

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»
Заведующий кафедрой _____

(Ф.И.О., ученая степень, звание)

« _____ » _____ 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Создание динамического интернет-портала для музыкантов на основе технологии Web 2.0

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Нухатова М.С. ВТ-10-3
(Фамилия и инициалы) группа

Научный руководитель Ержан А.А. д.ф.н.
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Еркешева З.Д. с.и.и.и.
(Фамилия и инициалы, ученая степень, звание)
Еркешева « 14 » 05 20 14 г.
(подпись)

по безопасности жизнедеятельности:

Приходько И.Г. Д.ф.н., профессор
(Фамилия и инициалы, ученая степень, звание)
Приходько « 14 » 05 20 14 г.
(подпись)

по применению вычислительной техники:

Ержан А.А. д.ф.н.
(Фамилия и инициалы, ученая степень, звание)
Ержан « 22 » мая 20 14 г.
(подпись)

Нормоконтролер: Тусупов Д.М.
(Фамилия и инициалы, ученая степень, звание)

Тусупов « 27 » мая 20 14 г.
(подпись)

Рецензент: _____
(Фамилия и инициалы, ученая степень, звание)

« _____ » _____ 20__ г.
(подпись)

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Информационных технологий
Специальность 5В070400 - Вычислительная техника и программное обеспечение
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Нухамба Марша Сергеевна
(фамилия, имя, отчество)

Тема проекта Создание динамического интернет-портала для музеев на основе технологии Web 2.0

утверждена приказом ректора № 115 от «24» сентября 2013 г.

Срок сдачи законченной работы «__» _____ 20__ г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Необходимо создать веб-сайт, на котором пользователи смогут делиться своим творчеством. При разработке сайта необходимо использовать Web 2.0.

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

1. Исследование существующих веб-порталов для музеев
2. Изучение языков PHP и JS, фреймворка Yii, CSS-фреймворка Twitter Bootstrap, технологии AJAX
3. Разработка схем создания прототипа
4. Создание финального версии продукта

Перечень графического материала (с точным указанием обязательных чертежей)

1. ER-диаграмма базы данных проекта.
2. Диаграмма преемств
3. Диаграмма классов
4. Диаграмма разграничений

Рекомендуемая основная литература

1. Флэнган Д. Что и зачем JavaScript. Подробное руководство
2. Курочкин М, Шиндлер И. PHP. Практика создания веб-сайтов
3. Кло Б. Библия jQuery. Руководство профессионала
4. Зандерс М. PHP: коды, шаблоны и методы программирования

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
БД/ИД	Джиродово И.Б.	11.04 - 15.05.14	Джиродово
Ассемблея	Зрешев З.Д.	6.05 - 14.05.14	Зрешев
Основная часть	Ертак А.А.	27.03 - 22.05.14	Ертак
Нормоконтроль	Тусупов Д.М.	27.05.14	Тусупов

Аннотация

В данном дипломном проекте проводится разработка динамического интернет-портала на основе технологии Web 2.0 для музыкантов.

Рассмотрены существующие веб-порталы для музыкантов. Web-интерфейс разработан на основе новейших интернет-технологий и отвечает всем современным требованиям к программному обеспечению.

Кроме этого, в двух последних главах рассматриваются вопросы безопасности жизнедеятельности, приводится технико-экономическое обоснование и рассчитывается цена разработки проекта.

Андатпа

Бұл дипломдық жобада музыканттар үшін Web 2.0 технологиялары негізінде динамикалық итернет-порталдың әзірлемесі жүргізілді. Музыканттар үшін қолданыстағы веб-порталдар қарастырылды.

Web-интерфейс ең жаңа интернет-технологиялар негізінде әзірленді және бағдарламалық қамтамасыз етуде барлық қазіргі талаптарға жауап береді.

Сонымен қатар, жұмыстың соңғы екі бөлімінде өміртіршілігі қаіпсіздігі мәселелері қарастырылды, техникалық-экономикалық негіздемесі келтірілді және жоба әзірлемесінің бағасы есептелді.

Abstract

In this project is carried out to provide a dynamic online portal based on Web 2.0 technologies for musicians.

The existing web portals for musicians. Web-based interface is designed based on the latest Internet technology and meets all the requirements of the software.

In addition, in the last two chapters deal with issues of life safety, provides a feasibility study and calculate the price of the project development. Web-based interface is designed based on the latest Internet technology and meets all the requirements of the software.

Содержание

Введение.....	13
1 Теоретическая часть.....	15
1.1 Постановка задачи	15
1.2 Этап начальной разработки БД	15
1.3 Проектирование БД	21
1.3.1 Концептуальное проектирование.....	21
1.3.2 Бизнес правила	22
1.3.3 Разработка и построение подробной ER-диаграммы.....	26
1.3.4 Анализ информационных задач и круга пользователей системы	28
1.3.5 Определение атрибутов каждой сущности	28
1.4 Составление реляционных отношений.....	328
1.4.1 Расчет места для хранения БД.....	351
1.4.2 Задание места хранения БД. Загрузка базы данных.....	417
1.5 Используемые технологии программирования.....	42
1.5.1 Подход Web 2.0.....	43
1.5.2 Технология JQuery	43
1.5.3 Twitter Bootstrap.....	44
1.6 Используемые языки программирования.....	484
1.6.1 HTML.....	44
1.6.2 Язык PHP	495
1.6.3 Язык JavaScript.....	495
2 Расчетная часть.....	528
2.1 Логическое проектирование	528
2.2 Физическое проектирование.....	50
2.2.1 Обоснование выбора СУБД.....	541
2.2.2 Меры по обеспечению безопасности.....	541
2.2.3 Создание пользователей, логинов и задание им паролей.....	541
2.3 Описание интерфейса интернет-портала для музыкантов	562
3 Техничко-экономическое обоснование проекта.....	64
3.1 Оборудование, используемое в работе.....	694
3.2 Коэффициенты расчета трудоемкости.....	64
3.3 Расчет.....	66
4 Безопасность жизнедеятельности	7875
4.1 Анализ потенциально опасных и вредных факторов	7975
4.2 Расчет кондиционирования помещения.....	Ошибка! Закладка не определена. 7
4.3 Расчет освещенности офиса.....	Ошибка! Закладка не определена. 2

Заключение	917
Список используемой литературы	928
Приложение А	939

Введение

Тема данного дипломного проекта – «Создание динамического интернет-портала на основе технологии Web 2.0 для музыкантов».

Интернет-портал – это сайт в сети Internet, предоставляющий пользователю разнообразные интерактивные услуги и сервисы, объединённые единым доменным именем, функционирующим благодаря DNS. Данный проект является динамическим порталом, так как процесс получения его содержимого происходит на основе трех принципов построения динамических сайтов: во-первых, портал он генерирует содержимое на стороне сервера; во-вторых, сгенерированная страница передается клиенту, и в-третьих, уже на клиентской стороне происходит формирование содержимого. Ключевым отличием динамических сайтов от статических является то, что во множестве однотипных страниц используется одна страница-шаблон, в которую подгружается соответствующее содержимое, благодаря чему редактирование внешнего вида заметно облегчается, ведь остается редактировать всего лишь один родительский шаблон, и все дочерние страницы изменятся автоматически.

В настоящее время существует огромное количество музыкальных групп различного направления, сочиняющих собственную музыку и желающих добиться успеха и зарабатывать деньги своим творчеством. В нашей стране для развития молодых и талантливых музыкантов альтернативных направлений современной музыки не создается практически никаких условий, исключения составляют изредка приводящиеся фестивали и концерты, организованные частными лицами. Таким образом, у музыкантов остается только один выход – распространять свои композиции с помощью сети Internet.

Основательница крупных компаний Circle и Spire Джанет Краус придумала интересную систему определения актуальности идей. Дело должно отвечать трем требованиям – быть кислородом, аспирином и драгоценностью для пользователя или потребителя. Кислород рассматривает продукты, жизненно необходимые для человека. В данном случае, мы рассматриваем музыкантов, и их главная потребность в данном случае – само выразаться посредством своего творчества. В психологии, в пирамиде потребностей Маслоу на верхней ступеньке лежит потребность в самореализации. Поэтому данная площадка напрямую выполняет психологические потребности человека, значит, является кислородом для них. А аспирин – это то, что помогает человеку, при этом сам человек может обойтись и без лекарства, однако оно значительно улучшит его существование. Конечно, молодые группы могут

добиваться признания и без участия в данном проекте, с помощью других сайтов и «по старинке» телефонные звонками продюсерам, но используя интернет и данную систему, набрать популярность и как следствие привлечь инвесторов получится гораздо проще. Драгоценность – это те продукты, которые можно назвать роскошью и излишествами, связанные с отдыхом и удовольствиями. Прослушивание музыки – это удовольствие, играть в группе – это тоже прежде всего удовольствие. На сайте будет создано радио с лучшими песнями участников, тем самым любой человек может просто расслабиться, слушая новые и качественные коллективы. Таким образом, данный продукт удовлетворяет всем трем критериям.

Цель данного программного продукта – создать ресурс для начинающих музыкантов, где они смогут выкладывать своё творчество, делиться со своими слушателями информацией о выступлениях, тем самым набирая среди людей, что в дальнейшем может привлечь продюсеров, инвесторов и представителей известных звукозаписывающих лейблов.

Данный портал позволит музыкантам искать группы по своим интересам, общаться с единомышленниками и, даже, собирать свои группы.

Зарегистрировавшись, пользователи смогут создавать страницы своих групп, загружать в них фотографии, клипы и, разумеется, песни.

Другие пользователи должны иметь возможность подписываться на песни тех групп, которые им нравятся, чтобы следить за их обновлениями.

Для того, чтобы пользователя было проще найти на сайте, он заполняет свой профиль. Юзер может указать какими музыкальными способностями он обладает, на каких инструментах умеет играть, в каких проектах участвует.

Структурно дипломная работа состоит из введения, четырех разделов и заключения.

В первом разделе определены требования, которым должен удовлетворять программный продукт. Разработана модель базы данных, рассмотрены технологии и языки программирования, используемые в работе.

Во втором разделе приведено описание работы веб-сайта.

В третьем разделе приведено технико-экономическое обоснование реализации проекта и вычислена его себестоимость.

В четвертом разделе проведен анализ потенциально опасных и вредных производственных факторов проектируемого объекта, воздействующих на персонал, расчет кондиционирования офиса и освещенности рабочего помещения.

1 Теоретическая часть

1.1 Постановка задачи

Задачей является создать динамический интернет-портал для музыкантов. Данный портал позволит музыкантам искать группы по своим интересам, общаться с единомышленниками и, даже, собирать свои группы.

Зарегистрировавшись, пользователи смогут создавать страницы своих групп, загружать в них фотографии, клипы и, разумеется, песни.

Другие пользователи должны иметь возможность подписываться на песни тех групп, которые им нравятся, чтобы следить за их обновлениями.

Для того чтобы пользователя было проще найти на сайте, он заполняет свой профиль. Юзер может указать какими музыкальными способностями он обладает, на каких инструментах умеет играть, в каких проектах участвует.

На портале должен быть предоставлен список репетиционных баз, концертных площадок. Будет создана страница афиши, на которой отображаются все предстоящие мероприятия. Все места и события будут показываться на карте.

Регистрация должна занимать меньше минуты и в том числе осуществляться с помощью популярных социальных сетей.

Также, на основе статистики прослушивания песен и просмотра страниц, будет формироваться общий чарт песен и групп.

Для привлечения на портал новичков будет создан раздел с новостями мира музыки и разделом с учебными материалами.

1.2 Этап начальной разработки БД

Анализ предметной области

Музыкальный коллектив — это некоторое количество людей, связанных одной целью, которая напрямую связана с музыкой. Число музыкантов в одной группе в среднем варьируется от двух до двенадцати человек в зависимости от жанра. Рок-группы, которые составляют большинство от общего числа всех создаваемых групп, имеют в своем составе от трех до семи человек. Практически в каждой группе, исполняющей рок или металл, есть ритм-секция, барабанщик и басист, а также гитарист (который по совместительству может

быть и вокалистом). Так как данный интернет-портал рассчитан в основном на рок-музыкантов, в дальнейшем будем рассматривать только их.

По статистике, рок-музыкантами в 85% случаев является представители мужского пола, поэтому данную особенность следует учитывать при разработке дизайна и интерфейса веб-сайта.

Новые группы чаще всего собираются учащимися старших классов, затем по интернету ищутся недостающие члены группы. Например, если группа организуется басистом и вокалистом, то подается объявление о поиске гитариста (или сразу двух: ритм - и соло-гитары) и, в некоторых случаях, клавишника, скрипача или саксофониста (в зависимости от стилистики и жанра, в котором группа планирует играть). Далее все участники коллектива собираются и репетируют на различных репетиционных базах. Чаще всего, впервые организованные коллективы, начинают свой творческий путь с каверов – переигранных песен отечественных или зарубежных исполнителей. Затем появляются первые песни собственного сочинения, и в среднем, через полгода, группа впервые выступает на публике. Для начинающих рок-музыкантов созданы специальные пабы и бары, где часто происходят подобные дебюты. Посетителями таких заведений в основном являются друзья и знакомые музыкантов, которые в этот день дают представление в его стенах. Музыкантам в таких заведениях ничего не платят. Примером такого заведения в нашей стране является бар «Кошкин дом». Более продвинутые коллективы выступают в ночных клубах и дорогих барах (например, бар «Чукотка» или ночной клуб «Жесть»), где за некоторую плату становятся официальными резидентами заведения, то есть уже зарабатывают своим творчеством. Изредка организовываются фестивали («Нейтрал», байк-фест).

Оповещение о выступлениях в основном происходит через социальные сети. Там же, немногочисленные фанаты могут посмотреть видео, фото с выступлений и послушать аудиозаписи. Песни чаще всего записываются и микшируются самостоятельно участниками группы (получаются так называемые «демо-записи»), или на недорогих студиях.

Базовые сущности данной базы данных:

1 Пользователи. Эта сущность служит для хранения информации о пользователях системы, об их личных данных и уровне доступа.

2 Аудиозаписи. Здесь хранятся музыкальные композиции, загруженные пользователями.

3 Фото. Здесь хранятся фотографии, загруженные на сервер.

4 Видео. Здесь хранятся видеозаписи, загруженные пользователями на сервер.

5 Группы. Данная сущность служит для хранения информации о музыкальных группах.

6 Концерты. Здесь хранится информация о проводимых мероприятиях, время и место проведения.

7 Клубы. Здесь хранится информация о местах проведения концертов, цены и адрес заведения.

8 Администраторы. Здесь хранятся данные о пользователях, которые следят за нарушителями правил пользования сайта, а также размещающих новости.

Обзор аналогов и конкурентов

В данное время единственным конкурентноспособным сайтом по сходной тематике является американский портал reverbnation.com. Но этот сайт реализован только на английском языке, и в плане дизайна морально устарел. К тому же в нем отсутствуют некоторые функции, осуществленные в данном проекте – такие как онлайн-радио с лучшими песнями зарегистрированных на сайте музыкантов, площадка для сбора средств в помощь музыкантам, новости музыки на главной странице, оценка аудиозаписей, а также уроки и интересные познавательные статьи. Главный акцент делается на американских пользователях с песнями на английском языке. В данной дипломной работе основной упор делается на русскоязычных музыкантов. Порталом будет удобно пользоваться жителям пост-советского пространства, в частности музыкантам России, Казахстана, Украины, Беларуси, Узбекистана и Киргизстана.

В России существует радио для любителей музыки в жанре рок и металл, которое называется «Наше Радио». В 2013 году они запустили проект «Наше 2.0», который предназначен для начинающих музыкантов и молодых исполнителей. Цель этого проекта – попасть в ротацию радио «Наше» наравне со старожилами русского рока и модными современными коллективами. Группа оставляет на сайте небольшое резюме, которое располагается там в открытом доступе. И в результате всеобщего голосования, а также голосования коллегии жюри, самые лучшие группы получают место на радио и контракты с продюсерами. При этом, на данном радио в основном крутятся песни в жанре поп-рок, то есть приемлемые для разного контингента слушателей. И у музыкальных коллективов, играющих хардкор, прогрессив-рок, дет-металл и другие экстремальные виды рок-музыки, шансы попасть в ротацию радиостанции практически равны нулю. Проект от радио можно не рассматривать как конкурентов, потому что предлагаемый в данной дипломной работе веб-портал охватывает более широкую аудиторию исполнителей и слушателей. Остальные российские музыкальные порталы существуют в виде мелких и устаревших сайтов, некрасивых и трудночитаемых, у которых практически нет посетителей. В Казахстане сайтов данной тематики нет вообще. Исключение составлял лишь портал Тенгри FM, который давал рок-музыкантам из Казахстана возможность заявить о себе, послав песню на радио. И каждую субботу в передаче «Сводный Чарт» уделялся один час для представления и дебюта молодых музыкантов, которые регистрировались на сайте и присылали свои песни администратору. Однако Тенгри FM в данное время практически не функционирует, сайт закрыт и не работает.

Мировым лидером в области распространения музыки является сервис soundcloud.com. Но его направленность несколько отличается от данного ресурса, так как на нем можно только выкладывать музыку, он является чем-то

вроде портала Twitter, только вместо текстовых сообщений, пользователь делится музыкой.

В данное время основным порталом, где русскоязычные музыканты создают группы и выкладывают информацию о своем творчестве является известный портал, созданный Павлом Дуровым – социальная сеть «ВКонтакте». Однако там нельзя оценивать песни, красиво оформлять страницы, и, в общем, этот портал не предназначен для музыкантов. Заполнить эту нишу позволит портал, разрабатываемый в данной дипломной работе.

UML диаграммы

Для моделирования статических объектов в объектно-ориентированной концепции UML существует 12 диаграмм, но так как большинство объектов в базе данных являются статическими, используем самые основные:

- 1 Диаграмма компонентов.
- 2 Диаграмма прецедентов.
- 3 Диаграмма классов.
- 4 Диаграмма развертывания.

Диаграмма компонентов

Диаграмма компонентов показывает набор компонентов и отношений между ними.

Компонент – это физически заменяемая часть системы, которая имеет набор интерфейсов и обеспечивает их реализацию.

Приложение обращается за получением или добавлением информации к базе данных MySQL. Но обращается не напрямую, а с помощью драйвера MySQL Driver for PHP. Таким образом, MySQL Driver for PHP является интерфейсом между приложением и базой данных, обеспечивая их взаимодействие. Диаграмма компонентов представлена на рисунке 1.1.

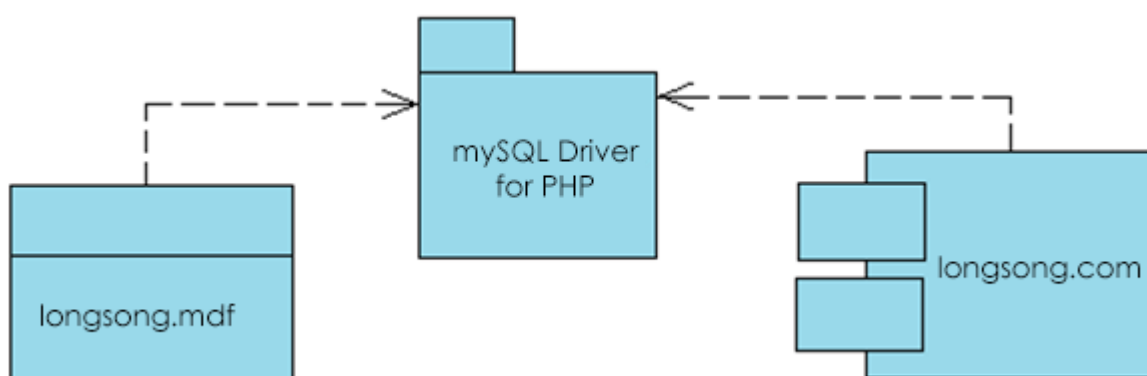


Рисунок 1.1 – Диаграмма компонентов

Диаграмма прецедентов

Для характеристики взаимодействия пользователей с базами данных применяется диаграмма прецедентов или вариантов использования. На данной диаграмме показано, какие пользователи могут взаимодействовать с базой и

какими способами. В данном случае имеется два типа пользователей, у каждого из которых различные права. При этом администратор может делать всё то, что может делать и обычный пользователь. То есть администратор обладает расширенными правами. Диаграмма прецедентов представлена на рисунке 1.2.



Рисунок 1.2 – Диаграмма прецедентов

Отсюда видно, что пользоваться базой могут 2 группы пользователей: администраторы и пользователи.

Диаграмма классов

Диаграмма классов, показанная на рисунке 1.3, и в UML является частным случаем ER-диаграммы. ER-диаграммы используются для логического проектирования баз данных. Главное их отличие: в ER-диаграмме уделяется внимание структуре данных, а в диаграмме классов - поведению классов.

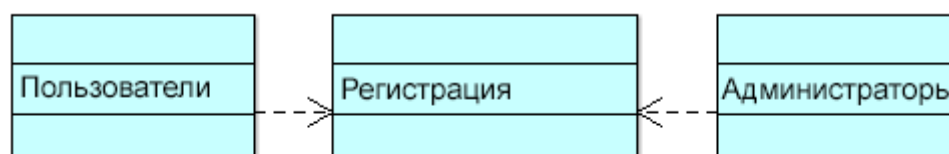


Рисунок 1.3 – Диаграмма классов

Диаграмма развертывания

Диаграмма развертывания, представленная на рисунке 1.4, по-другому называется «Deployment diagram» в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать web-сайт диаграмма развертывания должна показывать, какие аппаратные компоненты ("узлы") существуют (например, web-сервер, сервер базы данных, сервер приложения), какие программные компоненты ("артефакты") работают на каждом узле (например, web-приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, OLEDB, RMI). В данном случае присутствует сервис Internet, который изображается на

диаграмме в виде облака, посредством которого сервер взаимодействует с маршрутизатором.

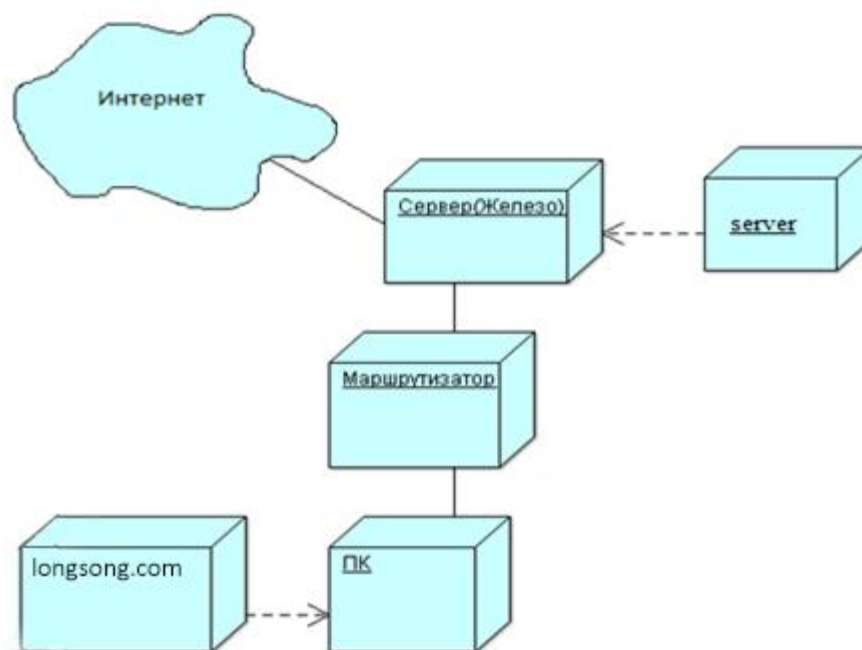


Рисунок 1.4 – Диаграмма развертывания

Узлы представляются как прямоугольные параллелепипеды с артефактами, расположенными в них, изображенными в виде прямоугольников. Узлы могут иметь подузлы, которые представляются как вложенные прямоугольные параллелепипеды. Один узел диаграммы развертывания может концептуально представлять множество физических узлов, таких как кластер серверов баз данных.

Существует два типа узлов:

- узел устройства;
- узел среды выполнения.

Узлы устройств – это физические вычислительные ресурсы со своей памятью и сервисами для выполнения программного обеспечения, такие как обычные ПК, мобильные телефоны. Узел среды выполнения – это программный вычислительный ресурс, который работает внутри внешнего узла и который предоставляет собой сервис, выполняющий другие исполняемые программные элементы.

Диаграмма развертывания нашей системы очень проста, так как не требует специального оборудования и любой пользователь, у которого есть выход в интернет, может просматривать информацию о нем. При этом узел среды выполнения – это программный вычислительный ресурс, который работает внутри внешнего узла и который предоставляет собой сервис, выполняющий другие программные модули.

1.3 Проектирование БД

1.3.1 Концептуальное проектирование

Учитывая параметры нашей системы, приведенные ранее, мы выделим основные сущности и связи между ними, изобразим их на предварительной ER-диаграмме, которая приведена на рисунке 1.5.

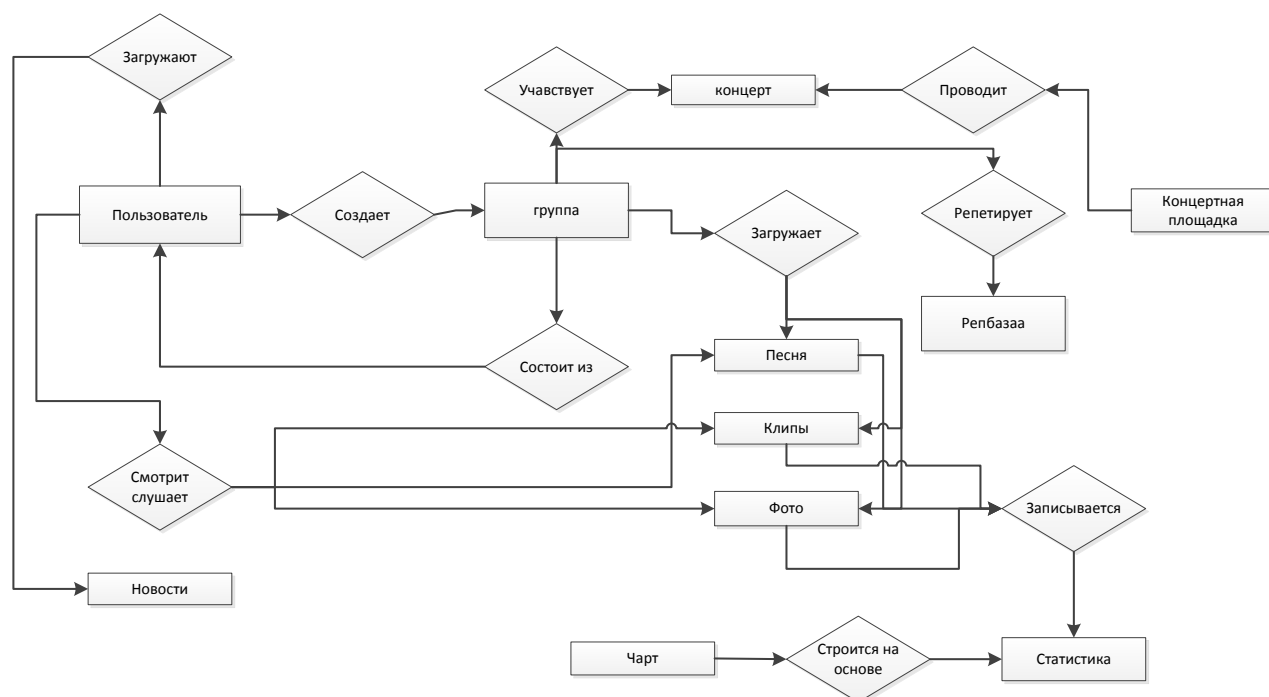


Рисунок 1.5 – Общая ER-диаграмма

Для построения ER-диаграммы воспользуемся программным обеспечением от компании Microsoft под названием Visio.

