

Департамент образования и науки Курганской области
Государственное бюджетное профессиональное образовательное учреждение
«Курганский педагогический колледж»

Защищена с оценкой _____

Протокол № ____ от _____

**Разработка интернет–магазина по продаже
фирменных аранжировок**

Дипломная работа

Студент группы № 43 _____ /Пухов А.А./

Руководитель _____ /Ревняков Е.Н./

Специальность: 09.02.07. Информационные системы и программирование

Руководитель
методического
объединения _____ /Тютрина М.М./

Курган 2022

Содержание

Введение	4
1. Анализ процессов предметной области	6
1.1 Описание предметной области	6
1.2. Общие требования к системе в целом.....	7
1.3. Основные функции информационной системы	7
1.4. Общие требования к программному продукту	8
1.5 Общее назначение	8
1.6 Требования к внешнему интерфейсу	8
1.7 Другие нефункциональные требования.....	9
1.7.1 Требования к безопасности	9
2. Проектная часть.....	10
2.1. Постановка задачи.....	10
2.2 Образ проекта	10
2.3 Концептуальная модель проектируемой информационной системы..	12
2.3.1 Базовые варианты использования	12
2.3.2 Диаграммы, описывающие поведение системы	14
2.3.3 Модель жизненного цикла информационной системы.....	14
2.4. Архитектура системы.	15
2.4.1 Базовые понятия	15
2.4.2 Диаграмма классов проектируемой Информационной системы ..	15
2.4.3 Информационные объекты и их атрибуты	17
2.4.4 Связи между объектами.....	17
2.4.5 Ограничения на характеристики объектов	17
2.4.6 Отображение концептуальной схемы в логическую схему.....	18
2.4.7 Выбор ключей.....	21
2.4.8 Создание базы данных и таблиц.....	21
2.5 Методы и средства разработки	21
2.5.1 Выбор платформы	21
2.5.2 Методы и инструменты проектирования	21
2.5.3 Выбор вспомогательных инструментов для работы с фреймворком	22

3. Описание информационной системы.....	26
3.1 Реализация программных модулей приложения	26
Заключение.....	33
Список использованных источников	35

ПРИЛОЖЕНИЕ 1

ПРИЛОЖЕНИЕ 2

ПРИЛОЖЕНИЕ 3

ПРИЛОЖЕНИЕ 4

ПРИЛОЖЕНИЕ 5

ПРИЛОЖЕНИЕ 6

Введение

В современном мире большой популярностью пользуются музыкальные магазины с готовыми аранжировками известных песен. Для обеспечения оперативности работы с клиентами необходима автоматизированная информационная система, основанная на современной базе данных. Использование базы данных и автоматизированной системы существенно сократит время обслуживания клиентов. В магазине необходимо хранить разнообразную информацию об аранжировках, чтобы оперативно можно было определить информацию о наличии в магазине некоторой аранжировки по определенному названию, исполнителю.

Основная цель информационной системы — организация хранения, обработки и передачи информации. Современные информационные технологии, основанные на использовании средств вычислительной техники и связи, нашли широкое применение и в музыкальной деятельности.

Актуальность данной темы продиктовано потребителями музыкальной продукции на интернет – ресурсов. Конечно, сейчас в интернете очень большое количество минусовок различного качества, но бывают такие ситуации, что именно той которая нужно в данный момент и не оказалось. Или не подходит формат аранжировки или нужна старая песня, но в современном тренде, или не устраивает качество найденной в просторах интернета минусовой фонограммы. Вот для этого и был создана Информационная Система, связывающая потребителя с музыкантом.

Объектом исследования является организация по продаже аранжировок.

Предметом исследования является предоставление информационных услуг для осуществления продаж и покупок аранжировок с использованием проектируемой информационной системы.

Цель дипломной работы – разработка информационной системы «SongPay».

Создание информационной системы позволит оказывать более качественные услуги клиентам, а также минимизировать затраты на обслуживания.

Задачи исследования:

1. изучить и применить на практике технические аспекты разработки информационной системы;
2. провести анализ и выбрать интернет – технологии для разработки информационной системы;
3. разработать структуру информационной системы;
4. разработать информационную систему с помощью фреймворка Django, и библиотеки ReactJS.

Сроки выполнения работы: сентябрь 2021 по июнь 2022 г.

Методы исследования – моделирование на языке UML, проектирование базы данных (отрисовка диаграмм), проектирование информационной системы, разработка информационной системы с помощью фреймворка Django и библиотеки ReactJS.

Выпускная квалификационная работа состоит из введения, двух разделов, заключения и приложений.

По срокам выполнения работ приводится диаграмма Ганта (рис 1).

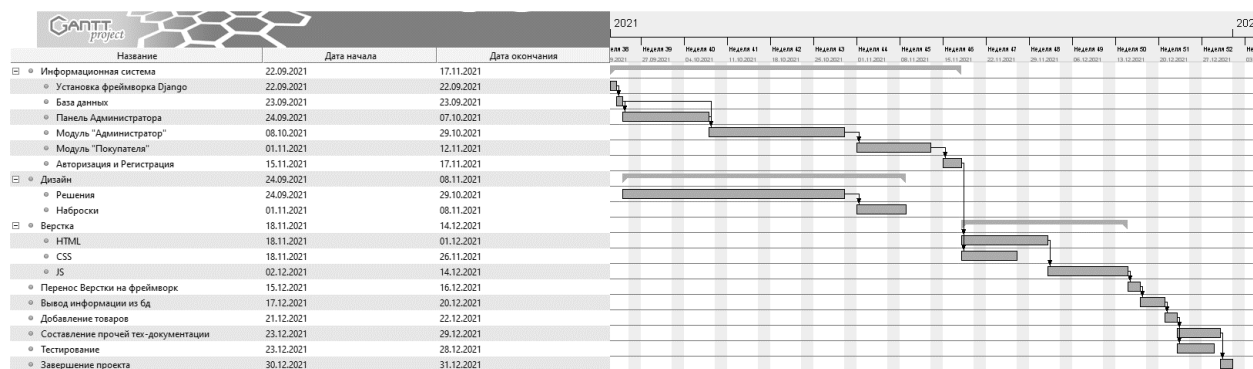


Рис. 1. Диаграмма Ганта по срокам реализации

1. Анализ процессов предметной области

1.1 Описание предметной области

Музыкальный магазин – это торговая площадка, где люди могут приобретать оригинальные и авторские аранжировки песен популярных исполнителей. Основным направлением работы любого магазина является: обслуживание клиентов.

Правильная организация фонда магазина облегчает клиенту пользование услугами. Обслуживание клиентов магазина осуществляется различными путями.

Клиент оформляет заказ по электронной почте у представителя магазина (консультанта) на определенный продукт, производит оплату.

В настоящее время такой подход несколько устарел, поэтому было принято решение создать информационную систему (ИС). Создание ИС также поможет магазину выйти на мировой уровень продаж.

Ниже приводится диаграмма (рис. 2) процесса покупки аранжировки без информационной системы.



Рис. 2. Диаграмма, отображающая процесс покупки аранжировки, без использования Информационной системы (AS-IS)

Как видно из диаграммы (Рисунок 2), такие продажи аранжировок имеют ряд ограничений и недостатков. Улучшить ситуацию можно, создав информационную систему с веб-интерфейсом.

Клиент сможет зарегистрироваться на сайте и выбирать товары. Администратор без проблем сможет просматривать и изменять статус заказа.

1.2. Общие требования к системе в целом

Информационная система музыкального магазина «SongPay» будет использоваться покупателями, а также сотрудниками магазина, в том числе, сотрудниками руководящего звена магазина.

Информационная система должна иметь веб-интерфейс. Необходимо, чтобы Информационная система была централизованной, то есть все данные располагались в центральном хранилище, использовать протокол TCP/IP в качестве протокола взаимодействия между компонентами Информационной системы на транспортно-сетевом уровне, использовать HTTPS (с использованием сертификата, к примеру, Let's Encrypt) для организации доступа пользователей к Информационной системе.

Информационный обмен между компонентами Системы должен осуществляться в синхронном взаимодействии, что обеспечивает оперативную передачу данных.

Информационная система должна включать в себя следующие прикладные функциональные компоненты (функциональные подсистемы и/или модули):

- модуль «Администратор системы»;
- модуль «Покупатель»;

1.3. Основные функции информационной системы

- 1) учёт музыкальных изданий;
- 2) просмотр информации о продаваемых аранжировках;
- 3) просмотр и изменение информации о заказанных аранжировках;
- 4) хранение информации о аранжировках, пользователях и заказах;

Для отдельных модулей Системы должны выполняться следующие функции:

Для модуля «Покупатель»:

1) возможность просмотра каталога (в т.ч. незарегистрированным пользователям);

2) оформление заказа (зарегистрированным пользователям).

Для модуля «Администратор системы»:

1) возможность удаленного доступа;

2) Управление продажами и каталогами;

3) Учет товара.

Для **всех** пользователей Информационной системы:

1) возможность регистрации;

2) возможность авторизации.

1.4. Общие требования к программному продукту

- простота пользования системой;
- удобный понятный интерфейс;
- защита информации в системе;
- возможность дополнять систему новыми функциями;
- приятный дизайн.

1.5 Общее назначение

Предметной областью разработки является предоставление информационных услуг для продажи товаров через Интернет. Областью внедрения разработки может быть музыкальный лейбл.

1.6 Требования к внешнему интерфейсу

Информационная система будет обладать веб-интерфейсом — это организованная рабочая зона с подключением к удаленному компьютеру (серверу), которая предоставляет пользователю доступ к информации в структурированном

виде. Помимо получения информации в целевых разделах интерфейса, пользователю предоставляется возможность совершать какие-либо действия для управления данными. Веб-интерфейс Информационной системы создается с целью:

- 1) удобства клиентов;
- 2) повышения уровня продаж аранжировок;
- 3) повышения работоспособности персонала.

