

**МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет компьютерных наук
Кафедра технологий обработки и защиты информации

Сайт по поиску музыкантов и групп «Believe»

Курсовой проект

09.03.02 Информационные системы и технологии
Обработка информации и машинное обучение

Обучающийся _____ М.О. Саввин, 3 курс, д/о

Обучающийся _____ Е.А. Стеблева, 3 курс, д/о

Обучающийся _____ М.А. Дынин, 3 курс, д/о

Оглавление

Введение.....	3
1. Постановка задачи.....	4
2. Анализ предметной области.....	5
2.1 Глоссарий.....	5
2.2 Анализ существующих решений.....	5
2.3 Анализ задачи.....	7
2.3.1 Варианты использования приложения.....	7
2.3.2 Взаимодействие компонентов системы.....	9
2.3.3 Варианты состояния системы.....	10
2.3.4 Варианты действия в системе.....	11
2.3.5 Развертывание приложения.....	13
3. Анализ средств реализации.....	15
4. Реализация.....	16
4.1 Сущности.....	16
4.2 Сценарии воронок конверсии.....	20
4.3 Графический интерфейс.....	20
4.3.1 Шапка сайта.....	20
4.3.2 Главная страница.....	20
4.3.3 Страница регистрации.....	21
4.3.4 Страница авторизации.....	22
4.3.5 Страница просмотра заявок музыкантов.....	22
4.3.6 Страница просмотра заявок групп.....	23
4.3.7 Страница создания заявки музыканта.....	24
4.3.8 Страница создания заявки группы.....	25
4.3.9 Страница личного кабинета.....	25
4.3.10 Страница редактирования личного кабинета.....	26
4.3.11 Страница администрирования.....	27
5. Тестирование.....	29
5.1 Дымовое тестирование.....	29
5.2 UI тесты.....	30
5.3 Юзабилити тесты.....	35
Заключение.....	37

Введение

В современном интернете есть множество различных сайтов и сервисов, которые упрощают людям поиск работы, недвижимости, автомобиля, вещей и многого другого. Но любому человеку нужен отдых, и самое очевидное и первостепенное, о чем думает каждый — это хобби, одним из которых является музыка. В настоящее время существуют группы и/или паблики в социальных сетях, форумы, посвященные поиску музыкантов в каком-либо конкретном городе, но у каждого из таких форумов различные функциональные возможности, разный интерфейс.

Желаемый сайт должен облегчить пользователям поиск музыкантов в группу и наоборот в любом городе, который их интересует. Тем самым необходимости в просмотре большого количества форумов и сайтов больше не будет. Желаемый сайт должен предоставлять основную необходимую функциональность:

- поиск музыкантом группы,
- поиск группой музыканта,
- ненагруженный, интуитивно понятный даже низкоуровневому пользователю интерфейс.

Данный курсовой проект посвящен разработке именно такого, простого в освоении, но в то же время выполняющего самые необходимые функции, сайта, способного уменьшить временные затраты каждого человека на поиск необходимых музыкантов и/или групп.

1. Постановка задачи

Цель курсовой работы: реализовать сайт, который отвечает следующим требованиям:

- стабильная работа на современных веб-браузерах,
- интуитивно понятный пользовательский интерфейс,
- адаптивность сайта под мобильные устройства,
- сдержанный внешний вид, выдержанный в едином стиле,
- отсутствие необходимости в справке для возможности

осуществления основных задач:

- 1) регистрация/Авторизация,
- 2) создание/Удаление заявок,
- 3) вывод заявки в топ при помощи VIP-кода,
- 4) просмотр и отклик на заявки,
- 5) редактирование информации в личном кабинете,
- 6) поиск заявок с помощью фильтра,
- 7) модерация заявок с уведомлением клиента о решении по почте,
- 8) генерация VIP-кодов администратором.

Для достижения данной цели были выделены следующие задачи:

- 1) разработка Front-end части сайта, находящиеся на телефоне/компьютере пользователя,
- 2) разработка Back-end части сайта, развернутой на удаленном сервере сайта,
- 3) создание связи между Front-end и Back-end частями сайта,
- 4) разработка базы данных, расположенной на удаленном сервере.

2. Анализ предметной области

2.1 Глоссарий

VIP-код — последовательность символов, вводимая при создании заявки, с помощью которой пользователь может вывести ее (заявку) в топ.

Заявка — заполненная пользователем информация, необходимая и достаточная для поиска и отбора музыкантов и групп.

Топ — расположение заявок в начале списка всех заявок.

Исполнитель — сольный артист (музыкант) или группа.

2.2 Анализ существующих решений

1. <https://myband.ru/>

Достоинства:

- большое разнообразие функций и возможностей,
- красивое оформление,
- возможность загрузки своих аудио-работ,
- синхронизация с различными соц. сетями.

Недостатки:

- ограниченное количество критериев для фильтрации,
- отсутствие структуры составления заявки,
- отсутствие кроссбраузерности.

2. <https://muzlk.com/>

Достоинства:

- присутствие личных сообщений,
- возможность добавления своих работ в формате аудио и видео,
- формат соц. сети.

Недостатки:

- плохая оптимизация и проблемы с входом на сервис,
- неоднозначный дизайн и подбор цветов,
- перебои в работе сайта,

- отсутствие модерирования,
- отсутствие возможности редактирования личного кабинета.

3. <https://www.realrocks.ru/>

Достоинства:

- приятный дизайн,
- большое разнообразие жанров,
- синхронизация с различными соц. сетями.

Недостатки:

- отсутствие возможности поиска группы,
- присутствие чужой рекламы на странице в случае обычного аккаунта,
- ограниченное количество критериев для фильтрации.

