**Министерство науки и высшего образования РФ**

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт – ИШИТР

Отделение – ИТ

Направление – ИСиТ

**Курсовая работа**

«БД дискографий (энциклопедия)»

по дисциплине: управление данными

Выполнил: студент гр. 8И82 \_\_\_\_\_\_\_\_\_\_\_ Санжиев Д.Б

(Номер группы) (Подпись) (Ф.И.О.)

Проверил: старший преподаватель ОИТ \_\_\_\_\_\_\_\_\_\_\_ Лепустин А.В.

(Подпись) (Ф.И.О.)

Дата 19.06.2020

Цель работы

Получить основные навыки проектирования и создания web-приложения на основе имеющейся базы данных.

Постановка задачи

На основе созданной раннее базы данных реализовать web-приложение, которое имеет следующие особенности:

1. Приложение должно выводить в «шапке» страницы следующие сущности: композиции, коллективы, альбомы, музыканты, жанры. На страницах этих сущностей выводить все атрибуты.
2. На web-странице определенного музыканта помимо его основной информации выводить название коллективов, в которых он состоит, а также его исполнения песен.
3. На web-странице определенной композиции помимо основной информации выводить жанр, в котором написан, его композиторов, авторов и исполнителей.
4. На web-странице определенного коллектива помимо основной информации выводить музыкантов, состоящих в коллективе, а также альбомы.
5. На web-странице определенного альбома помимо основной информации выводить композиции, входящие в него.
6. На web-странице определенного жанра помимо основной информации выводить композиции, входящие в него.
7. Должны быть реализована регистрация пользователя.
8. Зарегистрированные пользователи должны иметь возможность просматривать, редактировать и удалять данные. Незарегистрированные пользователи имеют возможность только просматривать данные.
9. При создании и редактировании композиций иметь возможность выбирать жанр, автора, композитора из выпадающего списка.
10. При создании, редактировании композиций иметь возможность выбирать жанр из выпадающего списка.

**Используемые технологии**

В ходе реализации web-приложения были использованы следующие технологии: язык программирования C#, СУБД MS SQL Server, ASP.NET, Entity Framework, LINQ, HTML.

**Описание БД**

Основными сущностями базы данных являются: Коллективы, Композиции, Музыканты, Лейбл, Альбомы (см. Приложение А)

Коллективы – сущность, хранящая в себе основную информацию о разных музыкальных коллективах. Атрибуты: ID коллектива, Страна, Название, Год создания.

Композиции – сущность, которая хранит в себе основную информацию о разных музыкальных композициях. Атрибуты: ID композиции, Название, Год выпуска.

Музыканты – сущность, которая хранит в себе основную информацию о разных музыкантах. Атрибуты: ID музыканта, Фамилия, Имя, Отчество, Гражданство.

Лейбл – сущность, которая хранит в себе основную информацию о разных лейблах. Атрибуты: ID лейбла, Название.

Альбомы – сущность, которая хранит в себе основную информацию о разных альбомах/сборниках. Атрибуты: ID альбома, Название, Дата выпуска

Музыканты могут вступать в коллективы, а также их покидать. Таблица «Музыканты в группах» хранит в себе даты вступления и ухода музыкантов из коллективов.

Также музыкальные коллективы могут вступать в лейблы и покидать их. Таблица «Коллективы в лейблах» хранит в себе даты их вступления и ухода.

У музыкальных композиций могут быть несколько авторов и композиторов. Для этого реализованы таблицы «Авторы» и «Композиторы»

Композиции могут исполнятся в разное время, разными коллективами, разными музыкантами. Таблица «Исполнение композиций» хранит в себе всю эту информацию. Исполненные композиции также могут входить в разные альбомы/сборники, для этого реализована таблица «ИК\_альбомы».

Также в базе данных есть таблицы-справочники: 1. Жанры – хранит в себе информацию о разных жанрах композиций.

2. Тип – данный справочник показывает во что входят исполненные композиции: в альбомы или в сборники.