В данном веб-интерфейсе клиент (пользователь) получит возможность:

- 1) просматривать музыкальные подборки;
- 2) совершать заказы (круглосуточно);
- 3) просматривать информацию о аранжировках (категория, цена, и т.д.)

Информационная система для музыкального магазина «SongPay» должна быть кроссплатформенной, что позволит клиенту (пользователю) иметь доступ к сайту с любого устройства (с любой операционной системой) и кросс браузерной, что позволит отображать страницы одинаково в любом браузере.

1.7 Другие нефункциональные требования

1.7.1 Требования к безопасности

Должна быть возможность регистрация пользователей в системе;

Обязательная авторизация пользователей в системе при заказе товара;

Пароль не должен храниться в базе данных в открытом виде, он должен быть изменен с помощью хеширования. Требования, которым должен удовлетворять хэш: стойкость к атакам перебора (прямой перебор и перебор по словарю), невозможность поиска одинаковых паролей разных пользователей по хешам.

Поля форм должны быть провалидированы для защиты от SQL-инъекций и прочих возможных атак на информационную систему, для формы регистрации должно быть поле повтора пароля.

Администратор должен иметь доступ ко всей ИС, может управлять доступом других пользователей, наделять их правами, а также должен иметь возможность вносить и редактировать информацию о аранжировках, заказах, и т.д.

2. Проектная часть

2.1. Постановка задачи

Цель работы – проектирование информационной системы для музыкального магазина «SongPay».

Задачи.

ИС «SongPay» должна обеспечивать возможность долговременного хранения данных. В Системе должно осуществляться соединение с сервером. ИС «SongPay» должна обеспечивать возможность многопользовательской работы.

В создаваемой Информационной системе предлагается выделить следующие подлежащие автоматизации задачи:

- 1) прием и учет музыкальных изданий;
- 2) просмотр и изменение информации о заказанных аранжировках;
- 3) ведение архива товаров.

2.2 Образ проекта

Ниже приводится таблица ролей пользователей ИС «SongPay»

Таблица 1

Роли пользователей в ИС «SongPay»

Роль в системе	Лица (кто конкретно)	Права и возможности
Покупатель	Покупатель	Совершать покупки на сайте магазина, просмотр товара.
Администратор/Продавец (admin)	Администратор ИС Продавец магазина	Управление правами доступа, расширение Информационной системы Добавлять сведения о новых поступлениях и т.д.

На рисунке 3 представлена диаграмма базового варианта использования для информационной системы «SongPay»

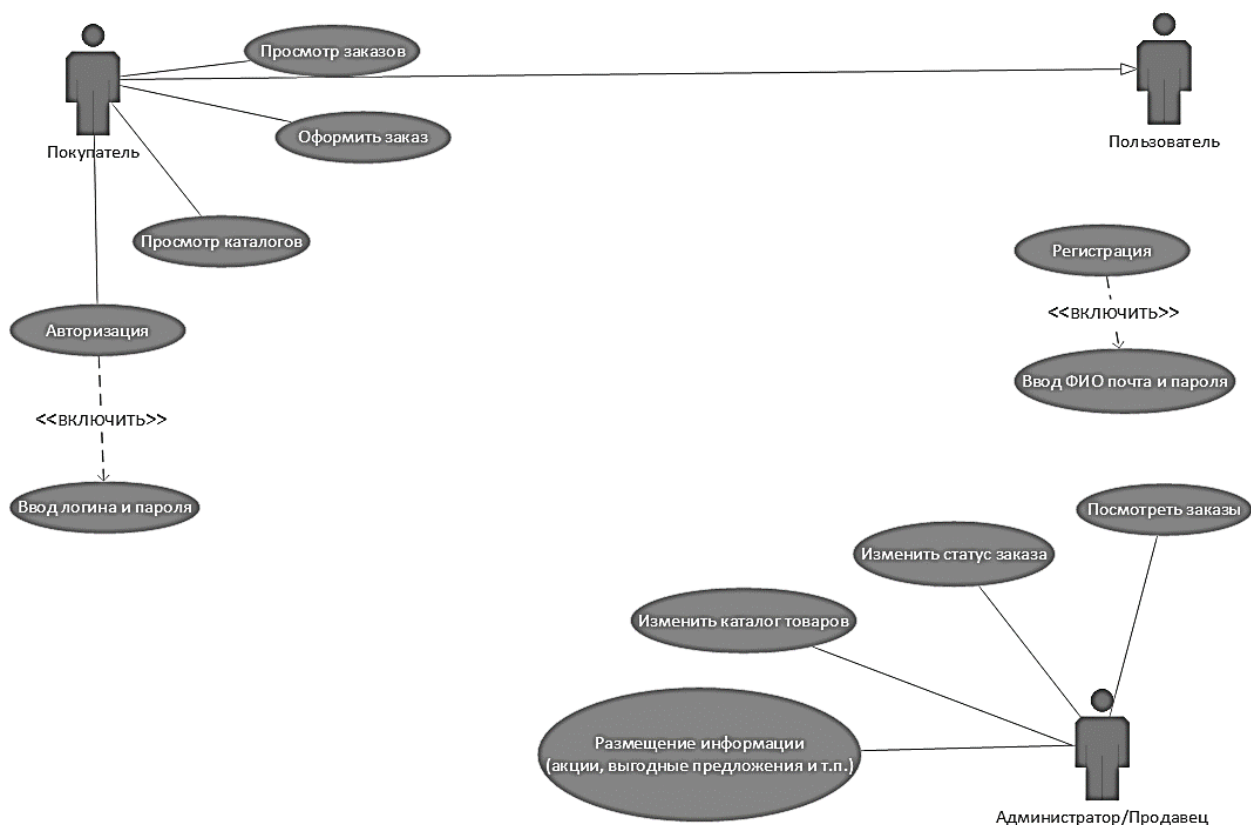


Рис. 3. Базовые варианты использования для ИС «SongPay» (Диаграмма To-Be).

В создаваемой информационной системе предлагается выделить следующие функции для **всех** пользователей системы:

- заполнение формы регистрации;
- отправка формы регистрации;
- заполнение формы авторизации;
- отправка формы авторизации;
- заполнение формы обратной связи;
- отправка формы обратной связи;

В создаваемой Информационной системе предлагается выделить следующие функции для следующих пользователей Системы:

Администратор системы:

- 1) изменение статуса заказа;
- 2) изменение каталога товаров;

- 3) размещение информации (акции и т.д.);
- 4) просмотр заказов.
- 5) добавление элемента авторизации;
- 6) изменение элемента авторизации;
- 7) удаление элемента авторизации;
- 8) чтение элемента авторизации;
- 9) добавление роли;
- 10) удаление роли;
- 11) наделение правами.

Пользователь:

- 1) просмотр каталога;
- 2) просмотр заказов (корзины);
- 3) оформление заказа;
- 4) отмена заказа;
- 5) добавление товара в корзину;

2.3 Концептуальная модель проектируемой информационной системы

2.3.1 Базовые варианты использования

Подробное описание некоторых базовых вариантов использования ИС «SongPay» с описанием действий пользователя и откликов Системы приведен в таблицах 2-7.

Таблица 2

«Добавление товара в корзину»

Действие актёров	Отклик системы
1. Пользователь заходит на сайт	2. Вывод товара
3. Ищет конкретный товар Исключение №1 такого товара нет в магазине	

Продолжение таблицы 2

4. Выбирает понравившийся товар Исключение №2 такого товара нет в наличии	5. Вывод информации о товаре
6. Добавляет товар в корзину	7. Сохранение товара в корзине

Таблица 3

Раздел «Обработка исключений»

Действия актеров	Отклик системы
Исключение №1 такого товара нет в магазине	Переход к п.2 «Вывод товара»
Исключение №2 такой аранжировки нет в магазине	Переход к п.2 «Вывод товара»

Таблица 4

Раздел «Авторизация» на сайте

Действия актеров	Отклик системы
1. Пользователь заходит на сайт	2. Вывод окна авторизации
3. Пользователь вводит логин Исключение №1 Пользователь ошибся в написании логина	
4. Пользователь вводит пароль Исключение №2 Пользователь ошибся в написании пароля	5. Проверка правильности введенных данных
6. Авторизация выполнена успешно Исключение №3 Пользователь не смог авторизоваться	7. Вход на сайт

Таблица 5

Раздел «Обработка исключений»

Действия актеров	Отклик системы
Исключение №1 Пользователь ошибка в написании логина	Предупреждение о неправильном вводе логина
Исключение №2 Пользователь ошибся в написании пароля	Предупреждение о неправильном вводе пароля
Исключение №3 Пользователь не смог авторизоваться	Переход к п.2 «Вывод окна авторизации»

Таблица 6

Раздел «Регистрация» на сайте

Действия актеров	Отклик системы
1. Пользователь заходит на сайт	2. Вывод окна регистрации
3. Пользователь придумывает логин Исключение №1 Такой логин уже используется	
4. Пользователь придумывает пароль Исключение №2 Пароль недостаточно надежен	5. Проверка правильности введенных данных
6. Регистрация выполнена успешно Исключение №3 Пользователь не смог зарегистрироваться	7. Вход на сайт

Таблица 7

Раздел «Обработка исключений»

Действия актеров	Отклик системы
Исключение №1 Такой логин уже используется	Вывод подсказки о других возможных вариантах написания логина
Исключение №2 пароль недостаточно надежен	Вывод подсказки: «Пароль должен содержать не менее 8-ми символов, а также включать в себя большие буквы»
Исключение №3 Пользователь не смог авторизоваться	Переход к п.2 «Вывод окна регистрации»

2.3.2 Диаграммы, описывающие поведение системы

Для описания поведений системы используются диаграммы деятельности и последовательности для некоторых базовых вариантов использования.