2.3 Анализ задачи

2.3.1 Варианты использования приложения

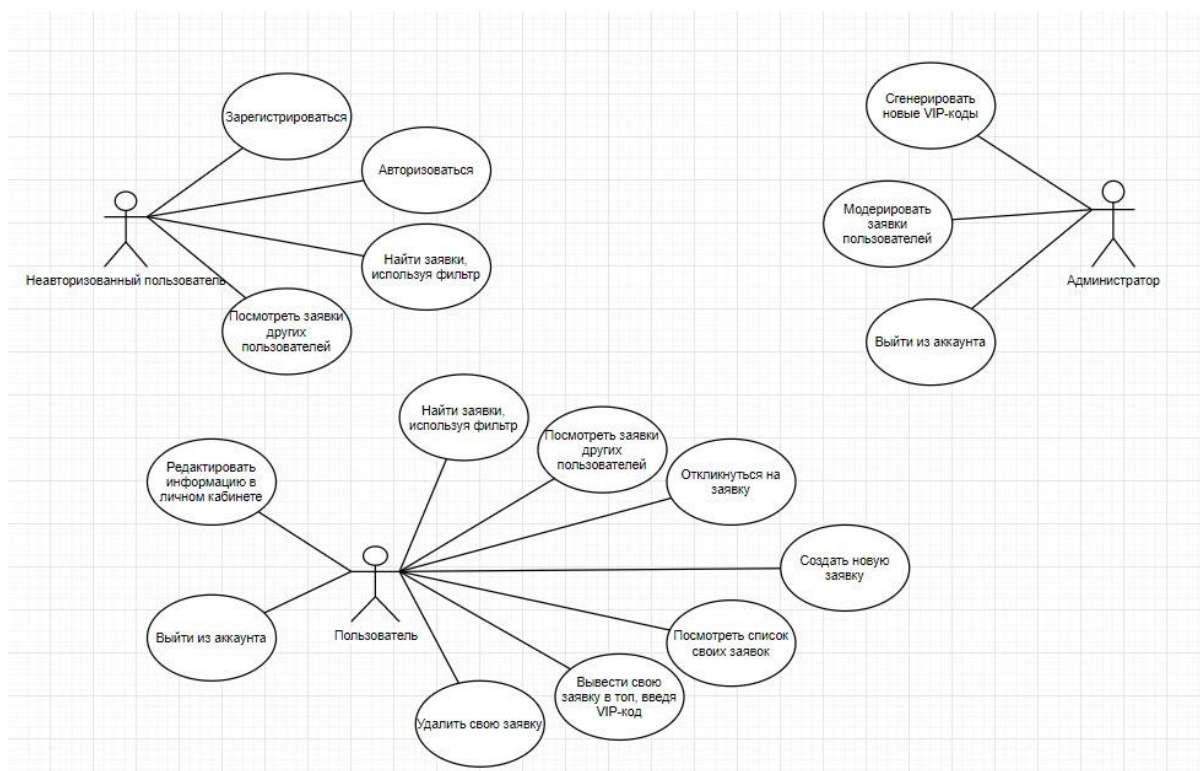


Рисунок 1 - Диаграмма прецедентов

При взаимодействии с сайтом у пользователя есть определенный список возможностей, который более наглядно изображен на рисунке 1:

- поиск заявок, используя фильтр,
- просмотр заявок других пользователей,
- отклик на заявку,
- создание заявки,
- просмотр списка своих заявок,
- вывод заявки в топ,
- удаление заявки,
- выход из аккаунта,
- редактирование информации в личном кабинете.

Неавторизованный пользователь имеет возможность:

- зарегистрироваться,
- авторизоваться,
- найти заявки, используя фильтр,
- посмотреть заявки других пользователей.

Администратор сервиса может:

- сгенерировать VIP-коды,
- модерировать заявки пользователей,
- выйти из аккаунта.

2.3.2 Взаимодействие компонентов системы

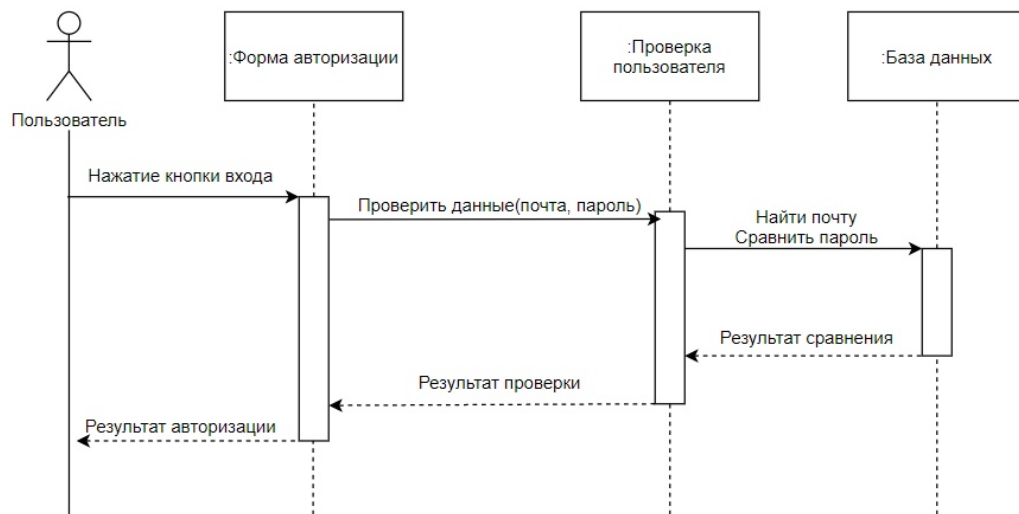


Рисунок 2 - Диаграмма последовательностей

На рисунке 2 показана диаграмма последовательности, на которой изображено упорядоченное во времени взаимодействие объектов при авторизации пользователя.

Для авторизации пользователь обращается к форме авторизации, которая передаёт введённые данные на проверку в модуль проверки пользователя. Тот в свою очередь проверяет существование данного пользователя в базе данных и совпадение введённого пароля с паролем, хранящимся в базе данных. Модуль проверки пользователя посылает статус проверки на форму авторизации, которая выводит пользователю результат авторизации.

2.3.3 Варианты состояния системы

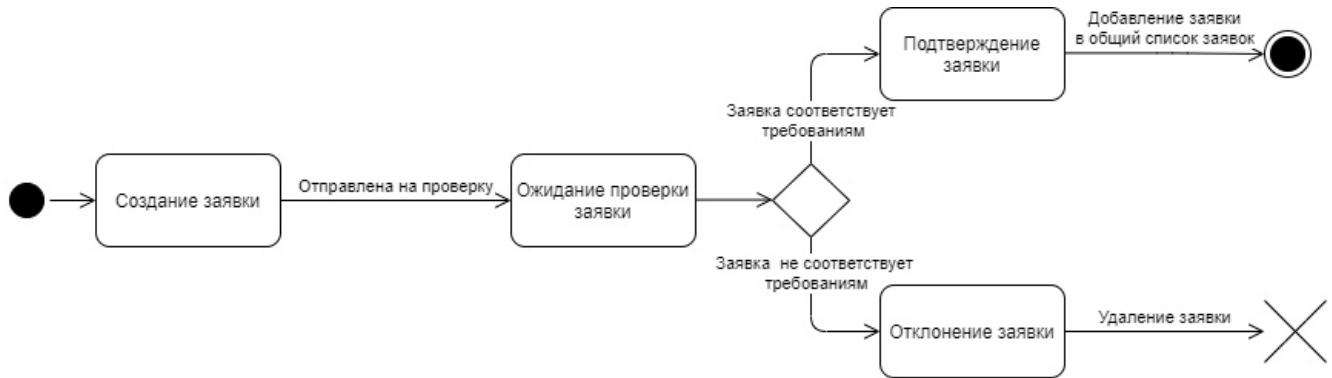


Рисунок 3 - Диаграмма состояний

Диаграмма состояний, изображенная на Рисунке 3, отражает возможные состояния заявки. После создания заявки она отправляется на проверку администратором и переходит в состояние ожидания проверки. Если заявка соответствует требованиям (проходит проверку), то она переходит в состояние подтверждения (является подтвержденной) и добавляется в общий список всех заявок нужной категории. Если заявка не соответствует требованиям (не проходит проверку), то она переходит в состояние отклонения (является отклоненной) и удаляется из списка ожидающих проверку заявок.