На основе данной диаграммы сформируем наши бизнес-правила. Бизнес-правилами называются такие правила, которые устанавливают ограничения для связей между сущностями. Бизнес-правила представляют собой специализированный вид логики, описывающей ограничения на образ действий, которые система или люди должны учитывать в своем поведении. Большинство бизнес-правил обнаруживаются в процессе накопления требований. При этом появляется естественное желание добавить еще одну секцию в описания сценариев использования и в нее добавлять бизнес-правила. Однако если бизнес-правило по своему существу не принадлежит к функции, описанной сценарием использования, то оно обычно охватывает несколько сценариев. Бизнес-правила ограничения проверяют значение транзакции и

операции на непротиворечивость, чтобы в итоге инвариантные правила обеспечили непротиворечивость итоговых данных в результате множественных изменений. Размещая бизнес-правила в соответствующем классе на диаграмме классов, мы гарантируем, что эти правила будут многократно использоваться во всех сценариях точно таким же образом, как и класс, к которому они принадлежат, в модели сценариев использования. Бизнес-правила изображают в виде небольших диаграмм. Моделирование и выделение бизнес-правил - необходимая часть разработки системы баз данных. Моделирование обеспечивает проверку бизнес-правил на достоверность еще до создания системы. Технология UML поддерживает моделирование бизнес-правил с помощью сценариев использования и диаграмм класса.

Целесообразно начинать с них, потому что этот тип требований выявляется на начальных этапах анализа требований. Бизнес-правила помогают определить рамки проекта и выявить связи между его компонентами. Цель бизнес-правила – управлять данными аспектами и влиять на него наложением структуры.

Далее установим связи между сущностями, опираясь на описание операций на предварительной ER-диаграмме. Точнее связи наших сущностей можно установить на основе бизнес-правил, которые, в свою очередь, построены на основе подробного описания операций.

1.3.2 Бизнес правила

1 Информация о пользователе заносится в две сущности. В одной из них находится техническая информация, а во второй личные данные пользователей, такие как имя, фамилия, URL-адрес аватарки, краткая биография и город (Рисунок 1.6).

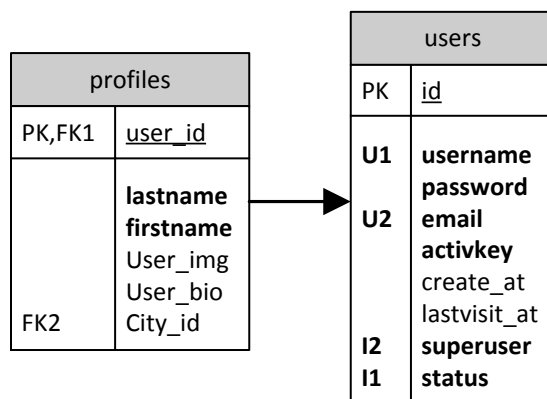


Рисунок 1.6 – Сегмент ER -диаграммы для бизнес-правила 1

2 Каждый город находится в определенной стране. В каждой стране может быть неограниченное количество городов (Рисунок 1.7).

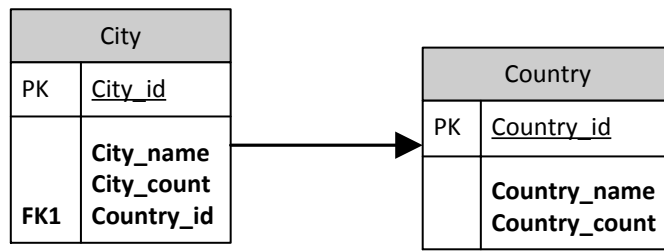


Рисунок 1.7 – Сегмент ER-диаграммы для бизнес-правила 2

3 В профиль пользователя заносится информация о том, в каком городе он живет (Рисунок 1.8).

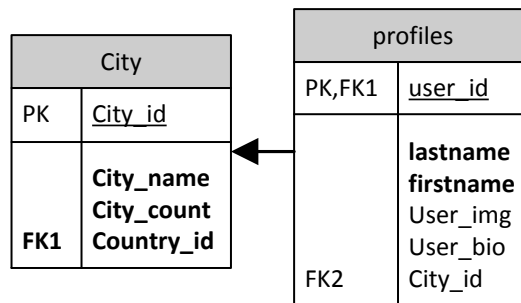


Рисунок 1.8 – Сегмент ER-диаграммы для бизнес-правила 3

4 Когда пользователи прослушивают какую-либо песню, информация об этом записывается в специальную таблицу для составления в будущем персональной статистики для групп и пользователей (Рисунок 1.9).

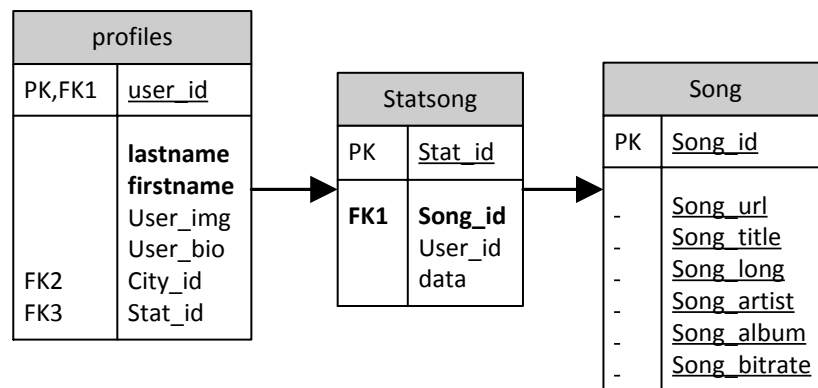


Рисунок 1.9 – Сегмент ER -диаграммы для бизнес-правила 4

5 Пользователи могут указывать информацию о своей специализации (гитарист, вокалист, композитор) (Рисунок 1.10).

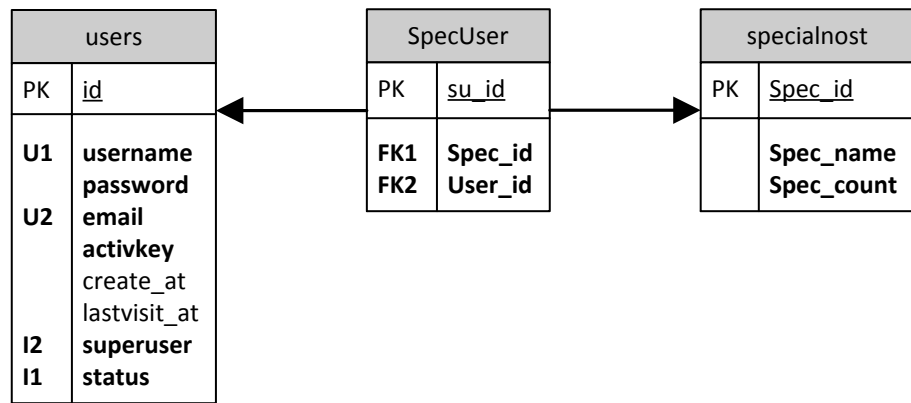


Рисунок 1.10 – Сегмент ER-диаграммы для бизнес-правила 5

6 Пользователи могут становиться фанатами групп, то есть имеют возможность подписаться на их обновления (Рисунок 1.11).

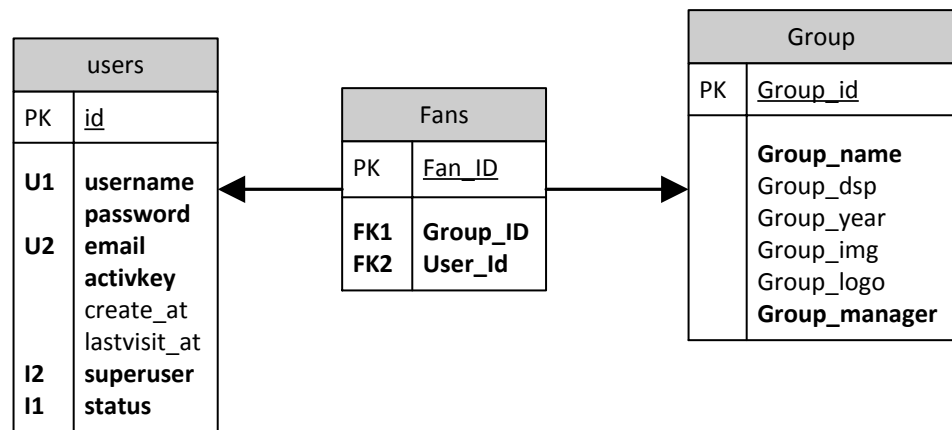


Рисунок 1.11 – Сегмент ER-диаграммы для бизнес-правила 6

7 В городах существуют репетиционные базы, у которых указан адрес, телефон и расценки (Рисунок 1.12).

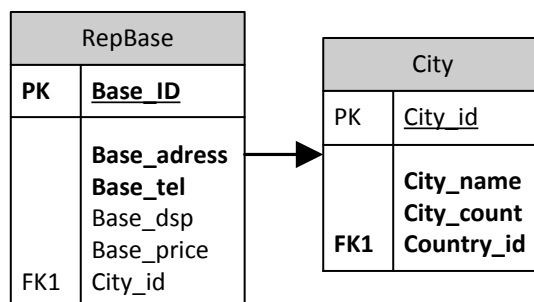


Рисунок 1.12 – Сегмент ER-диаграммы для бизнес-правила 7

8 О группах собирается общая статистика, которая учитывает когда и сколько раз пользователь просматривал страницу группы (Рисунок 1.13).

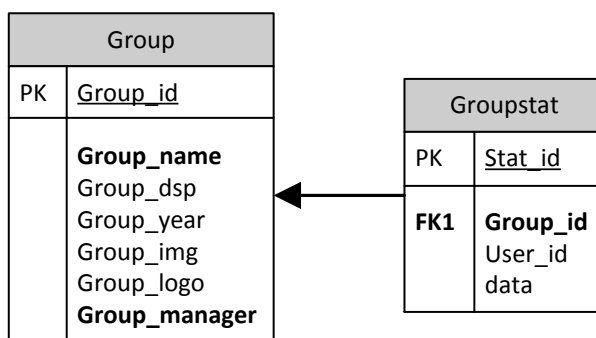


Рисунок 1.13 – Сегмент ER-диаграммы для бизнес-правила 8

9 В группе есть состав, где отмечаются пользователи, которые играют в музыкальной группе (Рисунок 1.14).

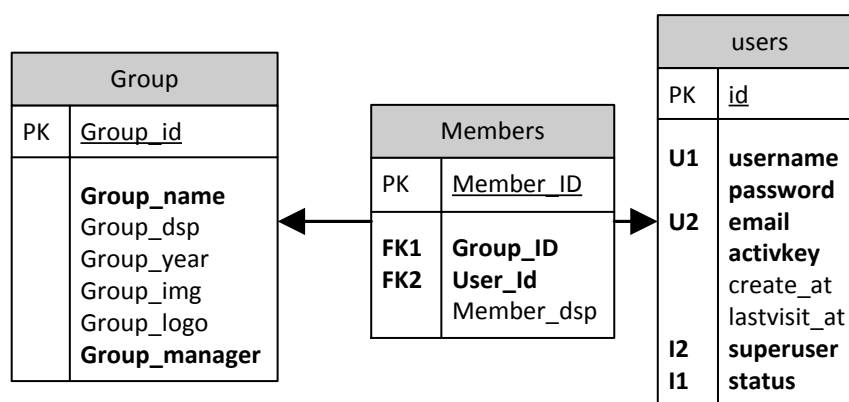


Рисунок 1.14 – Сегмент ER-диаграммы для бизнес-правила 9

10 В клубах проводятся концерты, цена билета устанавливается организаторами концерта (Рисунок 1.15).

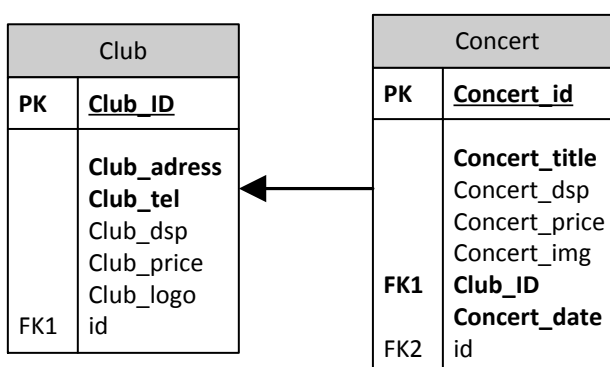


Рисунок 1.15 – Сегмент ER-диаграммы для бизнес-правила 10

11 На концертах регистрируются группы-участники, которые будут выступать на мероприятии (Рисунок 1.16).

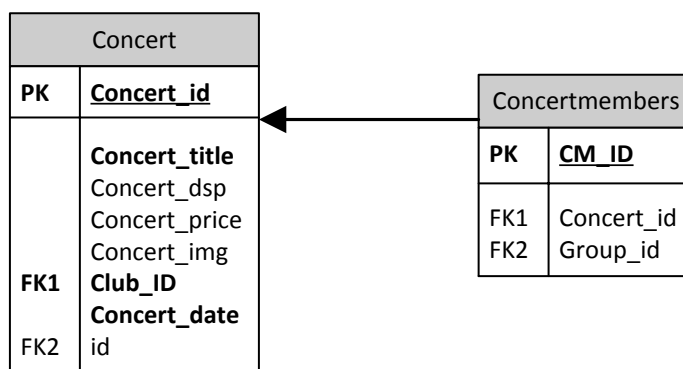


Рисунок 1.16 – Сегмент ER-диаграммы для бизнес-правила 11

12 В группе её участники, помимо песен, могут выкладывать фотографии и видеозаписи (Рисунок 1.17).

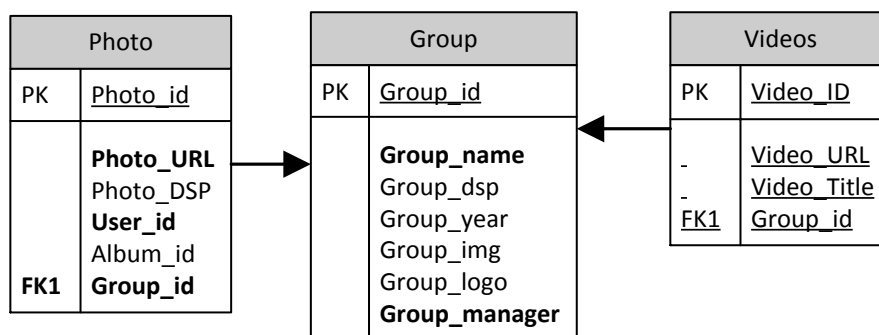


Рисунок 1.17 – Сегмент ER-диаграммы для бизнес-правила 12

13 Не может быть пользователей с одинаковыми логинами или электронными адресами.

14 Пользователи делятся на 2 уровня доступа к информации: пользователь, администратор.

1.3.3 Разработка и построение подробной ER-диаграммы на основании бизнес правил

Подробная ER-диаграмма складывается из вышеописанных бизнес-правил. Полученная модель представлена множеством понятий и связей между ними и определяет смысловую структуру рассматриваемой предметной области и её конкретного объекта. Концептуальная модель веб-системы для музыкантов представлена на рисунке 1.18.

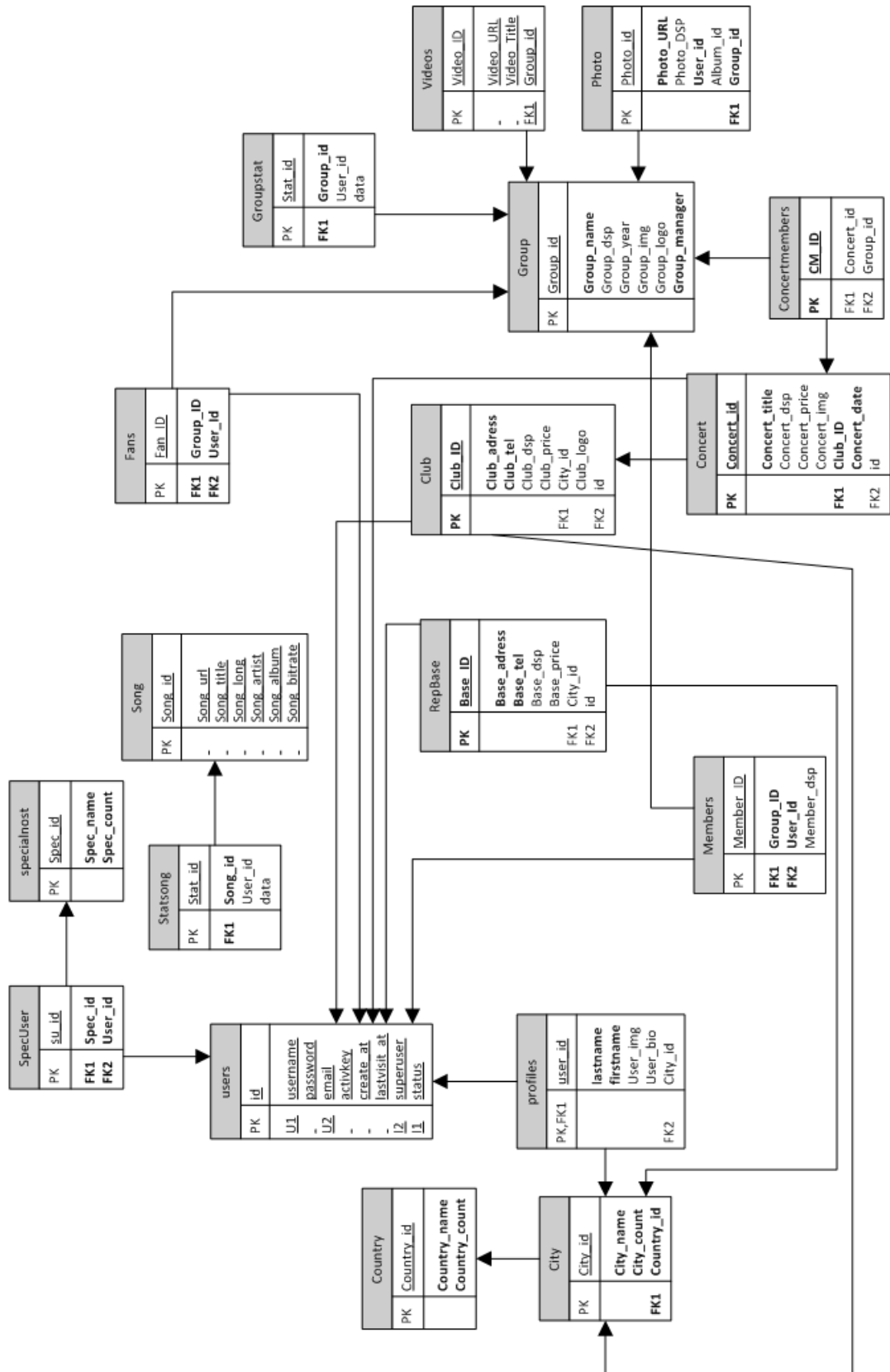


Рисунок 1.18 – Окончательный вариант концептуальной модели веб-системы для музыкантов

1.3.4 Анализ информационных задач и круга пользователей системы

С данным web-порталом могут работать следующие группы пользователей:

- администратор;
- пользователь.

При работе с системой администратор должен иметь возможность решать следующие задачи:

- 1 Добавлять новости на главную страницу;
- 2 Блокировать пользователей, которые так или иначе нарушили правила пользования сайта;
- 3 Регистрировать концерты и репетиционные базы;
- 4 Следить за работоспособностью всей системы.

Пользователь должен иметь возможность:

- 1 Создавать группы;
- 2 Подписываться на группы;
- 3 Добавлять контент – аудиозаписи, фотографии, видеозаписи;
- 4 Комментировать новости;
- 5 Просматривать новости;
- 6 Изменять свои личные данные.

1.3.5 Определение атрибутов каждой сущности

В таблице 1.1 представлены атрибуты каждой из сущностей созданной базы данных.

Т а б л и ц а – 1.1 Определение атрибутов каждой сущности

Сегмент ER модели		Описание
users		Таблица USERS (Пользователи) содержит следующие данные: ID – идентификатор пользователя USERNAME – логин для входа в систему EMAIL – имя владельца ACTIVKEY – уникальный ключ активации, который высылается пользователю при регистрации, CREATE_AT – дата и время создания аккаунта LASTVISIT_AT – дата и время последнего захода пользователя на сайт SUPERUSER – категория пользователя - 0 обычный пользователь 1 админ STATUS – статус активации аккаунта, 0 - не активизирован, 1- активизирован
PK	<u>id</u>	
<u>U1</u>	<u>username</u>	
-	<u>password</u>	
<u>U2</u>	<u>email</u>	
-	<u>activkey</u>	
-	<u>create_at</u>	
-	<u>lastvisit_at</u>	
<u>I2</u>	<u>superuser</u>	
<u>I1</u>	<u>status</u>	

Сегмент ER модели	Описание																
<table border="1"> <tr> <th colspan="2">SpecUser</th> </tr> <tr> <td>PK</td> <td><u>su_id</u></td> </tr> <tr> <td>FK1</td> <td>Spec_id</td> </tr> <tr> <td>FK2</td> <td>User_id</td> </tr> </table>	SpecUser		PK	<u>su_id</u>	FK1	Spec_id	FK2	User_id	<p>Таблица SPECUSER (Специализация пользователя) содержит следующие данные: SU_ID – номер записи в журнале специальностей SPEC_ID – идентификатор специализации USER_ID – идентификатор пользователя</p>								
SpecUser																	
PK	<u>su_id</u>																
FK1	Spec_id																
FK2	User_id																
<table border="1"> <tr> <th colspan="2">specialnost</th> </tr> <tr> <td>PK</td> <td><u>Spec_id</u></td> </tr> <tr> <td></td> <td>Spec_name</td> </tr> <tr> <td></td> <td>Spec_count</td> </tr> </table>	specialnost		PK	<u>Spec_id</u>		Spec_name		Spec_count	<p>Таблица SPECIALNOST (Типы специализаций) следующие данные: SPEC_ID – идентификатор специализации SPEC_NAME – название специализации SPEC_COUNT – число представителей этой специализации на сайте</p>								
specialnost																	
PK	<u>Spec_id</u>																
	Spec_name																
	Spec_count																
<table border="1"> <tr> <th colspan="2">Statsong</th> </tr> <tr> <td>PK</td> <td><u>Stat_id</u></td> </tr> <tr> <td>FK1</td> <td>Song_id</td> </tr> <tr> <td></td> <td>User_id</td> </tr> <tr> <td></td> <td>data</td> </tr> </table>	Statsong		PK	<u>Stat_id</u>	FK1	Song_id		User_id		data	<p>Таблица STATSONG (Статистика количества прослушиваний аудиозаписей) содержит следующие данные: STAT_ID – идентификатор сообщения SONG_ID – идентификатор аудиозаписи USER_ID – идентификатор пользователя, прослушавшего аудиозапись DATA – дата и время прослушивания конкретной аудиозаписи</p>						
Statsong																	
PK	<u>Stat_id</u>																
FK1	Song_id																
	User_id																
	data																
<table border="1"> <tr> <th colspan="2">Song</th> </tr> <tr> <td>PK</td> <td><u>Song_id</u></td> </tr> <tr> <td>-</td> <td>Song_url</td> </tr> <tr> <td>-</td> <td>Song_title</td> </tr> <tr> <td>-</td> <td>Song_long</td> </tr> <tr> <td>-</td> <td>Song_artist</td> </tr> <tr> <td>-</td> <td>Song_album</td> </tr> <tr> <td>-</td> <td>Song_bitrate</td> </tr> </table>	Song		PK	<u>Song_id</u>	-	Song_url	-	Song_title	-	Song_long	-	Song_artist	-	Song_album	-	Song_bitrate	<p>Таблица SONG (Аудиозаписи) содержит следующие данные: SONG_ID – идентификатор аудиозаписи SONG_URL – адрес аудиозаписи SONG_TITLE – название песни SONG_LONG – длительность аудиозаписи SONG_ARTIST – исполнитель SONG_ALBUM – альбом, к которому принадлежит песня SONG_BITRATE – битрейт песни</p>
Song																	
PK	<u>Song_id</u>																
-	Song_url																
-	Song_title																
-	Song_long																
-	Song_artist																
-	Song_album																
-	Song_bitrate																
<table border="1"> <tr> <th colspan="2">Fans</th> </tr> <tr> <td>PK</td> <td><u>Fan_ID</u></td> </tr> <tr> <td>FK1</td> <td>Group_ID</td> </tr> <tr> <td>FK2</td> <td>User_id</td> </tr> </table>	Fans		PK	<u>Fan_ID</u>	FK1	Group_ID	FK2	User_id	<p>Таблица FANS (Фанаты) содержит следующие данные: FAN_ID – идентификатор фаната GROUP_ID – идентификатор группы, на которую он подписался USER_ID – идентификатор пользователя</p>								
Fans																	
PK	<u>Fan_ID</u>																
FK1	Group_ID																
FK2	User_id																
<table border="1"> <tr> <th colspan="2">profiles</th> </tr> <tr> <td>PK,FK1</td> <td><u>user_id</u></td> </tr> <tr> <td></td> <td>lastname</td> </tr> <tr> <td></td> <td>firstname</td> </tr> <tr> <td></td> <td>User_img</td> </tr> <tr> <td></td> <td>User_bio</td> </tr> <tr> <td>FK2</td> <td>City_id</td> </tr> </table>	profiles		PK,FK1	<u>user_id</u>		lastname		firstname		User_img		User_bio	FK2	City_id	<p>Таблица PROFILES (Профиль пользователя) содержит следующие данные: USER_ID- идентификатор пользователя LASTNAME – Фамилия пользователя FIRSTNAME – Имя пользователя USER_IMG – адрес файла с аватаром пользователя USER_BIO – небольшая информация о пользователе CITY_ID – номер города</p>		
profiles																	
PK,FK1	<u>user_id</u>																
	lastname																
	firstname																
	User_img																
	User_bio																
FK2	City_id																

Сегмент ER модели	Описание						
<table border="1"> <thead> <tr> <th colspan="2">City</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td><u>City_id</u></td> </tr> <tr> <td>FK1</td> <td>City_name City_count Country_id</td> </tr> </tbody> </table>	City		PK	<u>City_id</u>	FK1	City_name City_count Country_id	<p>Таблицы CITY (город) содержит следующие данные: CITY_ID – номер города CITY_NAME – название города CITY_COUNT – количество зарегистрированных пользователей из этого города COUNTRY_ID – номер страны, в которой расположен город</p>
City							
PK	<u>City_id</u>						
FK1	City_name City_count Country_id						
<table border="1"> <thead> <tr> <th colspan="2">Country</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td><u>Country_id</u></td> </tr> <tr> <td></td> <td>Country_name Country_count</td> </tr> </tbody> </table>	Country		PK	<u>Country_id</u>		Country_name Country_count	<p>Таблица COUNTRY (Страна) содержит следующие данные: COUNTRY_ID – уникальный номер страны COUNTRY_NAME – название страны COUNTRY_COUNT – количество зарегистрированных пользователей из этой страны</p>
Country							
PK	<u>Country_id</u>						
	Country_name Country_count						
<table border="1"> <thead> <tr> <th colspan="2">Group</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td><u>Group_id</u></td> </tr> <tr> <td></td> <td>Group_name Group_dsp Group_year Group_img Group_logo Group_manager</td> </tr> </tbody> </table>	Group		PK	<u>Group_id</u>		Group_name Group_dsp Group_year Group_img Group_logo Group_manager	<p>Таблица GROUP (Группы) содержит следующие данные: GROUP_ID – идентификатор группы GROUP_NAME – название группы GROUP_DSP – описание группы GROUP_YEAR – год образования группы GROUP_IMG – постер группы GROUP_LOGO – логотип-аватарка группы GROUP_MANAGER – менеджер группы</p>
Group							
PK	<u>Group_id</u>						
	Group_name Group_dsp Group_year Group_img Group_logo Group_manager						
<table border="1"> <thead> <tr> <th colspan="2">Groupstat</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td><u>Stat_id</u></td> </tr> <tr> <td>FK1</td> <td>Group_id User_id data</td> </tr> </tbody> </table>	Groupstat		PK	<u>Stat_id</u>	FK1	Group_id User_id data	<p>Таблица GROUPSTAT (Статистика просмотров) содержит следующие данные: STAT_ID – идентификатор записи в журнале статистики GROUP_ID – идентификатор группы USER_ID – идентификатор пользователя, просмотревшего информацию по данной группе DATA – дата и время просмотра</p>
Groupstat							
PK	<u>Stat_id</u>						
FK1	Group_id User_id data						
<table border="1"> <thead> <tr> <th colspan="2">Videos</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td><u>Video_ID</u></td> </tr> <tr> <td>- FK1</td> <td><u>Video_URL</u> <u>Video_Title</u> <u>Group_id</u></td> </tr> </tbody> </table>	Videos		PK	<u>Video_ID</u>	- FK1	<u>Video_URL</u> <u>Video_Title</u> <u>Group_id</u>	<p>Таблица VIDEOS (Видео) содержит следующие данные: VIDEO_ID – идентификатор видеозаписи VIDEO_URL – адрес видеозаписи VIDEO_TITLE – название видеозаписи GROUP_ID – идентификатор группы, которая загрузила данную видеозапись</p>
Videos							
PK	<u>Video_ID</u>						
- FK1	<u>Video_URL</u> <u>Video_Title</u> <u>Group_id</u>						
<table border="1"> <thead> <tr> <th colspan="2">Photo</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td><u>Photo_id</u></td> </tr> <tr> <td>FK1</td> <td>Photo_URL Photo_DSP User_id Album_id Group_id</td> </tr> </tbody> </table>	Photo		PK	<u>Photo_id</u>	FK1	Photo_URL Photo_DSP User_id Album_id Group_id	<p>Таблица PHOTO (Фотографии) содержит следующие данные: PHOTO_ID – идентификатор фотографии PHOTO_URL – адрес фотографии PHOTO_DSP – описание под фотографией USER_ID – пользователь, загрузивший фото ALBUM_ID – идентификатор альбома с фотографиями GROUP_ID - идентификатор группы</p>
Photo							
PK	<u>Photo_id</u>						
FK1	Photo_URL Photo_DSP User_id Album_id Group_id						