Процесс Регистрации пользователя в ИС «SongPay» /ПРИЛОЖЕНИЕ 5/.

Процесс Авторизации пользователя в ИС «SongPay» /ПРИЛОЖЕНИЕ 6/.

2.3.3 Модель жизненного цикла информационной системы

Выбрана спиральная модель жизненного цикла, в дальнейшем планируется развивать систему, дополняя ее дополнительными функциями. Это будет реализоваться в следующих версиях информационной системы.

2.4. Архитектура системы.

2.4.1 Базовые понятия

Ниже приведено описание базовых понятий, которые встречаются в проекте.

Аранжировка в ИС «SongPay» — это музыкальное сопровождение песни, доработка и сочинение новых партий инструментов, а также определение жанра будущей композиции.

Жанр (музыкальный жанр) в ИС «SongPay» – это род музыки, музыкальных произведений, характеризующийся определёнными сюжетными, композиционными, стилистическими и др. признаками; а также отдельные разновидности этого рода.

Тональность в ИС «SongPay» — это закрепление положения музыкального лада за определёнными по высоте звучания музыкальными тонами, привязка к конкретному участку музыкального звукоряда.

2.4.2 Диаграмма классов проектируемой Информационной системы

Ниже приводится диаграмма классов проектируемой Информационной системы. Данная диаграмма будет использована в качестве модели для создания базы данных Информационной системы. Классы, представленные на диаграмме, являются сущностями проектируемой базы данных (Рисунок 4).

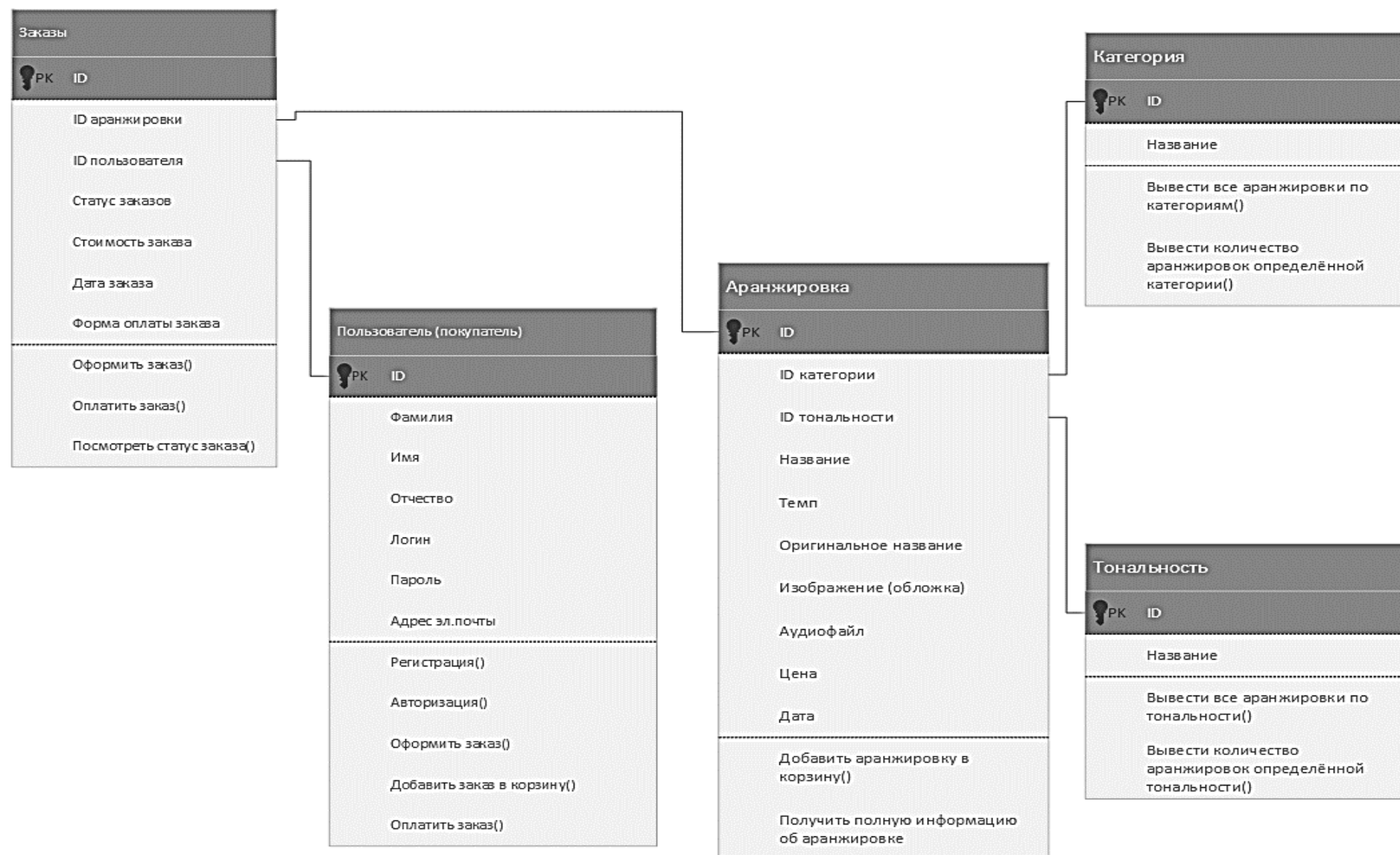


Рис. 4. Первоначальный макет диаграммы классов для проектируемой Информационной системы «SongPay»

2.4.3 Информационные объекты и их атрибуты

Ниже описаны основные объекты (сущности) проектируемой информационной системы. Для каждой сущности задан набор атрибутов (ключевые атрибуты подчеркнуты):

1. **Аранжировка** (ID, код категории, код тональности, название, темп, оригинальное название, изображение(обложка), аудиофайл, цена, дата).
2. **Заказ** (ID, код аранжировки, код пользователя, статус заказа, стоимость заказа, дата заказа, форма оплаты заказа).
3. **Пользователь** (ID, фамилия, имя, отчество, логин, пароль, номер телефона, email-адрес).
4. **Категория** (ID, название).
5. **Тональность** (ID, название).

2.4.4 Связи между объектами

Ниже описаны связи между объектами:

Аранжировка, пользователь (покупатель) – слушает, заказывает.

Аранжировка, продавец – продает, размещает.

Аранжировка, автор – пишет.

Пользователь, заказ – оформляет, получает, отменяет.

Продавец, заказ – проверяет, отменяет.

2.4.5 Ограничения на характеристики объектов

Ниже приводятся некоторые ограничения на характеристики объектов в Информационной системе:

- 1) Одному пользователю может быть присвоен только один номер заказа.
- 2) Один заказ может относиться к разным аранжировкам.

2.4.6 Отображение концептуальной схемы в логическую схему

Ниже в таблицах показан результат отображения полученной информационной схемы на реляционную базу данных, поддерживаемую СУБД MySQL. Каждая сущность описана в виде отношения. Первичные ключи помечены знаком #.

Таблица 8

Аранжировка (arrangement)

<u>Код аранжировки</u>	Код категории	Код тональности	Название	Оригинальное название	Изображение (обложка)	Аудиофайл
# Счётчик	Числовое, длинное целое	Числовое, длинное целое	Текстовое поле, 30 символов	Текстовое поле, 40 символов	Объект OLE	Объект OLE

Продолжение таблицы 8

Аранжировка (arrangement)

Цена	Дата
Денежный	Дата/время

Таблица 9

Заказ (order)

<u>Код заказа</u>	Код аранжировки	Код пользователя	Статус заказа	Дата заказа	Форма оплаты заказа
# Счётчик	Числовое, длинное целое	Числовое, длинное целое	Текстовое поле, 15 символов	Дата/Время	Текстовое поле, 15 символов

Таблица 10

Пользователь (user)

<u>Код пользователя</u>	Фамилия	Имя	Отчество	Логин	Адрес Эл. почты
# Счётчик	Текстовое поле, 30 символов	Текстовое поле, 30 символов	Текстовое поле, 30 символов	Текстовое поле, 30 символов	Текстовое поле, 30 символов

Таблица 11

Категория (category)

<u>Код категории</u>	Название
# Счётчик	Текстовое поле, 30 символов

Таблица 12

Тональность (key)

<u>Код тональности</u>	Название
# Счётчик	Текстовое поле, 30 символов

2.4.7 Выбор ключей

Определены первичные и вторичные ключи каждого отношения, а также связи между таблицами. Первичные ключи родительских таблиц связаны с вторичными ключами дочерних таблиц. Так как почти со всеми таблицами базы данных будут работать пользователи через созданные для этого интерфейсы, почти все первичные ключи определены как автоинкрементные.

2.4.8 Создание базы данных и таблиц

База данных и таблицы будут созданы с помощью ORM-таблиц. В будущем планируется использовать PostgreSQL.

2.5 Методы и средства разработки

2.5.1 Выбор платформы

В качестве серверной платформы выбран программный python-сервер Gunicorn и Web Server, так как он является одним из самых проверенных решений, способных работать с операционной системой Linux.

2.5.2 Методы и инструменты проектирования

Для проектирования информационной системы был выбран объектно-ориентированный метод. Проектируемая информационная система была смоделирована и описана в рамках стандарта «UML», который в настоящий момент является «стандартом номер один» в мире при разработке объектно-ориентированных систем.

Программы, используемые для проектирования ИС «SongPay» – «Microsoft Visio», «Díа».