3. Типа коллектива – данный справочник показывает к какому музыкальному формированию относится коллектив: ансамбль или группа.

Ход работы

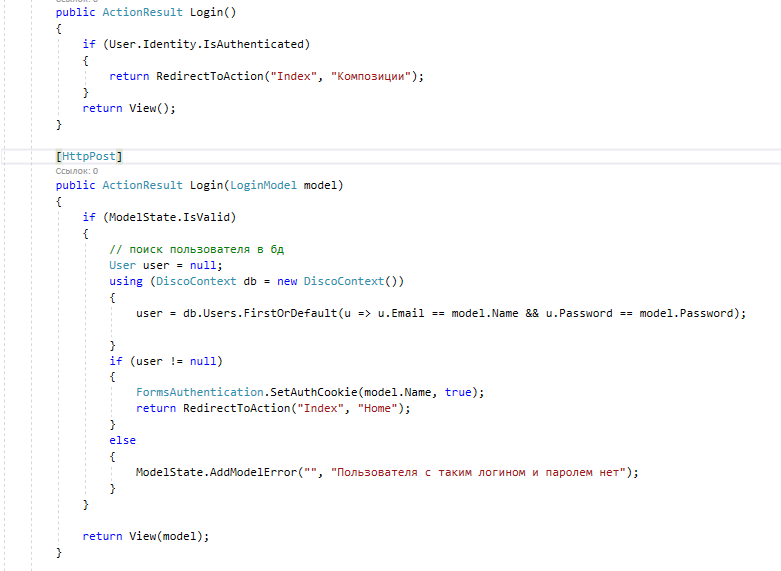
Диаграмма классов приложения представлена в приложении А.

**Описание контроллеров и их методов**

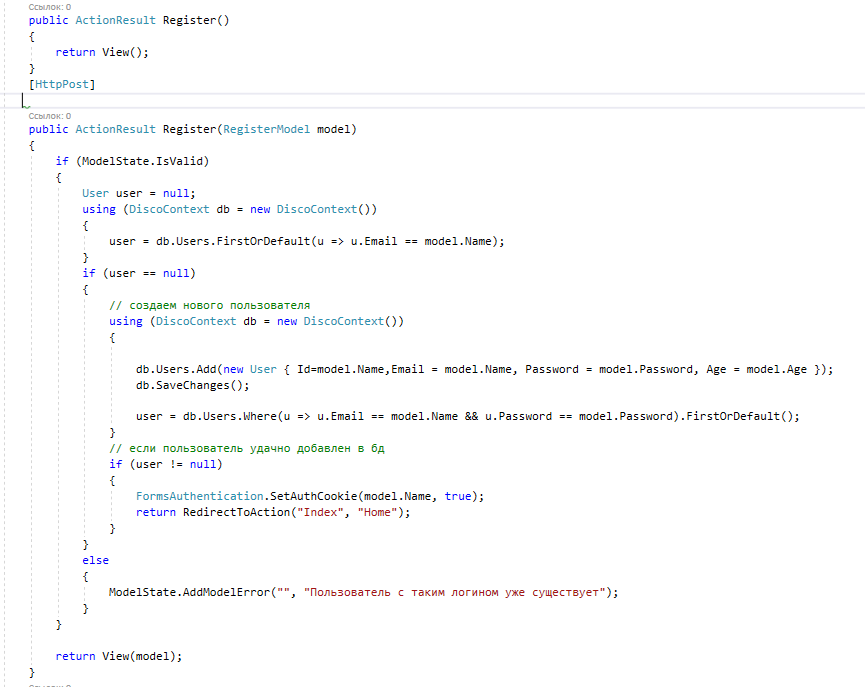
Контроллер «AccountController» отвечает за аутентификацию пользователя и имеет следующие методы:

1. Метод «Login» позволяет считывать введенный логин и пароль и искать пользователя в базе данных. Метод «Get Login» проверяет зарегистрирован ли пользователь, если зарегистрирован – перебрасывает на страницу «Композиции». Метод «Post Login» принимает объект класса «LoginModel», если пользователь найден в базе данных, то перебрасывает на домашнюю страницу, если не найден – возвращает сообщение.
2. Метод «Register» позволяет создавать нового пользователя. Метод «Post Register» принимает объект класса «RegisterModel», если пользователь с вводимыми данными не найден в базе данных, то идет сохранение данных и возвращение на домашнюю страницу, если найден – возвращает сообщение.
3. Метод «Logoff» позволяет выходить из учетной записи и перекидывает на главную страницу.

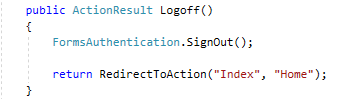
Все методы показаны на рисунке 1.



а)



б)



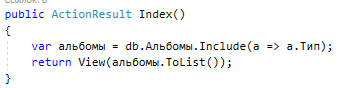
в)

Рисунок 1 – Методы контроллера «AccountController»: а – Login, б – Register, в – Logoff

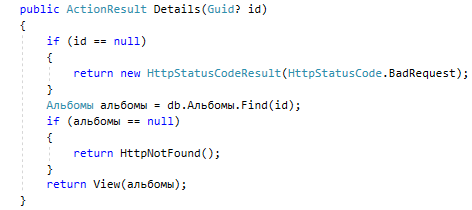
Контроллер «АльбомыController» позволяет отображать данные, связанные с сущностью «Альбомы» и имеет следующие методы:

1. Метод «Index» отображает список альбомов, существующих в базе данных, возвращает альбомы и типы альбомов.
2. Метод «Details» позволяет более подробно увидеть информацию о конкретном альбоме. Принимает Guid id и возвращает объект класса «Альбомы» в случае если он не равен значению null, если null- то выводит ошибку.
3. Метод «Create» позволяет создавать новые альбомы с выбором типа альбома, композитора – автора (композитор и автор – это одна личность). Метод «Post Create» принимает и возвращает объект класса «Альбомы».
4. Метод «Edit» позволяет вносить изменения в существующие альбомы. Метод «Get Edit» принимает значение Guid id. Если он равен значению null, то выдает ошибку, если нет – объект класса «Альбомы». Метод «Get Edit» принимает и возвращает объект класса «Альбомы».
5. Метод «Delete» позволяет удалять существующие альбомы. Метод «Get Delete» принимает значение Guid id. Если он равен значению null, то выдает ошибку, если нет – объект класса «Альбомы». Метод «Post Delete» принимает значение Guid id, удаляет данные равные этому id, сохраняет изменения и перенаправляет на домашнюю страницу.

Все методы показаны на рисунке 2.



а)



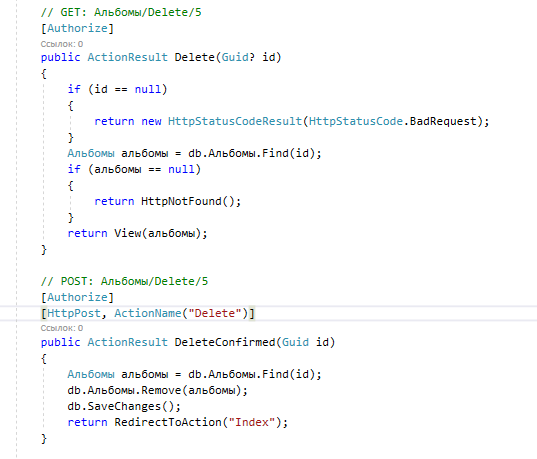
б)



в)



г)



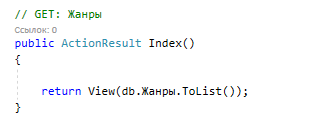
д)

Рисунок 2 – Методы контроллера «АльбомыController»: а – Index, б – Details, в – Create, г – Edit, д – Delete.