Сегмент ER модели	Описание								
<table border="1"> <thead> <tr> <th colspan="2">Club</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td><u>Club_ID</u></td> </tr> <tr> <td>FK1</td> <td>Club_adress Club_tel Club_dsp Club_price City_id Club_logo</td> </tr> <tr> <td>FK2</td> <td>id</td> </tr> </tbody> </table>	Club		PK	<u>Club_ID</u>	FK1	Club_adress Club_tel Club_dsp Club_price City_id Club_logo	FK2	id	<p>Таблица CLUB (Заведения) содержит следующие данные:</p> <p>CLUB_ID – идентификатор заведения CLUB_ADRESS – адрес заведения CLUB_DSP – описание заведения CLUB_PRICE - расценка услуг CITY_ID – идентификатор города CLUB_LOGO – логотип или фотграфия заведения ID – идентификатор профиля</p>
Club									
PK	<u>Club_ID</u>								
FK1	Club_adress Club_tel Club_dsp Club_price City_id Club_logo								
FK2	id								
<table border="1"> <thead> <tr> <th colspan="2">Concert</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td><u>Concert_id</u></td> </tr> <tr> <td>FK1</td> <td>Concert_title Concert_dsp Concert_price Concert_img Club_ID Concert_date</td> </tr> <tr> <td>FK2</td> <td>id</td> </tr> </tbody> </table>	Concert		PK	<u>Concert_id</u>	FK1	Concert_title Concert_dsp Concert_price Concert_img Club_ID Concert_date	FK2	id	<p>Таблица CONCERT (Выступления) содержит следующие данные:</p> <p>CONCERT_ID– идентификатор выступления CONCERT_TITLE – название мероприятия CONCERT_DSP – описание мероприятия CONCERT_PRICE – цена билета CONCERT_IMG – постер события CLUB_ID – идентификатор места, где будет проходить концерт CONCERT_DATE – дата и время проведения события ID – идентификатор профиля</p>
Concert									
PK	<u>Concert_id</u>								
FK1	Concert_title Concert_dsp Concert_price Concert_img Club_ID Concert_date								
FK2	id								
<table border="1"> <thead> <tr> <th colspan="2">Concertmembers</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td><u>CM_ID</u></td> </tr> <tr> <td>FK1</td> <td>Concert_id</td> </tr> <tr> <td>FK2</td> <td>Group_id</td> </tr> </tbody> </table>	Concertmembers		PK	<u>CM_ID</u>	FK1	Concert_id	FK2	Group_id	<p>Таблица CONCERTMEMBERS (Участники концерта) содержит следующие данные:</p> <p>CM_ID – номер участника CONCERT_ID – идентификатор концерта GROUP_ID – идентификатор группы, выступающей на данном концерте</p>
Concertmembers									
PK	<u>CM_ID</u>								
FK1	Concert_id								
FK2	Group_id								
<table border="1"> <thead> <tr> <th colspan="2">Members</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td><u>Member_ID</u></td> </tr> <tr> <td>FK1</td> <td>Group_ID</td> </tr> <tr> <td>FK2</td> <td>User_id Member_dsp</td> </tr> </tbody> </table>	Members		PK	<u>Member_ID</u>	FK1	Group_ID	FK2	User_id Member_dsp	<p>Таблица MEMBERS (Состав) содержит следующие данные:</p> <p>MEMBER_ID – идентификатор участника группы GROUP_ID – идентификатор группы USER_ID – идентификатор пользователя MEMBER_DSP – описание учатсника</p>
Members									
PK	<u>Member_ID</u>								
FK1	Group_ID								
FK2	User_id Member_dsp								
<table border="1"> <thead> <tr> <th colspan="2">RepBase</th> </tr> </thead> <tbody> <tr> <td>PK</td> <td><u>Base_ID</u></td> </tr> <tr> <td>FK1</td> <td>Base_adress Base_tel Base_dsp Base_price City_id</td> </tr> <tr> <td>FK2</td> <td>id</td> </tr> </tbody> </table>	RepBase		PK	<u>Base_ID</u>	FK1	Base_adress Base_tel Base_dsp Base_price City_id	FK2	id	<p>Таблица REPBASE (Репетиционные базы) содержит следующие данные:</p> <p>BASE_ID – идентификатор базы BASE_ADRESS – адрес базы BASE_TEL – телефон базы BASE_DSP – описание базы BASE_PRICE – расценки базы CITY_ID – город, в котором расположена база</p>
RepBase									
PK	<u>Base_ID</u>								
FK1	Base_adress Base_tel Base_dsp Base_price City_id								
FK2	id								

1.4 Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности и в него вносятся все атрибуты сущности. Для каждого отношения необходимо определить первичный ключ и внешние ключи (если они есть).

Отношения приведены в таблицах 1.2 – 1.19. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Обязательное поле для краткости обозначено not null, необязательное - null. Типы данных обозначаются так: int – числовой, varchar – символьный, datetime – дата.

Т а б л и ц а 1.2 – USERS

Содержание поля	Имя поля	Тип, длина	Примечание
ID пользователя	ID	int	Первичный ключ
Логин	USERNAME	Varchar(20)	Not null
Пароль	PASSWORD	Varchar(128)	Not null
Почта	EMAIL	Varchar(128)	Not null
Ключ активации	ACTIVKEY	Varchar(128)	Not null
Дата создания аккаунта	CREATE_AT	Datetime	Not null
Последний просмотр	LASTVISIT_AT	Datetime	Not null
Администратор	SUPERUSER	Int	Not null
Статус	STATUS	int	Not null

Т а б л и ц а 1.3 – SPECUSER

Содержание поля	Имя поля	Тип, длина	Примечание
ID записи	SU_ID	int	первичный ключ
ID специализации	SPEC_ID	int	Not null (внешний ключ)
ID пользователя	USER_ID	int	Not null (внешний ключ)

Т а б л и ц а 1.4 – SPECIALNOST

Содержание поля	Имя поля	Тип, длина	Примечание
ID специализации	SPEC_ID	int	Первичный ключ
Название	SPEC_NAME	varchar	Not null
Количество зарегистрированных	SPEC_COUNT	int	Not null

Т а б л и ц а 1.5 – STATSONG

Содержание поля	Имя поля	Тип, длина	Примечание
ID статистики	STAT_ID	int	Первичный ключ
ID аудиозаписи	SONG_ID	int	Not null(внешний ключ)
ID пользователя	USER_ID	int	Null (Внешний ключ)
Дата и время прослушивания	DATA	datetime	null

Т а б л и ц а 1.6 – SONG

Содержание поля	Имя поля	Тип, длина	Примечание
ID аудиозаписи	SONG_ID	int	Первичный ключ
Адрес	SONG_URL	varchar	Not null
Название трека	SONG_TITLE	Varchar	Not null
Длительность	SONG_LONG	int	Null
Исполнитель	SONG_ARTIST	varchar	Not null
Альбом	SONG_ALBUM	varchar	Null
Битрейт	SONG_BITRATE	int	null

Т а б л и ц а 1.7 – FANS

Содержание поля	Имя поля	Тип, длина	Примечание
ID слушателя	FAN_ID	int	Первичный ключ
ID группы	GROUP_ID	int	Внешний ключ
ID пользователя	USER_ID	int	Внешний ключ

Т а б л и ц а 1.8 – REPBASE

Содержание поля	Имя поля	Тип, длина	Примечание
ID базы	BASE_ID	Int	Первичный ключ
Адрес	BASE_ADRESS	Varchar(255)	Not null
Телефон	BASE_TEL	Varchar(12)	Not null
Описание	BASE_DSP	text	Null
Цены	BASE_PRICE	int	Null
Город	CITY_ID	int	Внешний ключ

Т а б л и ц а 1.9 – CLUB

Содержание поля	Имя поля	Тип, длина	Примечание
ID заведения	CLUB_ID	Int	Первичный ключ
Адрес	CLUB_ADRESS	Varchar(255)	Not null
Телефон	CLUB_TEL	Varchar(12)	null
Описание	CLUB_DSP	text	null
Цены	CLUB_PRICE	Int	null
ID города	CITY_ID	int	null
Логотип	CLUB_LOGO	Varchar(255)	null
ID профиля	ID	int	Внешний ключ

Т а б л и ц а 1.10 – GROUP

Содержание поля	Имя поля	Тип, длина	Примечание
ID группы	GROUP_ID	int	Первичный ключ
Название	GROUP_NAME	varchar	Not null
Описание	GROUP_DSP	text	null
Год	GROUP_YEAR	Datetime	Null
Аватара	GROUP_IMG	varchar	null
Логотип	GROUP_LOGO	int	null
Менеджер	GROUP_MANAGER	varchar	Not null

Т а б л и ц а 1.11 – GROUPSTAT

Содержание поля	Имя поля	Тип, длина	Примечание
ID статистики	STAT_ID	int	Первичный ключ
ID группы	GROUP_ID	int	Внешний ключ
ID пользователя	USER_ID	int	Внешний ключ
Дата просмотра	DATA	datetime	Null

Т а б л и ц а 1.12 – CITY

Содержание поля	Имя поля	Тип, длина	Примечание
ID города	CITY_ID	Int	Первичный ключ
Название города	CITY_NAME	Nchar(200)	Not null
Количество зарегистрированных	CITY_COUNT	bigint	Null
ID страны	COUNTRY_ID	int	Not null(Внешний ключ)

Т а б л и ц а 1.13 – COUNTRY

Содержание поля	Имя поля	Тип, длина	Примечание
ID страны	COUNTRY_ID	int	Первичный ключ
Название страны	COUNTRY_NAME	Nchar(200)	Not null
Количество зарегистрированных	COUNTRY_COUNT	bigint	null

Т а б л и ц а 1.14 – CONCERT

Содержание поля	Имя поля	Тип, длина	Примечание
ID концерта	CONCERT_ID	int	Первичный ключ
Название	CONCERT_TITLE	Varchar(255)	Not null
Описание	CONCERT_DSP	text	null
Цена билета	CONCERT_PRICE	int	null
Изображение	CONCERT_IMG	Varchar(255)	null
Место проведения	CLUB_ID	int	Not null(Внешний ключ)
Дата	CONCERT_DATE	datetime	null
ID профиля	ID	int	null

Т а б л и ц а 1.15 – PROFILES

Содержание поля	Имя поля	Тип, длина	Примечание
ID пользователя	USER_ID	int	Первичный ключ
Фамилия	LASTNAME	varchar(50)	Not null
Имя	FIRSTNAME	varchar(50)	Not null
Аватара	USER_IMG	varchar(255)	null
Сведения о себе	USER_BIO	Text	Null
ID города	CITY_ID	Int	Null(Внешний ключ)

Т а б л и ц а 1.16 – CONCERTMEMBERS

Содержание поля	Имя поля	Тип, длина	Примечание
ID записи участника	CM_ID	Int	Первичный ключ
ID концерта	CONCERT_ID	Int	Not null(Внешний ключ)
ID группы	GROUP_ID	Int	Not null(Внешний ключ)

Т а б л и ц а 1.17 – VIDEO

Содержание поля	Имя поля	Тип, длина	Примечание
ID видеозаписи	VIDEO_ID	int	Первичный ключ
Адрес	VIDEO_URL	Varchar(255)	Not null
Название	VIDEO_TITLE	Varchar(255)	Null
ID группы	GROUP_ID	int	Not null(Внешний ключ)

Т а б л и ц а 1.18 – PHOTO

Содержание поля	Имя поля	Тип, длина	Примечание
ID фото	PHOTO_ID	int	Первичный ключ
Адрес	PHOTO_URL	Varchar(255)	Not null
Описание	PHOTO_DSP	text	Null
ID пользователя	USER_ID	int	Not null(Внешний ключ)
ID альбома	ALBUM_ID	int	Null(Внешний ключ)
ID группы	GROUP_ID	int	Not Null(Внешний ключ)

Т а б л и ц а 1.19 – MEMBERS

Содержание поля	Имя поля	Тип, длина	Примечание
ID участника	MEMBER_ID	int	Первичный ключ
ID группы	GROUP_ID	int	Not null(Внешний ключ)
ID пользователя	USER_ID	int	Null(Внешний ключ)
Описание	MEMBER_DSP	text	null

1.4.1 Расчет места для хранения БД

Перед запуском БД необходимо выставить необходимые размеры первоначальной БД и ее роста.

На этом этапе, необходимо знать какой объем памяти будет занимать создаваемая база данных. Объем внешней памяти, необходимый для функционирования системы, складывается из двух составляющих: память, занимаемая модулями СУБД (ядро, утилиты, вспомогательные программы), и память, отводимая под данные (M_d). Наиболее существенным обычно является M_d . Объем памяти, занимаемый программными модулями пользователя, обычно невелик по сравнению с объемом самих данных, поэтому может не учитываться. В проекте рассчитывается предполагаемый максимальный объем памяти занимаемой БД. Расчет физической памяти приводится в таблицах 1.20 – 1.37.

Т а б л и ц а 1.20 – Расчет физической памяти для таблицы USERS

Имя поля	Тип, длина	Длина(байт)
ID	int	4
USERNAME	Varchar(20)	20
PASSWORD	Varchar(128)	128
EMAIL	Varchar(128)	128
ACTIVKEY	Varchar(128)	128
CREATE_AT	Datetime	3
LASTVISIT_AT	Datetime	3
SUPERUSER	Int	4
STATUS	int	4

Общая длина строки: 422 байт

Число строк: ~ 1000

Общий объем требуемой памяти: ~ 422 000 байт

Т а б л и ц а 1.21 – Расчет физической памяти для таблицы SPECUSER

Имя поля	Тип, длина	Длина (байт)
SU_ID	int	4
SPEC_ID	int	4
USER_ID	int	4

Общая длина строки: 12 байт

Число строк: ~ 1500

Общий объем требуемой памяти: ~ 18 000 байт

Т а б л и ц а 1.22 – Расчет физической памяти для таблицы SPECIALNOST

Имя поля	Тип, длина	Длина (байт)
SPEC_ID	int	4
SPEC_NAME	Varchar(255)	255
SPEC_COUNT	int	4

Общая длина строки: 263 байт

Число строк: ~ 20

Общий объем требуемой памяти: ~ 5260 байт

Т а б л и ц а 1.23 – Расчет физической памяти для таблицы STATSONG

Имя поля	Тип, длина	Длина (байт)
STAT_ID	int	4
SONG_ID	int	4
USER_ID	int	4
DATA	datetime	3

Общая длина строки: 15 байт

Число строк: ~ 200 000

Общий объем требуемой памяти: ~ 3000000 байт

Т а б л и ц а 1.24 – Расчет физической памяти для таблицы SONG

Имя поля	Тип, длина	Длина (байт)
SONG_ID	int	4
SONG_URL	Varchar(255)	255
SONG_TITLE	Varchar(255)	255
SONG_LONG	int	4
SONG_ARTIST	Varchar(255)	255
SONG_ALBUM	Varchar(255)	255
SONG_BITRATE	int	4
SONG_LONG	int	4
SONG_ARTIST	Varchar(255)	255
SONG_ALBUM	Varchar(255)	255
SONG_BITRATE	int	4

Общая длина строки: 1036 байт

Число строк: ~ 20 000

Общий объем требуемой памяти: ~ 20 720 000 байт

Т а б л и ц а 1.25 – Расчет физической памяти для таблицы FANS

Имя поля	Тип, длина	Длина (байт)
FAN_ID	int	4
GROUP_ID	int	4
USER_ID	int	4

Общая длина строки: 12 байт

Число строк: ~ 50 000

Общий объем требуемой памяти: ~ 600 000 байт

Т а б л и ц а 1.26 – Расчет физической памяти для таблицы REPBASE

Имя поля	Тип, длина	Длина (байт)
BASE_ID	Int	4
BASE_ADRESS	Varchar(255)	255
BASE_TEL	Varchar(12)	12
BASE_DSP	text	500
BASE_PRICE	int	4
CITY_ID	int	4

Общая длина строки: 779 байт

Число строк: ~ 50

Общий объем требуемой памяти: ~ 38 950 байт

Т а б л и ц а 1.27 – Расчет физической памяти для таблицы CLUB

Имя поля	Тип, длина	Длина (байт)
CLUB_ID	Int	4
CLUB_ADRESS	Varchar(255)	255
CLUB_TEL	Varchar(12)	12

Имя поля	Тип, длина	Длина (байт)
CLUB_DSP	text	500
CLUB_PRICE	Int	4
CITY_ID	int	4
CLUB_LOGO	Varchar(255)	255

Общая длина строки: 1038 байт

Число строк: ~ 50

Общий объем требуемой памяти: ~ 51 900 байт

Т а б л и ц а 1.28 – Расчет физической памяти для таблицы GROUP

Имя поля	Тип, длина	Длина (байт)
GROUP_ID	int	4
GROUP_NAME	Varchar(255)	255
GROUP_DSP	text	500
GROUP_YEAR	Datetime	3
GROUP_IMG	Varchar(255)	255
GROUP_LOGO	int	4
GROUP_MANAGER	Varchar(255)	255

Общая длина строки: 1276 байт

Число строк: ~ 500

Общий объем требуемой памяти: ~ 638 000 байт

Т а б л и ц а 1.29 – Расчет физической памяти для таблицы GROUPSTAT

Имя поля	Тип, длина	Длина (байт)
STAT_ID	int	4
GROUP_ID	int	4
USER_ID	int	4
DATA	datetime	3

Общая длина строки: 15 байт

Число строк: ~ 50 000

Общий объем требуемой памяти: ~ 750 000 байт

Т а б л и ц а 1.30 – Расчет физической памяти для таблицы CITY

Имя поля	Тип, длина	Длина (байт)
CITY_ID	Int	4
CITY_NAME	Nchar(200)	200
CITY_COUNT	bigint	8
COUNTRY_ID	int	4

Общая длина строки: 216 байт

Число строк: ~ 50

Общий объем требуемой памяти: ~ 10 800 байт

Т а б л и ц а 1.31 – Расчет физической памяти для таблицы COUNTRY

Имя поля	Тип, длина	Длина (байт)
COUNTRY_ID	int	4
COUNTRY_NAME	Nchar(200)	200
COUNTRY_COUNT	bigint	8

Общая длина строки: 212 байт

Число строк: ~ 10

Общий объем требуемой памяти: ~ 2 120 байт

Т а б л и ц а 1.32 – Расчет физической памяти для таблицы CONCERT

Имя поля	Тип, длина	Длина (байт)
CONCERT_ID	int	4
CONCERT_TITLE	Varchar(255)	255
CONCERT_DSP	text	500
CONCERT_PRICE	int	4
CONCERT_IMG	Varchar(255)	255
CLUB_ID	int	4
CONCERT_DATE	datetime	3
ID	int	4

Общая длина строки: 1029 байт

Число строк: ~ 1000

Общий объем требуемой памяти: ~ 1029000 байт

Т а б л и ц а 1.33 – Расчет физической памяти для таблицы PROFILE

Имя поля	Тип, длина	Длина(байт)
USER_ID	int	4
LASTNAME	varchar(50)	50
FIRSTNAME	varchar(50)	50
USER_IMG	varchar(255)	255
USER_BIO	Text	500
CITY_ID	Int	4

Общая длина строки: 863 байта

Число строк: ~ 1000

Общий объем требуемой памяти: ~ 863 000 байт

Т а б л и ц а 1.34 – Расчет физической памяти для таблицы CONCERTMEMBERS

Имя поля	Тип, длина	Длина (байт)
CM_ID	Int	4
CONCERT_ID	Int	4
GROUP_ID	Int	4

Общая длина строки: 12 байта
 Число строк: ~ 5000
 Общий объем требуемой памяти: ~ 60 000 байт

Т а б л и ц а 1.35 – Расчет физической памяти для таблицы VIDEO

Имя поля	Тип, длина	Длина(байт)
VIDEO_ID	int	4
VIDEO_URL	Varchar(255)	255
VIDEO_TITLE	Varchar(255)	255
GROUP_ID	int	4

Общая длина строки: 518 байта
 Число строк: ~ 1000
 Общий объем требуемой памяти: ~ 518 000 байт

Т а б л и ц а 1.36 – Расчет физической памяти для таблицы PHOTO

Имя поля	Тип, длина	Длина (байт)
PHOTO_ID	int	4
PHOTO_URL	Varchar(255)	255
PHOTO_DSP	text	500
USER_ID	int	4
ALBUM_ID	int	4
GROUP_ID	int	4

Общая длина строки: 771 байта
 Число строк: ~ 100 000
 Общий объем требуемой памяти: ~ 77 100 000 байт

Т а б л и ц а 1.37 – Расчет физической памяти для таблицы MEMBERS

Имя поля	Тип, длина	Длина (байт)
MEMBER_ID	int	4
GROUP_ID	int	4
USER_ID	int	4
MEMBER_DSP	text	500

Общая длина строки: 512 байта
 Число строк: ~ 2500
 Общий объем требуемой памяти: ~ 1 280 000 байт

Таким образом, из полученных данным можно рассчитать приблизительный максимальный объем базы данных. Объем будет равен сумме объемов всех таблиц.

$M_d = 422\,000 + 18\,000 + 5\,260 + 30\,000\,000 + 20\,720\,000 + 600\,000 + 38\,950 + 51\,900 + 638\,000 + 750\,000 + 10\,800 + 2\,120 + 1029\,000 + 863\,000 + 60\,000 + 518\,000 + 77\,100\,000 + 1\,280\,000 = 107\,107\,030$ байт = **102** (Мб).

Объём памяти, занимаемый программными модулями пользователя, обычно невелик по сравнению с объёмом самих данных, поэтому может не учитываться. Требуемый объём оперативной памяти определяется на основании анализа интенсивности запросов и объёма результирующих данных.

1.4.2 Задание места хранения БД. Загрузка базы данных

Исходя из данных, полученных во время физического проектирования можно сделать вывод, что размер данных БД за первый год работы базы будет равен **102** Мб. На следующий год этот размер будет увеличен в 2 раза. Перед запуском БД необходимо выставить необходимые размеры первоначальной БД и ее роста.

1.5 Используемые технологии программирования

1.5.1 Проектирование на основе WEB 2.0

Web 2.0 – это такой способ проектирования сайтов, при котором пользователь сам заполняет их контентом, таким образом развивая и улучшая его. Чем больше людей пользуется сайтом, тем он становится качественнее и лучше. Главный принцип этой технологии – привлечение пользователей для наполнения ресурса информацией. Все социальные сети, блоги и некоторые открытые онлайн-энциклопедии являются представителями систем, созданных на основе Web 2.0. Однако использовать такую технологию – это значит зависеть от сторонних сервисов и засорять важную информацию потоком неконтролируемых высказываний от анонимных пользователей или посетителей, скрывающихся за псевдонимами («затопление вздором» или «BS-flooding»). Многие пользователи интернета не доверяют сайтам, построенным по принципам Web 2.0, так как сбор персональных данных, информация о карьере, о друзьях и интересах позволяет государству манипулировать людьми. Но реалии современной жизни таковы, что, несмотря на этот негативный фактор, люди испытывают огромную потребность в виртуальной жизни в интернете.

Впервые термин Web 2.0 придумал в 2005 году основатель издательской компании O'Reilly Media Тим О'Рейли, обозначив им новую веху в истории развития интернета [1]. На смену статичным персональным сайтам Web 1.0 пришли интерактивные блоги, во главу угла которых ставится общение с читателями и обмен мнениями.

1.5.2 Фреймворк Yii

Фреймворк — это каркас для программы, объединяющий разрозненные элементы компьютерной системы. Особенностью фреймворков является то, что такая конфигурация системы состоит из двух частей: каркаса и сменных модулей. Каркасы не меняются в зависимости от конфигурации и архитектуры системы, и в их гнездах располагаются сменные модули, которые иначе называются точками расширения. Иногда подключаются дополнительные программы и библиотека кода, объединение которых происходит благодаря единому API.

API расшифровывается как интерфейс программирования приложений. Это необходимый инструмент для прикладного программирования, состоящий из наборов классов и процедур, записанных в библиотеки и различные сервисы. API является встраиваемым набором модулей, который используется во внешних программах. При этом API представляет собой иерархическую структуру функциональности, при которой высокоуровневые компоненты используют более низкоуровневые, а те в свою очередь используют элементы еще более низкого уровня. Интерфейс включает в себя сигнатуру функций и их семантику. Обычно сигнатурой метода или функции является её имя и последовательность параметров. То есть сигнатура позволяет отличить одну функцию от другой и вызывать её для выполнения задач. А описанием того, что же делает данная функция, занимается семантика. В дипломном проекте использован Web API и представляет собой определенный набор HTTP-запросов [2]. С помощью Web API функционирует связь компонентов фреймворка Yii, который был выбран для данной работы.

При подготовке к созданию программы были рассмотрены фреймворки Zend Framework, Django и Yii. Zend Framework – свободный каркас на PHP, основанный на идеях MVC (шаблон проектирования Модель-Представление-Контроллер) представляет мощную функциональность, однако имеет довольно сложный интерфейс и документацию [3]. Django – также свободный фреймворк, однако веб-разработка ведется на языке Python, поэтому был выбран фреймворк Yii, который дает огромные возможности проектирования, при этом прост и удобен в обращении. К достоинствам данного фреймворка стоит отнести: активное и постоянное развитие, автоматическую генерацию кода, огромную библиотеку готовых расширений, простую документацию и поддержку JQuery.

Фреймворк Yii появился в 2008 году, и сразу вырвался в отрыв в негласном рейтинге веб-каркасов на PHP, так как имел по сравнению с ними более высокую производительность [4]. Большую популярность ему принесла и парадигма MVC, без которой современные фреймворки уже не могут конкурировать друг с другом.

MVC – это шаблон проектирования, разделенный на три компонента, при том, что изменение одного элемента не ведет за собой изменение других. Другими словами MVC старается минимизировать воздействие компонентов

друг на друга. В итоге, получается своеобразный архитектурный каркас, где сходятся теория и практика в предметной области. Модель и бизнес-логика отделяются от визуализация. Это очень удобно, так как к одной модели можно добавить несколько видов, при этом реализация модели не пострадает.

В данной модели присутствуют три основных компонента:

1 Модель (англ. Model). По-другому этот элемент можно назвать знаниями о данных, знаниями о методах работы с данными, как отвечать на запросы.