Для разработки информационной системы был выбран язык «Python».

Одна из самых привлекательных черт языка Python — невероятная простота использования в веб-проектах. Практический характер Python обусловлен четырьмя важными характеристиками:

- 1) Простота использования
- 2) Легкость изучения
- 3) Легкость чтения
- 4) Несравненная гибкость

«Django» — свободный фреймворк для веб-приложений на языке «Python», использующий шаблон проектирования MVC. Проект поддерживается организацией «Django Software Foundation».

Django содержит огромное количество функционала для решения большинства задач веб-разработки. Вот некоторые из высокоуровневых возможностей Django:

1. ORM
2. Миграции базы данных
3. Аутентификация пользователя
4. Панель администратора
5. Формы
6. Django REST Framework для API

2.5.3 Выбор вспомогательных инструментов для работы с фреймворком

Для создания Информационной системы будет использоваться интегрированная среда разработки «PyCharm».

При разработке использовалась библиотека ReactJS. Это JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов.

ReactJS разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций. React может использоваться для разработки одностраничных и мобильных приложений.

2.6 Дизайн сайта

При создании интерфейса Информационной системы музыкального магазина большое внимание должно быть уделено UX части. Именно поэтому, было принято решение использовать технологию юзабилити.

Согласно терминологии, usability (юзабилити) сайта — это степень эффективности, продуктивности и удобства взаимодействия человека с интерфейсом. Так же зачастую под данным словом объединяют методы повышения эффективности работы web-сайта. Владелец любого сайта желает видеть на своем ресурсе как можно большее количество посетителей.

При проектировании дизайна информационной системы были приняты следующие правила:

1) Гибкий дизайн – На смену «жесткому» дизайну, пришел гибкий, или как его еще называют – резиновый. Данный вид предусматривает наличие разного размера экранов и разрешений: все элементы привязаны к странице в процентном соотношении, таким образом, на больших экранах сайт будет автоматически растягиваться на всю ширину, а на маленьких соответственно, сужаться

2) Минимализм - Он очень распространен в реальности, добрался и до веб-пространства. Основной акцент — на простом и неброском оформлении. Здесь не должно быть ничего лишнего! Информативность проявляется не в обилии контента, а наоборот. Иногда весь текст на минималистичных сайтах заменен на видео или сжат до пары фраз или призыва к действию

3) Актуальность - дизайн сайта должен соответствовать последним тенденциям в web-дизайне, но в тоже время не изобилловать графическими эффектами дабы не отвлекать пользователей от интересующей информации и услуг клиники.

4) Легкость - скорость загрузки страницы должна быть максимальной, но в то же время качество графических элементов необходимо сохранить на предельно высоком уровне.

5) Универсальность - страницы сайта должны корректно отображаться как на стационарных компьютерах, так и на мобильных устройствах с выходом в интернет.

Ниже на рисунках показаны шаблоны интерфейса главной страницы (Рисунок 5, 6).

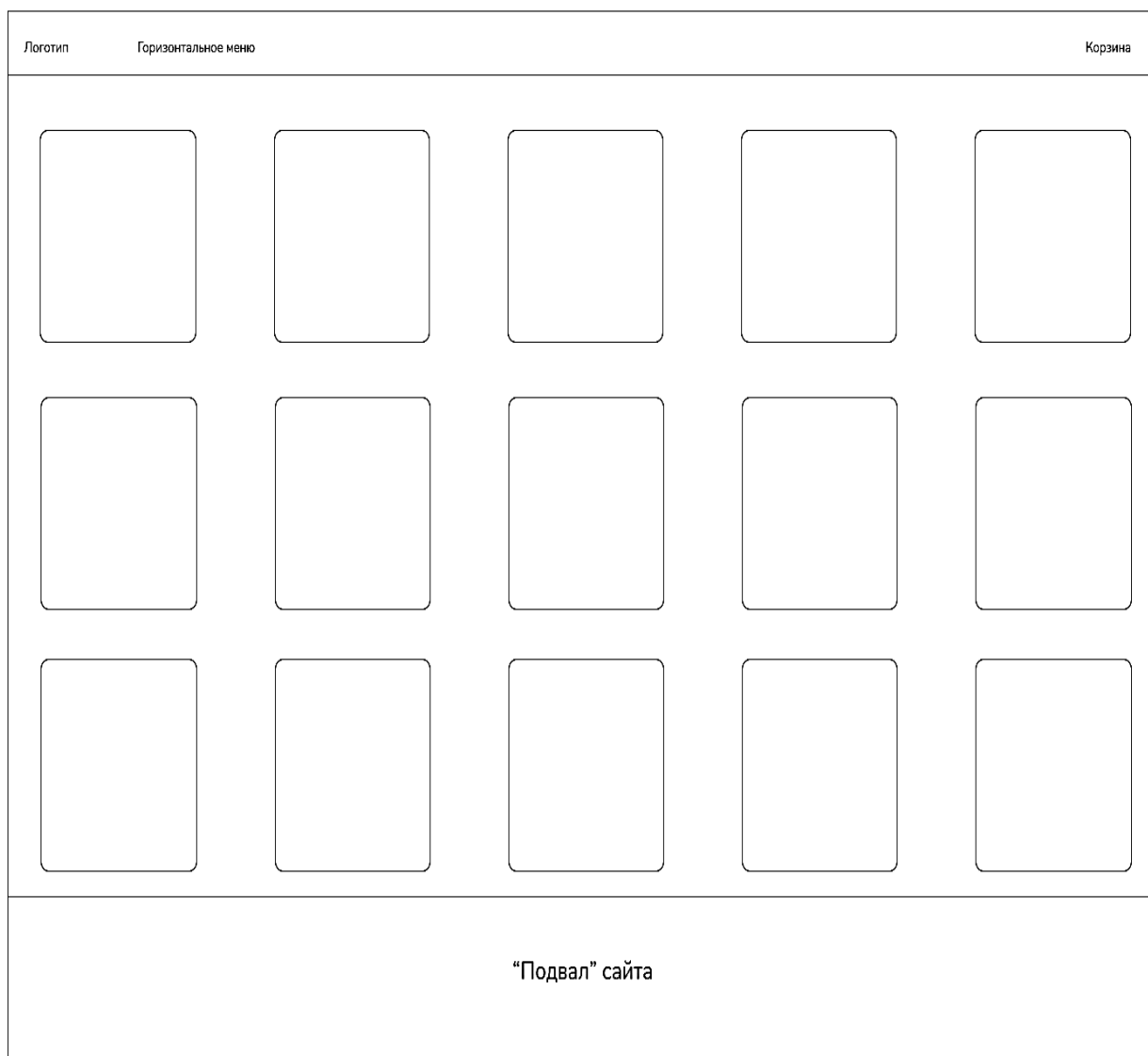


Рис. 5. Макет дизайна главной страницы для музыкального магазина «SongPay»



Рис. 6. Логотип

Скорость работы сайта

Необходимо обеспечить стабильно высокую скорость обработки запросов и загрузку страницы, занимая при этом минимальное количество ресурсов сервера.

Удобство навигации

Простота навигации по сайту это один из важнейших критериев при разработке проекта. Пользователь должен интуитивно и беспрепятственно перемещаться по сайту, получать необходимую информацию или услугу.

Внутреннее содержание сайта

Ресурс должен содержать только качественный и полно изложенный материал с правильно подобранными «ключевыми словами».

Необходимо было реализовать простой, понятный и легкий интерфейс как для пользователей административной панели, так и для клиентов сайта.

Выводы по разделу:

При разработке любого программного обеспечения значимым фактором является точная формулировка цели проекта и анализ предметной области. В данном разделе проанализированы основные бизнес-процессы музыкального магазина «SongPay», рассмотрены основные технические решения, проведен анализ информационной базы. Также, были установлены основные задачи на разработку информационной системы.

3. Описание информационной системы

3.1 Реализация программных модулей приложения

Информационная система разделена на три модуля:

- модуль «Администратор системы»;
- модуль «Пользователь»;

Модуль пользователя позволяет клиентам перемещаться по сайту и просматривать товары. Демонстрация главной страницы представлена на рисунке 7.