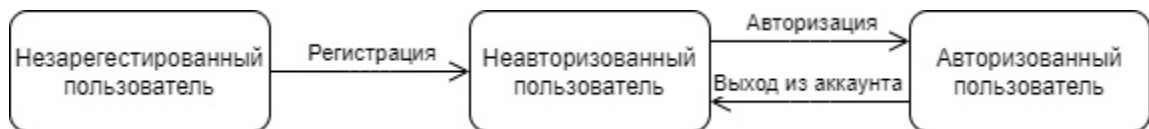


Рисунок 4 - Диаграмма состояний

Диаграмма состояний, изображенная на Рисунке 4, отражает возможные состояния пользователя. Изначально любой пользователь, не прошедший регистрацию, находится в состоянии незарегистрированного пользователя. После прохождения регистрации пользователь переходит в состояние неавторизованного пользователя. После авторизации пользователь переходит в состояние авторизованного пользователя. Если пользователь выходит из аккаунта он возвращается к состоянию неавторизованного пользователя и может авторизоваться снова.

2.3.4 Варианты действия в системе

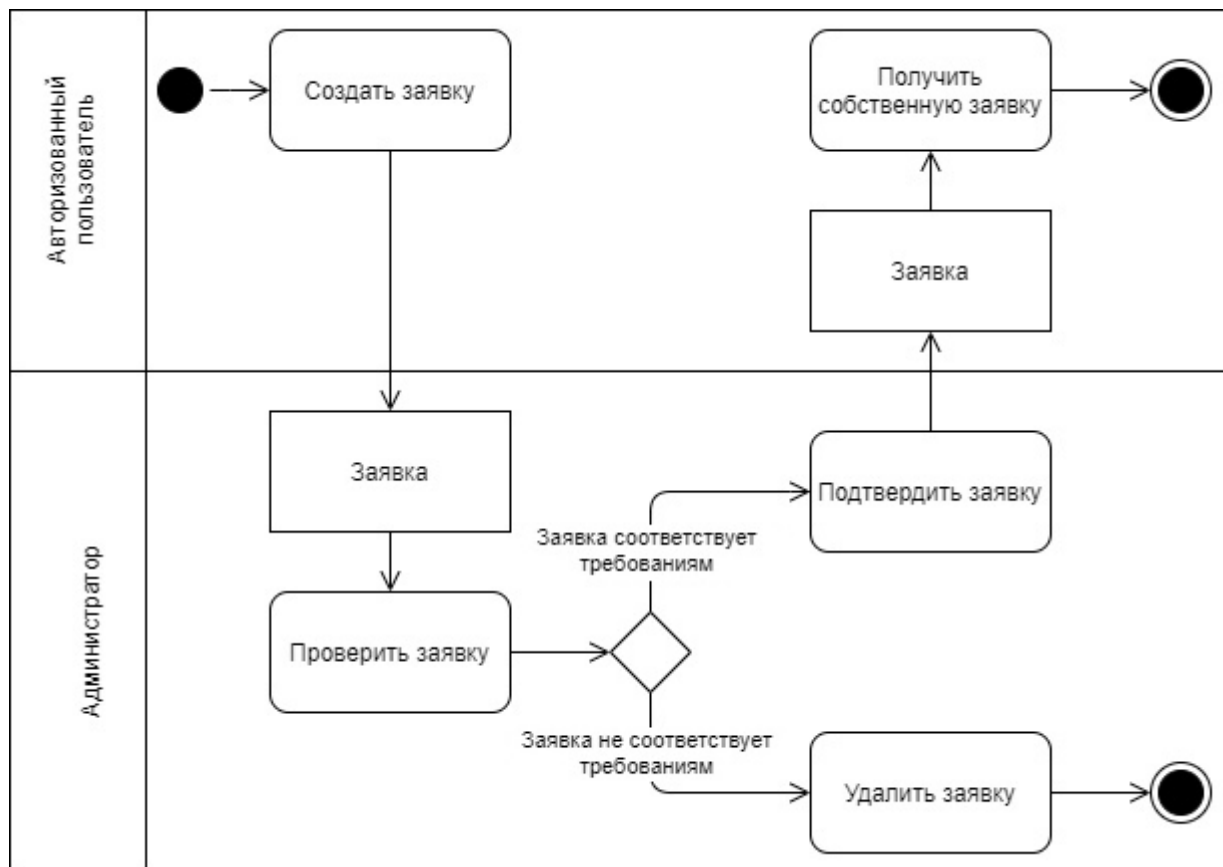


Рисунок 5 - Диаграмма активности

Диаграмма активности, изображенная на Рисунке 5, отражает действия авторизованного пользователя и администратора при создании заявки. После того, как авторизованный пользователь создает заявку, она переходит к администратору. При получении заявки администратор должен ее проверить. Если заявка не соответствует требованиям, то администратор удаляет ее из списка ожидающий проверку заявок. Если заявка соответствует требованиям, то администратор подтверждает ее, и она (заявка) переходит тому авторизованному пользователю, который ее создал. Авторизованный пользователь получает собственную заявку, и она в свою очередь закрепляется за ним.

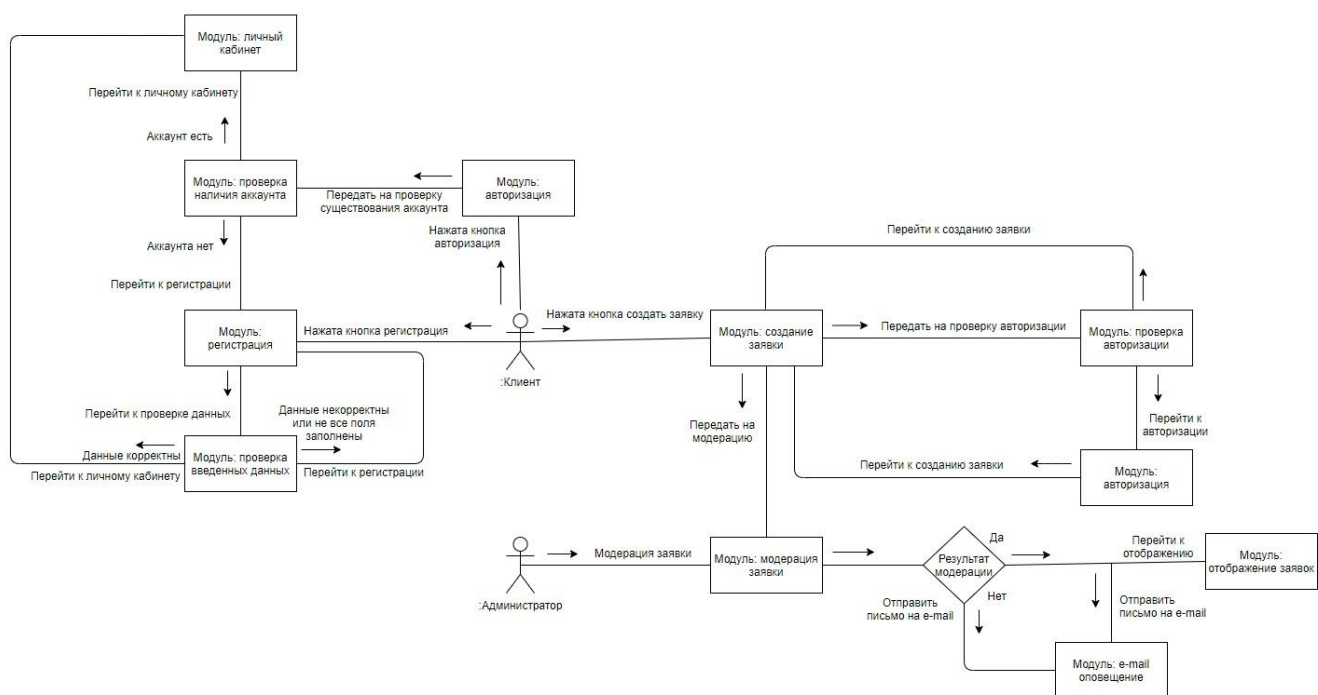


Рисунок 6 - Диаграмма взаимодействий

На Рисунке 6 представлена диаграмма взаимодействий. Она отражает возможные действия пользователя и системы.