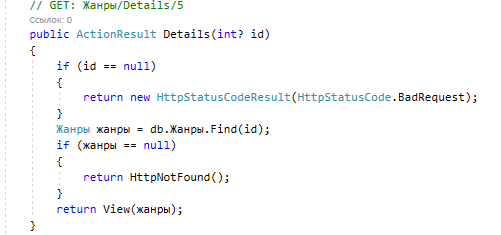
Контроллер «ЖанрыController» позволяет отображать данные, связанные с сущностью «Жанры» и имеет следующие методы:

1. Метод «Index» отображает список жанров, существующих в базе данных, возвращает жанры.
2. Метод «Details» позволяет более подробно увидеть информацию о конкретном жанре. Принимает Int id и возвращает объект класса «Жанры» в случае если он не равен значению null, если null- то выводит ошибку.
3. Метод «Create» позволяет создавать новый жанр. Метод «Post Create» принимает и возвращает объект класса «Жанр».
4. Метод «Edit» позволяет вносить изменения в существующие жанры. Метод «Get Edit» принимает значение Int id. Если он равен значению null, то выдает ошибку, если нет – объект класса «Жанры». Метод «Get Edit» принимает и возвращает объект класса «Жанры».
5. Метод «Delete» позволяет удалять существующие жанры. Метод «Get Delete» принимает значение Int id. Если он равен значению null, то выдает ошибку, если нет – объект класса «Жанры». Метод «Post Delete» принимает значение Int id, удаляет данные равные этому id, сохраняет изменения и перенаправляет на домашнюю страницу.

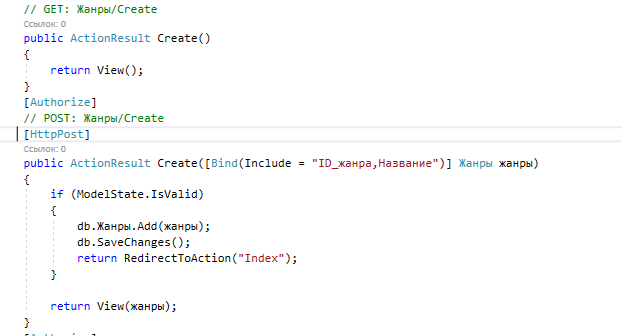
Все методы показаны на рисунке 3.



а)



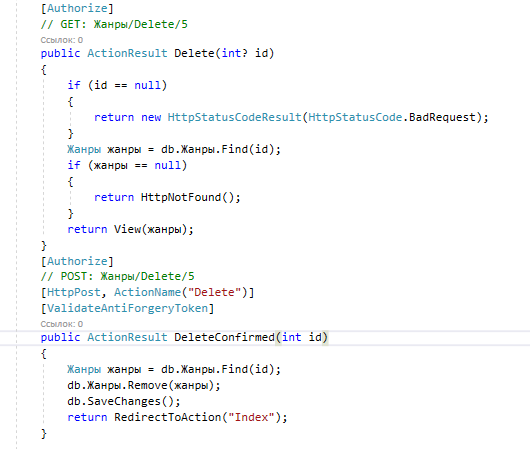
б)



в)



г)



д)

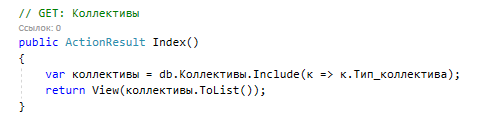
Рисунок 3 – Методы контроллера «ЖанрыController»: а – Index, б – Details,

в – Create, г – Edit, д – Delete.