2 Представление, вид (англ. View). Показывает, как визуализировать данные с помощью графических элементов.

3 Контроллер (англ. Controller). Контролирует сообщения между пользователем и системой, используя две предыдущие компоненты для создания нужной реакции.

Таким образом, фреймворк Yii отвечает всем современным стандартам веб-программирования и отлично подходит для данной дипломной работы в качестве каркаса для интернет-портала.

1.5.3 Библиотека JQuery

JQuery – это расширение для Javascript, позволяющее этому языку программирования проще взаимодействовать с HTML. Для данной работы был выбрана именно эта библиотека, а не просто Javascript, потому что расширение предоставляет огромное количество необходимых функций, которые уже реализованы и остается лишь подставить нужные переменные и параметры. С появлением этой библиотеки в веб-программировании осуществилась html-javascript шаблонизация, которая раньше казалась невозможной и неуместной.

Однако, как показало время, jQuery оказалось революционным средством, которое разделило логику и представление не на серверном языке (PHP), а уже на уровне javascript. Таким образом, у веб-разработчиков появились новые возможности для создания пользовательского интерфейса и визуальных эффектов, которые становятся все популярнее и популярнее с каждым днем. Интерактивный и необычный дизайн сейчас является ключом к успеху веб-проекта. В эпоху Web 1.0 сайты были довольно однотипные и статичные, но их было не так много, поэтому пользователю в первую очередь важен был уникальный контент. Конечно, и сейчас содержимое сайта имеет ключевую роль в формировании интереса пользователя, но веб-порталов стало гораздо больше, и из двух ресурсом со схожими предложениями, спрос будет у того, чей дизайн более удобный и красивый.

Колоссальное количество пользовательских расширений предоставляют практически не ограниченные возможности, поэтому программисту остается лишь собирать плагины как конструктор, и от идеи до реализации остается всего один шаг. В целом, данное расширение обеспечивает пользователю полнофункциональную и удобную программу для создания интерактивных качественных приложений в короткие сроки.

1.5.4 Twitter Bootstrap

Twitter Bootstrap – это специальный набор элементов для удобства проектирования интерфейса пользователя. Если JQuery отвечает за анимацию и обработку событий, то Bootstrap предназначен для создания панелей навигации и блоков. Готовые наборы компонентов и дизайнерских решений превратят обычный HTML-документ в красивый сайт. Достаточно всего лишь добавить к тегу какой-либо класс, задействованный в Twitter Bootstrap, и вместо десятков строк кода в таблице стилей, который иначе пришлось бы набирать вручную, мы, всего лишь подключив внешнюю таблицу bootstrap.css, сразу получаем отличную визуализацию.

Twitter Bootstrap является фронт-энд фреймворком, то есть не затрагивающим серверную часть. С каждым годом все больше корпораций используют его для разработки дизайна, включая сайт NASA. Впервые Bootstrap появился в 2011 году, когда разработчики известной сети twitter.com Марк Отто и Джейкоб Торнтон создавали фреймворк для связки всех внутренних элементов этого ресурса [5].

Основные инструменты Bootstrap

Сетки (колонки) – это объекты с заранее заданными размерами, которые удобно использовать для разметки документа. В новой версии Twitter Bootstrap 3.0 класс данных объектов называется.col-md-* – для устройств с экранами средней величины, .col-xs-* – для устройств с экстремально-маленькими экранами (например, смартфонов и мобильных телефонов), .col-sm-* – для устройств с маленькими экранами (у планшетов) и.col-lg-* для устройств с большими экранами (телевизоры и широкодюймовые мониторы), где за коротким дефисом следуют простые числа от 1 до 12. Такой диапазон значений используется потому, что в концепции разработчиков Twitter Bootstrap экран делится на 12 маленьких частей. Таким образом, .col-md-2 означает что объект данного класса будет занимать 2 /12 или 16% от ширины экрана. Разумеется, в пикселях значение класса.col-* будет разным для мониторов разных размеров.

Шаблоны – к данному типу относятся фиксированный и резиновый макеты сайта или веб-приложения. Под фиксированным макетом понимается неизменяемая ширина макета сайта. Такой шаблон фиксирован по своим размерам, он не изменяется в зависимости от параметров окна браузера или экрана. Данный тип разработки очень прост, так как пользователи с различными устройствами будут видеть одно и то же представление сайта или веб-приложения. Информация не будет искажаться, но будет отображаться строго в соответствии с заданными фиксированными разработчиком параметрами. К преимуществам фиксированного шаблона можно отнести то, что задуманный вид сайта будет одинаково отображаться всегда и везде – на любых системах с любыми экранами, какое-либо искажение в данном случае почти невозможно. Однако, как и у всего на этой планете, у фиксированного шаблона есть и недостатки. Например, если шаблон отлично подходит для окон

средней величины, такой негибкий дизайн будет не эстетично смотреться на маленьких или больших мониторах, к тому же возможно появление полос прокрутки, которые закроют собой полезную информацию, затруднят или полностью сведут на нет навигацию сайта или веб-приложения, что в конечном итоге отпугнёт пользователей. Резиновый дизайн – это совершенно другая концепция.

При изменении параметров окна – также изменяется и внешний вид шаблона. Такая интересная вещь происходит из-за масштабирования объектов. То есть масштабирует объект не сервер, а клиент. Неважно – на мобильном телефоне или на экране домашнего кинотеатра просматривается приложение – картинка всегда будет автоматически подстраиваться под размер монитора, становясь то уже, то шире. Благодаря этому, все элементы будут видны и ничего не будет скрыто от пользователя. Такая открытая и дружелюбная, можно сказать даже умная приспособляемость располагает к себе клиента, вызывая у него чувство эстетического удовольствия. Однако, при неумелом проектировании разработчиком макета, чувство наслаждения может быстро перерасти в раздражение и даже злобу. Например, текст может стать нечитабельным – если нет фиксированных границ, строки слишком растянутся на больших мониторах, что быстро утомляет шейные мышцы и глаза. Такая непредсказуемость негативно воздействует на ЦНС человека. Графика может неожиданно оказаться не в том месте, где пользователю предстояло её увидеть. К тому же некоторые браузеры работают с резиновым кодом отлично от других, таким образом, внешний вид приложения отобразится некорректно. Такая непредсказуемость негативно воздействует на ЦНС человека. Приложения с резиновым макетом грузятся дольше фиксированных, и это ожидание так же расстраивает посетителя.

Типографика – это описания шрифтов и определение классов для них. Очень удобным инструментом является выделение цитат и оформление кода особым шрифтом. Специальные теги избавляют программиста от утомительного написания отдельных CSS классов для цитат и кода. Революционный тег `<blockquote>` заменит десятки свойств (таких как отсчитывание отступов, выделение полужирным шрифтом и курсивом и т.п.), а с тегом `<code>` набранный текст превратится в удобно оформленный для чтения код [6].

С помощью Twitter Bootstrap значительно упростился процесс оформления таблиц, вплоть до добавления функциональности сортировки. Например, с помощью класса `table-striped` к таблице добавляется полосатый фон строк – серая полоса и белая. Вот так, всего лишь одним классом можно во много раз повысить удобочитаемость таблицы – строки не сливаются, и смотреть на них – одно удовольствие. А класс `table-bordered` – добавляет псевдо-трехмерные границы для всех ячеек таблицы, таким образом, выделяя их и подчеркивая значимость информации, которую они содержат. Класс `table-condensed` в наше время актуален как никогда – он позволяет визуально сузить таблицу, тем самым приводя её в красивый и аккуратный вид, это очень

ценится в крупных корпорациях, таких как Apple и Google. Но самое главное удобство в оформлении таблиц Twitter Bootstrap привнес с помощью контекстных классов. С их помощью, в зависимости от контекста, можно акцентировать внимание на отдельных строках или столбцах таблицы. Успешное действие пользователя над таблицей окрасится в позитивный зеленый цвет при помощи класса.success, а предупредительная строка класса.warning станет красноватой. Немалую долю успеха бутстрапу принес уникальный класс .table-responsive, который позволяет появляться полосам прокрутки внутри самой таблицы. На маленьких устройствах это просто незаменимое свойство.

Twitter Bootstrap создал большое число классов для оформления форм, то есть таких элементов приложения как блоки ввода текста, паролей и других заполняемых пользователем полей. С помощью данных нововведений легко можно расположить форму горизонтально или вертикально, настроить ширину и высоту каждого поля и установить фокус указателя мыши на определенное поле. Но больше всего полезных функций создано для оформления различных кнопок, табов, вкладок, меню и тулбара – то есть объектов для навигации по сайту. Например, раскрасить кнопку, которая обязательно должна привлекать внимание целесообразнее в красный цвет, а кнопку с предупреждением лучше всего сделать оранжевой. Для этого в Twitter Bootstrap созданы несколько классов, рисующих кнопки определенного цвета. Практически в каждом макете для бустрапа есть готовая панель навигации – вертикальная или горизонтальная, с выпадающим меню или без. Для того чтобы вместо стандартной белой панели добавить на страницу черную панель, совсем не обязательно вручную менять цвет фона, цвет шрифта и задний фон. Всё, что нужно – это добавить к тегу класс.inverse. Twitter Bootstrap добавили очень много полезных функций, которые берегут время разработчика и позволяют ему сделать дизайн веб-приложения гораздо быстрее и лучше. Поэтому в данной работе я решил использовать именно эту замечательную технологию.

1.5.5 AJAX

AJAX – это особый подход к построению веб-программ, основанный на интерактивности интерфейсов пользователя. Название AJAX расшифровывается по-русски как «асинхронный JavaScript и XML». Обмен данными между сервером и клиентской частью в этом случае происходит в фоновом режиме. Обновляя страницу, пользователь будет с удовольствием наблюдать, как быстро и удобно перезагружаются данные. При этом часть данных не перегружается полностью, за счёт чего и появляется ускорение. Концепция данной технологии в том, что она использует некоторое количество смежных технологий, то есть не является полностью самостоятельной.

AJAX имеет в своём основании два главных принципа, описанных разработчиками еще в самом начале появления этой революционной технологии.

Во-первых, это применение особой технологии динамического обращения к серверу, не перезагружая полностью страницы. Этого можно добиться несколькими способами. Например, можно динамически создавать дочерние фреймы или тег `<script>`. Компания Google в своей программе Google Analytics, которая анализирует продажи и конверсии, а также позволяет следить за статистикой пользователей на сайтах, использует для более быстрой работы динамическую загрузку тега ``. Однако, самый распространенный способ – это воспользоваться помощью специального API, который называется XMLHttpRequest, доступное в скриптовых языках, таких как JavaScript. API использует запросы протоколов передачи гипертекста, в том числе и защищенных соединений, применяя их напрямую к веб-серверу, а его ответ загружает в скрипт, написанный на JavaScript. Информация передается в формате XML или HTML. Так как снимается нагрузка на сервер, страницы не перезагружаются полностью и открываются быстрее. Такие запросы XMLHttpRequest являются главной частью технологии AJAX, позволяющий быстро реагировать на запросы пользователя. Именно этот API применяется для асинхронного взаимодействия с сервером. XMLHttpRequest используется ведущими сайтами мира, такими как Facebook и Google.

Во-вторых, AJAX – это использование DHTML. DHTML представляет собой сочетание статичного HTML, встраиваемого языка JavaScript, каскадных таблиц стилей CSS и объектной модели документа DOM. В основном такой способ создания динамики на сайте используется для упрощения навигации и создания интерактивности внутри веб-приложения. Например, с его помощью удобно создавать динамическую технологию Drag-and-drop, или попросту перетаскивать объекты с места на место, или делать небольшие браузерные видеоигры. Также, благодаря этой методике работает система S5 для показа презентаций, выполняемых онлайн в браузерах. К тому ж, DHTML отлично работает с базами данных. Можно даже сказать, что подключение к базе данных – это одна из необходимых вещей для нормального функционирования приложения, написанного с использованием DHTML [7]. DHTML потихоньку вытесняет с рынка конкурирующие ресурсы Adobe Flash и Silverlight как Архимед выталкивал объемом своего тела лишние кубометры воды.

AJAX производит действия над элементами DOM, ведь именно они обрабатывают данные и могут изменять их представление. Объектная модель документа не зависит от платформы и является по сути программным интерфейсом, способным менять содержимое HTML-документов. Эта модель представляет документ как дерево, состоящее из веток и листьев, где каждая ветвь представляет собой какой-то компонент и объект. Ветви бывают дочерними и родительскими.

Значительные преимущества этой технологии в том, что её использование значительно экономит трафик, потому загружается только изменившаяся часть страницы. Похожий метод используется и в видео-компрессии. В тоже время этот же набор данных можно получить в формате XML, который JavaScript превратит в нужную пользователю информацию. Уменьшение

нагрузки на сервер из-за того, что одинаковые элементы дизайна не грузятся каждый раз при переходе на другую страницу сайта, позволяет использовать HTTP-соединения гораздо выгоднее. Пользователь замечает, что сайт более отзывчивый и значительно быстрее реагирует на нажатие кнопки и переходы по внутренним ссылкам ресурса.

Однако бездумное использование AJAX может привести к негативным для пользователя последствиям. Так как динамически генерируемые страницы не регистрируются поисковыми системами, кнопка «назад» на панели браузера не будет работать. Например, этим недостатком страдает сайт о клингонском языке, что затрудняет его изучение, мешает скачивать материалы и попросту запутывает пользователя. При этом нельзя даже сохранить страницу с нужной информацией в закладки. К тому же, такие сайты не будут продвигаться вверх в рейтинге поиска, так как Google или Яндекс их просто не видят. Это замедлит или вовсе сведет на нет посещение сайта новыми пользователями. А если у клиента на компьютере выключен JavaScript в настройках браузера, то сайт для такого пользователя вообще не будет работать.

Так что данную технологию нужно использовать с умом. В руках профессионала или просто умного человека, этот метод будет способствовать улучшению работы веб-сайта, ускорению загрузки страниц и экономии трафика посетителя, а следовательно и его денег.

1.6 Используемые языки программирования

1.6.1 HTML5

HTML не является языком программирования в строгом смысле этого слова, но мы будем рассматривать его в этом разделе. HTML – это язык разметки документов. Он интерпретируется браузерами и визуализирует информацию на экран. Этот язык был разработан Тимом Бернерсом-Ли в 1991 году [8]. Языком разметки HTML является, так как использует специальные семантические и структурные элементы – дескрипторы, которые иначе зовутся «тегами». В настоящее время стандартизируется и разрабатывается пятая версия языка HTML, широко известная HTML5. Цель этой версии – улучшение поддержки мультимедиа и добавление единого API. По сути HTML5 – это объединения норм HTML и XHTML. В данную версию были добавлены теги <video> и <audio>, которые позволяют более оперативно взаимодействовать соответственно с видеофайлами и аудиозаписями, причем отлично реализована функция добавления видео со сторонних ресурсов. Тег <canvas> позволяет рисовать двумерную растровую графику с помощью javascript [9].

В данной дипломной работе язык разметки HTML используется для создания каркаса сайта. Если рассматривать программный продукт как живой организм, то HTML – это его кости, его скелет и опора. Без него получится лишь бесформенный ком. JavaScript – это мышцы, благодаря которым тело может двигаться, CSS – это одежда, которая позволяет телу выглядеть достойно

и, если постараться, красиво. И, наконец, PHP и серверная часть с базами данных – это мозги, которые управляют мышцами, поэтому тело делает осмысленные движения, обладает возможностью реагировать на раздражители в виде запросов базы данных, а также получает память, где хранит воспоминания о пользователях и не забывает их просто так. Без какого-либо из этих компонентов, программа не будет являться полноценной, а без некоторых вообще не будет функционировать.

1.6.2 Язык PHP

PHP (англ. PHP: Hypertext Preprocessor – «PHP: гипертекста препроцессор»; первоначально Personal Home Page Tools – «Инструменты для создания персональных веб-страниц»; произносится пи-эйч-пи) – скриптовый язык программирования общего назначения, интенсивно для разработки веб-приложений применяется. Поддерживается в подавляющем большинстве провайдеров, и в настоящее время – это один из лидеров среди языков программирования динамических веб-сайтов. Группой энтузиастов язык и его интерпретатор разрабатываются в рамках проекта с открытым кодом. Проект распространяется под собственной лицензией GNU GPL.

PHP – это типизированный динамический язык взаимодействия человека с компьютером, в котором нет необходимости объявлять переменные и их тип. Преобразования между скалярными типами зачастую осуществляются без дополнительных усилий неявно (хотя, PHP дает сильные инструменты и для явного преобразования типов).

Синтаксис PHP подобен синтаксису языка Си. Некоторые элементы, массивы и цикл `foreach`, `if`, `then` и ассоциативные массивы, взяты из Perl.

1.6.3 Язык JavaScript

JavaScript – скриптовый язык, который очень часто используется при создании различных алгоритмов, впоследствии генерируемых в таблицы. Данный язык не что иное, как разновидность языка ECMAScript. Этот язык был разработан Бренданом Айком, работающим в компании-производителе браузеров Netscape, и естественно этот язык был встроен в обозреватель Нетскэйп Навигатор, начиная с версии 2.0. Произошло это великое событие зимой 1995 года. Изначально язык назывался Mocha, далее он был переименован в LiveScript [10]. На синтаксис оказали влияние языки Си и Java, и, поскольку технология Java была в то время очень популярной, LiveScript переименовали в JavaScript, получив соответствующую лицензию у Sun. Реализация компании Microsoft получила название JScript. Internet Explorer поддерживает JScript, начиная с версии 3.0, выпущенной в августе 1996 г. Стандартизированная версия имеет название ECMAScript, описывается стандартом ECMA-262. Первая версия данного стандарта примерно

соответствовала JavaScript 1.1 [11]. Стандартизация языка ассоциацией ECMA инициирована компанией Netscape.

Заголовок «JavaScript» есть зарегистрированный товарный знак компании Sun Microsystems, Inc.

Область применения языка

Среди сторонних продуктов, например, Java, начиная с версии 6, этот продукт содержит встроенный интерпретатор JavaScript на базе Rhino. Сценарии JavaScript поддерживаются в таких приложениях Adobe, как Adobe Photoshop, Adobe Dreamweaver, Adobe Illustrator или Adobe InDesign.

JavaScript находит применение в качестве скриптового языка доступа к объектам приложений. Платформа Mozilla (XUL/Gecko) использует JavaScript.

JavaScript в данный момент полностью занимает нишу браузерных языков. Несмотря на то, что по слухам некоторые разработчики браузеров встраивают (или уже встроили) в дополнение к JavaScript-у такой язык как Python, для динамического изменения веб-страниц на стороне клиента, официальной информации по этому вопросу нет [12].

О языке

Популярность JavaScript во многом обусловлена именно тем, что написанный на нем сценарий может реагировать на действия пользователя и другие внешние события. Каждое из событий связано с тем или иным объектом: формой, гипертекстовой ссылкой или даже с окном, содержащим текущий документ.

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

- объекты, с возможностью интроспекции и динамического изменения типа через механизм прототипов;
- функции как объекты первого класса;
- автоматическое приведение типов;
- автоматическая сборка мусора;
- анонимные функции.

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

Обработка события производится с помощью специально предназначенного для этого фрагмента кода, называемого обработчиком события. Для каждого события JavaScript предоставляет свой обработчик. Однако при построении сценария можно создавать собственный обработчик события и использовать его вместо обработчика, заданного по умолчанию.

Имя обработчика определяет, какое событие он должен обрабатывать. Так, для того чтобы сценарий нужным образом отреагировал на щелчок мышью, используется обработчик с именем `onClick`, для обработки события, заключающегося в получении фокуса ввода, - обработчик `onFocus`.

JavaScript обладает рядом свойств объектно-ориентированного языка, но благодаря прототипированию поддержка объектов в нём отличается от традиционных ОО языков. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам - функции как объекты первого уровня, объекты как списки, карринг (`currying`), анонимные функции, замыкания (`closures`) - что придает языку дополнительную гибкость [13].

Использование в HTML

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом простым для использования непрограммистами. Языком JavaScript какая-либо компания или организация не владеет, что его от языков программирования ряда отличает, используемых в веб-разработке.

Встраиваемый язык JavaScript используется как, обычно для доступа к объектам приложений программного. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. Основные архитектурные черты: слабая типизация, динамическая типизация, управление памятью автоматическое, функции как объекты первого класса, прототипное программирование.

Практикой для такого рода вещей является отделение javascript от разметки документа, что достигается упорным трудом и напористостью. Для этого его помещают внутрь тега `head`, а в теле страницы по возможности оставляется.

JavaScript используется как, обычно для доступа к объектам приложений программного. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. Основные архитектурные черты: слабая типизация, динамическая типизация, управление памятью автоматическое, функции как объекты первого класса.

Кроме этого, JavaScript обладает рядом свойств объектно-ориентированного языка, но благодаря прототипированию поддержка объектов в нём отличается от традиционных ОО языков [14]. Помимо всего прочего, JavaScript имеет ряд свойств, присущих функциональным языкам - функции как объекты первого уровня, объекты как списки, карринг, анонимные функции, замыкания, что придает языку особое удобство.

2 Расчетная часть

2.1 Логическое проектирование

Задача логического проектирования состоит в создании реляционных табличных структур на языке DDL.

Ниже представлен код создания нескольких таблиц:

```
CREATE TABLE IF NOT EXISTS `City` (  
    `City_id` int(11) NOT NULL AUTO_INCREMENT,  
    `City_name` varchar(255) NOT NULL,  
    `City_count` int(11) NOT NULL DEFAULT '0',  
    `Country_id` int(11) NOT NULL,  
    PRIMARY KEY (`City_id`))  
ENGINE=MyISAM DEFAULT CHARSET=cp1251 AUTO_INCREMENT=2 ;
```

```
CREATE TABLE IF NOT EXISTS `CLUB` (  
    `CLUB_ID` int(11) NOT NULL AUTO_INCREMENT,  
    `NAME` varchar(255) NOT NULL,  
    `CLUB_X` varchar(50) NOT NULL,  
    `CLUB_Y` varchar(50) NOT NULL,  
    `CLUB_DSP` text,  
    `CLUB_ADRES` varchar(255) DEFAULT NULL,  
    `CLUB_FOTO` varchar(255) DEFAULT NULL,  
    `CITY_ID` int(11) DEFAULT NULL,  
    `USER_ID` int(11) DEFAULT NULL,  
    PRIMARY KEY (`CLUB_ID`))  
ENGINE=MyISAM DEFAULT CHARSET=cp1251 AUTO_INCREMENT=3 ;
```

Далее представлен код заполнения некоторых начальных таблиц:

```
INSERT INTO `Videos` (`Video_ID`, `Video_URL`, `Video_Title`,  
    `Group_id`) VALUES  
(14, '//www.youtube.com/embed/vSkb0kDacjs', NULL, 12),  
(12, '//www.youtube.com/embed/7-7knsP2n5w', NULL, 2),  
(13, '//www.youtube.com/embed/vSkb0kDacjs', NULL, 2),  
(10, '//www.youtube.com/embed/1-pUaogoX5o', NULL, 2);  
  
INSERT INTO `specialnost` (`Spec_id`, `Spec_name`,  
    `Spec_count`) VALUES  
(1, 'Гитара', 0),  
(2, 'Басист', 0),  
(3, 'Вокал', 0),  
(4, 'Клавиши', 0);
```

На рисунке 2.1 представлена диаграмма физической модели базы данных.

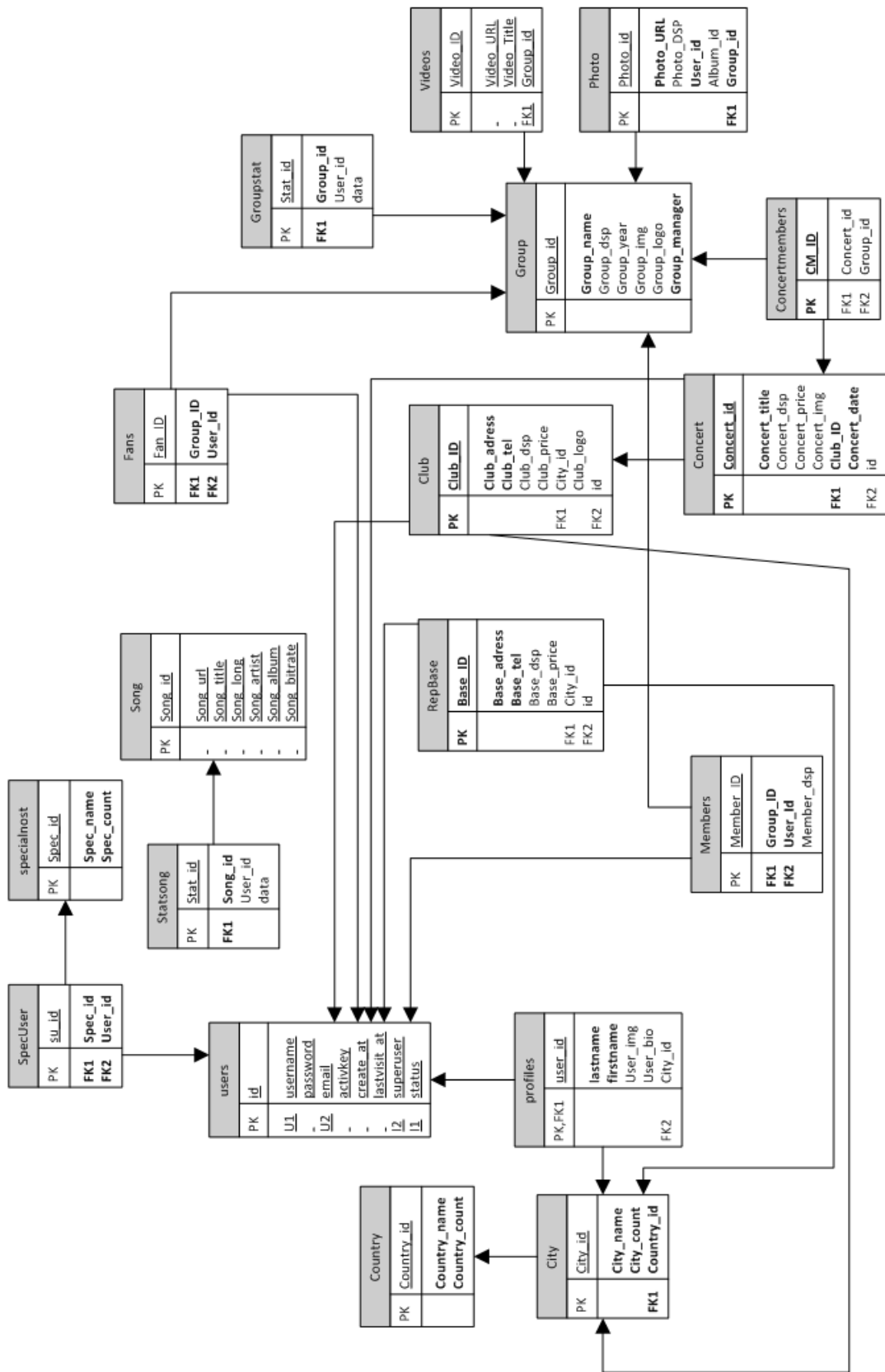


Рисунок 2.1 – Диаграмма физической модели данной базы данных

2.2 Физическое проектирование

2.2.1 Обоснование выбора СУБД

Для разработки базы данных была выбрана СУБД MySQL, которую разрабатывает и поддерживает компания Oracle. Данная система была выбрана, потому что имеет хорошую поддержку языка PHP, который используется в проектировании серверной части данной работы и имеет гибкость в поддержке проектирования таблиц. Устойчивость, большая скорость работы и простота использования также являются главными характеристиками данной СУБД. К тому же в данной работе используется программа для сборки серверов Денвер, в состав которой входит MySQL. Данная СУБД не подходит для работы с большими объемами данных, но для интернет-порталов это идеальное решение. MySQL распространяется бесплатно в соответствии с лицензией GNU [14]. Приложение на PHP, использующее базу данную работает быстрее, чем приложение, использующее для хранения файлы. Основное достоинство MySQL заключается в том, что она эффективно работает с жестким диском, так как язык C++, на котором написана база данных, является компилятором, а язык PHP – интерпретатором. Разумеется, компиляция выполняется гораздо быстрее.