Если пользователь захочет оставить заявку, заполнив анкету, система проверит статус авторизации клиента, если он не авторизирован, то система предупредит его об этом и предоставит возможность авторизоваться. Если пользователь авторизован, заявка попадает на модерацию, где администратор проверяет корректность предоставленных пользователем данных и выносит вердикт. Если заявка одобрена, то клиент получает уведомление об одобрении заявки, и она появляется в списке всех заявок, в случае отказа, пользователь получает уведомление об отказе с указанием причины.

Если пользователь захочет пройти процесс регистрации, заполнив необходимые поля, система проверит входящие данные на корректность и в случае ошибки выдаст предупреждение. Если данные корректны, пользователь перейдет в личный кабинет.

При желании пользователя авторизоваться, заполнив необходимые поля, система так же проверит корректность введенных данных, в случае, если пользователя с такими данными нет, система предложит зарегистрироваться.

2.3.5 Развертывание приложения

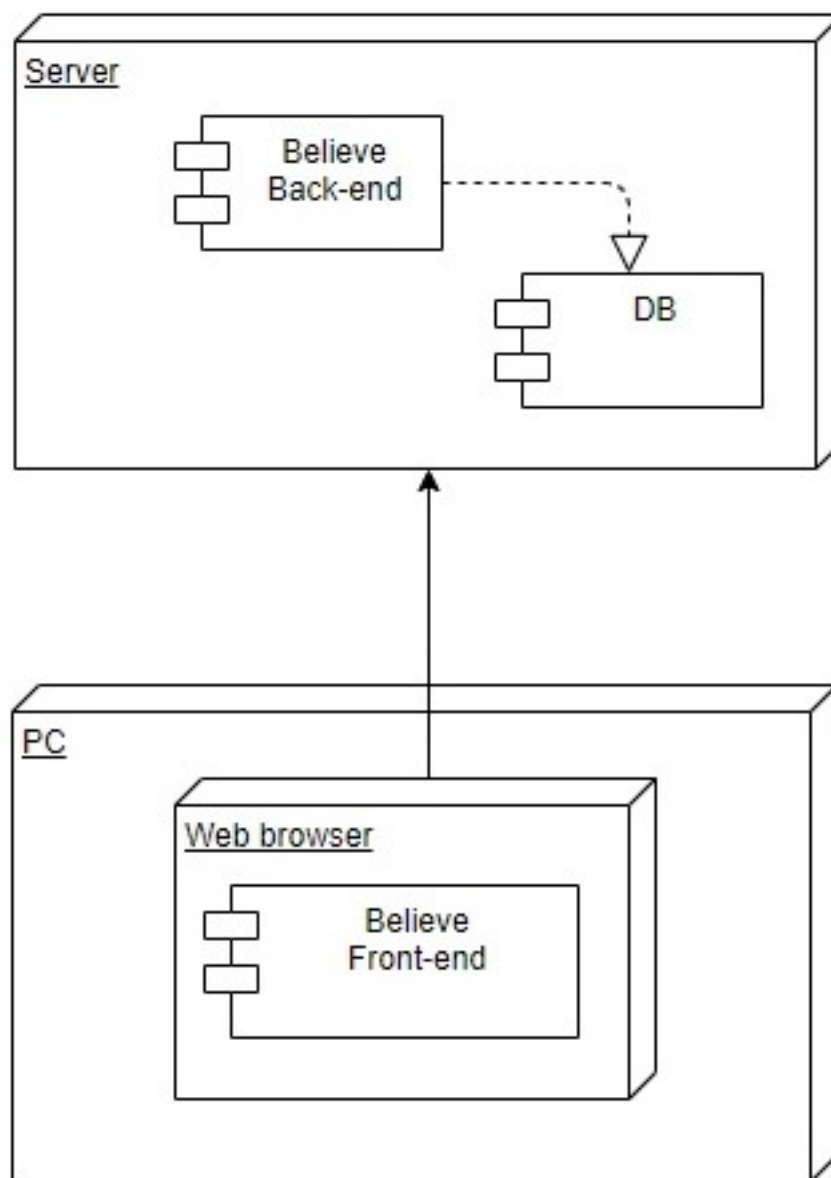


Рисунок 7 - Диаграмма развертывания

На Рисунке 7 представлена диаграмма развертывания, чтобы определить какие аппаратные компоненты («узлы») существуют, какие программные компоненты работают на каждом узле и как различные части этого комплекса соединяются друг с другом. Для разрабатываемого web-приложения

узлом устройства является персональный компьютер и сервер, а в качестве узла среды выполнения выступает web-браузер. В браузере развернут front-end приложения, а на серверной части back-end и база данных.

3. Анализ средств реализации

В качестве средств реализации приложения были выбраны следующие технологии:

— HTML, CSS, JS и сторонние библиотеки JS – web ориентированные языки HTML и CSS необходимы для разработки front-end части. JS и сторонние библиотеки дают доступ к огромному количеству инструментов, заточенных под разные задачи и упрощающие процесс разработки,

— в качестве СУБД была выбрана MySQL. Она является хорошо масштабируемой, в равной степени легко может быть использована для работы, как с малыми, так и с большими объемами данных. А за счет упрощения некоторых используемых в ней стандартов система имеет высокую производительность,

— в качестве языка разработки в back-end использовался язык PHP и библиотека Redbean.