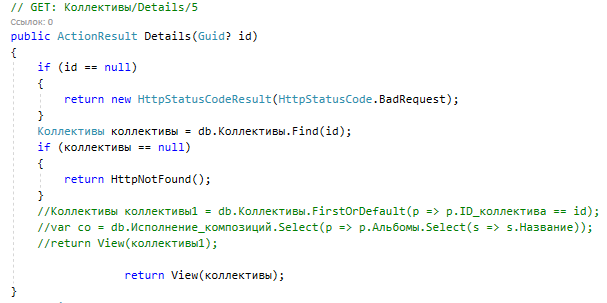
Контроллер «КоллективыController» позволяет отображать данные, связанные с сущностью «Коллективы» и имеет следующие методы:

1. Метод «Index» отображает список коллективов, существующих в базе данных, возвращает коллективы и типы коллектива.
2. Метод «Details» позволяет более подробно увидеть информацию о конкретном коллективе. Принимает Guid id и возвращает объект класса «Коллективы» в случае если он не равен значению null, если null- то выводит ошибку.
3. Метод «Create» позволяет создавать новые коллективы с выбором типа коллектива. Метод «Post Create» принимает и возвращает объект класса «Коллективы».
4. Метод «Edit» позволяет вносить изменения в существующие коллективы. Метод «Get Edit» принимает значение Guid id. Если он равен значению null, то выдает ошибку, если нет – объект класса «Коллективы». Метод «Get Edit» принимает и возвращает объект класса «Коллективы».
5. Метод «Delete» позволяет удалять существующие коллективы. Метод «Get Delete» принимает значение Guid id. Если он равен значению null, то выдает ошибку, если нет – объект класса «Коллективы». Метод «Post Delete» принимает значение Guid id, удаляет данные равные этому id, сохраняет изменения и перенаправляет на домашнюю страницу.

Все методы показаны на рисунке 4.



а)



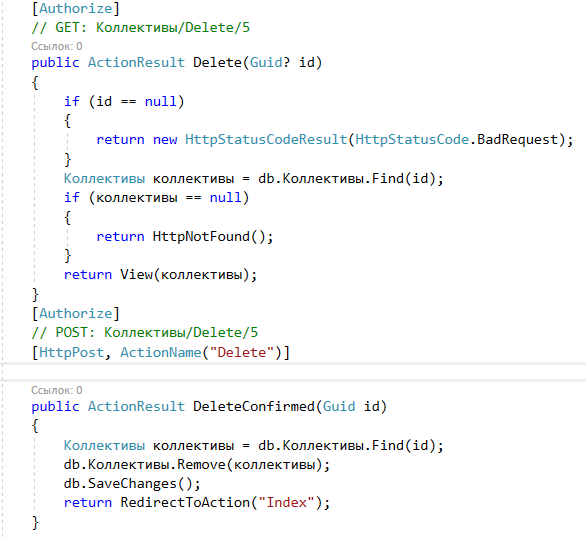
б)



в)



г)



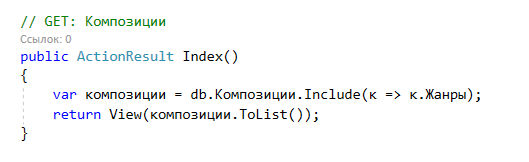
д)

Рисунок 4 – Методы контроллера «КоллективыController»: а – Index, б – Details, в – Create, г – Edit, д – Delete.

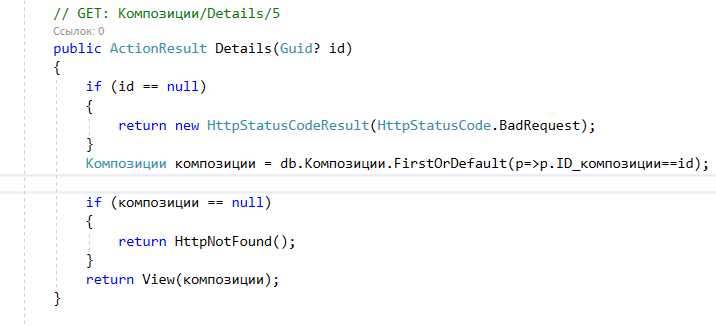
Контроллер «КомпозицииController» позволяет отображать данные, связанные с сущностью «Композиции» и имеет следующие методы:

1. Метод «Index» отображает список композиций, существующих в базе данных, возвращает композиции и жанры.
2. Метод «Details» позволяет более подробно увидеть информацию о конкретной композиции. Принимает Guid id и возвращает объект класса «Композиции» в случае если он не равен значению null, если null- то выводит ошибку.
3. Метод «Create» позволяет создавать новые композиции с выбором жанра. Метод «Post Create» принимает и возвращает объект класса «Композиции».
4. Метод «Edit» позволяет вносить изменения в существующие композиции. Метод «Get Edit» принимает значение Guid id. Если он равен значению null, то выдает ошибку, если нет – объект класса «Композиции». Метод «Get Edit» принимает и возвращает объект класса «Композиции».
5. Метод «Delete» позволяет удалять существующие композиции. Метод «Get Delete» принимает значение Guid id. Если он равен значению null, то выдает ошибку, если нет – объект класса «Композиции». Метод «Post Delete» принимает значение Guid id, удаляет данные равные этому id, сохраняет изменения и перенаправляет на домашнюю страницу.

Все методы показаны на рисунке 5.



а)



б)



в)



г)



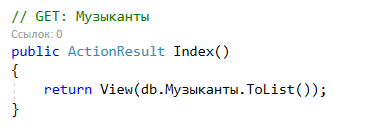
д)

Рисунок 5 – Методы контроллера «КомпозицииController»: а – Index, б – Details, в – Create, г – Edit, д – Delete.

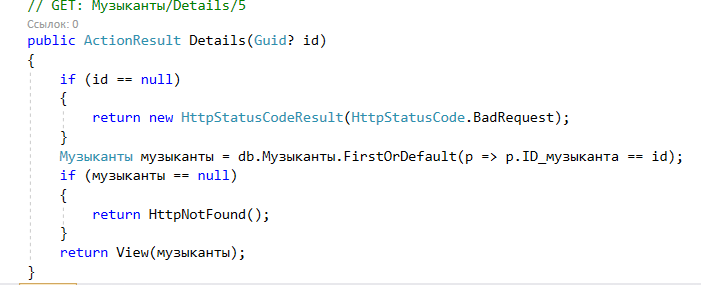
Контроллер «МузыкантыController» позволяет отображать данные, связанные с сущностью «Музыканты» и имеет следующие методы:

1. Метод «Index» отображает список музыкантов, существующих в базе данных, возвращает музыканты.
2. Метод «Details» позволяет более подробно увидеть информацию о конкретном музыканте. Принимает Guid id и возвращает объект класса «Музыканты» в случае если он не равен значению null, если null- то выводит ошибку.
3. Метод «Create» позволяет создавать новых музыкантов. Метод «Post Create» принимает и возвращает объект класса «Музыканты».
4. Метод «Edit» позволяет вносить изменения в существующие композиции. Метод «Get Edit» принимает значение Guid id. Если он равен значению null, то выдает ошибку, если нет – объект класса «Музыканты». Метод «Get Edit» принимает и возвращает объект класса «Музыканты».
5. Метод «Delete» позволяет удалять существующих музыкантов. Метод «Get Delete» принимает значение Guid id. Если он равен значению null, то выдает ошибку, если нет – объект класса «Музыканты». Метод «Post Delete» принимает значение Guid id, удаляет данные равные этому id, сохраняет изменения и перенаправляет на домашнюю страницу.