MySQL является реляционной базой данных, то есть состоит из таблиц, которые, в свою очередь, состоят из строк и столбцов. Название «реляционные» произошло от того, что в таких СУБД таблицы всегда связаны друг с другом разными отношениями.

2.2.2 Меры по обеспечению безопасности

Одной из наиболее важных частей в БД является разработка прав доступа к ней, т.к. нужна защита от несанкционированного доступа и защита от доступа. Для защиты от сбоев разрабатывается стратегия резервного копирования. В моем случае, из-за специфики web-приложений ограничения доступа к информации осуществляется через само приложение

2.2.3 Создание пользователей, логинов и задание им паролей

Создание пользователя muz:

```
CREATE USER muz identified by oper  
QUOTA 200M on gaitbs;
```

Создание роли muzikant:

```
create role muzikant;
```

Присвоение пользователю oper роль operator:

```
grant create session to muzikant;  
grant alter session to muzikant;  
grant create table to muzikant;  
grant create synonym to muzikant;  
grant create view to muzikant;  
grant create sequence to muzikant;  
grant create trigger to muzikant;  
grant create procedure to muzikant;  
grant create user to muzikant;  
grant drop user to muzikant;  
grant create view to muzikant;  
grant create sequence to muzikant;  
grant create trigger to muzikant;  
grant create role to muzikant;  
grant alter user to muzikant;  
grant create session to muzikant;  
grant alter session to muzikant;  
grant create table to muzikant;  
grant create synonym to muzikant;  
grant create view to muzikant;  
grant create sequence to muzikant;  
grant create trigger to muzikant;  
grant create procedure to muzikant;  
grant create user to muzikant;  
grant drop user to muzikant;
```

```
grant mizikant to muz;
```

Создание пользователя admin:

```
CREATE USER admin identified by simple  
QUOTA 200M on gaitbs;
```

Создание роли administ:

```
create role administ;
```

Присвоение пользователю admin роль administ:

```
grant create session to administ;  
grant select on oper.ADMINISTR_STATIYI to administ;  
grant select on oper.Doverennosti to administ;  
grant select on oper.Doverennosti2 to administ;  
grant select on oper.KOLVO_SHTRAFOV to administ;  
grant select on oper.MASHINI_I_SHTRAFI to administ;  
grant select on oper.PRAVA_I_VLADELCY to administ;  
grant select on oper.VLADELCY_I_AVTO to administ;  
grant select on oper.VODITELI_I_SHTRAFI to administ;  
grant select on oper.VOZHDENIE_PO_DOVEREN to administ;  
grant select on oper.KOLVO_SHTRAFOV to administ;
```

```
grant select on oper.MASHINI_I_SHTRAFI to administ;  
grant select on oper.PRAVA_I_VLADELCEY to administ;  
grant select on oper.Doverennosti to administ;  
grant select on oper.Doverennosti2 to administ;  
grant select on oper.KOLVO_SHTRAFOV to administ;  
grant select on oper.VLADELCEY_I_AVTO to administ;  
grant select on oper.VODITELI_I_SHTRAFI to administ;  
grant administ to admin;
```

2.3 Описание интерфейса интернет-портала для музыкантов

Работа с приложением

Данный web-портал разработан с применением самых современных разработок в сфере пользовательского интерфейса. Код приложения представлен в приложении А.

С помощью моего портала, музыкантам будет проще заниматься творческим саморазвитием.

В интерфейсе сайта используются темные цвета для фона, полупрозрачные блоки, а также выделение форм белым цветом для контраста. Это позволяет снизить нагрузку на глаза пользователя и акцентировать внимание тех, кто использует систему, именно, на ее содержимом. Противоположностью моему интерфейсу можно назвать сквеморфизм, который уступил господство в интернете в начале 2013 года. Окончательная победа такого типа интерфейсов состоялась с выходом iOS 7 от компании Apple.

Без авторизации пользователи могут свободно просматривать материала сайта, но не добавлять свои материалы, для этого необходимо зарегистрироваться и авторизоваться. В системе предусмотрено разделение пользователей на 2 группы по уровню доступа: простой пользователь и администратор.

После авторизации все пользователи попадают на страницу своего профиля. На странице указаны данные о группах в которых он играет, личные данные, его умения. Также на этой странице отображаются песни групп, на которые пользователь подписался.

На рисунке 2.2 представлена главная страница сайта. В верхней части, на всю ширину контента располагается большая фотография из аннотации к последней загруженной новости. Если нажать на текст, расположенный слева на фотографии, то откроется страница самой новости.

Ниже большой фотографии расположен блок с аннотациями к последним новостям и статьям.

Помимо новостей, справа расположен блок с самыми популярными песнями на сайте за последнее время, а еще ниже лучшие клипы, загруженные пользователями.

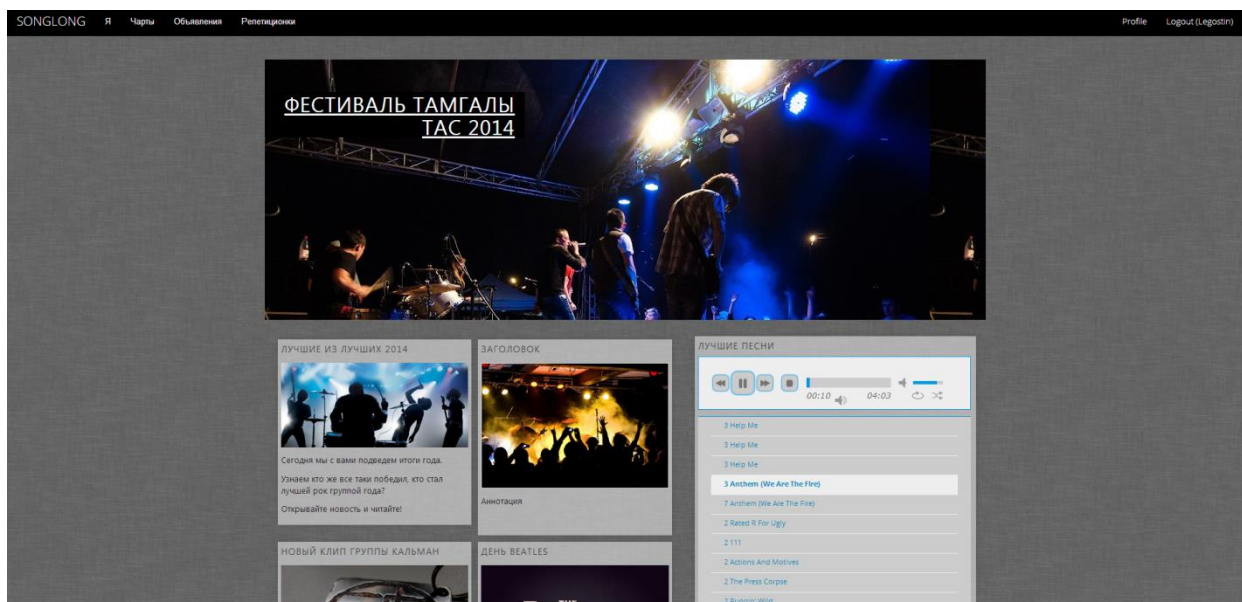


Рисунок 2.2 – Главная страница портала

Нажав на заголовок любой новости, посетитель портала попадет на страницу этой записи (Рисунок 2.3). Сама статья может содержать как текстовые материалы, так и произвольную HTML разметку и такие мультимедийные материалы как: аудиозаписи, фотографии, видеоролики. Также, зарегистрированные пользователи могут оставлять свои комментарии к записям.

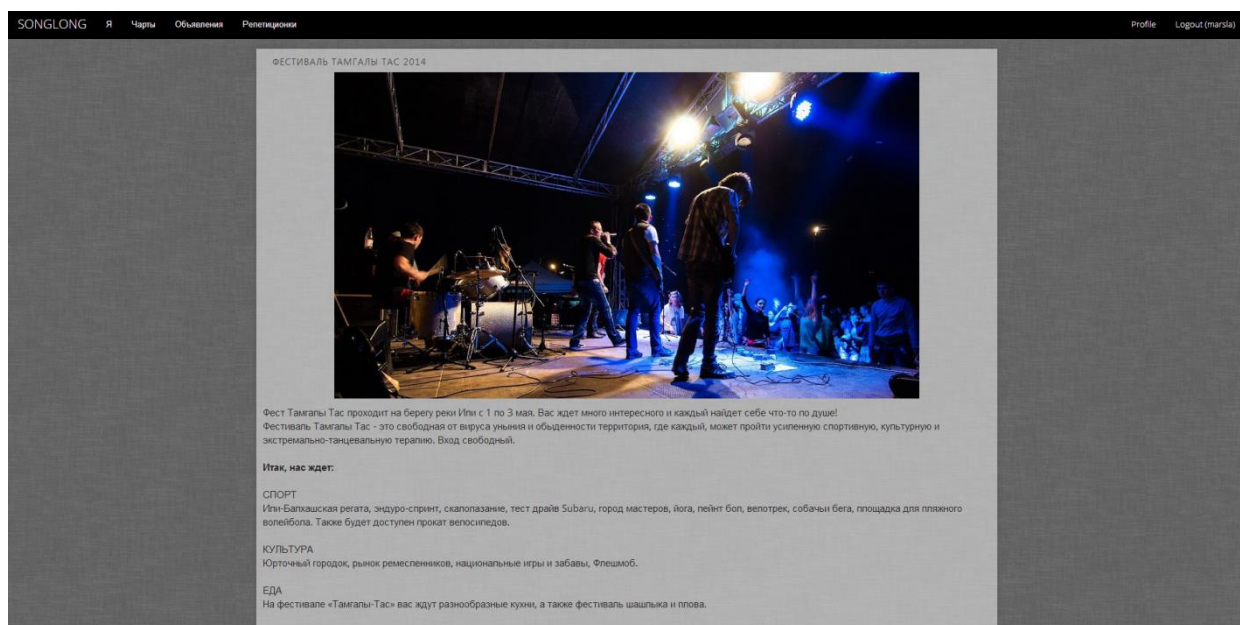


Рисунок 2.3 – Страница одной из новостей

Также, на странице новостей расположен блок с комментариями. Система комментариев создана с использованием модуля Disqus, что позволяет снизить

нагрузку на базу и хостинг, поскольку все комментарии хранятся на стороннем сервере (Рисунок 2.4).

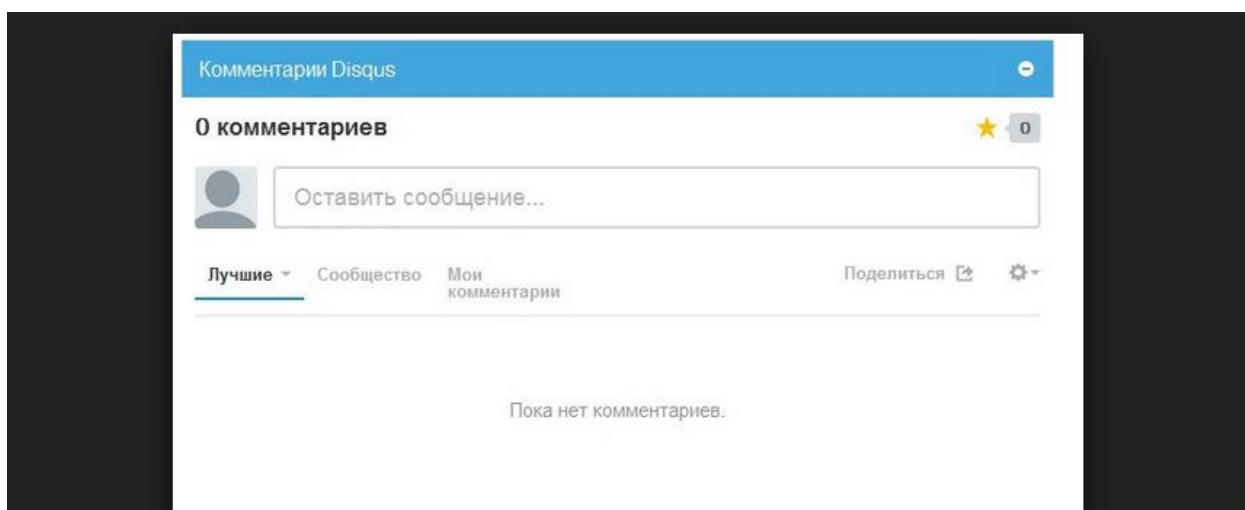


Рисунок 2.4 – Блок отправки комментария к статье

Для того чтобы самому начать пользоваться всем функционалом сайта, необходимо зарегистрироваться. На рисунке 2.5 представлена форма регистрации. Количество полей минимально, нет необходимости активировать свою запись через электронную почту, в дальнейшем имеется возможность подключить к профилю на сайте учетную запись одной или нескольких социальных сетей. Красивая и простая форма регистрации сразу-же настраивает пользователей на благоприятное отношение к сайту. Время, необходимое чтобы пройти всю регистрацию, составляет менее 30 секунд.

Также, имеется встроенная AJAX валидация формы.

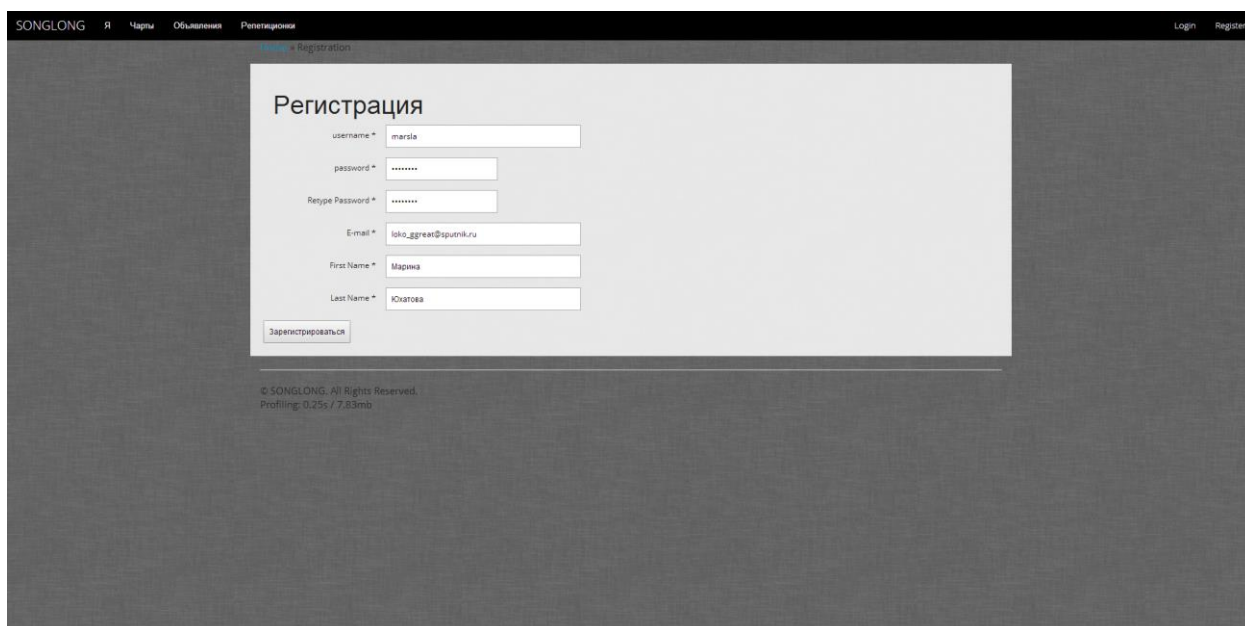


Рисунок 2.5 – Форма регистрации

После регистрации пользователь попадает в свой пустой профиль, пока не заполнены ни какие данные, кроме имени и фамилии. Но прямо отсюда, всего за несколько секунд можно улучшить свою страницу, с помощью форм различных настроек и добавления контента (Рисунок 2.6).

Все настройки применяются мгновенно и результат виден сразу же, это стало возможным, благодаря, глубоко интегрированному использованию AJAX запросов к системе.

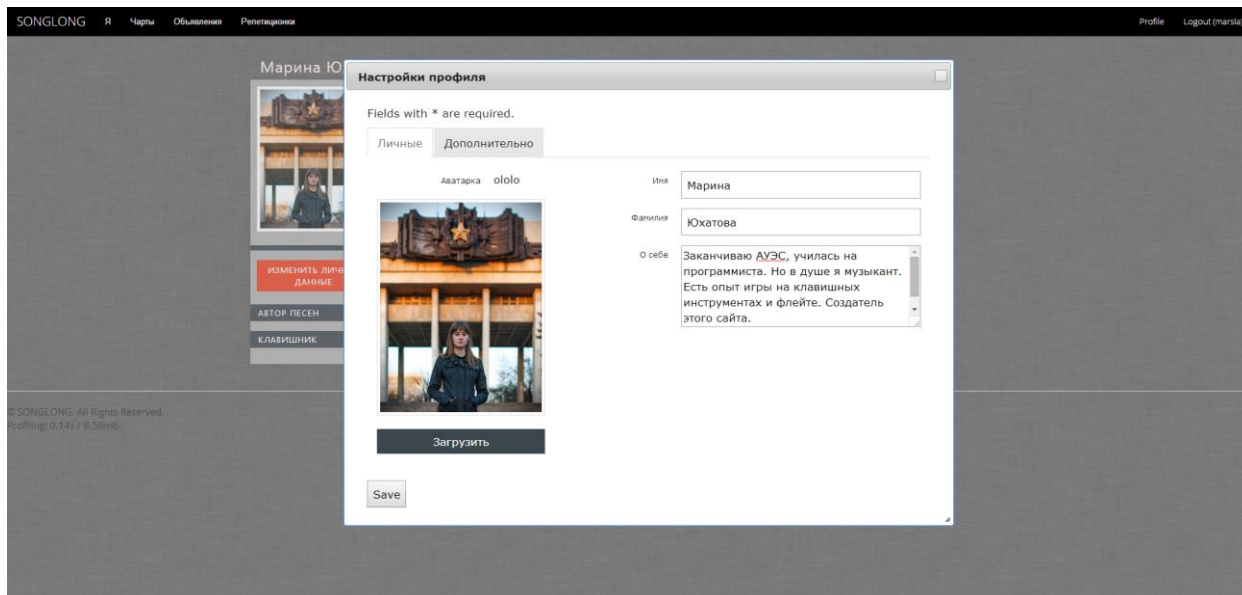


Рисунок 2.6 – Меню настройки личных данных пользователя

Прямо из профиля можно создать новую группу. Как можно увидеть, форма на рисунке 2.7 имеет всего одно поле, в котором вводится название группы. Использование столь простых форм сильно улучшает опыт использования сайта. Всего через несколько секунд после начала, группа создана, и можно приступать к ее настройке и добавлению контента.

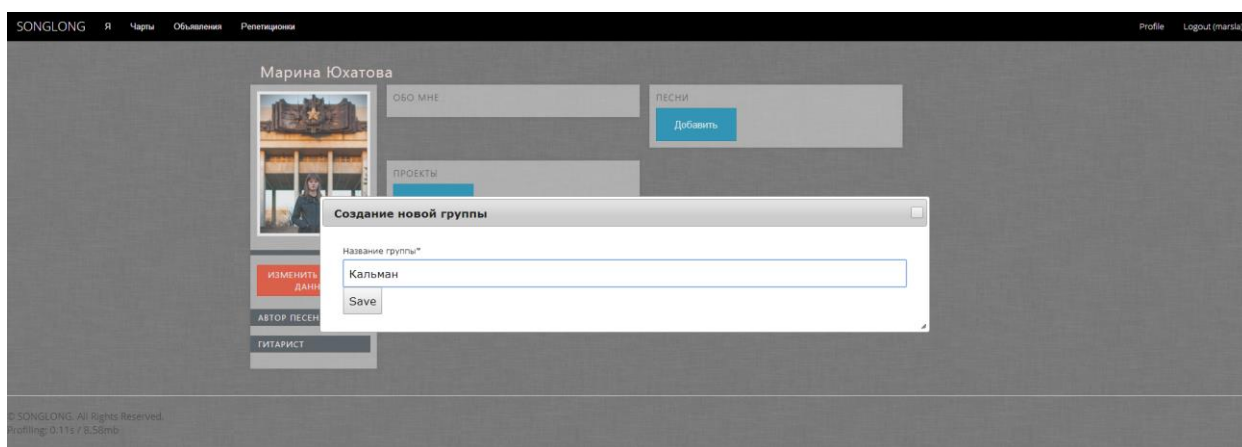


Рисунок 2.7 – Форма создания новой группы

Сразу после того как пользователь нажмет на кнопку сохранения, группа отобразится в списке проектов пользователя. Естественно, что кроме названия не будет выведено никакой информации (Рисунок 2.8). Однако, нажав на название группы, мы тут же попадем на страницу только что созданной группы.

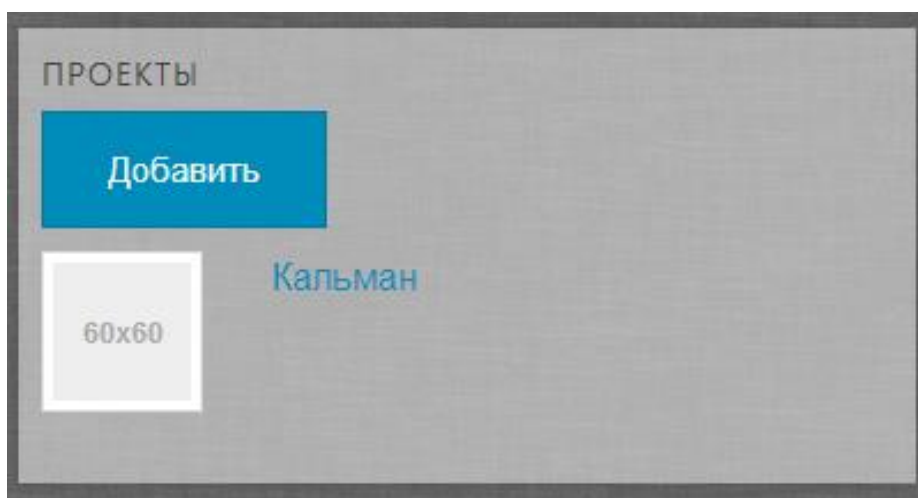


Рисунок 2.8 – Блок с только что созданной группой

После того как пользователь нажал на название группы и попал на ее страницу, он увидит совершенно пустую страницу группы, с которой можно будет начинать добавлять контент на страничку (Рисунок 2.9). Осуществляется этот процесс через вызов необходимых форм с помощью кнопок на странице.

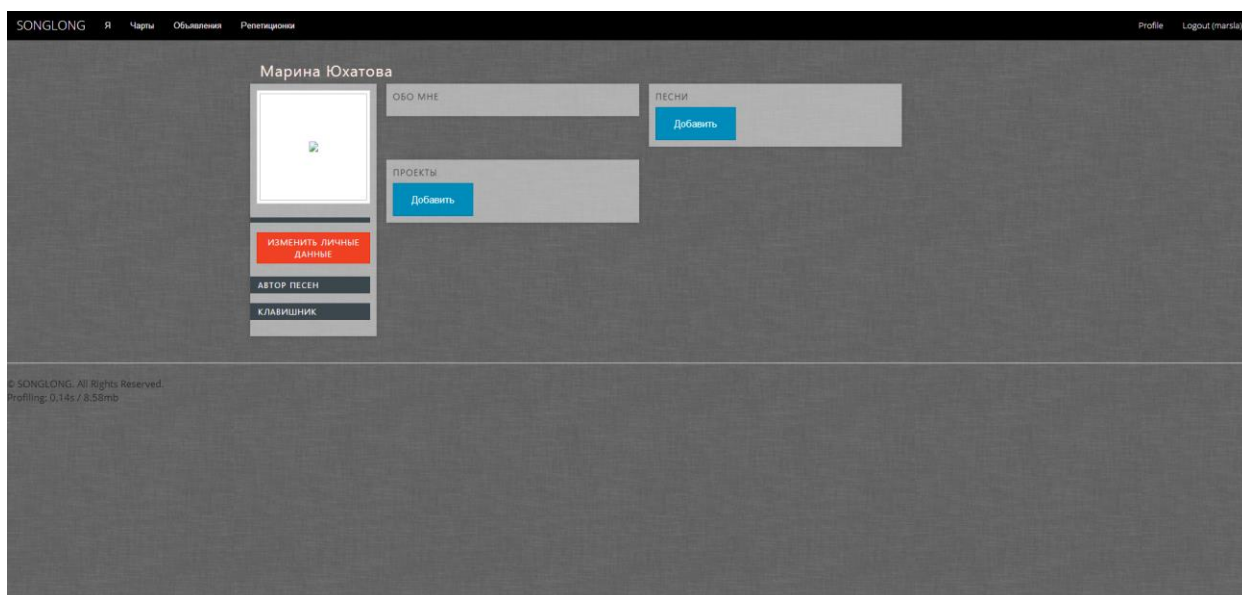


Рисунок 2.9 – Пустая страница только что созданной группы

Нажав на кнопку добавления материалов, в блоке песен откроется специальная форма для загрузки аудиозаписей (Рисунок 2.10). От пользователя требуется лишь выбрать mp3 файл на своем компьютере, чтобы его загрузить.

Все остальные параметры записи, такие как: название песни, альбом, битрейт, жанр и год записи автоматически будут считаны с помощью парсера id3 тегов из mp3 файлов и записаны в базу, для облегчения последующей классификации записей.

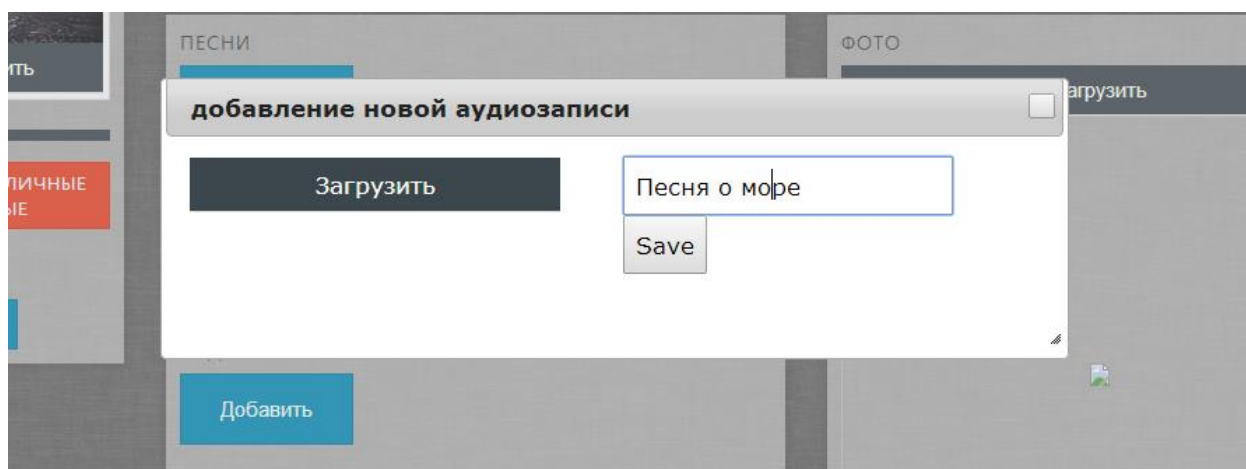


Рисунок 2.10 – Форма загрузки аудиозаписи на странице группы

После загрузки песня тут же появится в списке воспроизведения на странице. Сам плеер выполнен с использованием HTML5 технологий, а следовательно является кроссплатформенным (Рисунок 2.11). То есть работает как на стационарных компьютерах и ноутбуках, так и на планшетах с мобильными телефонами.