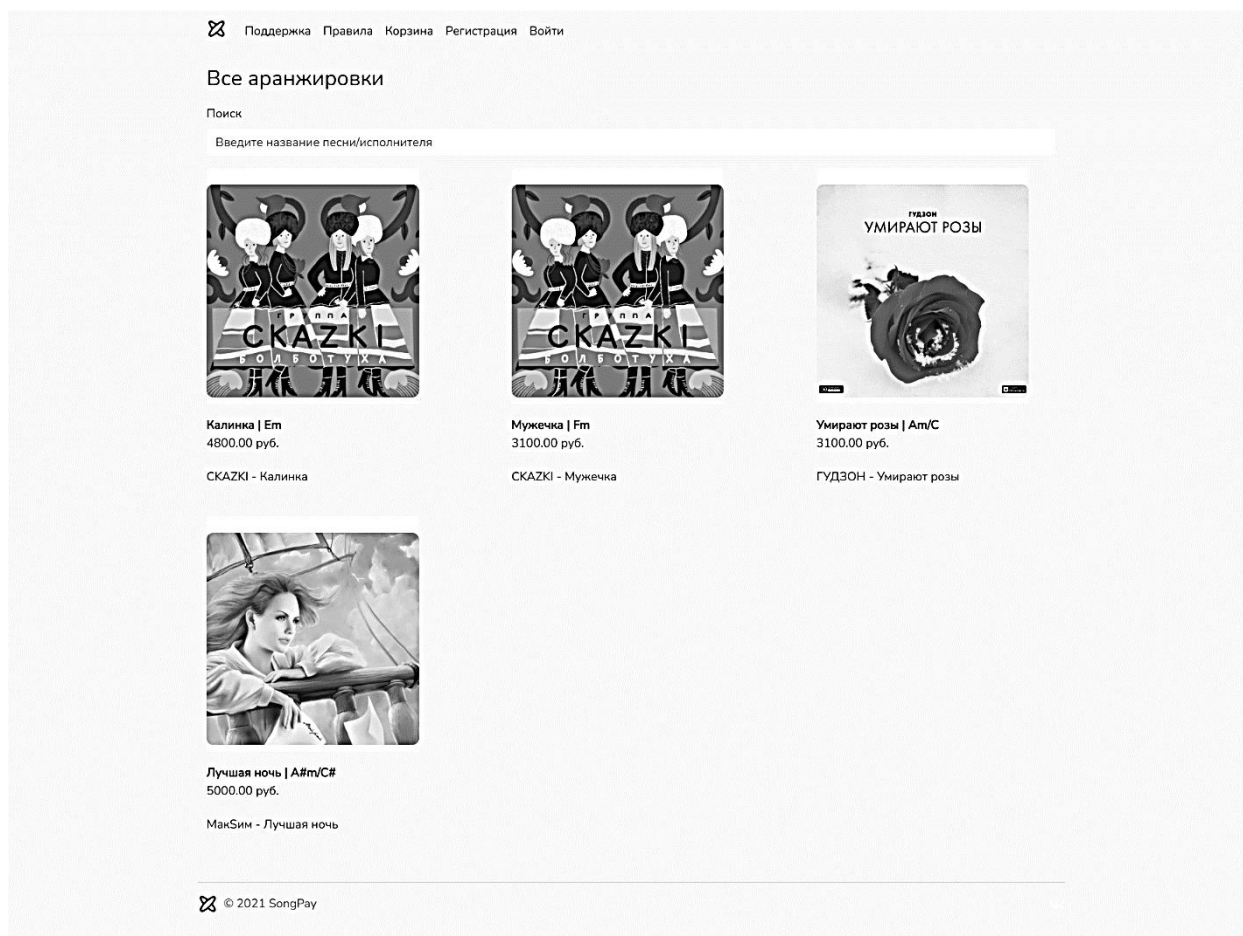


Рис. 7. Главная страница сайта

На стартовой странице информационной системы находится общая информация о назначении информационной системы. На верхней панели страницы шесть кнопок: «Главная (логотип)», «Поддержка» (Рисунок 8), «Правила», «Корзина», «Регистрация» и «Авторизация».

Демонстрация страницы «Поддержка»

Рис. 8. Страница «Поддержка»

Для создания функционала сайта была создана база данных. /ПРИЛОЖЕНИЕ 2/. Данные о пользователях, зарегистрированных на сайте, о аранжировках, продаваемых в магазине, о сделанных покупателями заказах будут храниться в базе данных.

Незарегистрированные пользователи могут просматривать каталог (главная страница).

После регистрации информация, введенная в форму, сохраняется в таблице «lkarr_userprofile» базы данных. Пароли и логины пользователей могут состоять только из латинских букв и цифр, так как некоторые символы, например, кавычки, могут использоваться злоумышленниками. Для обеспечения безопасности пароли хранятся в базе данных не в открытом виде, а в виде хеш-кодов, полученных с использованием криптографического алгоритма. Если пользователь при авторизации забудет ввести пароль, на форме авторизации появится сообщение с просьбой ввода пароля. Работа формы авторизации продемонстрирована на рисунке 9.

Рис. 9. Форма авторизации

Чтобы пользователю стала доступна возможность заказа товаров, необходимо пройти процедуру регистрации на сайте. Сделать это можно, кликнув по разделу «Регистрация», после чего пользователь перейдёт на страницу регистрации. Система «Регистрации» пользователя продемонстрирован на рисунке 10.

Регистрация

Email
alex@mail.ru

Имя пользователя
Имя пользователя

Имя
Имя

Фамилия
Фамилия

Пароль
Пароль

Регистрация

Рис. 10. Форма Регистрации

Для того, чтобы интересующая информация была отправлена администратору, а после сохранена в базу пользователю требуется заполнить форму обратной связи (поддержка). Сделать это можно, перейдя на соответствующий пункт главного меню. Форма обратной связи представлена на рисунке 11.

Поддержка

Если у вас есть вопросы или идеи - заполните эту форму и отправьте!
Мы свяжемся с вами в ближайшее время, если это будет необходимо.
Спасибо, что выбрали нас!

Email
Email

Ваше обращение
Ваше обращение

Отправить

Рис. 11. Страница «Поддержка»

Администрирование доступно после авторизации. через административный модуль. Администратору доступно управление пользователями, через административную панель, которая встроена в фреймворк Django (Рисунок 12, 13,

14).