4. Реализация

4.1 Сущности

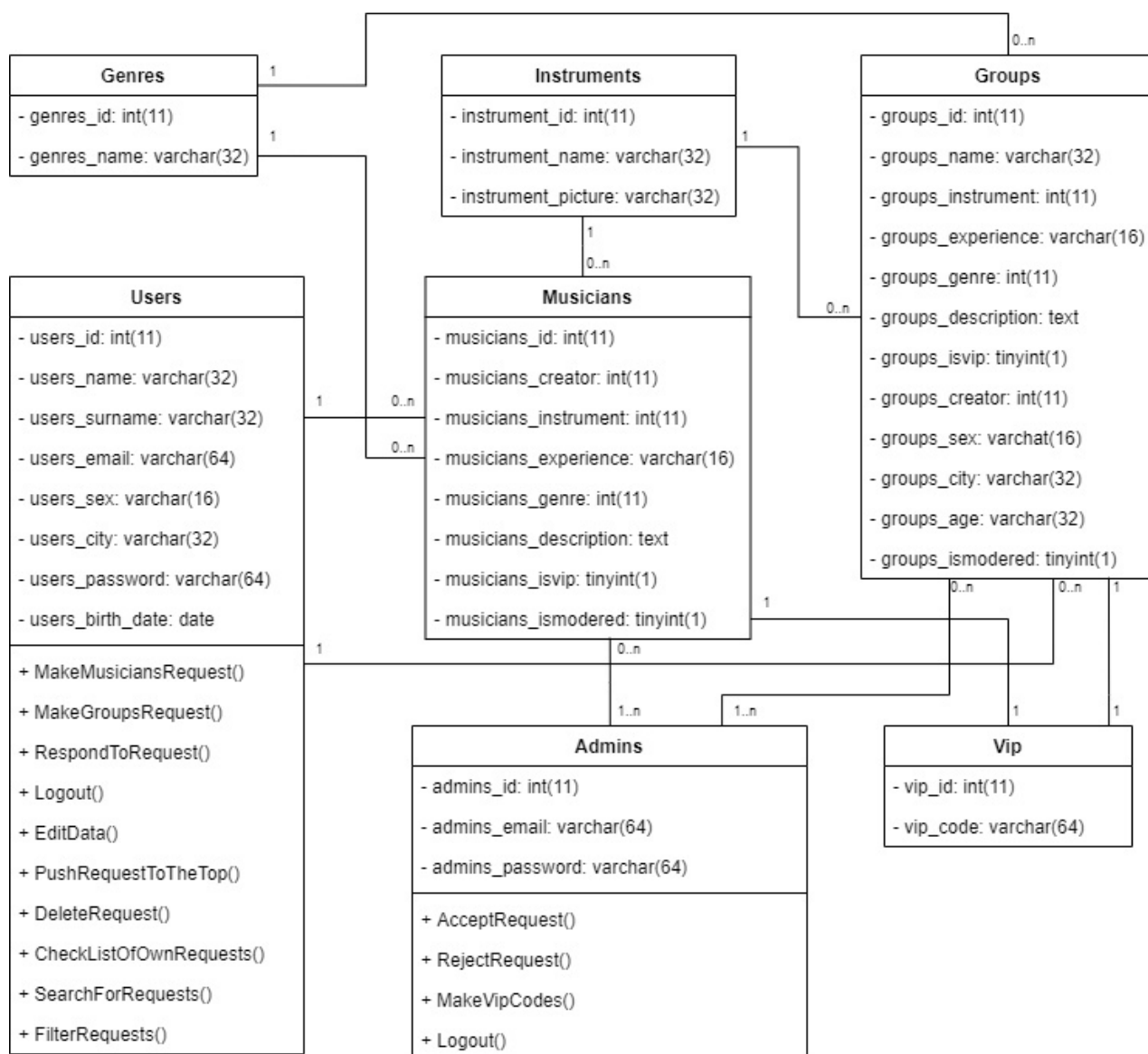


Рисунок 8 - Диаграмма классов

На Рисунке 8 изображена диаграмма классов, отражающая их отношения.

1. Класс «Genres» – представляет собой отражение сущности жанр. Класс имеет следующие свойства:

— «genres_id» – уникальный идентификатор,

— «genres_name» – название жанра.

2. Класс «Instruments» – представляет собой отражение сущности инструмент. Класс имеет следующие свойства:

— «instrument_id» – уникальный идентификатор,

— «instrument_name» – название инструмент,

— «instrument_picture» – название картинки инструмента.

3. Класс «Vip» – представляет собой отражение сущности вип-код.

Класс имеет следующие свойства:

— «vip_id» – уникальный идентификатор,

— «vip_code» – вип-код,

4. Класс «Musicians» – представляет собой отражение сущности заявка музыканта. Класс имеет следующие свойства:

— «musicians_id» – уникальный идентификатор,

— «musicians_creator» – создатель заявки музыканта (FK),

— «musicians_instrument» – выбранный инструмент в оставляемой им заявке музыканта (FK),

— «musicians_experience» – опыт игры на выбранном инструменте,

— «musicians_genre» – предпочитаемый жанр (FK),

— «musicians_description» – дополнительное описание,

— «musicians_isvip» – наличие вип-статуса заявки музыканта,

— «musicians_ismodered» – находится ли заявка в модерации.

5. Класс «Groups» – представляет собой отражение сущности заявка группы. Класс имеет следующие свойства:

— «groups_id» – уникальный идентификатор,

— «groups_name» – название группы,

— «groups_instrument» – требуемый группе исполнитель, играющий на данном инструменте(FK),

- «groups_experience» – желаемый опыт игры на выбранном инструменте,
- «groups_genre» – играемый жанр (FK),
- «groups_description» – дополнительное описание,
- «groups_creator» – создатель заявки группы (FK),
- «groups_sex» – желаемый группе исполнитель данного пола,
- «groups_city» – желаемый группе исполнитель, проживающий в данном городе,

- «groups_age» – желаемый группе исполнитель данного возраста,
- «groups_isvip» – наличие вип-статуса заявки группы,
- «groups_ismodered» – находится ли заявка в модерации.

6. Класс «Users» – представляет собой отражение сущности пользователь. Класс имеет следующие свойства:

- «users_id» – уникальный идентификатор,
- «users_name» – имя пользователя,
- «users_surname» – фамилия пользователя,
- «users_email» – email пользователя,
- «users_sex» – пол пользователя,
- «users_city» – город пользователя,
- «users_password» – пароль пользователя,
- «users_birth_date» – дата рождения пользователя.

7. Класс «Admins» – представляет собой отражение сущности администратор. Класс имеет следующие свойства:

- «admins_id» – уникальный идентификатор,
- «admins_email» – email администратора,
- «admins_password» – пароль администратора.