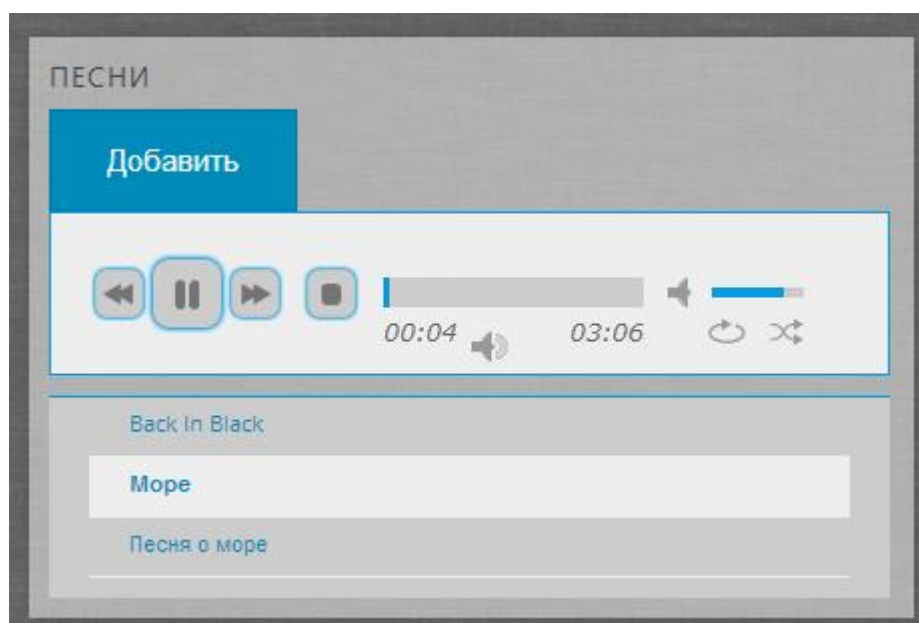


Рисунок 2.11– Блок со списком загруженных песен

Там же можно изменить всю информацию о группе, если нажать на соответствующую кнопку. (Рисунок 2.12). Настройки группы, как и другие

формы можно менять без перезагрузки страницы. Использование AJAX запросов позволяет организовать работу пользователя с сайтом, чтобы музыка, которую он слушает, никогда не прерывалась. Все асинхронные запросы к серверу осуществляются с помощью jQuery, JavaScript фреймворка, который на данный момент является неотъемлемой частью, практически любого мелкого или крупного проекта.

Для безопасности личных данных, пересылаемых на сервер, а также снижения риска передачи конфиденциальной информации используется метод POST, который гораздо сложнее перехватить, чем GET.

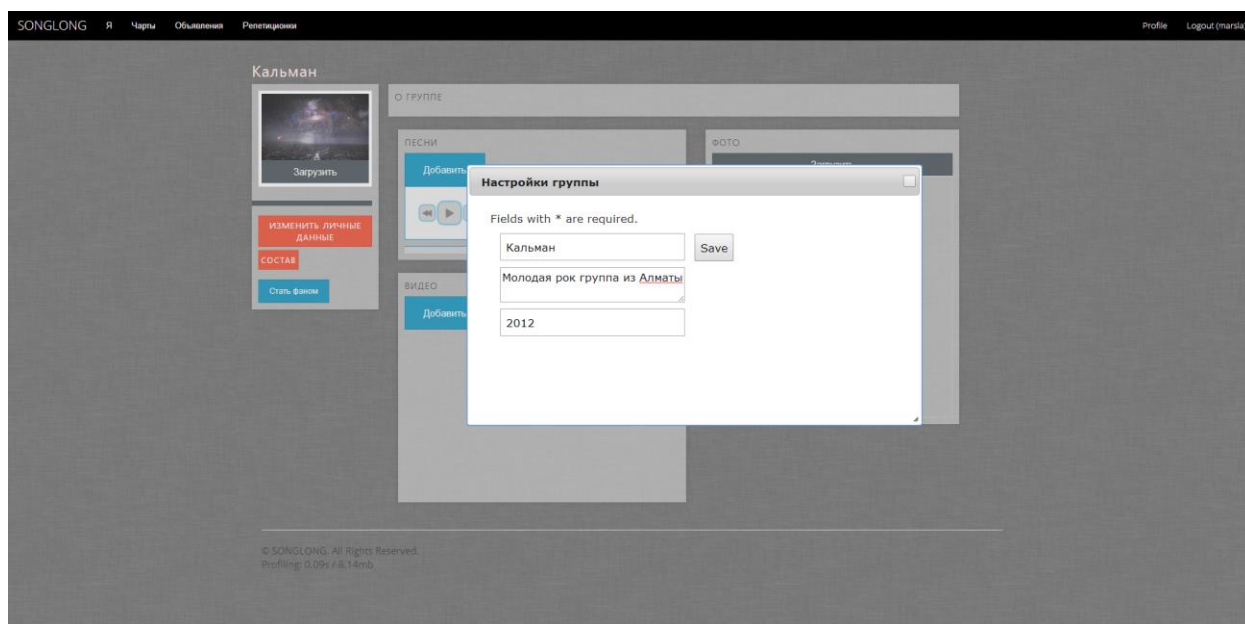


Рисунок 2.12– Форма настроек для группы

Также на странице группы можно добавлять видеоклипы, если нажать на соответствующую кнопку в блоке видеоклипов. Форма загрузки видео имеет всего одно поле, куда нужно вставить ссылку на ролик с ютуба. Встроенный в систему парсер сам сформирует код для последующей вставки видеоматериалов на сайт (Рисунок 2.13). Парсер использует специальную библиотеку, написанную на языке PHP и распространяемой по свободной лицензии.

Использование подобного приема также, в значительной мере улучшит восприятия пользователем системы, а, следовательно, он останется на сайте на большее время, загрузит больше материалов и вероятнее, что расскажет о моем портале своим друзьям.

Именно из-за таких мелочей и формируется восприятия сайта его пользователями. Чем оно лучше, тем больше вероятность удержания юзеров на портале на большее время.

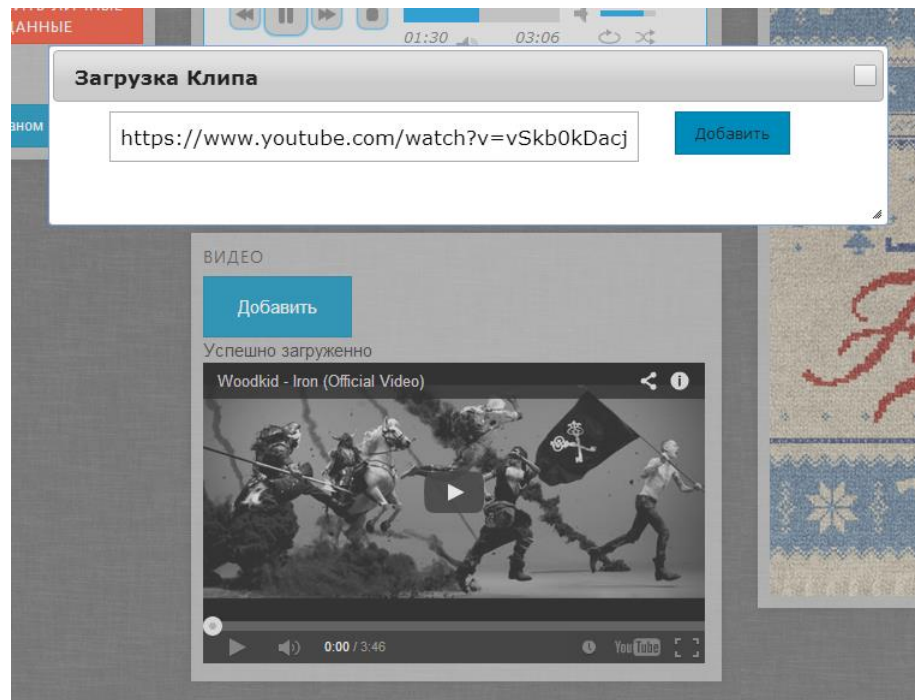


Рисунок 2.13 – Форма загрузки видеоклипа

Одной из привлекательных особенностей моего портала является справочник репетиционных баз, в котором собрана вся информация о них, включая цену, адрес и описание (Рисунок 2.14).

Все репетиционные базы отображаются на карте, если нажать на любой из маркеров, то откроется сообщение со всей необходимой информацией об этом месте. Для улучшения восприятия графической информации, близко расположенные маркеры объединяются в кластеры, нажав на которые, можно поучить информацию о каждом из заведений.

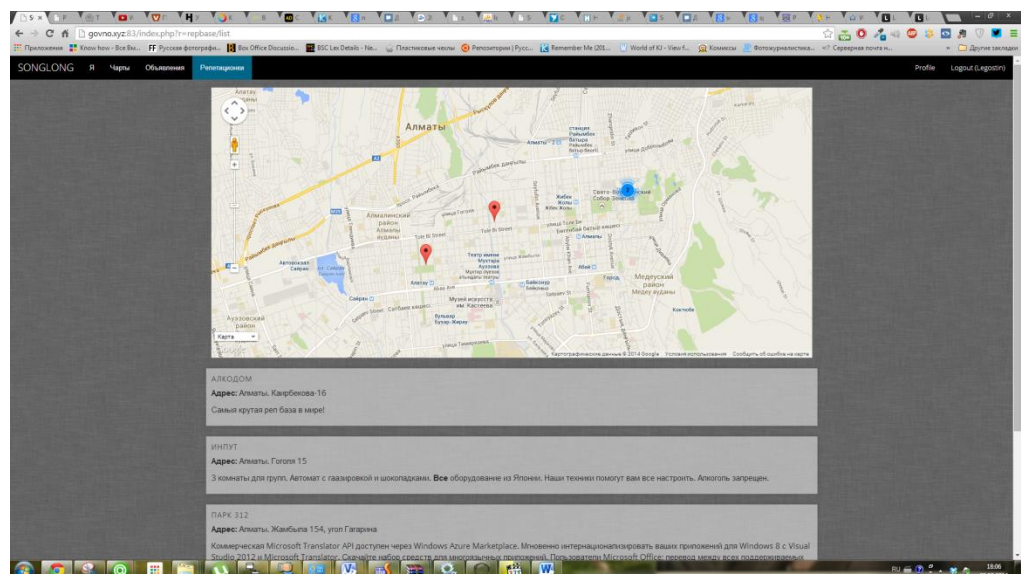


Рисунок 2.14 – Страница с каталогом репетиционных баз

Все то же самое можно сказать и про заведения, в которых поведутся различные концерты и мероприятия. В базу моего проекта заносится информация о точном месторасположении объекта, его название, описание, логотип, а также цена за вход в обычные дни. Сама страница очень похожа на страницу из предыдущего пункта, за исключением наличия здесь логотипа заведения (Рисунок 2.15).

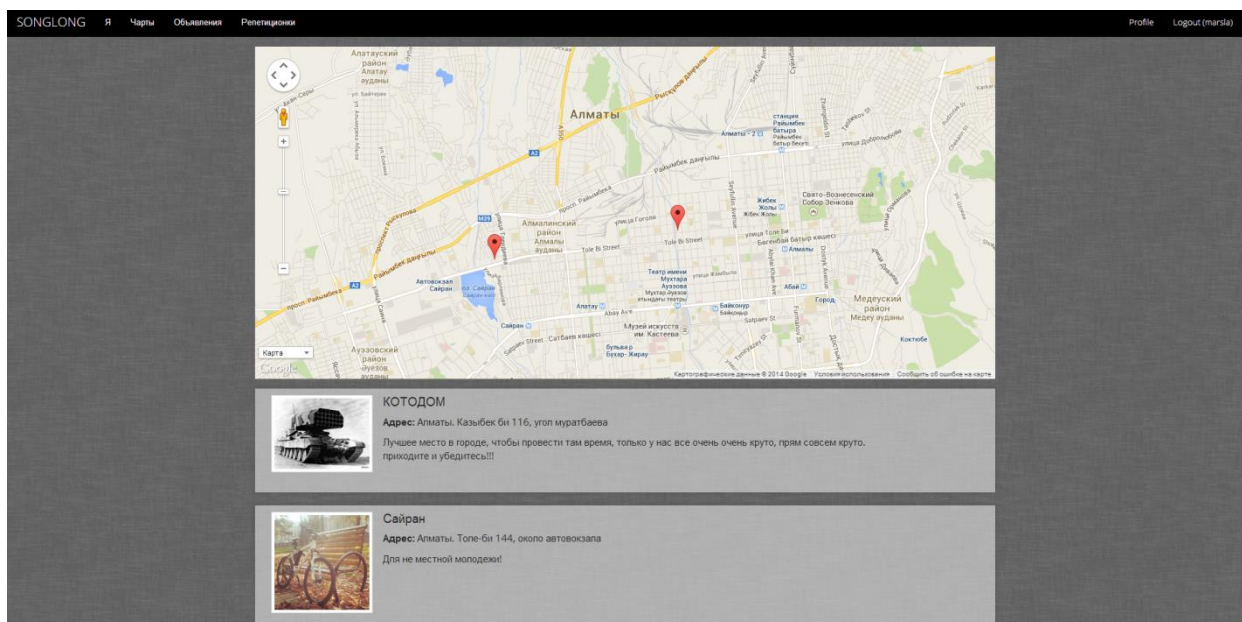


Рисунок 2.15– Страница с каталогом концертных площадок

Все фотографии на сайте можно открыть и более подробно рассмотреть. Перелистывать их можно, как нажимая мышкой на большие кнопки на экране, так и с помощью кнопок на клавиатуре (Рисунок 2.16).

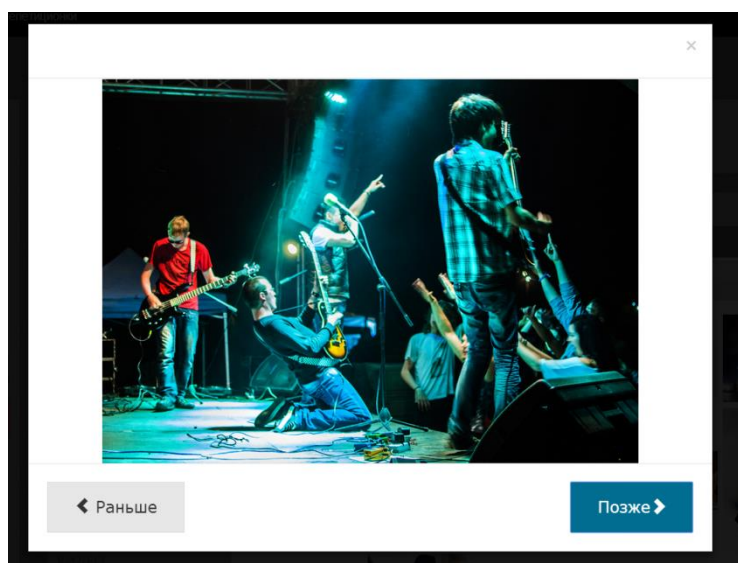
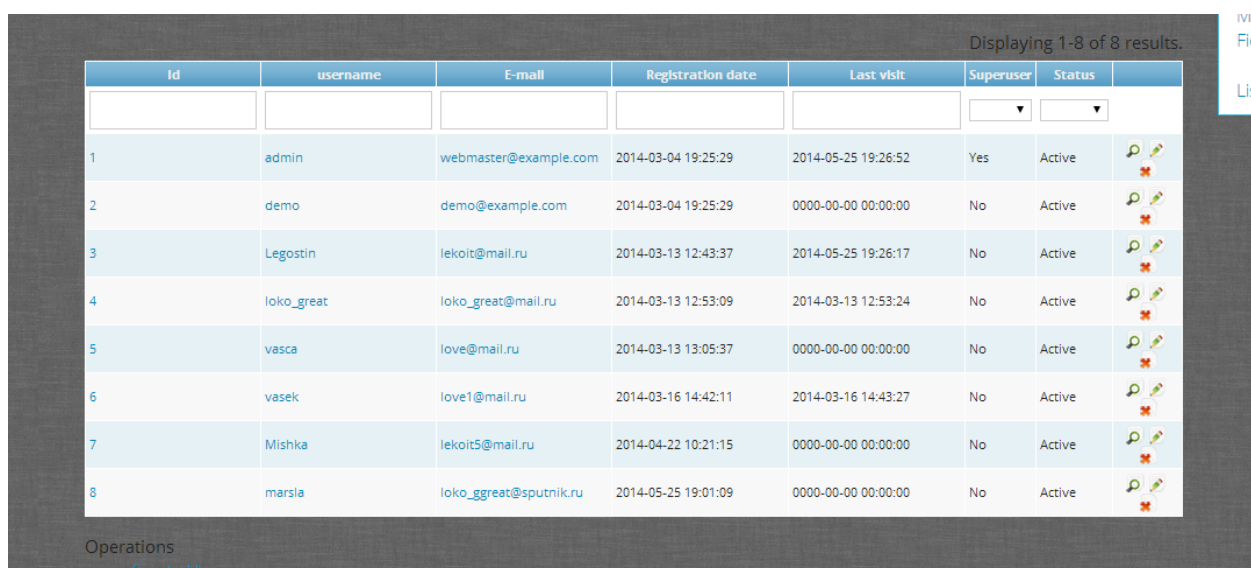

















Рисунок 2.16 – Просмотр одной из фотографий

Рассмотрим сайт с точки зрения администратора. На рисунке 2.17 представлена форма для управления всеми пользователями системы.

На форме представлены широкие возможности для поиска и фильтрации необходимых аккаунтов.



Displaying 1-8 of 8 results.

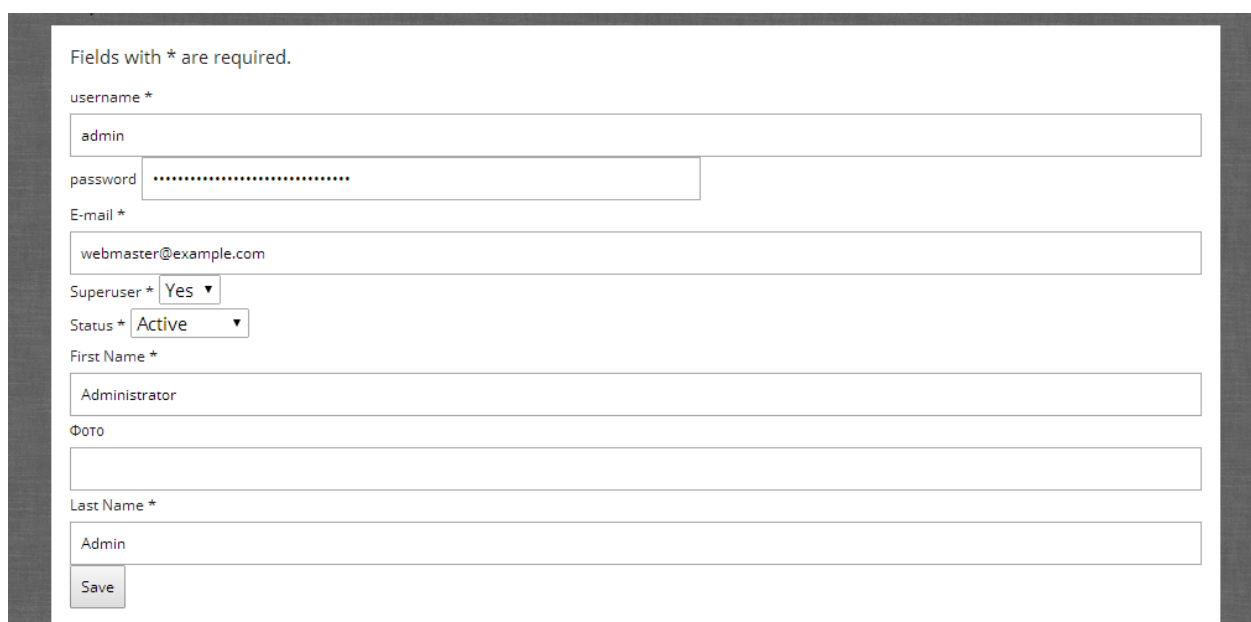
Id	username	E-mail	Registration date	Last visit	Superuser	Status	
1	admin	webmaster@example.com	2014-03-04 19:25:29	2014-05-25 19:26:52	Yes	Active	 
2	demo	demo@example.com	2014-03-04 19:25:29	0000-00-00 00:00:00	No	Active	 
3	Legostin	lekoit@mail.ru	2014-03-13 12:43:37	2014-05-25 19:26:17	No	Active	 
4	loko_great	loko_great@mail.ru	2014-03-13 12:53:09	2014-03-13 12:53:24	No	Active	 
5	vasca	love@mail.ru	2014-03-13 13:05:37	0000-00-00 00:00:00	No	Active	 
6	vasek	love1@mail.ru	2014-03-16 14:42:11	2014-03-16 14:43:27	No	Active	 
7	Mishka	lekoit5@mail.ru	2014-04-22 10:21:15	0000-00-00 00:00:00	No	Active	 
8	marsla	loko_ggreat@sputnik.ru	2014-05-25 19:01:09	0000-00-00 00:00:00	No	Active	 

Operations
Create User

Рисунок 2.17 – Панель администратора

Нажав на иконку с карандашиком администратор может изменить все параметры профиля, даже такие, какие сам пользователь не в праве менять.

Админ имеет право наделять пользователя правами суперпользователя, что в контексте данного проекта является синонимом слова администратор (Рисунок 2.18).



Fields with * are required.

username *
admin

password *
.....

E-mail *
webmaster@example.com

Superuser * Yes ▾

Status * Active ▾

First Name *
Administrator

Фото
[Empty field]

Last Name *
Admin

Save

Рисунок 2.18 – Панель настроек пользователя администратором

На рисунке 2.19 показан общий список с информацией о пользователях. Администратор использует данный список, когда ему необходимо просмотреть и сравнить информацию сразу о нескольких пользователях системы.

username	Registration date	Last visit
admin	2014-03-04 19:25:29	2014-05-25 19:26:52
demo	2014-03-04 19:25:29	0000-00-00 00:00:00
Legostin	2014-03-13 12:43:37	2014-05-25 19:26:17
loko_great	2014-03-13 12:53:09	2014-03-13 12:53:24
vasca	2014-03-13 13:05:37	0000-00-00 00:00:00
vasek	2014-03-16 14:42:11	2014-03-16 14:43:27
Mishka	2014-04-22 10:21:15	0000-00-00 00:00:00
marsla	2014-05-25 19:01:09	0000-00-00 00:00:00

Operations

- Manage Users
- Manage Profile Field

Рисунок 2.19 – Список пользователей

На рисунке 2.20 показана форма для изменения информации об одной из репетиционных баз. Как вы видите, сверху расположена большая карта, кликнув по любой точке которой, можно установить маркер, координаты которого в дальнейшем запишутся в базу, как месторасположение объекта.

SONGLONG Я Чарты Объявления Репетиционные Profile Logout (marsla)

Redaktirovaniye Input

Operations

- List Rebases
- Create Rebase
- View Rebase
- Manage Rebases

Название:

Адрес:

Цена:

Описание:

Город:

Save

Operations

- Manage Users
- Manage Profile Field

Рисунок 2.20 – Форма изменения одной из репетиционных баз

На рисунке 2.21 показана форма добавления концертной площадки. Также как и на форме репетиционных баз, в верхней ее части располагается карта, с помощью которой задаются координаты объекта. Также, на этой

странице есть кнопка для загрузки фотографии. Загрузка осуществляется асинхронным образом, с помощью модуля EAJaxUpload. Сразу после загрузки фотографии, она отображается на форме.

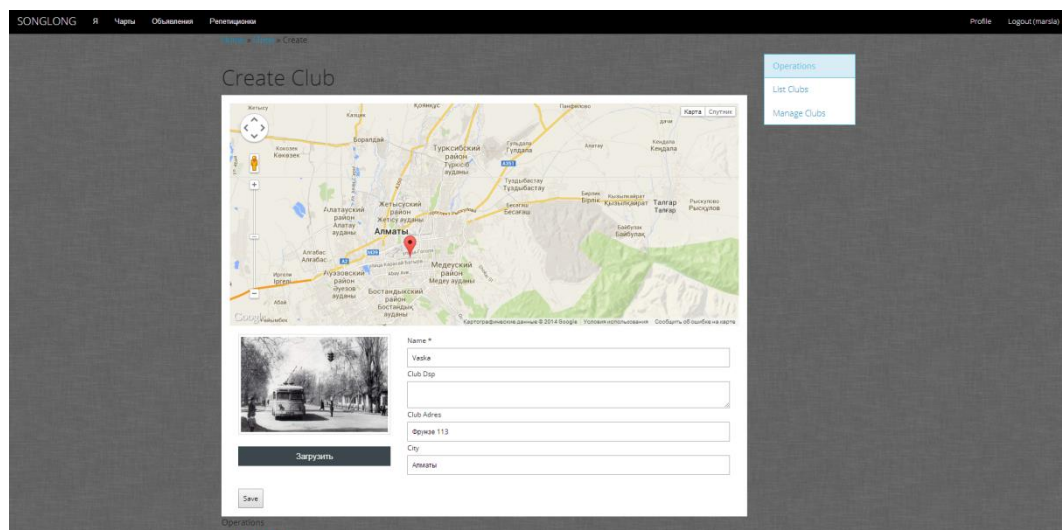


Рисунок 2.21– Форма изменения одного из клубов

На рисунке 2.22 показана форма для добавления администратором новости на сайт. Помимо основного материала, необходимо вводить анонс с фотографией большого разрешения, чтобы новость красиво смотрелась на главной странице. Публикацию записи можно отложить до любого момента, для этого нужно просто щелкнуть по полю с датой и выбрать в появившемся календаре необходимое число и месяц.

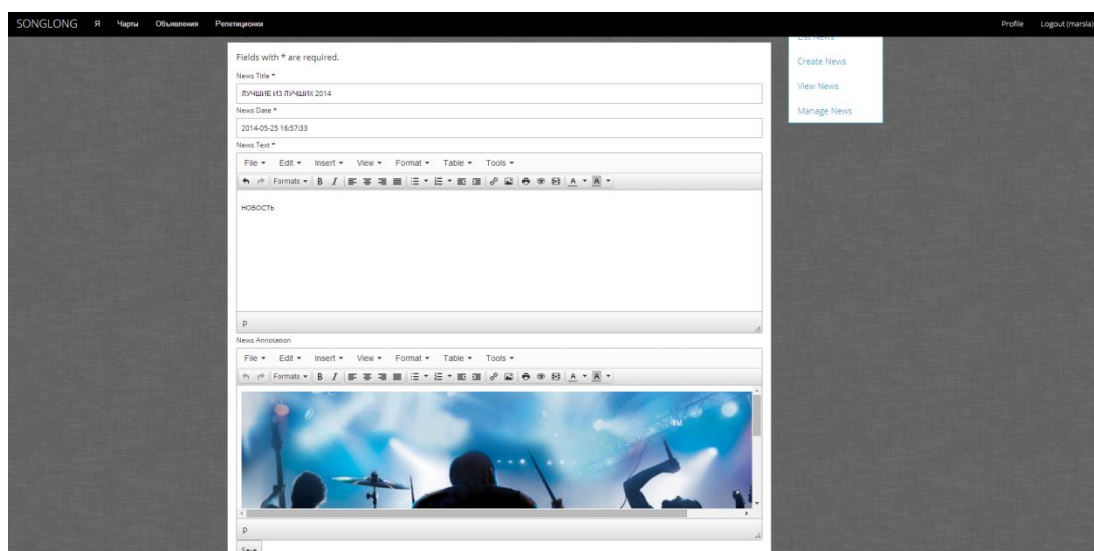


Рисунок 2.22 – Форма добавления новости

3 Технико-экономическое обоснование проекта

3.1 Описание работы и обоснование необходимости

Тема данного дипломного проекта «Разработка динамической интернет-платформы на основе технологии Web 2.0».

Цель данного проекта – разработать web-портал для музыкантов, позволяющий им выкладывать в интернет своё творчество, создавать интернет-сообщества собственных музыкальных групп, участвовать в общем рейтинге музыкантов, чтобы привлечь большую аудиторию слушателей, а так же внимание продюсеров и инвесторов.

Так как в пространстве пост-советских стран существует множество музыкальных групп различного направления, но нет единого интернет-портала и единой базы музыкальных коллективов, было решено сделать данный продукт.

