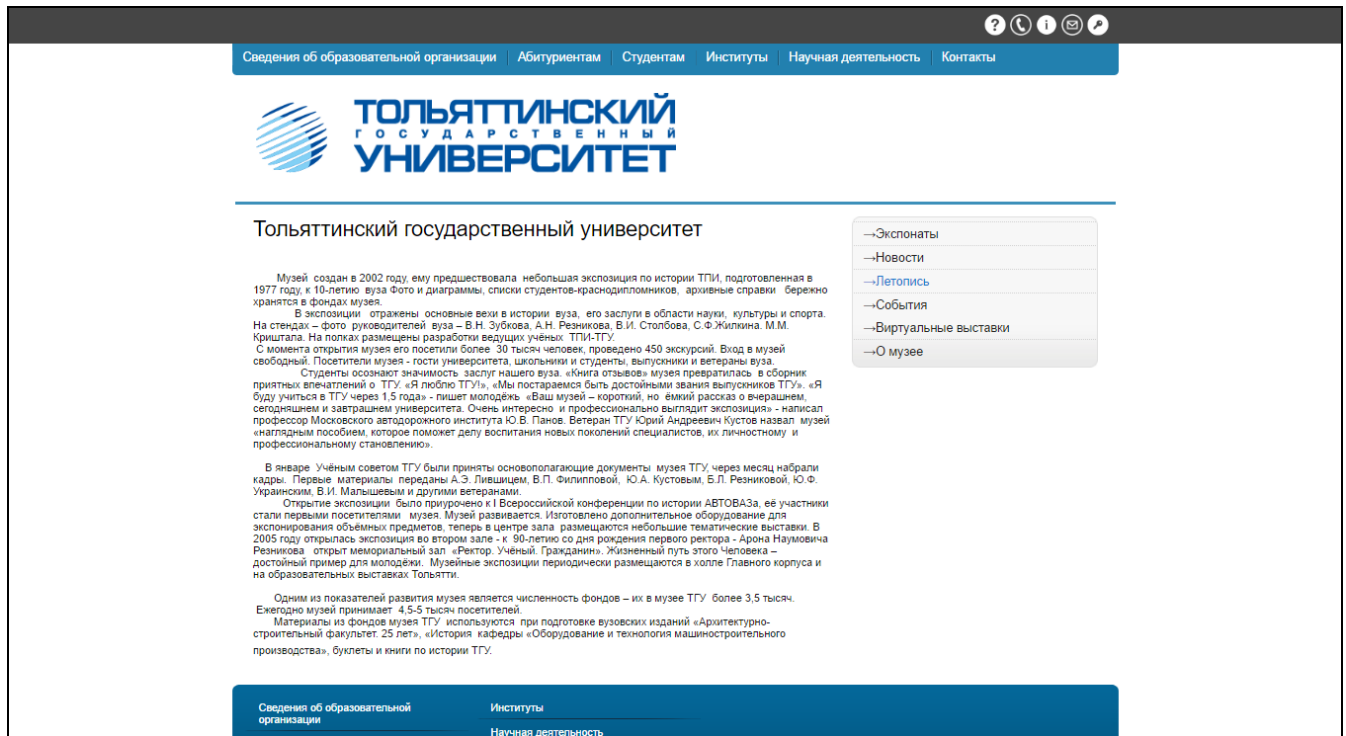


Рисунок А.1 – Главная страница информационной системы музея ТГУ

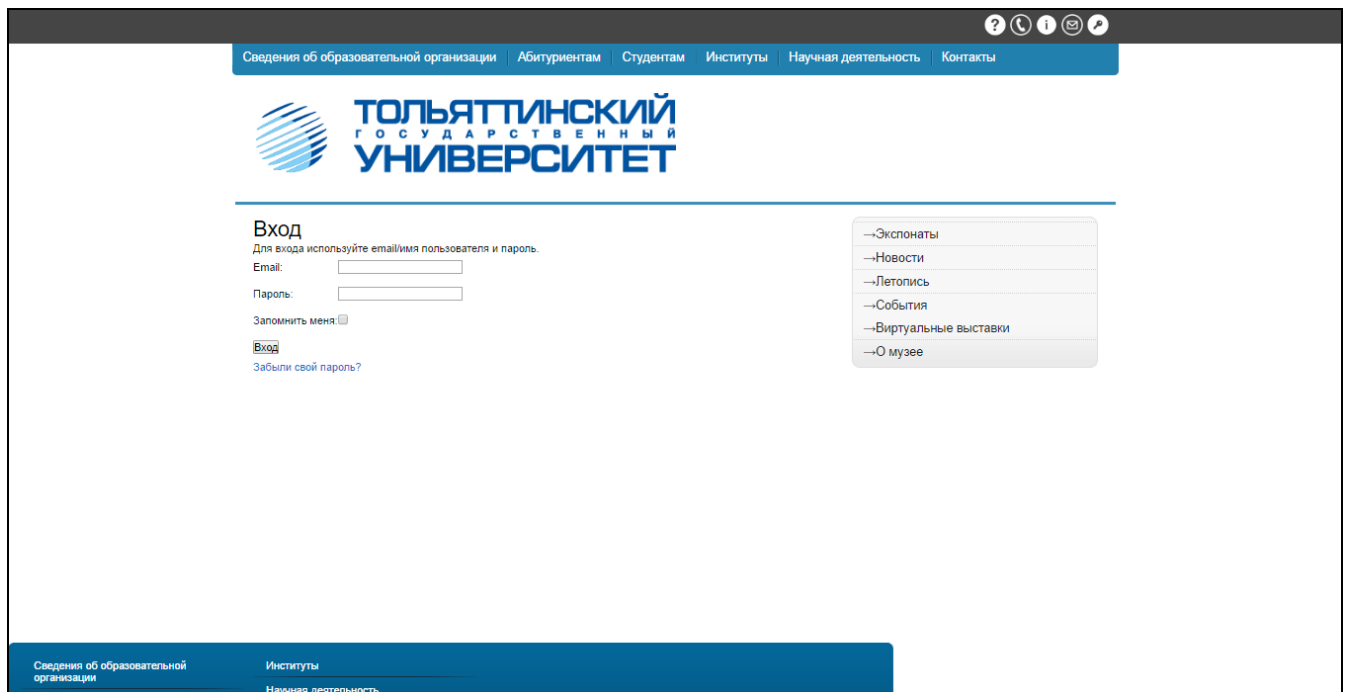


Рисунок А.2 – форма регистрации администраторов ИС музея ТГУ

Сведения об образовательной организации | Абитуриентам | Студентам | Институты | Научная деятельность | Контакты

ТОЛЬЯТТИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Экспонаты | Новости | События | Летопись | Виртуальные выставки

Редактирование экспоната

ИД: 1701 | Перфильева А.А.

Название: Деревянные бруски для каллиграфии" - подарок от делегации университета г.Шаосин (Китай)

Год: 2015 г. Китай

Категория: дерево, краска | Имя обработки: гравировка

Размеры: 1.5 x 4 x 30 (см)

Лотная: []

Реставрационные меры: []

Место хранения:

Комната: [] | Шкаф (стеллаж): [] | Полка: [] | Ящик: []

Дата поступления: 1.10.2015 | Дата окончания: 31.10.2015

Выберите файл | Файл не выбран | Выберите файл | Файл не выбран

Выберите файл | Файл не выбран | Выберите файл | Файл не выбран

В картонной коробочке с малым геометрическим рисунком коричнево-бежевых тонов. Бруски - предмет удлиненной формы, сечение 4-х угольное, грани-плоские