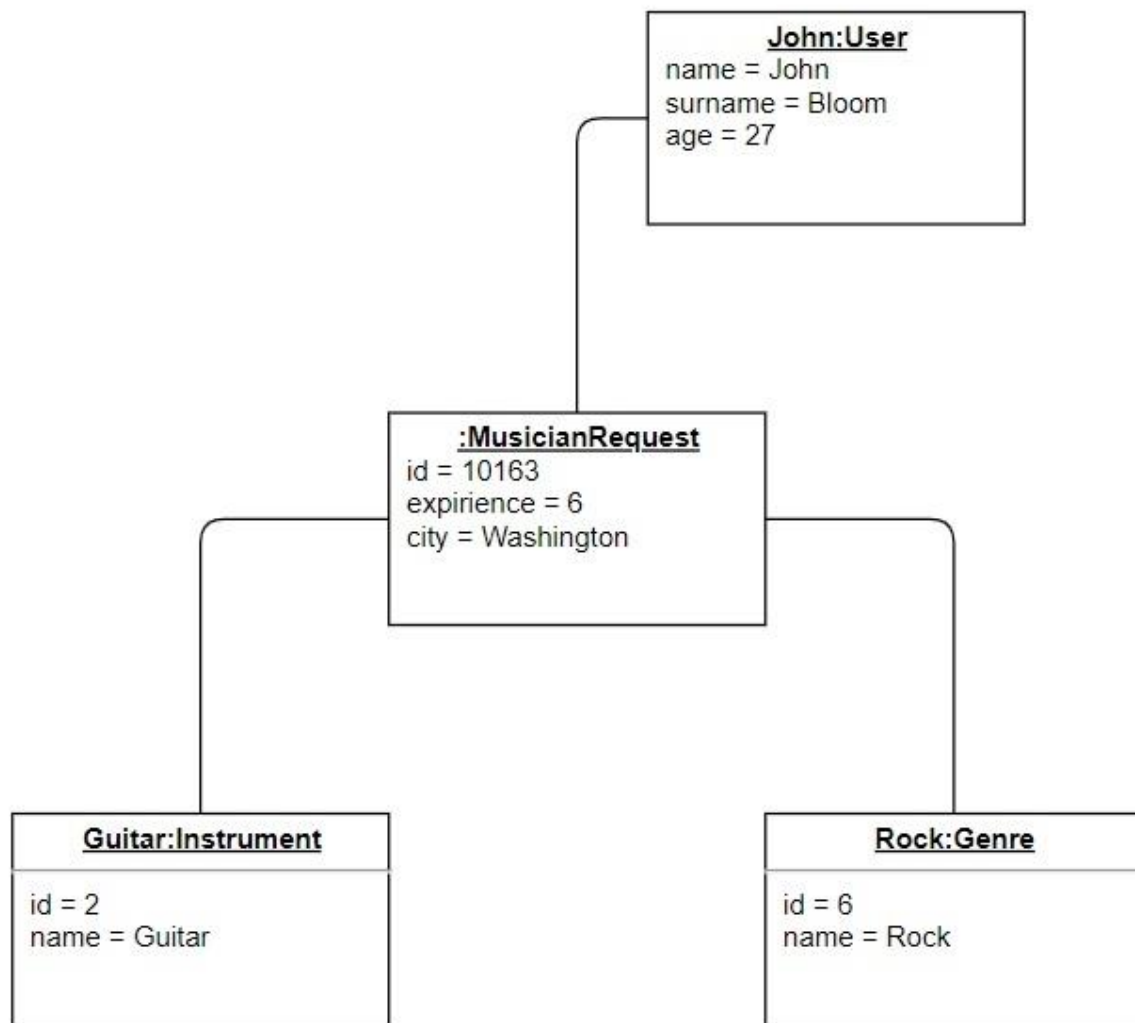


Рисунок 9 - Диаграмма объектов

На Рисунке 9 изображена диаграмма объектов, которая отражает множество экземпляров классов и отношений между ними в некоторый момент времени. На ней изображён экземпляр класса “Авторизованный пользователь”, экземпляр класса “Заявка музыканта”, которая была создана пользователем и хранит в себе ссылку на него. В свою очередь экземпляр класса “Заявка музыканта” содержит в себе поле Инструмент, являющееся экземпляром класса “Инструмент” и поле Жанр, являющееся экземпляром класса “Жанр”,

которые также отражены на диаграмме, и показана их связь с другими объектами.

4.2 Сценарии воронок конверсии

1) Посетил главную страницу - Авторизовался - Перешел на страницу создания заявки музыканта - Нажал кнопку "Добавить".

2) Посетил главную страницу - Авторизовался - Перешел на страницу создания заявки группы - Нажал кнопку "Добавить".

3) Посетил главную страницу - Авторизовался - Выбрал нужные фильтры - Нажал кнопку "Найти".

4.3 Графический интерфейс

4.3.1 Шапка сайта

На определенных страницах сайта есть «шапка», которая предоставляет следующие возможности пользователю:

- 1) перейти на страницу регистрации,
- 2) перейти на страницу авторизации,
- 3) перейти на страницу просмотра заявок музыкантов,
- 4) перейти на страницу просмотра заявок групп,
- 5) перейти на главную страницу.

Предоставляет следующие дополнительные возможности только авторизованному пользователю, кроме возможности перехода на страницы регистрации и авторизации:

- 1) перейти на страницу личного кабинета,
- 2) перейти на страницу администрирования (только для администратора),
- 3) выйти из аккаунта.

4.3.2 Главная страница

Предоставляет пользователю возможность перейти на страницу просмотра заявок музыкантов или страницу просмотра заявок групп. Главная страница изображена на рисунке 10.



Рисунок 10 – Главная страница

4.3.3 Страница регистрации

Предоставляет пользователю возможность регистрации и возможность перейти на страницу авторизации.

Страница регистрации изображена на рисунке 11.

Имя

Фамилия

Email

Пол ☐ Мужчина ☐ Женщина

Дата рождения: дд.мм.гггг

Город

Пароль

Повторите пароль

Уже есть аккунт? Тогда быстрее авторизируйся!

[ЗАРЕГИСТРИРОВАТЬСЯ](#)

Рисунок 11 – Страница регистрации

4.3.4 Страница авторизации

Предоставляет пользователю возможность авторизации и возможность перейти на страницу регистрации.

Страница авторизации изображена на рисунке 12.

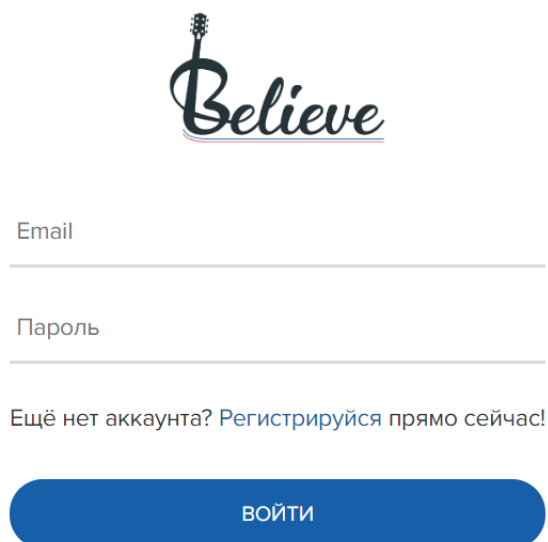


Рисунок 12 – Страница авторизации

4.3.5 Страница просмотра заявок музыкантов

Дополнительно предоставляет следующие возможности пользователю:

1) просмотреть все одобренные заявки музыкантов, с возможностью их фильтрации по предоставленным критериям.

Дополнительно предоставляет следующие возможности авторизованному пользователю:

- 1) перейти на страницу создания заявки музыканта,
- 2) перейти на страницу создания заявки группы,
- 3) откликнуться на заявку музыканта, т. е. увидеть почту оставившего данную заявку пользователя.

А также возможности «шапки» сайта, описанные в п.4.3.1.

Страница просмотра заявок музыкантов изображена на рисунке 13.