В данном разделе приводится рассмотрение экономической составляющей реализации данного продукта, отражающее временные, трудовые и финансовые затраты на проект.

3.2 Трудовые ресурсы, используемые в работе

В данном дипломном проекте используется интеллектуальный труд, стоимость затрат которого выше, чем физического труда. В работе задействованы:

- руководитель – постановка задачи, разработка схемы сайта;
- инженер-разработчик – разработка алгоритмов и программирование, сопровождение и инструктаж персонала сопровождения;
- консультант по экономической части и по части ОТиБЖД.

Общее количество сотрудников и их заработная плата представлены в таблице 3.1.

Т а б л и ц а 3 . 1 – Количество задействованных в проекте работников, и их заработная плата

Исполнитель	Количество, человек	Месячная заработная плата, тенге
Руководитель	1	80000
Консультант по части “Экономика”	1	65000
Консультант по части “Безопасность жизнедеятельности”	1	60000
Инженер-разработчик	1	100000
Итого	4	305000

3.3 Оборудование, используемое в работе

Оборудование, использованное при разработке сайта, представлено в таблице 3.2.

Т а б л и ц а 3.2 – Необходимое оборудование для разработки ПП

Наименование	Модель	Стоимость, тенге
Ноутбук	Lenovo Z570	110000
Компьютерная мышь	Genius SuperPower X5	2500
Модем	Zyxel 3000	5 000
Итого		117500

3.4 Сроки реализации проекта

В таблице 3.3 представлены сроки реализации данного проекта.

Т а б л и ц а 3.3 – Сроки реализации проекта

Перечень работ		Ожидаемая длительность	Обозначение
1 этап	Постановка задачи	4	$T_{ПЗ}$
	Описание задачи, выбор технологий	3	$T_{ОЗ}$
	Разработка алгоритма	6	T_A
2 этап	Создание интерфейса и дизайна сайта	2	T_C
3 этап	Создание базы данных, настройка связей	9	$T_{БД}$
4 этап	Разработка основной функциональной части сайта	6	$T_{ОСН}$
5 этап	Тестирование ПО	2	$T_{ТЕСТ}$
	Отладка ПО	6	$T_{ОТ}$
	Внедрение ПО	2	$T_{ВН}$
6 этап	Подготовка раздела «Экономика»	7	$T_{Э}$
	Подготовка раздела «БЖД»	7	$T_{БЖ}$

3.5 Расчет стоимости работы по проектированию и разработке

Разработка многомодульного программного продукта – сложный и трудоемкий процесс, требующий наряду с интеллектуальными, техническими

затратами и финансовых затрат. Поэтому необходимым является произведение расчета стоимости этой разработки. Затраты на разработку данного программного комплекса определяется по формуле

$$C = \Phi OT + C_H + A + \text{Э} + C_{np} + H \quad (3.1)$$

где ΦOT – фонд оплаты труда;

C_H – социальный налог;

A – амортизационные отчисления;

Э – затраты на электроэнергию;

C_{np} – прочие расходы;

H – накладные расходы.

Необходимый фонд оплаты труда рассчитывается по формуле

$$\Phi OT = Z_{осн} + Z_{доп} \quad (3.2)$$

где: $Z_{осн}$ – основная заработная плата;

$Z_{доп}$ – дополнительная заработная плата.

Труд сотрудников университета, задействованных в работе, оплачивается согласно положению о заработной плате АУЭС, труд программиста-разработчика принят условно, на договорной основе в размере 100000 тенге.

Базовый показатель для определения составляющих затрат труда вычисляется по формуле

$$Q = q * c \quad (3.3)$$

где Q – условное число команд,

$q=5100$ – коэффициент, учитывающий условное число команд в зависимости от типа задачи,

$c=1,19$ – коэффициент, учитывающий новизну и сложность программы.

$$Q = 5100 * 1,19 = 6069$$

Выбрать значение коэффициента q можно из таблицы 3.4.

Т а б л и ц а 3 . 4 – Значения коэффициента q

Тип задачи	Пределы изменений коэффициента
Задачи учета	От 1400 до 1500
Задачи оперативного управления	От 1500 до 1700
Задачи планирования	от 3000 до 3500
Многовариантные задачи	от 4500 до 5000
Комплексные задачи	от 5000 до 5500

Программные продукты по степени новизны могут быть отнесены к одной из 4-х групп:

- группа А – разработка принципиально новых задач;
- группа Б – разработка оригинальных программ;
- группа В – разработка программ с использованием типовых решений;
- группа Г – разовая типовая задача.

Коэффициент c определяется из таблицы 3.5, на пересечении групп сложности и степени новизны.

Т а б л и ц а 3 . 5 – Коэффициенты расчета трудоемкости

Язык программирования	Группа сложности	Степень новизны				Коэффициент B
		А	Б	В	Г	
Высокого уровня	1	1,38	1,26	1,15	0,69	1,2
	2	1,30	1,19	1,08	0,65	1,35
	3	1,20	1,10	1,00	0,60	1,5
Низкого уровня	1	1,58	1,45	1,32	0,79	1,2
	2	1,49	1,37	1,24	0,74	1,35
	3	1,38	1,26	1,15	0,69	1,5

Далее необходимо рассчитать время на создание программного продукта. Так как участники, задействованные в проекте, работают в различные промежутки времени, в течение которого реализуется проект, необходимо произвести расчет дневной и почасовой оплаты труда.

Заработная плата каждого работника за один рабочий день рассчитывается по формуле

$$D = \frac{O}{n} \quad (3.4)$$

где O – оклад работника в тенге;

n – количество дней в рабочем месяце (это 24 дня – шестидневная рабочая неделя).

Для руководителя

$$D = \frac{80000}{24} = 3333 \text{ тенге/день}$$

Для консультанта по части “Экономика”

$$D = \frac{65000}{24} = 2708 \text{ тенге/день}$$

Для консультанта по части “БЖД”

$$D = \frac{60000}{24} = 2500 \text{ тенге/день}$$

Для инженера разработчика

$$D = \frac{100000}{24} = 4167 \text{ тенге/день}$$

Заработная плата за один час рассчитывается по формуле

$$H = \frac{D}{z} \quad (3.5)$$

где D – заработная плата работника за один рабочий день,
 z – количество часов рабочего дня (8 часов).

Для руководителя

$$H = \frac{3333}{8} = 416,6 \text{ тенге/час}$$

Для консультанта по части “Экономика”

$$H = \frac{2708}{8} = 338,5 \text{ тенге/час}$$

Для консультанта по части “БЖД”

$$H = \frac{2500}{8} = 312,5 \text{ тенге/час}$$

Для инженера разработчика

$$H = \frac{4167}{8} = 520,8 \text{ тенге/час}$$

Время рассчитывается в человеко-часах, причем $T_{ПЗ}$ берется по фактически отработанному времени, а время остальных этапов определяется по расчётам, по условному числу команд Q [15].

Определяем время, затраченное на каждый этап создания программного продукта:

1 $T_{ПЗ}$ (время на подготовку описания задачи), берется по факту и составляет (от 3-х до 5-ти дней по 8 часов)

$$T_{ПЗ} = 32 \text{ чел/час};$$

2 T_{oz} (время на описание задачи) определяется по формуле

$$T_{oz} = \frac{Q * B}{50 * K} \quad (3.6)$$

где $B=1,35$ – коэффициент учета изменений задачи, коэффициент B в зависимости от сложности задачи и числа изменений выбирается в интервале от 1,2 до 1,5 (Таблица 3.6)

$K=1$ – коэффициент, учитывающий квалификацию программиста.

$$T_{oz} = \frac{6069 * 1,35}{50 * 1} = 163,8 \text{ чел/час};$$

Т а б л и ц а 3.6 – Коэффициенты квалификации программиста

Опыт работы	Коэффициент квалификации
До двух лет	0.8
2-3 года	1
3-5 лет	1.1 – 1.2
5-7 лет	1.3 – 1.4
более 7 лет	1.5 – 1.6

3 T_A (время на разработку алгоритма) рассчитываем по формуле

$$T_A = \frac{Q}{50 * K} \quad (3.7)$$

$$T_A = \frac{6069}{50 * 1} = 121,4 \text{ чел/час};$$

4 $T_{лс}$ (время на создание интерфейса сайта) определяется аналогично T_A

$$T_C = 121,4 \text{ чел/час};$$

5 $T_{бд}$ (время на проектирование и создание БД) определяется по формуле

$$T_{БД} = \frac{3500 * 1,35}{50 * 1} = 94,5 \text{ чел/час};$$

6 T_M (время на разработку основной функциональности) определяется по формуле

$$T_{ОСН} = \frac{3500 * 1,35}{50 * 1} = 94,5 \text{ чел/час};$$

7 $T_{ИН}$ (время на отладку программы) определяется по формуле

$$T_{ИН} = \frac{6069 \cdot 1,35}{50 \cdot 1} = 163,7 \text{ чел/час};$$

8 $T_{ОСН}$ (время на внедрение), определяется по формуле

$$T_{ВН} = \frac{4500 \cdot 1,35}{50 \cdot 1} = 121,5 \text{ чел/час};$$

9 $T_{ТЕСТ}$ (время на тестирование), определяется по формуле

$$T_{ТЕСТ} = \frac{3000 \cdot 1,35}{50 \cdot 1} = 81 \text{ чел/час};$$

10 $T_{ТЕХ}$ (время на составление технической документации), берется по факту и составляет (от 3-х до 5-ти дней по 8 часов)

$$T_{ТЕХ} = 16 \text{ чел/час};$$

11 $T_{Э}, T_{БЖ}$ (время на подготовку разделов Экономика и БЖ), берется по факту и составляет (от 7-и до 10-ти дней по 8 часов)

$$T_{Э} = T_{БЖ} = 56 \text{ чел/час};$$

12 Т.к. в проекте задействованы руководители, заработная плата которых отличается, то необходимо рассчитать время $T_{РУК}$. Время берется среднее и оно равно

$$T_{РУК} = 60 \text{ чел/час},$$

$$T_{Э} = T_{БЖ} = 20 \text{ чел/час}.$$

Суммарные затраты труда рассчитываются как сумма составных затрат труда по формуле

$$T_{СТ} = T_{ПЗ} + T_{ОЗ} + T_{А} + T_{ЛС} + T_{БД} + T_{М} + T_{ИН} + T_{ОСН} + T_{ТЕСТ} + T_{ТЕХ} + T_{Э} + T_{БЖ} \quad (3.8)$$

$$T_{СТ} = 32 + 163,8 + 121,4 + 121,4 + 94,5 + 94,5 + 163,7 + 121,5 + 81 + 16 + 56 + 56 = \\ = 1105,8 \text{ чел/час}$$

Таким образом, суммарная основная заработная плата составит

$$Z_{ОСН} = Z_{ИНДЖ} + Z_{РУК} \quad (3.9)$$

$$Z_{ОСН} = 1105,8 * 520,8 + 20 * 312,5 + 20 * 338,5 + 60 * 416,6 = 613695 \text{ тенге}$$

Дополнительная заработная плата в среднем определяется в размере 10% от основной заработной платы и рассчитывается по формуле

$$Z_{ДОП} = Z_{ОСН} * 10\% \quad (3.10)$$

$$Z_{ДОП} = 613695,5 * 10\% = 61370 \text{ тенге}$$

Общий фонд оплаты труда согласно формуле 3.2

$$\PhiОТ = 613695 + 61370 = 675065 \text{ тенге}$$

Социальный налог составляет 11% от ФОТ и рассчитывается по формуле

$$C_H = (\PhiОТ - ПО) * 11\% \quad (3.11)$$

где ПО (пенсионные отчисления) составляют 10% от ФОТ и рассчитываются по формуле

$$ПО = \PhiОТ * 10\% \quad (3.12)$$

Размер пенсионных отчислений согласно формуле 3.12 составляет

$$ПО = 675065 * 10\% = 67506 \text{ тенге}$$

Отчисления по социальному налогу согласно формуле 3.11

$$C_H = (675065 - 67506) * 11\% = 66831,5 \text{ тенге}$$

3.6 Расчет затрат на амортизацию

Амортизационные отчисления производятся по установленным нормам амортизации, выражаются, в процентах к балансовой стоимости оборудования и рассчитываются по формуле

$$A = \frac{C_{обор} * H_A * N}{100 * 12 * t} \quad (3.13)$$

где H_A — норма амортизации;

$C_{обор}$ — первоначальная стоимость оборудования;

N — количество дней на выполнение работ;
 t — общее время использования персонального компьютера.
 Норма амортизации H_A , рассчитывается по формуле

$$H_A = \frac{C_{ОБОР} - C_{ЛИКВ}}{T_{НОРМ} * C_{ОБОР}} * 100\% \quad (3.14)$$

где $C_{ЛИКВ}$ — ликвидационная стоимость, составляет 5.61% от стоимости оборудования;

$T_{НОРМ}$ — нормативный срок службы (для персонального компьютера — 4 года).

Так как ликвидационная стоимость составляет 5.61% от оборачиваемой стоимости, следовательно

$$C_{ЛИКВ} = 0,056 * 117500 = 6580 \text{ тенге}$$

Общее время использования персонального компьютера учитывает лишь время работы на компьютере и рассчитывается по формуле

$$t = T_{ЛС} + T_{БД} + T_M + T_{ИН} + T_{ОСН} + T_{ТЕСТ} \quad (3.15)$$

$$t = 121,4 + 94,5 + 94,5 + 163,7 + 121,5 + 81 = 676,6 \text{ чел/час}$$

$$H_A = \frac{117500 - 6580}{3 * 117500} = 31,3 \%$$

$$A = \frac{117500 * 31,3 * 1105,8}{100 * 12 * 676,6} = 5009 \text{ тенге}$$

Затраты на электроэнергию вычисляется по формуле

$$\mathcal{E} = M * k_z * T * C_{кВт\cdotч} \quad (3.16)$$

где M — мощность ЭВМ (600 Вт=0,6 кВт);

k_z — коэффициент загрузки (0.8);

$C_{кВт\cdotч}$ — 14,935 тг — стоимость 1 кВт-час электроэнергии;

T — время работы.

$$\mathcal{E} = 0,6 * 0,8 * 14,935 * 676,6 = 4850 \text{ тенге}$$

Расходы на материалы и комплектующие, используемые в процессе написания программного продукта ($C_{МуК}$), а также затраты на техническое обслуживание и ремонт ($C_{ТО}$) составляют, соответственно, 2,06% и 2,266% от стоимости оборудования — формулы

$$C_{мик} = 0,0206 * C_{обор}, \quad (3.17)$$

$$C_{то} = 0,02266 * C_{обор} \quad (3.18)$$

$$C_{Мик} = 0,0206 * 117500 = 2420,5 \text{ тенге,}$$

$$C_{ТО} = 0,02266 * 117500 = 2662,5 \text{ тенге}$$

$$C_{пр} = C_{мик} * C_{то} \quad (3.19)$$

$$C_{ПР} = 2420,5 + 2662,5 = 5083 \text{ тенге}$$

Накладные расходы, связанные с управлением и обслуживанием, содержанием и эксплуатацией оборудования и прочими дополнительными затратами на обеспечение процессов производства и обращения, составляют 50% от всех затрат, вычисляются по формуле

$$H = (\Phi OT + C_H + A + Э + C_{пр}) * 0,5 \quad (3.20)$$

$$H = (675065 + 66831,5 + 5009 + 4850 + 5083) * 0,5 = 378420 \text{ тенге}$$

Таким образом, затраты на разработку данного программного комплекса согласно формуле 3.1 равны

$$C = 675065 + 66831,5 + 5009 + 4850 + 5083 + 378420 = 1135257,5 \text{ тенге}$$

Сводные результаты расчета себестоимости программного продукта предоставлены в виде таблицы 3.7, с указанием статьи расходов, суммы и их доли в общей стоимости разработки.

Т а б л и ц а 3.7 – Результирующая себестоимость разработки динамического интернет-портала на основе технологии Web 2.0 для музыкантов

Статья расходов	Сумма, тенге	В процентах от общей суммы, %
ФОТ	675065	60
C _н	66831,5	5,8
A	5009	0,4
H	378420	33
Итого:	1135257,5	100

На рисунке 3.1 представлена диаграмма структуры себестоимости программного продукта, разрабатываемого в данной дипломной работе.

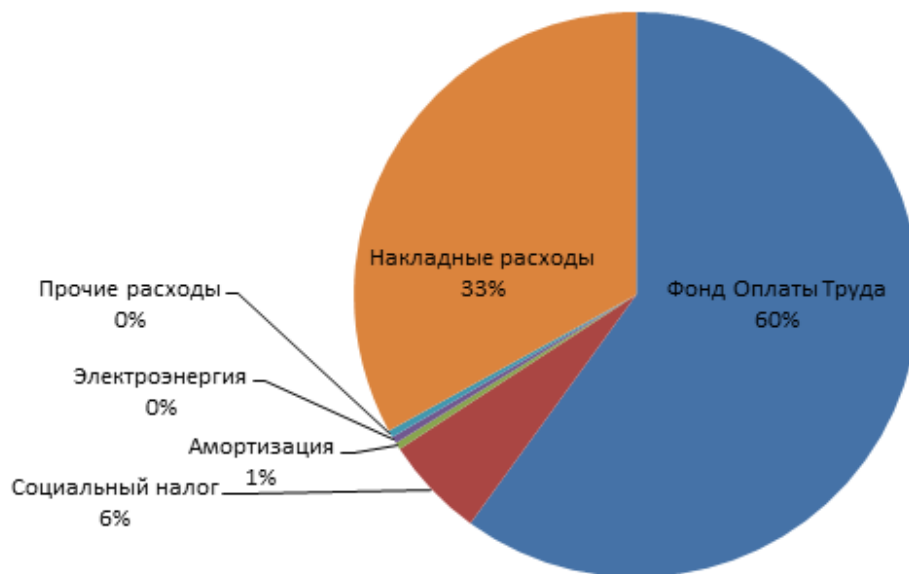


Рисунок 3.1 — Структура себестоимости программного обеспечения

3.7 Цена программного продукта

Минимальная цена программного продукта рассчитывается по следующей формуле

$$C_{min} = (C_{пп} + C_{мд} + C_з)(1 + P/100) \quad (3.21)$$

где $C_{пп}$ – себестоимость разработки программного продукта;

$C_{мд}$ – цена магнитного диска;

$C_з$ – стоимость записи;

P – планируемый процент рентабельности по отношению к себестоимости.

Тогда по формуле 3.21

$$C_{min} = (1135257,5 + 200 + 155) * (1 + 0,1885) = 1349675 \text{ тенге}$$

Далее определяется цена реализации с учётом налога на добавленную стоимость (НДС) по формуле

$$C_p = C_{min} + \text{НДС} \quad (3.22)$$

$$C_p = 1349675 + 0,12 * 1349675 = 1511636 \text{ тенге}$$

4 Безопасность жизнедеятельности

4.1 Анализ потенциально опасных и вредных производственных факторов проектируемого объекта, воздействующих на персонал

Работающий за компьютером человек длительное время подвергается опасным воздействиям.

Так как человек должен сохранять относительно неподвижное положение, это негативно сказывается на позвоночнике и циркуляции крови во всем организме (застой крови). Особенно сильно застой крови выражен на уровне органов малого таза и конечностей. При длительных нарушениях циркуляции крови нарушается питание тканей и повреждаются стенки сосудов, что в свою очередь приводит к их необратимому расширению. Такое расширение сосудов наблюдается, например, при геморрое.

Чтение информации с монитора вызывает перенапряжение глаз. Возникает это главным образом потому, что во время чтения с монитора расстояние от текста до глаз постоянно остается одним и тем же, из-за этого мышцы глаз, регулирующие аккомодацию, находятся в постоянном напряжении. Со временем это может привести к нарушению аккомодативной способности глаз и, следовательно, к нарушениям зрения.

Длительная работа на клавиатуре приводит к перенапряжению суставов кисти и мышц предплечья.

Работа за компьютером предполагает переработку большого массива информации и постоянную концентрацию внимания, поэтому при длительной работе за компьютером нередко развивается умственная усталость и нарушение внимания.

Работа за компьютером нередко поглощает все внимание работающего человека и потому, такие люди часто пренебрегают нормальным питанием и работают впроголодь весь день. Неправильное питание приводит не только к нарушениям работы органов пищеварительного тракта, но и к возникновению минеральной и витаминной недостаточности.

Гиподинамия, стресс, вредные привычки и неправильное питание являются основными причинами сердечно-сосудистых заболеваний и диабета. Таким образом, человек длительное время работающий за компьютером подвергается реальному риску данных заболеваний, различных заболеваний глаз, двигательного аппарата, органов желудочно-кишечного тракта, психических расстройств.

Длительная работа за компьютером может стать причиной серьезных нервно-мышечных расстройств. Особенно чувствительными участками тела являются пальцы, кисти рук и предплечья. При длительной работе за компьютером (на клавиатуре) нервные окончания пальцев подвергаются постоянному раздражению. Со временем это приводит к истощению нервных путей осуществляющих связь пальцев с корой головного мозга. В результате возникают нарушения координации движений пальцев и судороги кисти и предплечья.

Часто длительная работа за компьютером может стать причиной нарушений осанки или искривления позвоночника.

Работа за компьютером может быть причиной головных болей. Известно несколько типов головных болей, которые могут быть спровоцированы работой за компьютером. Одним из факторов провоцирующим появление головных болей является хроническое перенапряжение, важное значение имеет и постоянное напряжение черепных мышц и мышц лица.

Расстройства внимания и невозможность концентрироваться являются следствием хронического переутомления. Иногда из-за длительной работы за компьютером может возникнуть шум в ушах, головокружение, тошнота. При возникновении этих симптомов нужно обратиться за советом к врачу и временно прервать работу за компьютером.

Помимо описанных выше заболеваний длительное пребывание за компьютером может быть причиной возникновения гастритов, язвы желудка, простатита.

4.2 Микроклимат рабочей зоны программиста

Параметры микроклимата могут меняться в широких пределах, в то время как необходимым условием жизнедеятельности человека является поддержание постоянства температуры тела благодаря терморегуляции, т.е. способности организма регулировать отдачу тепла в окружающую среду. Принцип нормирования микроклимата – создание оптимальных условий для теплообмена тела человека с окружающей средой.

Вычислительная техника является источником существенных тепловыделений, что может привести к повышению температуры и снижению относительной влажности в помещении. В помещениях, где установлены компьютеры, должны соблюдаться определенные параметры микроклимата. В санитарных нормах СН-245-71 [16] установлены величины параметров микроклимата, создающие комфортные условия. Эти нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера производственного помещения, указанные в таблице 4.1.

Объем помещений, в которых размещены работники вычислительных центров, не должен быть меньше $19,5\text{м}^3/\text{человека}$ с учетом максимального числа одновременно работающих в смену. Нормы подачи свежего воздуха в помещения, где расположены компьютеры, приведены в таблице 4.2.

Т а б л и ц а 4 . 1 – Параметры микроклимата для помещений, где установлены компьютеры

Период	Параметр микроклимата	Величина
--------	-----------------------	----------

года		
Холодный	Температура воздуха в помещении	22...24°C
	Относительная влажность	40...60%
	Скорость движения воздуха	до 0,1м/с
Теплый	Температура воздуха в помещении	23...25°C
	Относительная влажность	40...60%
	Скорость движения воздуха	0,1...0,2м/с

Т а б л и ц а 4 . 2 – Нормы подачи свежего воздуха в помещения, где расположены компьютеры

Характеристика помещения	Объемный расход подаваемого в помещение свежего воздуха, м ³ /на одного человека в час
Объем до 20м ³ на человека	Не менее 30
20...40м ³ на человека	Не менее 20
Более 40м ³ на человека	Естественная вентиляция

Окраска помещений и мебели должна способствовать созданию благоприятных условий для зрительного восприятия, хорошего настроения. Источники света, такие как светильники и окна, которые дают отражение от поверхности экрана, значительно ухудшают точность знаков и влекут за собой помехи физиологического характера, которые могут выразиться в значительном напряжении, особенно при продолжительной работе. Отражение, включая отражения от вторичных источников света, должно быть сведено к минимуму. Для защиты от избыточной яркости окон могут быть применены шторы и экраны.

В зависимости от ориентации окон рекомендуется различная окраска стен и пола.

Окна ориентированы на юг: стены зеленовато-голубого или светло-голубого цвета; пол – зеленый;

Окна ориентированы на север: стены светло-оранжевого или оранжево-желтого цвета; пол – красновато-оранжевый;

Окна ориентированы на восток: стены желто-зеленого цвета; пол зеленый или красновато-оранжевый;

Окна ориентированы на запад: стены желто-зеленого или голубовато-зеленого цвета; пол зеленый или красновато-оранжевый.

В помещениях, где находится компьютер, необходимо обеспечить следующие величины коэффициента отражения: для потолка: 60...70%, для стен: 40...50%, для пола: около 30%. Для других поверхностей и рабочей мебели: 30...40%.

4.3 Расчет кондиционирования в офисе

Задача кондиционирования воздуха состоит в поддержании таких параметров воздушной среды, при которых каждый человек благодаря своей индивидуальной системе автоматической терморегуляции организма чувствовал бы себя комфортно, т.е. не замечал влияния этой среды.

Исходные данные приводятся в таблице 4.3.

Т а б л и ц а 4 . 3 – Исходные данные

Название параметра		Значение
Город		Алматы
Параметры помещения	Длина	9 м
	Ширина	5 м
	Высота	3,5 м
Данные по оборудованию	Мощность	NOС=40 Вт/м ²
	Вид источника	лампы люминесцентные
	Количество штук	8
	КПД	0,95
Данные по источнику света	Мощность	РОБ=0,5 кВт/ч
Окна	Количество	1
	Площадь окна	3 м ²
	Расположение	Ю
	Вид	жалюзи, металл. загрязнение незначительное
Расчетное время		11-12 ч
Температура в помещении	Летом	26 °С
	Зимой	20 °С
Вид положения работы		сидя

Расчет

Рассчитаем расход воздуха (L , [м³/час]) по формуле

$$L = \frac{Q_{изв} \cdot 860}{C \cdot \Delta t \cdot \gamma} \quad (4.1)$$

где $C=0,24$ ккал/(кг х °С) – теплоемкость воздуха,
 $\gamma=1,206$ кг/м³ – удельная масса приточного воздуха.

$$Q_{изв} = Q_{ОВ} + Q_{\pi} + Q_{ОВ} + Q_{РАД} - Q_{ОГР} \quad (4.2)$$

Количество тепла от оборудования рассчитывается по формуле

$$Q_{ОВ} = P_{ОВ} \cdot \mu \cdot n_{ОВ} \quad (4.3)$$

$$Q_{ОВ} = 9 \cdot 0,5 \cdot 0,95 = 4,5, \text{ кВт}$$

Количество тепла от осветительных устройств рассчитывается по формуле

$$Q_{ОВ} = \mu \cdot N_{ОУ} \cdot F \quad (4.4)$$

где F – площадь помещения (м^2).

Количество тепла от людей

$$Q_{\pi} = q_{\text{я}} \cdot n \quad (4.5)$$

где $q_{\text{я}}$ – тепловыделение явное (Вт).