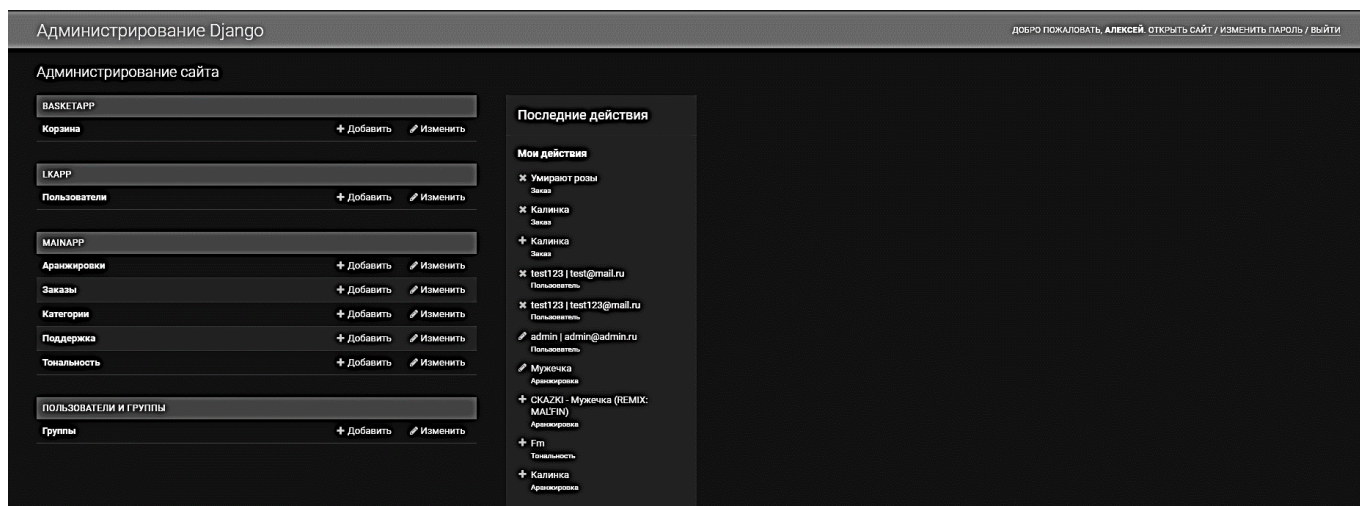


Рис. 12. Панель Администратора

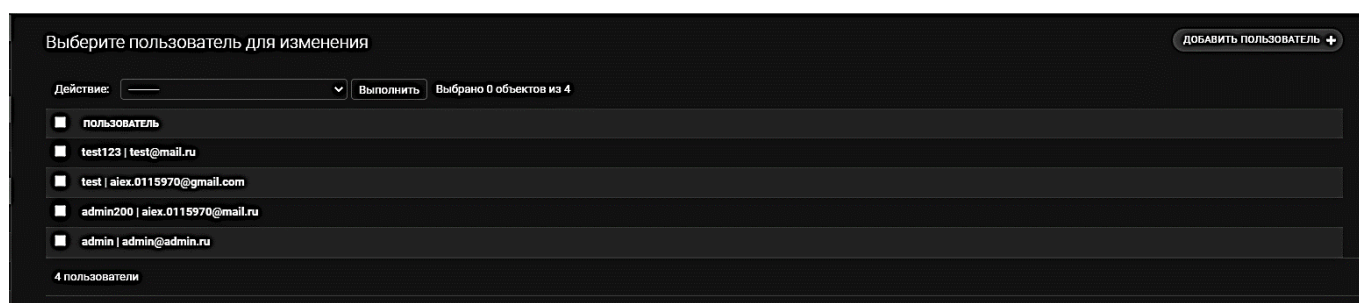


Рис. 13. Управление пользователями

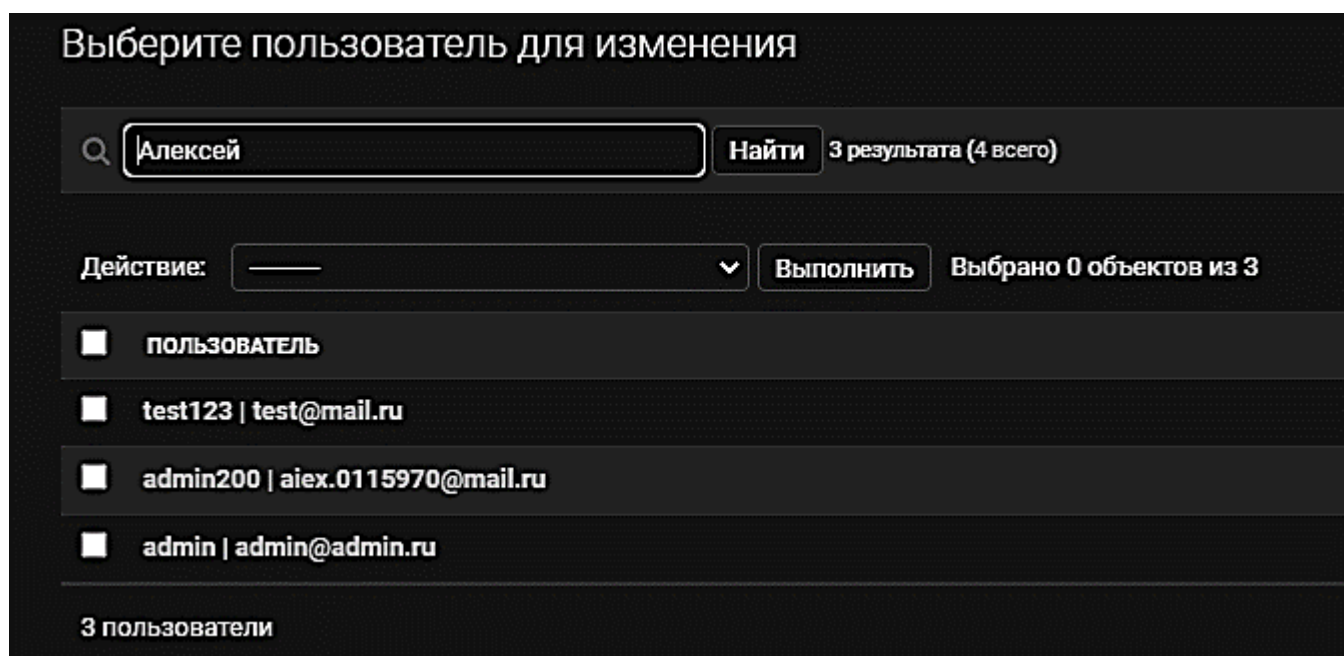


Рис. 14. Фильтрация пользователей по имени

Для модуля продавца доступна функция добавления товара (аранжировок), для её осуществления необходимо выбрать раздел в меню «MAINAPP» из выпадающего

списка выбрать «Аранжировки», после чего открывается страница с аранжировками. В данном разделе можно фильтровать аранжировки по категории, тональности. Также имеется возможность просмотра, удаления, редактирования (Рисунок 15, 16, 17).

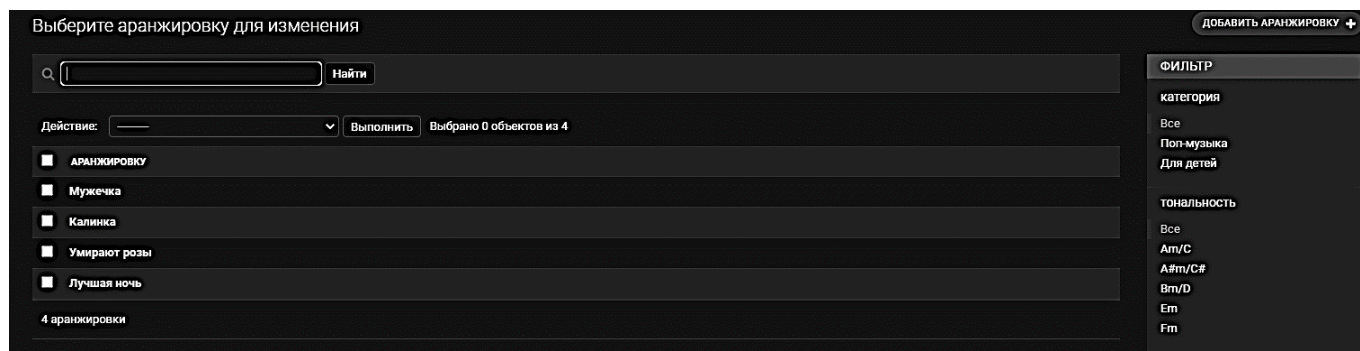


Рис. 15. Страница «Аранжировка»

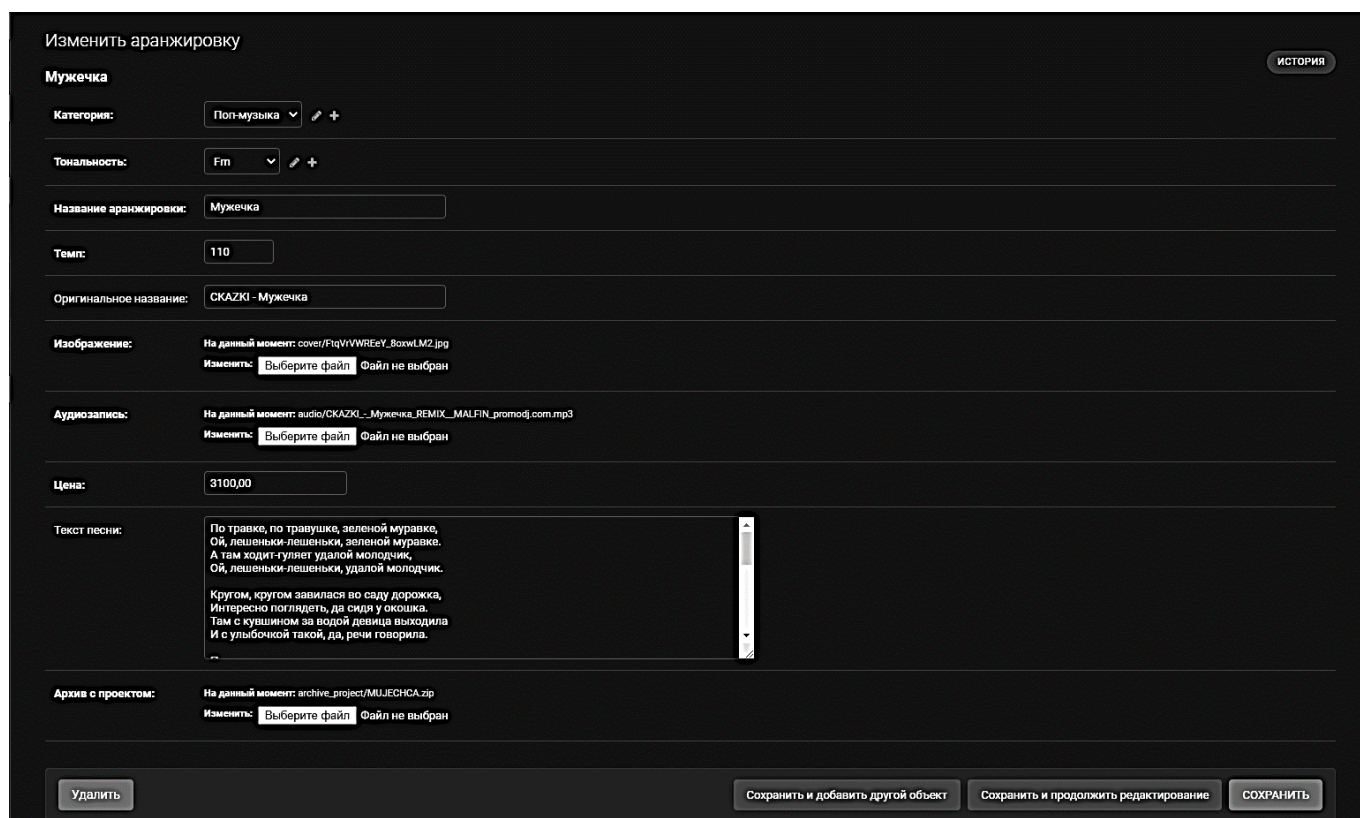


Рис. 16. Редактирование аранжировки

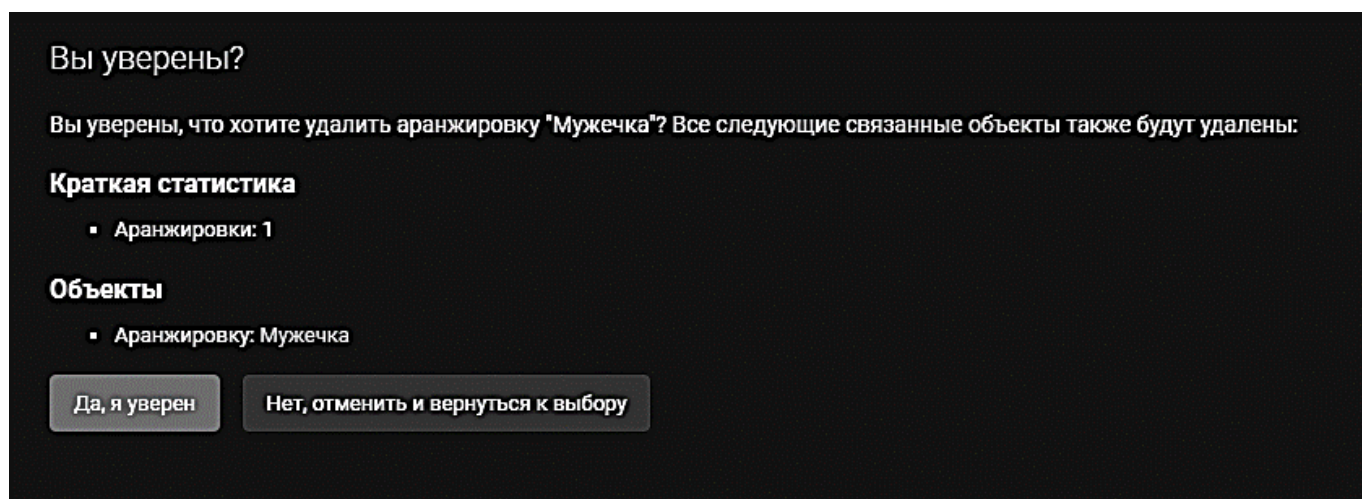


Рис. 17. Удаление аранжировки

Чтобы добавить в базу данных новую аранжировку, надо заполнить необходимые поля и нажать кнопку «Сохранить» или «Сохранить и добавить другой объект» или «Сохранить и продолжить редактирование». Процесс добавления нового товара представлен на рисунке 18.