Фильтр

Инструмент ▾

Пол ▾

Опыт ▾

Жанр ▾

Город

Возраст ▾

Очистить

Найти


Оставить заявку на поиск

Поиск музыканта

Поиск группы

Подберите для себя лучшего музыканта

Стеблева Екатерина



Пол: Женский
 Возраст: 20
 Инструмент: Электрогитара
 Опыт: < 1 года
 Жанр: Рок
 Город: Воронеж

О себе

Я только начинающий гитарист, но быстро учусь!

Авторизуйтесь, чтобы ответить на заявку

Рисунок 13 – Страница просмотра заявок музыкантов

4.3.6 Страница просмотра заявок групп

Дополнительно предоставляет следующие возможности пользователю:

1) просмотреть все одобренные заявки групп, с возможностью их фильтрации по предоставленным критериям.

Дополнительно предоставляет следующие возможности авторизованному пользователю:

- 1) перейти на страницу создания заявки музыканта,
- 2) перейти на страницу создания заявки группы,
- 3) откликнуться на заявку группы, т. е. увидеть почту оставившего данную заявку пользователя.

А также возможности «шапки» сайта, описанные в п.4.3.1.


Страница просмотра заявок групп изображена на рисунке 14.

Фильтр

Оставить заявку на поиск

Подберите для себя лучшую группу

Песни-пляски



Пол: Мужской
 Возраст: 30 - 40 лет
 Инструмент: Фортепиано
 Опыт: 1 - 3 года
 Жанр: Шансон
 Город: Липецк

О группе

Ищем мужчину, играющего на фортепиано. Срочно!

Авторизуйтесь, чтобы ответить на заявку

Скрипышь



Пол: Любой
 Возраст: Любой
 Инструмент: Скрипка

О группе


Ищем скрипача в группу

Рисунок 14 – Страница просмотра заявок групп

4.3.7 Страница создания заявки музыканта

Доступна только для авторизованного пользователя и предоставляет ему возможность создания заявки музыканта. А также возможности «шапки» сайта, описанные в п.4.3.1.

Страница создания заявки музыканта изображена на рисунке 15.



Главная [Музыканты](#) [Группы](#) [Аккаунт](#) | [Выход](#)

Кажется Вы хотите найти себе группу... Не вопрос!

Заполните несколько полей и мы обязательно подберем для Вас группу

Расскажите о себе, нам очень интересно!

Хотите чтобы Вашу заявку увидели первой? Введите VIP-код

Промокод вводить сюда!

Рисунок 15 – Страница создания заявки музыканта

4.3.8 Страница создания заявки группы

Доступна только для авторизованного пользователя и предоставляет ему возможность создания заявки группы. А также возможности «шапки» сайта, описанные в п.4.3.1.

Страница создания заявки группы изображена на рисунке 16.

Believe

Главная Музыканты Группы Аккаунт | Выход

Ищете музыканта? Тогда Вы по адресу!
Заполните несколько полей и мы обязательно найдем для Вас музыканта

Название группы Город

Требуемые музыкальные данные Требуемые физические данные

Инструмент Пол

Опыт Возраст

Жанр

Расскажите о Вашей группе, нам очень интересно!

Хотите чтобы Вашу заявку увидели первой? Введите VIP-код!

Промокод вводить сюда!

Добавить

Рисунок 16 – Страница создания заявки группы

4.3.9 Страница личного кабинета

Доступна только для авторизованного пользователя и предоставляет следующие возможности авторизованному пользователю:

- 1) просмотреть все собственные заявки с возможностью их удаления,
- 2) перейти на страницу редактирования личного кабинета.

А также возможности «шапки» сайта, описанные в п.4.3.1.

Страница личного кабинет изображена на рисунке 17.

Личный кабинет

Моя анкета

Имя	Михаил
Фамилия	Дынин
Email	dynin.michael21@gmail.com
Пол	Мужской
Дата рождения	23.08.1999
Город	Воронеж
Редактировать	

Мои заявки

Заявка №1

Одобрена

Поиск музыканта

[Подробнее](#)

Имя: Михаил
Опыт: > 5 лет
Инструмент: Барабаны
Жанр: Рок

О себе

Играю на барабанах с детства, ищу группу!

[Промокод вводить сюда!](#) [В топ](#)

[Удалить](#)

Заявка №2

Одобрена

Поиск группы

[Подробнее](#)

[Удалить](#)

Рисунок 17 – Страница личного кабинета

4.3.10 Страница редактирования личного кабинета

Доступна только для авторизованного пользователя и предоставляет ему возможности редактировать личные данные. А также возможности «шапки» сайта, описанные в п.4.3.1. Страница редактирования личного кабинета изображена на рисунке 18.

Редактирование данных личного кабинета

Имя	Иван
Фамилия	Петров
Email	petrov@mail.ru
Пол:	<input checked="" type="radio"/> Мужчина <input type="radio"/> Женщина
Дата рождения	01.09.1986
Город	Воронеж
<input type="button" value="Сохранить"/>	

Рисунок 18 – Страница редактирования личного кабинета

4.3.11 Страница администрирования

Доступна только для администратора и предоставляет ему возможности:

- 1) сгенерировать vip-коды в определенном количестве,
- 2) одобрить или отклонить находящиеся на рассмотрении (модерации) заявки,
- 3) перейти на главную страницу.

Страница администрирования изображена на рисунке 19.



Панель администратора

Сколько VIP-кодов сгенерировать?

10

Сгенерировать

Саввин Максим

Пол: Мужской
Возраст: 21
Инструмент: Флейта
Опыт: < 1 года
Жанр: Джаз
Город: Нововоронеж

О себе

Одобрить

Отклонить

Некорректные данные
Некорректные данные
Нецензурные выражения
Оскорбления
Спам

Саввин Максим

Пол: Мужской
Возраст: 21
Инструмент: Акустическая гитара
Опыт: 1 - 3 года
Жанр: Блюз
Город: Нововоронеж

О себе
вфыв

Одобрить

Отклонить

Некорректные данные

Рисунок 19 – Страница администрирования

5. Тестирование

После реализации всех задач, был проведен запланированный набор тестов. Он включает 3 вида тестирования:

- дымовое тестирование,
- UI тесты,
- юзабилити тесты

5.1 Дымовое тестирование

Для данного тестирования необходимо было проверить работоспособность сайта на следующих основных сценариях:

- создание заявки музыканта,
- создание заявки группы,
- удаление заявки,
- регистрация,
- авторизация,
- одобрение/отклонение заявки администратором,
- редактирование информации в личном кабинете,
- генерация vip-кодов,
- фильтрация заявок.

Дымовое тестирование проводилось ручным способом, в следующих браузерах: Mozilla Firefox, Google Chrome, Yandex Browser, Opera с включенным WI-FI для связи с back-end частью сайта. Результаты, полученные в ходе тестирования представлены в таблице 1.

Таблица 1. Результаты дымового тестирования.

Сценарий	Результат
Создание заявки музыканта	Пройден
Создание заявки группы	Пройден

Удаление заявок	Пройден
Регистрация	Пройден
Авторизация	Пройден
Одобрение/отклонение заявки администратором	Пройден
Отправка уведомления о решении по электронной почте	Пройден
Редактирование информации в личном кабинете	Пройден
Фильтрация заявок	Пройден

По итогу дымового тестирования было установлено, что сайт проходит все основные утвержденные сценарии.