4 июня 2015 года ТГУ встречал официальную делегацию университета г.Шаосин Китай. Гости подарили ректору ТГУ И.Кристалу символичный подарок - особые деревянные

Сохранить

- Экспонаты
- Новости
- Летопись
- События
- Виртуальные выставки
- О музее

Рисунок А.3 – Форма заполнения «карточки» экспоната музея ТГУ

Сведения об образовательной организации | Абитуриентам | Студентам | Институты | Научная деятельность | Контакты

ТОЛЬЯТТИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Экспонаты | Новости | События | Летопись | Виртуальные выставки

Поиск по экспонатам

Найти

Список экспонатов

Добавить экспонат

№ КПП	Название	Действия
1701	"Деревянные бруски для каллиграфии" - подарок от делегации университета г.Шаосин (Китай)	
1698	Копия "Соглашение о сотрудничестве между ФГБОУ ВПО "Тольяттинский государственный университет" и Дацзиний педагогический институт"	
1702	"Приз победителю 1 в мире конкурса массовых открытых онлайн-курсов "EdCrunchAward 2015"	
875	Образец "Хлорбутилкаучук"	

- Экспонаты
- Новости
- Летопись
- События
- Виртуальные выставки
- О музее

Рисунок А.4 – Страница ИС с полным списком экспонатов музея ТГУ (для администратора)