Добавить аранжировку

Категория:

Тональность:

Название аранжировки:

Темп:

Оригинальное название:

Изображение:

Аудиозапись:

Цена:

Текст песни:

Архив с проектом:

Рис. 18. Добавление аранжировки

Выводы по разделу:

На этапе физического проектирования были реализованы отдельные модули Информационной системы «SongPay»:

1. Созданы отдельные АРМ-ы для двух категорий пользователей ИС: пользователя (клиента) и администратора;
2. Создана реляционная база данных;
3. Реализована система регистрации и авторизации пользователей;
4. Выполнена стартовая страница с оригинальным оформлением;
5. Для модуля «Администратор системы» реализована административная панель с удобным интерфейсом;
6. Для просмотра и удобного управления контентом разработан модуль продавца (администратор):

В процессе работы над проектом создано 8 моделей таблиц, 6 контроллеров и CRUD, а также 3 модели форм, контроллеров и видов к ним.

Решены все поставленные задачи в полной мере. Реализованы все основные функции в двух модулях информационной системы.

Заключение

В ходе выполнения выпускной квалификационной работы была достигнута основная цель – разработан проект и реализованы основные модули информационной системы с веб-интерфейсом для музыкального магазина.

Для достижения поставленной цели был проведен анализ предметной области, после чего была построена концептуальная модель проектируемой информационной системы.

В ходе построения модели было выполнено следующее:

- Построены модели вариантов использования (useCaseModel) as-is и to-be;
- Написаны текстовые сценарии для каждого базового варианта использования;
- Построены диаграммы деятельности для трех базовых вариантов использования:
- Построены диаграммы последовательности и состояний;
- Продумана интеграция ИС «SongPay» с другими Информационными системами и построена диаграмма пакетов;
- Спроектирована и разработана база данных для информационной системы музыкального магазина «SongPay».

Далее был осуществлен выбор методов и программных средств для реализации проекта.

Для программной реализации были выбраны следующие средства:

- 1) серверная СУБД: SQLite;
- 2) веб-интерфейс: фреймворк Django, Bootstrap5, ReactJS.

На этапе создания спроектированной информационной системы были реализованы системы регистрации и авторизации пользователей,

созданы следующие модули ИС:

- модуль «Администратор системы»;
- модуль «Покупатель»;

Таким образом, все поставленные задачи по реализации работы успешно решены, а разработанная информационная система «SongPay» соответствует требованиям, изложенным в исходных данных к работе.

Список использованных источников

Монографическая и учебная литература

1. Компьютерная графика и WEB-дизайн: практикум/Т.И. Немцова и др.; под ред. Л.Г. Гагариной. - Москва: ФОРУМ; ИНФРА-М, 2015. - 288 с.
2. Селезнёв, В.А. Компьютерная графика: учебник и практикум для СПО/ В.А. Селезнёв, С.А. Дмитроченко. - 2-е изд., испр. и доп.- Москва: Юрайт, 2019.-218 с. - (Профессиональное образование).
3. Крейг, Дж. Шрифт и дизайн. Современная типографика /Дж. Крейг, И. Скала.-Санкт-Петербург: Питер, 2016. -176 с.
4. Маркин, А. В. Программирование на SQL: учебное пособие / А. В. Маркин. – Москва: Юрайт, 2022. – 435с. – (Профессиональное образование). – URL: <https://urait.ru/viewer/programmirovanie-na-sql-495666> (дата обращения: 06.05.2022).
5. Новожилов, О.П. Архитектура компьютерных систем в 2 ч.: учебное пособие для СПО / О.П. Новожилов. - Москва: Юрайт, 2020. - 276 с. - (Профессиональное образование).
6. Полуэктова, Н. Р. Разработка веб-приложений: учебное пособие для среднего профессионального образования / Н. Р. Полуэктова. — Москва: Издательство Юрайт, 2022. — 204 с. — (Профессиональное образование). — ISBN 978-5-534-14744-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/496693> (дата обращения: 27.09.2021).
7. Рекомендации по написанию выпускной квалификационной работы: методические материалы для преподавателей и обучающихся по специальностям технического профиля / авт.-сост. Л.П. Тишкова. – Курган, 2021. – 29с.
8. Ташков, П.А. Веб-мастеринг на 100%: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка / П.А. Ташков. – СПб.: Питер, 2010. - 512 с. – (Серия «На 100%»).
9. Тозик, В.Т. Компьютерная графика и дизайн: учебник/ В.Т.Тозик, Л.М. Корпан. - 9-е изд. стереотип. - Москва: Академия, 2019.-208 с. – (Среднее профессиональное образование).

Интернет-ресурсы

1. Bootstrap. - URL: <https://getbootstrap.com/> (дата обращения: 01.03.2022).

2. Django. – URL: <https://www.djangoproject.com/> (дата обращения: 26.11.2021).
3. GitHub. – URL: <https://github.com/> (дата обращения 28.11.2021).
4. Googlefonts. - URL: <https://fonts.google.com/> (дата обращения: 12.01.2022).
5. JavaScript. - URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата обращения: 5.09.2021).
6. ReactJS. - URL: <https://ru.reactjs.org/> (дата обращения: 20.02.2022)
7. Полное руководство по Django. – URL: <https://docs.djangoproject.com/en/4.0/> (дата обращения: 30.11.2021).
8. Руководство по Python. – URL: <https://docs.python.org/3/> (дата обращения: 11.10.2021).

ПРИЛОЖЕНИЕ 1

Код древовидного меню (используется React JS)

Файл: Header.js

```
import {Link} from 'react-router-dom';
```

```
import {Fragment} from "react";
```

```
const Header = ({user}) => {
```

```
  const logout = () => {
```

```
    localStorage.setItem('token', null);
```

```
    localStorage.setItem('refreshToken', null);
```

```
    localStorage.setItem('user', null);
```

```
    window.location = '/'
```

```
  }
```

```
  return (
```

```
    <nav className="navbar navbar-expand-lg navbar-light bg-light">
```

```
      <div className="container">
```

```
        <Link className="navbar-brand" to={"/"}>
```

```
          <svg width="30" height="24" viewBox="0 0 74 74" fill="none"
```

```
xmlns="http://www.w3.org/2000/svg"
```

```
      className="d-inline-block align-text-top">
```

```
        <path
```

```
          d="M45.6257 26.8194C31.7764 12.9702 12.4452 -0.301828
```

```
5.80925 5.32432C-0.826742 10.9505 12.3729 29.6088 22.5437 40.9572C36.2487
```

```
56.2489 60.1962 75.0031 67.5537 67.9342C74.9111 60.8652 59.475 40.6686
```

```
45.6257 26.8194Z"
```

```
      stroke="#212529" strokeWidth="8"/>
```

```
        <path
```

```
          d="M46.6515 45.6255C60.5007 31.7763 73.7728 12.4451 68.1466
```

```
5.80906C62.5205 -0.826924 43.8622 12.3727 32.5138 22.5435C17.222 36.2485 -
```

1.53213 60.196 5.5368 67.5535C12.6057 74.911 32.8024 59.4748 46.6515
45.6255Z"

```

        stroke="#212529" strokeWidth="8"/>
    </svg>
</Link>
<button      className="navbar-toggler"      type="button"      data-bs-
toggle="collapse"
        data-bs-target="#navbarSupportedContent"      aria-
controls="navbarSupportedContent"
        aria-expanded="false" aria-label="Toggle navigation">
    <span className="navbar-toggler-icon"/>
</button>
<div      className="collapse      navbar-collapse"
id="navbarSupportedContent">
    <ul className="navbar-nav me-auto mb-2 mb-lg-0">
        <li className="nav-item">
            <Link      to={"/support"}      className="nav-link      px-2
link_gray">Поддержка</Link>
        </li>
        {/*<li className="nav-item">*/}
        {/*      <Link      to={"/catalog"}      className="nav-link      px-2
link_gray">Каталог</Link>*/}
        {/*</li>*/}
        <li className="nav-item">
            <Link      to={"/terms"}      className="nav-link      px-2
link_gray">Правила</Link>
        </li>
        <li className="nav-item">
            <Link      to={"/cart"}      className="nav-link      px-2
link_gray">Корзина</Link>

```

```

    </li>

    { !user &&
    <Fragment>
        <li className="nav-item">
            <Link      to={"/register"}      className="nav-link      px-2
link_gray">Регистрация</Link>
            </li>
            <li className="nav-item">
                <Link      to={"/login"}      className="nav-link      px-2
link_gray">Авторизация</Link>
            </li>
        </Fragment>
    }
    {user &&
    <Fragment>
        <li className="nav-item">
            <Link      to={"/lk"}      className="nav-link      px-2
link_gray">Личный кабинет</Link>
            </li>
            <li className="nav-item"><Link to={"#"}
                className="nav-link      px-2      link_gray"
onClick={logout}>Выйти
                | {user && user.username}</Link></li>
        </Fragment>
    }
</ul>
</div>
</div>