$q_{\text{я}}$ женщин составляет 85% $q_{\text{я}}$ мужчин.

$$Q_{\pi\text{л}} = 3 \cdot 63 + 2 \cdot 63 \cdot 0,85 = 0,47, \text{ кВт},$$

$$Q_{\pi\text{з}} = 3 \cdot 92 + 2 \cdot 92 \cdot 0,85 = 0,56, \text{ кВт}$$

$Q_{\text{РАД}}$ – количество тепла от солнечного излучения

$$Q_{\text{РАД}} = m \cdot F_0 \cdot (q^I + q^{II}) \cdot \beta_{\text{с.з.}} \cdot k_1 \cdot k_2, \quad (4.6)$$

где m – количество окон;

F_0 – площадь окна;

q^I, q^{II} – тепловые потоки от прямой и рассеянной радиации ($\text{Вт}/\text{м}^2$);

$\beta_{\text{с.з.}}$ – коэффициент теплопропускания;

k_1 – коэффициент затемнения остекления;

k_2 – коэффициент загрязнения стекла.

$$Q_{\text{РАД}} = 1 \cdot (214 + 79 + 73 + 77) \cdot 0,15 \cdot 0,95 \cdot 1,28 = 0,73, \text{ кВт}$$

$Q_{\text{ОГР}}$ – теплопоступления и теплопотери в результате разности температур.

$$Q_{\text{ОГР ЛЕТО}} = 0, \text{ кВт}$$

$$Q_{\text{ОГР ЗИМА}} = X_0 \cdot V_{\text{ПОМ}} \cdot (t_{\text{Нрасч}} - t_{\text{Врасч}}) \quad (4.7)$$

где $V_{\text{ПОМ}}$ – объем помещения (м^3);

$X_0=0.42$ Вт/(м³ × °С) – удельная тепловая характеристика;
 $t_{Нрасч}, t_{Врасч}$ – выбираем из приложения 8 по параметру Б для города Алматы

$$Q_{ОГР\ ЗИМА} = 0,42 \cdot 128 \cdot (31,2 - 25) = 0,33, \text{ кВт}$$

Составим тепловой баланс помещения

Лето

$$Q_{ИЗВ} = 4,75 + 0,41 + 1,47 + 0,73 - 0 = 7,36, \text{ кВт}$$

Зима

$$Q_{ИЗВ} = 4,75 + 0,6 + 1,46 + 0,73 - 0,33 = 7,21, \text{ кВт}$$

Так как тепловой баланс для лета больше зимнего теплового баланса, то рассчитаем теплонапряженность воздуха по формуле

$$Q_H = \frac{Q_{ИЗВ\ ЛЕТО} \cdot 860}{V_{ПОМ}} \quad (4.8)$$

$$Q_H = \frac{7,36 \cdot 860}{128} = 49,5 \text{ ккал/м}^3.$$

При $Q_H > 20$ ккал/м³, $\Delta t = 8$ °С,
 при $Q_H < 20$ ккал/м³, $\Delta t = 6$ °С.

Определение количества воздуха, необходимое для поступления в помещение по формуле

$$L = \frac{Q_{ИЗВ} \cdot 860}{C \cdot \Delta t \cdot \gamma} \quad (4.9)$$

$$L = \frac{7,36 \cdot 860}{0,24 \cdot 8 \cdot 1,206} = 2733 \text{ м}^3/\text{час}$$

Определение кратности воздухообмена

$$n = \frac{L}{V_{ПОМ}} \quad (4.10)$$

$$n = \frac{2733}{128} = 21,35 \text{ час}$$

По таблице 4.4 подбираем соответствующую модель кондиционера.

Это необходимо для того чтобы, помещение обеспечивалось свежим воздухом в достаточной мере.

Т а б л и ц а 4 . 4 – Подбор модели кондиционера

Модель с верхней подачей SUA			0501
Электропитание		В/Ф/Гц	400/3/50+N
Расход воздуха	Макс.	м ³ /ч	4720
	Мин.	м ³ /ч	2950
Максимальная холодопроизводительность	(24 °С, 50%)	кВт	16,7
	(26 °С, 50%)	кВт	17,7
Мощность компрессора		кВт	4,5
Электронагреватель		кВт	6,6
Увлажнитель		кВт	2,3
Расход пара		кг/ч	3,0
Выносной стандартный конденсатор малошумящий		CAL	0511
		CAL	0661
Размеры	Высота	мм	1740
	Ширина	мм	1200
	Глубина	мм	450
Масса		кг	260
Соединительные линии	Газовая	мм	16
	Жидкостная	мм	16
	Дренажная	мм	21

На рисунке 4.1 представлена схема расположения выбранного кондиционера с верхней подачей SUA в помещении офиса.

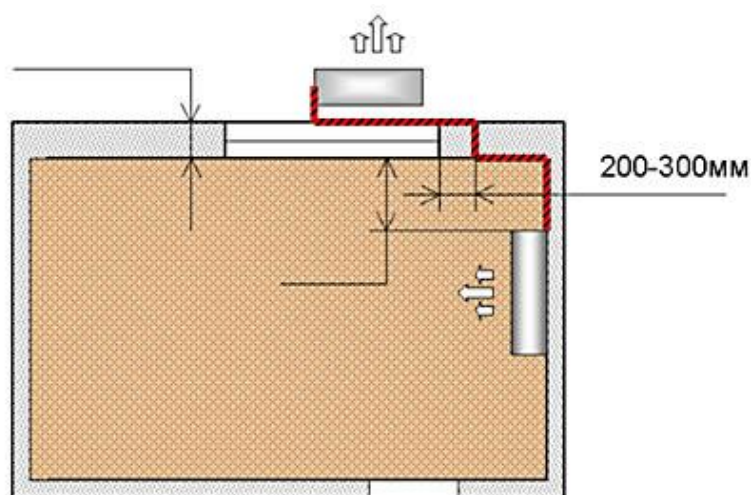


Рисунок 4.1 – Схема расположения кондиционера

В данной работе был произведен расчет кондиционирования производственного помещения с параметрами 9x5x4.5 м. Были получены

данные по количеству тепла и все необходимые характеристики для выбора кондиционера.

С учетом полученных данных был выбран кондиционер SUA 0501 с верхней подачей воздуха.

4.4 Расчет искусственного освещения помещения

Условия искусственного освещения в офисе оказывают большое влияние на зрительную работоспособность, физическое и моральное состояние людей, а, следовательно, на производительность труда, качество продукции и производственный травматизм.

Для создания благоприятных условий труда освещение должно отвечать следующим требованиям.

Освещенность на рабочем месте должна соответствовать характеру выполняемой работы по СНиПП-2.04-05-2002 «Естественное и искусственное освещение. Общие требования» [17].

Яркость на рабочей поверхности и в пределах окружающего пространства должна распределяться по возможности равномерно.

Резкие тени на рабочей поверхности должны отсутствовать.

Освещение должно обеспечивать необходимый спектральный состав света для правильной цветопередачи.

Система освещения не должна являться источником других вредных факторов (шум и т.д.), а также должна быть электро- и пожаробезопасной.

Искусственное освещение применяется при отсутствии или недостаточности естественного освещения, осуществляется путем использования таких источников света как лампы накаливания, газоразрядные лампы, плоские и щелевые световоды.

Нормирование искусственного освещения производится в соответствии со СНиПП-2.04-05-2002, освещенность на рабочих местах нормируется в зависимости от условий выполнения зрительных работ, вида источника света и системы освещения.

Точечный метод базируется на основном законе светотехники, и в зависимости от светового прибора (точечный, линейный, прожектор) или характеристики объекта (закрытое помещение, улица, площадь) расчетные формулы различны.

$$E_B = \frac{I_\alpha \cdot \cos \beta}{R^2} \quad (4.11)$$

где I_α – сила света в направлении от источника к точке, кд;

$\cos \beta$ – косинус угла падения луча на плоскость;

R – расстояние между источником и точкой, м.

Расчету освещенности должен предшествовать выбор типа осветительных приборов, а также определение расположения и высоты подвеса

их в помещении (h_p), определено нормируемое значение освещенности (E_n). Расчетная точка освещается практически всеми светильниками, находящимися в помещении, которые создают в расчетной точке относительную суммарную освещенность Σe , однако обычно учитывается действие ближайших светильников.

Трудно точно определить, какие светильники следует считать ближайшими и учитывать в Σe .

Во всех случаях при определении Σe не должны учитываться светильники, реально не создающие освещенности в контрольной точке из-за ее затенения оборудованием или самим рабочим при его нормальном фиксированном положении на рабочем месте.

В качестве контрольных точек выбираются те точки освещаемой поверхности, в которых Σe имеет наименьшее значение. Не следует выискивать самую малую освещенность (у стен или в углах): если в подобных точках есть рабочие места, задача обеспечения здесь нормируемых значений освещенности может быть решена увеличением мощности ближайших светильников или установкой дополнительных светильников.

Определение e для каждой контрольной точки производится с помощью пространственных изолюкс условной горизонтальной освещенности, на которых находится точка с заданным d и h_p . Если расчетная точка не совпадает точно с изолюксами, то e определяется интерполированием между ближайшими изолюксами.

Пусть суммарное действие светильников создает в контрольной точке условную освещенность Σe . Действие более далеких светильников и отраженная составляющая приближенно учитываются коэффициентом μ . Тогда для получения в этой точке освещенности E с коэффициентом запаса K_3 лампы в каждом светильнике должны иметь поток:

$$\Phi = \frac{1000 \cdot E_n \cdot K_3}{\mu \cdot \Sigma e_r} \quad (4.12)$$

где 1000 лм – условный поток лампы;

K_3 – коэффициент запаса;

E_n – нормированная освещенность;

$\mu=1.2$ – коэффициент добавочной поверхности, учитывая воздействие удаленных светильников и отраженных световых потоков;

Σe_r – сумма относительных условных освещенностей от ближайших светильников, лк.

Последовательность расчета осветительной установки точечным методом:

– находят минимальную нормированную освещенность;

– выбирают типы источника света и светильника, рассчитывают размещение светильников по помещению;

- на плане помещения с указанными светильниками намечают контрольные точки, в которых освещенность может оказаться наименьшей;
- вычисляют условную освещенность в каждой контрольной точке и точку с наименьшей условной освещенностью принимают за расчетную;
- по справочным таблицам устанавливают коэффициенты запаса и добавочной освещенности;
- по формуле находят световой поток лампы;
- по световому потоку из таблиц выбирают ближайшую стандартную лампу, световой поток которой отличается от расчетного не более чем на -10 или $+20$ %, и определяют ее мощность;
- подсчитывают электрическую мощность всей осветительной установки.

Очень важно при вычислении светового потока ламп правильно выбрать расчетную точку. В качестве ее на освещаемой поверхности, в пределах которой должна быть обеспечена нормированная освещенность, берут точку с минимальной освещенностью. Такую точку следует брать в центре поля или посередине одной стороны крайнего поля - пространства, ограниченного четырьмя ближайшими светильниками.

По точечному методу при кругло симметричных точечных излучателях принимается, что световой поток лампы в каждом светильнике равен 1000 лк. Создаваемую таким светильником освещенность называют условной. Световой поток лампы в каждом светильнике определяется по формуле 4.10.

Согласно СНиП РК 2.04-05-2002 для разряда зрительной работы III(б) нормированная освещенность $E_n = 300$ лк.

Точечным методом проверим соответствие данного количества и типа светильников нормируемой величине.

Высота светильника: $h_c = 0,332$ м.

Рабочая поверхность над полом: $h_p = 1$ м.

Высота помещения: $h = 4,5$ м.

Тогда расчетная высота составит

$$h_{расч} = 4,5 - 0,332 - 1 = 3,168 \text{ м}$$

Все светильники должны быть равномерно распределены по площади помещения.

Возьмем 2 ряда по 2 светильника.

Расстояние между светильниками и рядами измеряется по формуле

$$L_B = \frac{B}{2} \quad (4.13)$$

$$L_B = \frac{12}{2} = 6 \text{ м}$$

Расстояние между рядами

$$L_A = \frac{6}{2} = 3 \text{ м}$$

Расстояние от стены до ряда

$$d_B = \frac{3}{2} = 1,5 \text{ м}$$

Расстояние от стены начало ряда

$$d_B = \frac{6}{2} = 3 \text{ м}$$

Наметим контрольную точку А, для нее определяем суммарную условную освещенность всех светильников.

Находим проекцию расстояния на потолок от точки А до светильника d.

$$d = \sqrt{3^2 + 1,5^2} = 3,354 \text{ м}$$

Расчетная схема расположение светильников для точеного метода представлена на рисунке 4.2.

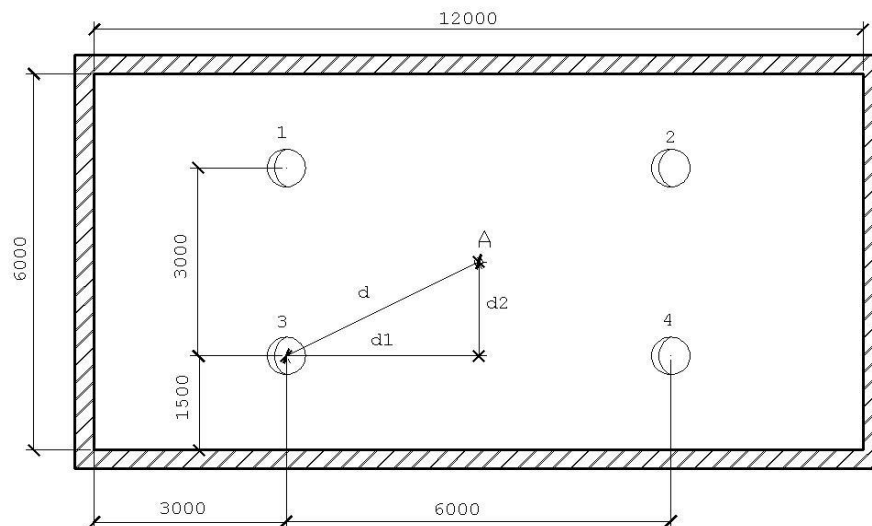


Рисунок 4.2 – Расчетная схема освещенности помещения

Расчет горизонтальной освещенности производится в следующей последовательности:

1 Определяем $tg \alpha$ по заданной высоте подвеса светильника из выражения

$$\operatorname{tg} \alpha = \frac{3,354}{3,168} = 1,059$$

Так как точка А находится на одинаковом расстоянии до значение $\operatorname{tg} \alpha$ будет для всех одним и тем же.

2 По найденному $\operatorname{tg} \alpha$ определяют угол α и $\cos^3 \alpha$

$$\alpha = \operatorname{arctg}(1,059) = 46,6^\circ \approx 45^\circ,$$

$$\cos^3 \alpha = \cos^3 45^\circ = 0,354$$

3 Определим значение силы света I_α по найденному углу α .

$$I_\alpha = 158 \text{ кд}$$

4 По расчетной формуле определим условную горизонтальную освещенность e_{AG} (для лампы в 1000 лм)

$$e_{AG} = \frac{158 \cdot 0,354}{3,354^2 \cdot 1,5} = 3,315 \text{ лк}$$

где $K_z = 1,5$ – коэффициент запаса.

5 Найденную условную освещенность пересчитаем с учетом потока лампы, установленной в светильнике. В нашем случае для светильника лампа накаливания мощностью 100 Вт., с световым потоком 1340 лм.

$$E_{AG} = \frac{1,1 \cdot 1340}{1000 \cdot 1,5} \cdot 4 \cdot 3,315 = 13,03 \text{ лк}$$

где Φ_n – световой поток лампы по ГОСТу [18].

Точный метод позволяет определять освещенность в различных местах помещения. Рассчитав таким методом заданное помещение, видно, что мощность и количество ламп недостаточно согласно СНиП РК 2.04.-05-2002. Т.к. результат расчета освещенности получился меньше 300 лк. Поэтому производится реконструкции заданного помещения [19]. Выбраны лампы мощностью 100 Вт с световым потоком 2400 лм. После этого получается количество светильников равное 9 шт., что вполне приемлемо для помещения с размерами 6x12x4,5.

Заключение

В данной дипломной работе был создан динамический интернет-портал на основе технологии Web 2.0. При проектировании работы, были учтены и решены все основные проблемы данного веб-сайта. База данных для интернет-портала проектировалась на MySQL, так как эта среда проектирования заточена именно под веб-приложения, быстро реагирует на запросы, удобна и проста.

Серверная часть создавалась на языке PHP, который представляет собой оптимальное средство для работы с базами данных в среде интернет. Клиентская часть выполнена с помощью технологии Web 2.0, применяя JQuery в качестве плагина для JavaScript и Twitter Bootstrap в качестве графической оболочки.

Дизайн выполнен в духе современного интернета, в стиле flat UI, что гарантирует отличное понимание работы системы всеми категориями пользователей.

В разработке использовались новейшие технологии HTML 5 и CSS 3. Система отлично работает как на компьютерах, так и на любых современных мобильных устройствах, дизайн автоматически адаптируется к разрешению экрана.

Список используемой литературы

- 1 Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление. – 5-е изд. – СПб.: БХВ–Петербург, 2004. – 1040 с.
- 2 Котеров Д., Костарев А. PHP. В подлиннике. – СПб.: БХВ–Петербург, 2005. – 1120 с.
- 3 Зандстра М. PHP: объекты, шаблоны и методики программирования. – 3-е изд. – М.: Вильямс, 2010. – 560 с.
- 4 Кузнецов М., Симдянов И. PHP . Практика создания Web-сайтов. – 2-е изд. – СПб.: БХВ–Петербург, 2008. – 1264 с.
- 5 Флэнаган Д. Чего не может JavaScript. Подробное руководство / Перевод Киселева А. – 5-е изд. – СПб.: Символ-Плюс, 2008. – 280 с.
- 6 Christian J. Test Driven JavaScript Development. – Addison: Wesley Professional, 2010. – 26 p.
- 7 Гукасян Г.М. Экономика от «А» до «Я»: Тематический справочник. – М.: ИНФРА–М, 2009. – 480 с.
- 8 Иванов И.Н. Экономика промышленного предприятия: Учебник. – М.: ИНФРА–М, 2011. – 395 с.
- 9 Адам Ф. jQuery для профессионалов = Pro jQuery. – 3-е изд. – СПб: Питер, 2012. – 689 с.
- 10 Сайт <http://php.ru/manual/>
- 11 Кузнецов М., Симдянов И. PHP . Практика создания Web-сайтов. – 2-е изд. – СПб.: БХВ–Петербург, 2008. – 1264 с.
- 12 Рофе А.И. Экономика труда: Учебник. – М.: КноРус, 2010. – 400 с.
- 13 Сайт <http://getbootstrap.com/css/>
- 14 Скляренко В.К., Прудников В.М. Экономика предприятия: Учебник. – М.: ИНФРА–М, 2006. – 528 с.
- 15 Сайт http://scholar.urc.ac.ru/ped_journal/numero4/pedag/tsit3.html
- 16 СНиП РК 4.02-05-2001. Вентиляция. Общие требования. – Астана: Издательство стандартов, 2001.
- 17 СНиПП-2.04-05-2002. Естественное и искусственное освещение. Общие требования. – Москва: Издательство стандартов, 2002.
- 18 Белов С.В., Ильницкая А.В., Козьяков А.Ф. Безопасность жизнедеятельности: Учебник для вузов. – М.: Высшая школа, 2005. – 448 с.
- 19 Хакимжанов Т.Е. Сборник задач по охране труда и безопасности жизнедеятельности: Учебное пособие для вузов. – Алматы: Эвер, 2007. – 274 с.

Приложение А

Листинг программы

```
<?php
class GroupController extends GxController {
    public function actionView($id) {
        $isfollow=Fans::model()-
>countByAttributes(array('Group_ID'=>$id, 'User_ID'=>Yii::app()-
>user->id));
        $followers=Fans::model()-
>findAllByAttributes(array('Group_ID'=>$id));
        $mysongs=SONG::model()-
>findAllByAttributes(array('Song_artist'=>$id));
        $photos = Photo::model()-
>findAllByAttributes(array('Group_id'=>$id),array('order'=>'Photo_
id DESC'));
        $videos = Videos::model()-
>findAllByAttributes(array('Group_id'=>$id),array('order'=>'Video_
id DESC'));
        $this->render('view', array(
            'model' => $this->loadModel($id, 'Group'),
            'id'=>$id,
            'mysongs'=>$mysongs,
            'photos'=>$photos,
            'isfollow'=>$isfollow,
            'followers'=>$followers,
            'videos'=>$videos,
        ));
    }
    public function actionCreate() {
        $model = new Group;
        if (isset($_POST['Group'])) {
            $model->setAttributes($_POST['Group']);
            if ($model->save()) {
                if (Yii::app()->getRequest()-
```

Продолжение приложения А

```
>getIsAjaxRequest())
        Yii::app()->end();
    else
        $this->redirect(array('view', 'id' =>
$model->Group_id));
    }
}
$this->render('create', array( 'model' => $model));
}
public function actionUpdate($id) {
    $model = $this->loadModel($id, 'Group');
    if (isset($_POST['Group'])) {
        $model->setAttributes($_POST['Group']);

        if ($model->save()) {
            $this->redirect(array('view', 'id' => $model->
>Group_id));
        }
    }
    $this->render('update', array(
        'model' => $model,
    ));
}
public function actionDelete($id) {
    if (Yii::app()->getRequest()->getIsPostRequest()) {
        $this->loadModel($id, 'Group')->delete();

        if (!Yii::app()->getRequest()->getIsAjaxRequest())
            $this->redirect(array('admin'));
    } else
        throw new CHttpException(400, Yii::t('app', 'Your
request is invalid.));
}
public function actionIndex() {
```

Продолжение приложения А

```
$dataProvider = new CActiveDataProvider('Group');
    $this->render('index', array(
        'dataProvider' => $dataProvider,
    ));
}
public function actionAdmin() {
    $model = new Group('search');
    $model->unsetAttributes();
if (isset($_GET['Group']))
    $model->setAttributes($_GET['Group'])
    $this->render('admin', array(
        'model' => $model,
    ));
}
public function actionUpload()
{
    Yii::import("ext.EAjaxUpload.qqFileUploader");
    $folder = Yii::getPathOfAlias('webroot') . '/songs/'; //
folder for uploaded files
    $allowedExtensions = array("mp3");
//array("jpg","jpeg","gif","exe","mov" and etc...
    $sizeLimit = 20 * 1024 * 1024; // maximum file size in
bytes
    $uploader = new qqFileUploader($allowedExtensions,
$sizeLimit);
    $newfilename = md5(microtime());
    $filename = $newfilename . '.' . $upload['extension'];
    $result = $uploader->handleUpload($folder);

    $fileSize = filesize($folder . $result['filename']);
//GETTING FILE SIZE
    // $fileName=$result['filename'];//GETTING FILE NAME
    $result = htmlspecialchars(json_encode($result),
ENT_NOQUOTES);
```

Продолжение приложения А

```
        echo $result; //
    }
    public function actionFan()
    {
        $model=new Fans();
        $model->User_ID=$_POST['id'];
        $model->Group_ID=$_POST['group'];
        if ($model->save())
        {
            /* echo CHtml::ajaxButton(
                'PhC,PiPëCÍP°C,СЪСÍСЦ',
                CController::createUrl('group/unfan'),
                array(
                    'type' => 'POST', // method
                    'data' => array('update' => TRUE,
                        'group' => $_GET['id'],
                        'id' => Yii::app()->user->id,
                    ), // DATA
                    'update' => '.follow-btn', // C±C,Ps
                ), array('class' => 'btn btn-primary btn-sm check
                short', 'id'=>'unfollow'));*/
        }
    }
    public function actionUnfan()
    {
        $id=$_POST['id'];
        $group=$_POST['group'];
        $model=Fans::model()-
        >FindByAttributes(array('Group_ID'=>$group,'User_ID'=>$id));
        if ($model->delete())
        {
            /* echo CHtml::ajaxButton(
```

Продолжение приложения А

```
'PŸC,P°C,СБ С,,P°PSPsPj',
    CController::createUrl('group/fan'),
    array(
        'type' => 'POST', // method
        'data' => array('update' => TRUE,
            'group' => $_GET['id'],
            'id' => Yii::app()->user->id,
        ), // DATA
        'update' => '.follow-btn',
    ),
array('class' => 'btn btn-primary btn-sm check short',
'id'=>'follow'));
    };
}
public function actionID3($filename)
{
    $model = new SONG;
    if (isset($_POST['SONG'])) {
        $model->setAttributes($_POST['SONG']);

        if ($model->save()) {
            if (Yii::app()->getRequest()->getIsAjaxRequest())
                Yii::app()->end();
            else
                $this->redirect(array('view', 'id' => $model-
>Song_id));
        }
    } else {

        $f = fopen('songs/' . $filename, 'rb');
        rewind($f);
        fseek($f, -128, SEEK_END);
        $tmp = fread($f, 128);
        if ($tmp[125] == Chr(0) and $tmp[126] != Chr(0)) {
            // ID3 v1.1
```

Продолжение приложения А

```
$format =
'a3TAG/a30NAME/a30ARTISTS/a30ALBUM/a4YEAR/a28COMMENT/x1/C1TRACK/C1
GENRENO';

    } else {
        // ID3 v1
        $format =
'a3TAG/a30NAME/a30ARTISTS/a30ALBUM/a4YEAR/a30COMMENT/C1GENRENO';
    }

    $id3tag = unpack($format, $tmp);

    $model->Song_title = $id3tag['NAME'];
    $model->Song_url = $filename;
    $json = json_encode($id3tag);
    print_r($id3tag);
    $this->renderPartial('_formdetail', array(
        'model' => $model,
        'id3tag' => $id3tag,
    ));
    //echo ("xxxx");
}
}

public function actionSongCreate()
{
    $model = new SONG;
    if (isset($_POST['SONG'])) {
        $model->setAttributes($_POST['SONG']);

        if ($model->save()) {
            if (Yii::app()->getRequest()->getIsAjaxRequest())
                Yii::app()->end();
            else
                $this->redirect(array('view', 'id' => $model->Song_id));
        }
    }
}
```


Продолжение приложения A

```
$this->render('create', array('model' => $model));
}
public function actionLoadAva()
{
    $id=$_GET['group'];
    $model = Group::model()->findByPk($id);
    $model->Group_logo=$_GET['filename'];
    $model->save();
}
}
<?php

class UserController extends Controller
{
    /**
     * @var string the default layout for the views. Defaults to
     * '//layouts/column2', meaning
     * using two-column layout. See
     * 'protected/views/layouts/column2.php'.
     */
    public $layout='//layouts/column2';

    /**
     * @return array action filters
     */
    public function filters()
    {
        return array(
            'accessControl', // perform access control for CRUD
operations
            'postOnly + delete', // we only allow deletion via
POST request
        );
    }
}
```

Продолжение приложения А

```
* Specifies the access control rules.
* This method is used by the 'accessControl' filter.
* @return array access control rules
*/
public function accessRules()
{
    return array(
        array('allow', // allow all users to perform
'index' and 'view' actions
            'actions'=>array('index','view', 'upload'),
            'users'=>array('*'),
        ),
        array('allow', // allow authenticated user to
perform 'create' and 'update' actions
            'actions'=>array('create','update',
'updateFast'),
            'users'=>array('@'),
        ),
        array('allow', // allow admin user to perform
'admin' and 'delete' actions
            'actions'=>array('admin','delete'),
            'users'=>array('admin'),
        ),
        array('deny', // deny all users
            'users'=>array('*'),
        ),
    );
}

/**
 * Displays a particular model.
 * @param integer $id the ID of the model to be displayed
 */
public function actionView($id)
```

Продолжение приложения А

```
$powers=Specialnost::model()->findAll();
$cspec=SpecUser::model()-
>findByAttributes(array('User_id' => $id));
$specuser=new SpecUser;
$model=$this->loadModel($id);
$city=City::model()->findByPk($model->City_id);
$this->render('view',array(
    'model'=>$model, 'city'=>$city, 'powers'=>$powers,
'cspec'=>$cspec
));
}

/**
 * Creates a new model.
 * If creation is successful, the browser will be redirected
to the 'view' page.
 */
public function actionCreate()
{
    $model=new User;

    // Uncomment the following line if AJAX validation is
needed
    // $this->performAjaxValidation($model);

    if(isset($_POST['User']))
    {
        $model->attributes=$_POST['User'];
        if($model->save())
            $this->redirect(array('view','id'=>$model-
>User_id));
    }

    $this->render('create',array(
```

Продолжение приложения А

```
'model'=>$model,
    ));
}

/**
 * Updates a particular model.
 * If update is successful, the browser will be redirected to
the 'view' page.
 * @param integer $id the ID of the model to be updated
 */
public function actionUpdate($id)
{
    $model=$this->loadModel($id);

    // Uncomment the following line if AJAX validation is
needed
    // $this->performAjaxValidation($model);

    if(isset($_POST['User']))
    {
        $model->attributes=$_POST['User'];
        if($model->save())
            $this->redirect(array('view','id'=>$model-
>User_id));
    }

    $this->render('update',array(
        'model'=>$model,
    ));
}
```