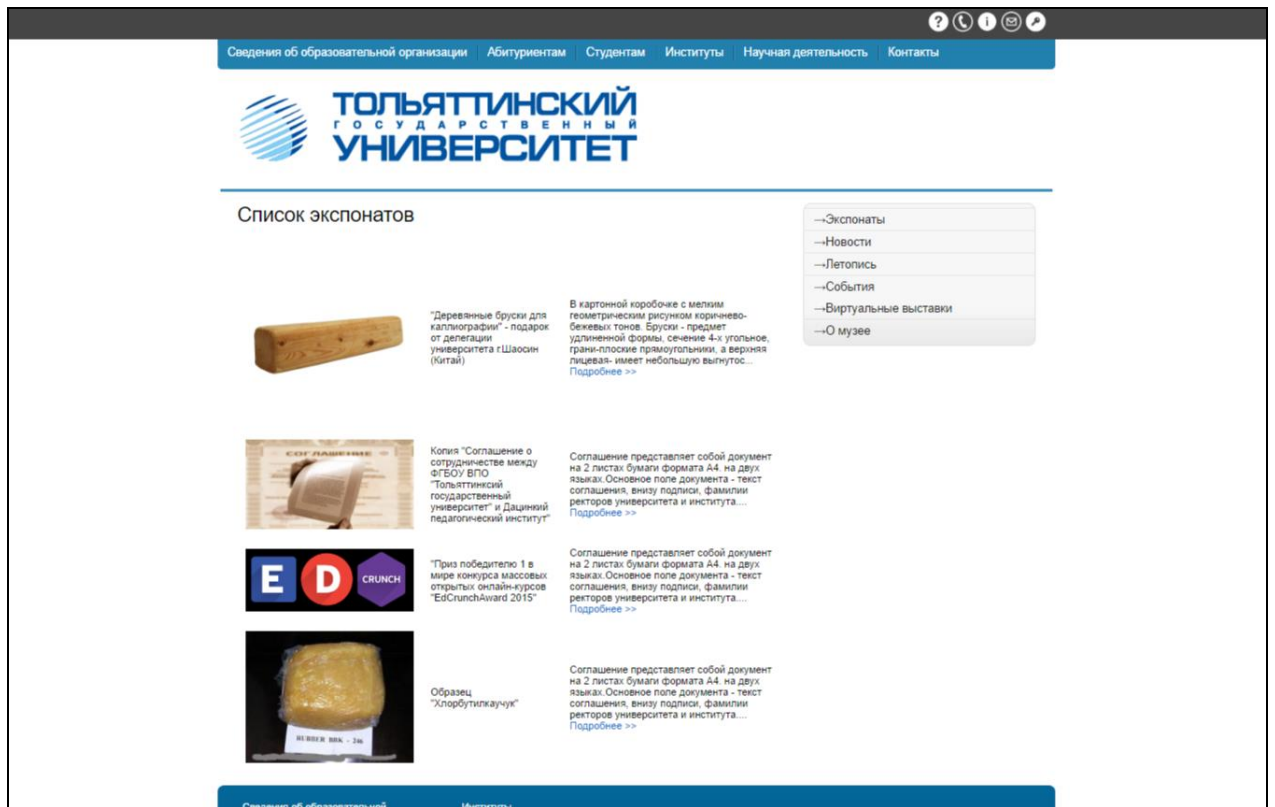


Рисунок А.5 – Страница ИС с полным списком экспонатов музея ТГУ (для посетителей)

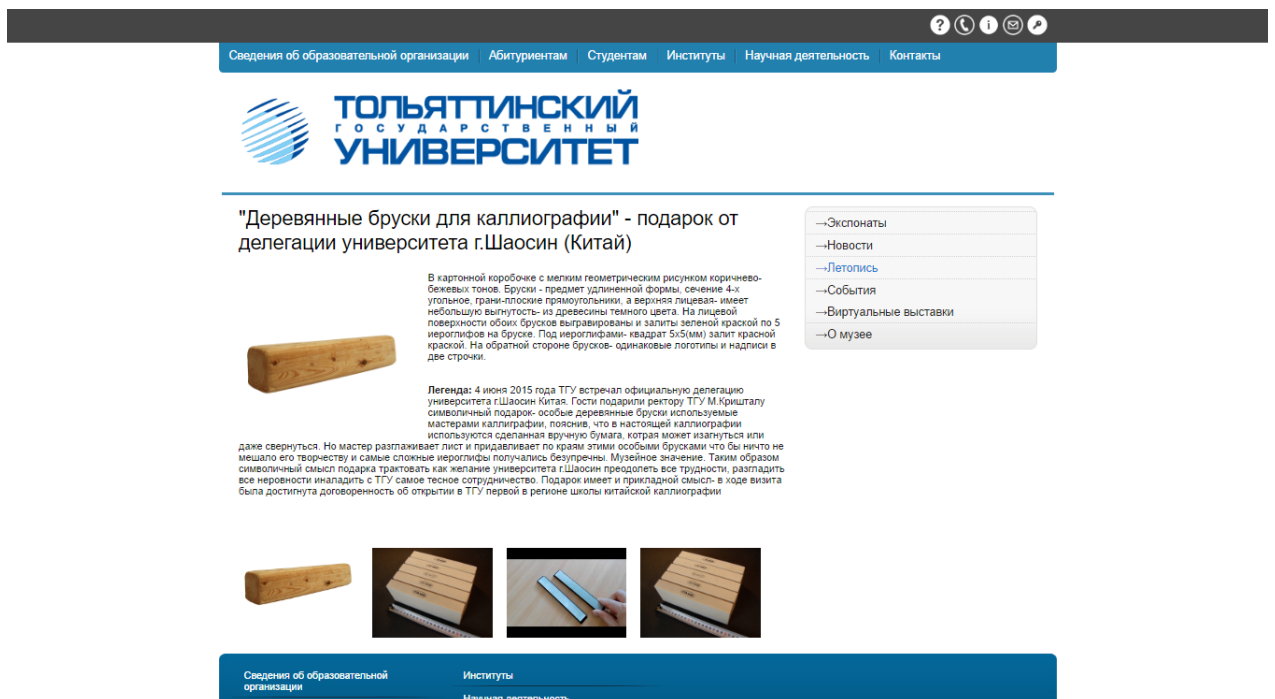


Рисунок А.6 – Страница «карточки» экспоната (для пользователей)