5.2 UI тесты

В результате UI тестирования было выполнено 27 тестов, охватывающие основные возможности сайта.

Таблица 2. Результаты UI тестирования.

Шаги теста	Ожидаемый результат	Статус
1. Нажимается изображение «Музыканты» на главной странице	Открылась страница со списком заявок музыкантов	Пройден
1. Нажимается изображение «Группы» на главной странице	Открылась страница со списком заявок групп	Пройден
1. Нажимается кнопка «Главная»	Открылась вкладка главной страницы	Пройден

1. Нажимается кнопка «Музыканты»	Открылась страница со списком заявок музыкантов	Пройден
1. Нажимается кнопка «Группы»	Открылась страница со списком заявок групп	Пройден
1. Нажимается кнопка «Регистрация» 2. В поля вводятся корректные данные 3. Нажимается кнопка «Зарегистрироваться»	1. Открылась страница с регистрацией 2. Открылась страница личного кабинета	Пройден
1. Нажимается кнопка «Регистрация» 2. В одно или несколько полей вводятся некорректные данные или не во все поля вводятся данные 3. Нажимается кнопка «Зарегистрироваться»	1. Открылась страница с регистрацией 2. Появилось сообщение о том, что необходимо ввести поле	Пройден
1. Нажимается кнопка «Вход» 2. В поля вводятся корректные данные 3. Нажимается кнопка «Войти»	1. Открылась страница с авторизацией 2. Открылась страница личного кабинета	Пройден
1. Нажимается кнопка «Вход» 2. В поля вводятся	1. Открылась страница с авторизацией 2. Появилось сообщение	Пройден

<p>некорректные данные</p> <p>3. Нажимается кнопка «Войти»</p>	<p>о том, что пользователя с такими данными не существует</p>	
<p>1. Нажимается кнопка «Вход»</p> <p>2. Не во все поля вводятся данные</p> <p>3. Нажимается кнопка «Войти»</p>	<p>1. Открылась страница с авторизацией</p> <p>2. Появилось сообщение о том, что необходимо ввести поле</p>	Пройден
<p>1. Нажимается кнопка «Аккаунт»</p>	<p>Открылась страница личного кабинета</p>	Пройден
<p>1. Нажимается кнопка «Выход»</p>	<p>Открылась страница главной страницы</p>	Пройден
<p>1. Нажимается кнопка «Редактировать»</p> <p>2. В поля вводятся корректные данные</p> <p>3. Нажимается кнопка «Сохранить»</p>	<p>1. Открылась страница с редактированием данных</p> <p>2. Открылась страница личного кабинета с сохраненными измененными данными</p>	Пройден
<p>1. Нажимается кнопка «Редактировать»</p> <p>2. В одно или несколько полей вводятся некорректные данные или не во все поля вводятся данные</p> <p>3. Нажимается кнопка «Сохранить»</p>	<p>1. Открылась страница с редактированием данных</p> <p>2. Появилось сообщение о том, что необходимо ввести поле</p>	Пройден

1. Нажимается кнопка «Откликнуться»	Выводится почта создавшего заявку человека	Пройден
1. Нажимается кнопка «Авторизуйтесь, чтобы ответить на заявку»	Открылась страница с авторизацией	Пройден
1. Нажимается кнопка «Подробнее»	Выводится подробная информация о заявке	Пройден
1. Нажимается кнопка «Удалить заявку»	Страница перегружается, заявка пропадает из списка «Мои заявки»	Пройден
1. Выбираются поля фильтрации и/или вводится город. 2. Нажимается кнопка «Найти»	Страница перегружается, выводится список заявок по выбранным критериям	Пройден
1. Нажимается кнопка «Сгенерировать»	Загружается текстовый файл со сгенерированными vip-кодами	Пройден
1. Нажимается кнопка «Админ»	Открылась страница с личным кабинетом администратора	Пройден
1. В поле «Промокод вводить сюда!» вводится недействительный vip-код.	Появилось сообщение о том, что промокод недействителен	Пройден
1. Нажимается кнопка «Одобрить»	Страница перегружается, заявка пропадает из списка	Пройден

1. Нажимается кнопка «Отклонить»	Страница перегружается, заявка пропадает из списка	Пройден
1. Нажимается кнопка «Поиск музыканта» 2. В поля вводятся корректные данные 3. Нажимается кнопка «Добавить»	Открылась страница с заявками музыкантов	Пройден
1. Нажимается кнопка «Поиск группы» 2. В поля вводятся корректные данные 3. Нажимается кнопка «Добавить»	Открылась страница с заявками групп	Пройден
1. Нажимается кнопка «Поиск музыканта» 2. В одно или несколько полей вводятся некорректные данные или не во все обязательные поля вводятся данные 3. Нажимается кнопка «Добавить»	1. Открылась страница с заявками музыкантов 2. Появилось сообщение о том, что необходимо ввести поле	Пройден
1. Нажимается кнопка «Поиск группы» 2. В одно или несколько полей вводятся некорректные данные или не во все обязательные поля вводятся данные	1. Открылась страница с заявками групп 2. Появилось сообщение о том, что необходимо ввести поле	Пройден

3. Нажимается кнопка «Добавить»		
---------------------------------	--	--

5.3 Юзабилити тесты

Для проведения юзабилити тестирования было случайно отобрано 3 человека, не пользовавшиеся заранее сайтом. Для данного тестирования необходимо проверить следующие основные сценарии взаимодействия пользователя с сайтом:

- регистрация,
- авторизация,
- создание заявки музыканта,
- создание заявки группы,
- удаление заявки,
- редактирование информации в личном кабинете,
- фильтрация заявок,
- отклик на заявку,
- подробный просмотр своих заявок,
- выход из аккаунта.

Результаты тестирования отображены в таблице 3.

Сценарий	Пользователь 1	Пользователь 2	Пользователь 3
Регистрация	Пройден	Пройден	Пройден
Авторизация	Пройден	Пройден	Пройден
Создание заявки музыканта	Пройден	Пройден	Пройден
Создание заявки группы	Пройден	Пройден	Пройден

Удаление заявки	Пройден	Пройден	Пройден
Редактирование информации в личном кабинете	Пройден	Пройден	Пройден
Фильтрация заявок	Пройден	Пройден	Пройден
Отклик на заявку	Пройден	Пройден	Пройден
Подробный просмотр своих заявок	Пройден	Пройден	Пройден
Выход из аккаунта	Пройден	Пройден	Пройден

Заключение

В результате работы был разработан сайт по поиску музыкантов и групп. Были выполнены следующие задачи:

- 1) разработана Front-end часть сайта, находящаяся на компьютере/телефоне пользователя,
- 2) разработана Back-end часть сайта, развернутая на удаленном сервере,
- 3) была создана связь между Front-end и Back-end частями приложения,
- 4) разработана база данных, расположенная на удаленном сервере.

Приложение отвечает всем заявленным требованиям.