```

```
    </nav>
  )
}
export default Header
```

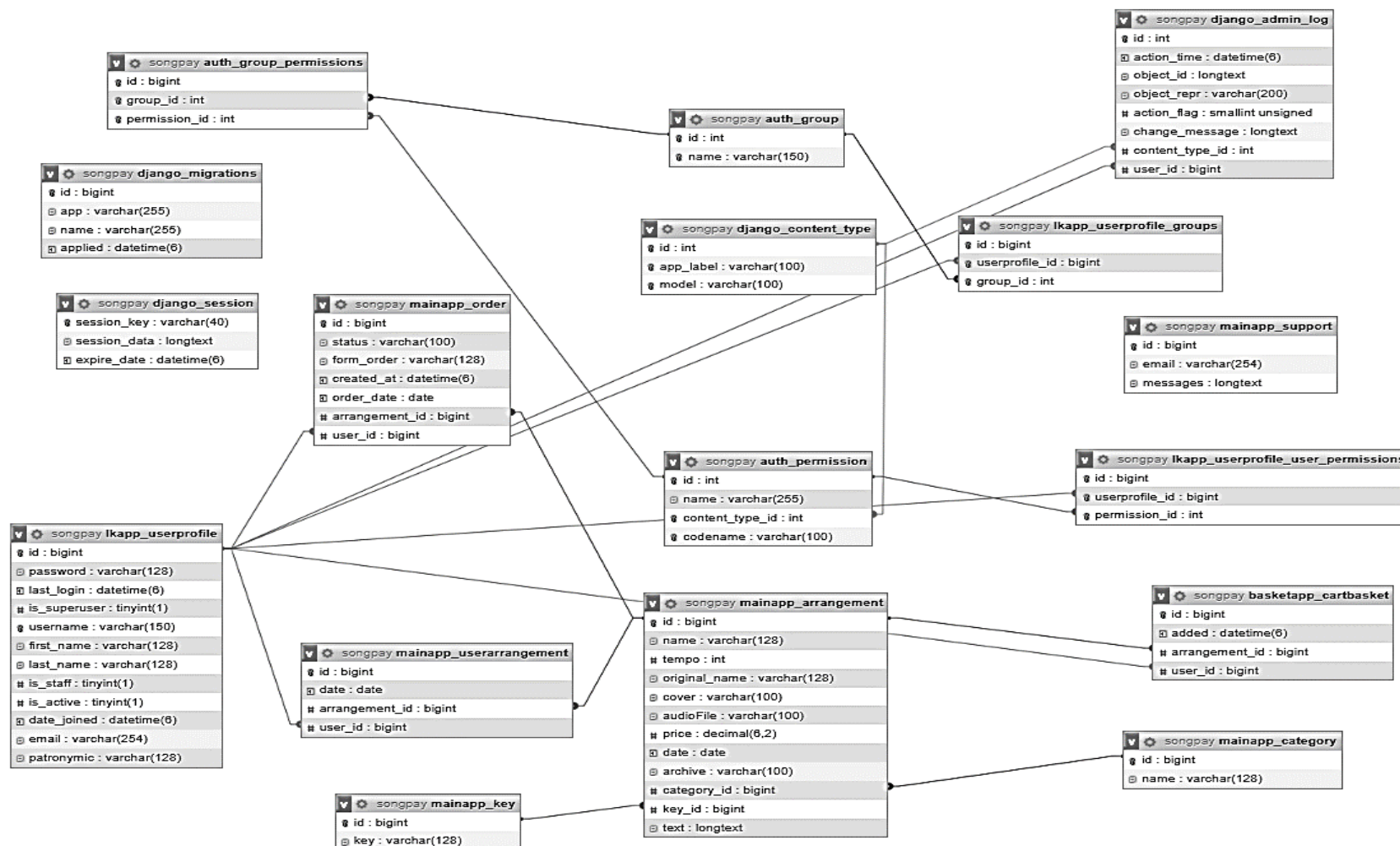



Рис. 19. Схема базы данных Информационной системы «SongPay»

ПРИЛОЖЕНИЕ 3

```

- {
  id: 3,
  - category: {
    id: 1,
    name: "Поп-музыка"
  },
  - key: {
    id: 10,
    key: "Em/G"
  },
  name: "Калинка",
  tempo: 87,
  original_name: "СКАЗКИ - Калинка",
  cover: "https://api.malfinbeats.ru/media/cover/FtqVrVwREeY.jpg",
  audioFile: "https://api.malfinbeats.ru/media/audio/kal_.mp3",
  price: "4800.00",
  text: "Сама иду по камению,
  Коня веду – конь лён топчет.
  Лёлюшки-лёли,
  Коня веду – конь лен то...у!

  Коня веду – конь лён топчет,
  Напереди воробейчик.
  Лёлюшки-лёли,
  Напереди воробей.

  Припев:
  Ох да, калинка моя,
  В саду ягода-малинка моя.
  Ох да, калинка моя,
  В саду ягода-малинка моя,
  Малинка моя.

  Воробей-воробейчик,
  Скажи-ка мне всю правдочку.
  Лёлюшки-лёли,
  Скажи-ка всю правда...у!

  Скажи-ка мне всю правдочку:
  Кому воля, кому нет воли?
  Лёлюшки-лёли,
  Кому воля, кому нет во...

  А девичья воля – да за коника во поле,
  А девичья воля – за платочек да в таночек,
  А бабская доля – на печи казан кипит,
  А бабская доля – дед ворчит, дитя кричит.

  Припев",
  date: "2022-06-03",
  archive: "https://api.malfinbeats.ru/media/archive_project/Kalinka.zip"
},

```

Рис. 20. Ответ Api

```
import React, { useEffect, useState } from 'react';  
import axios from "axios";
```

Рис. 21. Подключение хуков и асинхронной библиотеки

```
const [cards, setCards] = useState([]);  
const [loading, setLoading] = useState(true);
```

Рис. 22. Использование хука «useState»

```
useEffect(() => {  
    axios.get(`https://api.malfinbeats.ru/api/v1/arrangement/`)  
        .then((result) => {  
            setCards(result.data)  
            setLoading(false)  
        })  
}, [])
```

Рис. 23. Получение данных GET запросом к API-серверу



Рис. 24. Диаграмма регистрации пользователя

Приложение 6

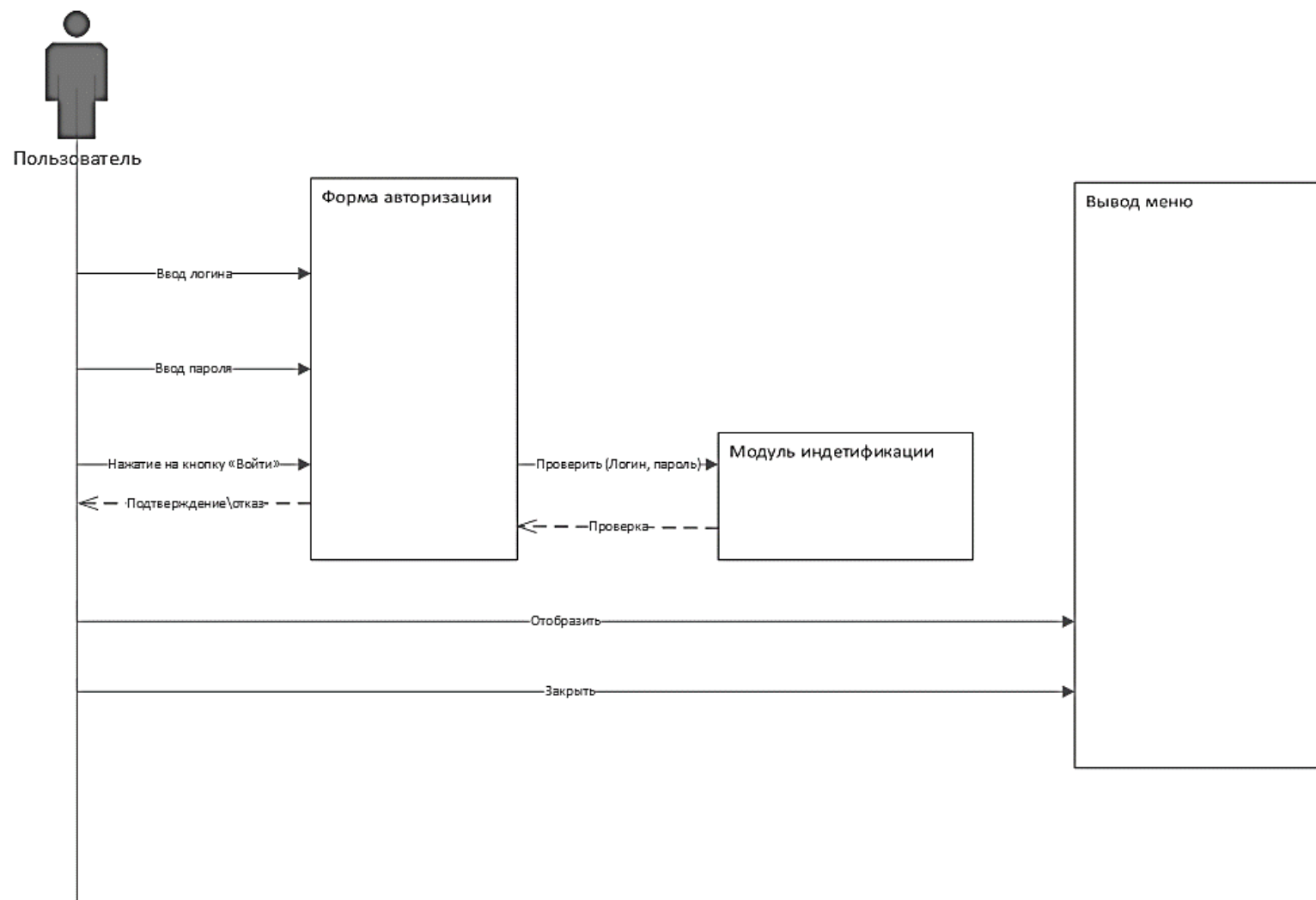


Рис. 25. Диаграмма последовательности для авторизации пользователя