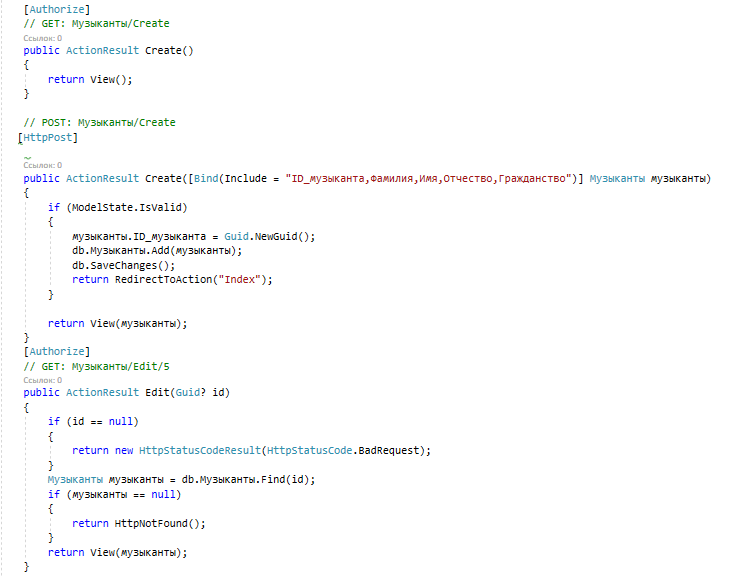
Все методы показаны на рисунке 6.



а)



б)



в)



г)



д)

Рисунок 6 – Методы контроллера «МузыкантыController»: а – Index, б – Details, в – Create, г – Edit, д – Delete.

**Описание реализованного функционала**

При запуске web – приложения, web – приложение предлагает нам авторизоваться (Рисунок 7).

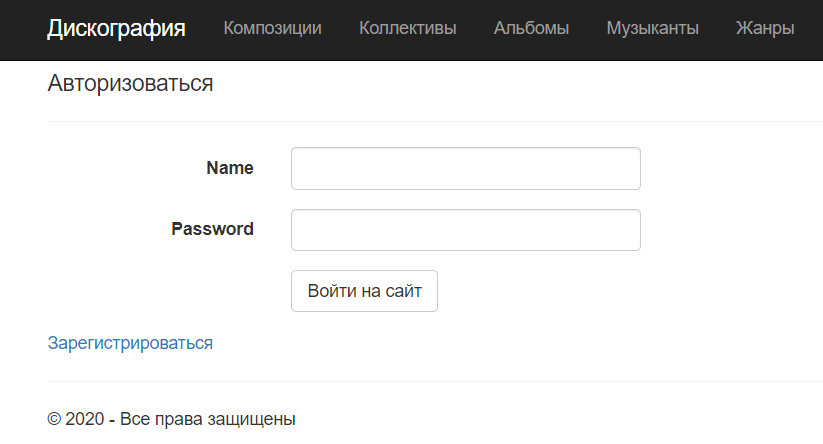


Рисунок 7 – Начальная страница

При нажатии на кнопку «Зарегистрироваться» нас перекидывает на страницу с регистрацией (Рисунок 8).

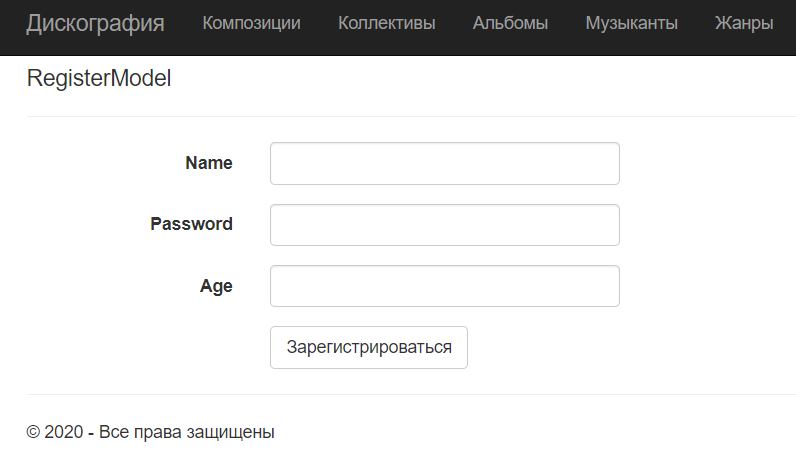


Рисунок 8 – Страница с регистрацией

В шапке сайта находится 5 раздела, при нажатии «Композиции» мы переходим на страницу, где показывается список композиций (Рисунок 9)

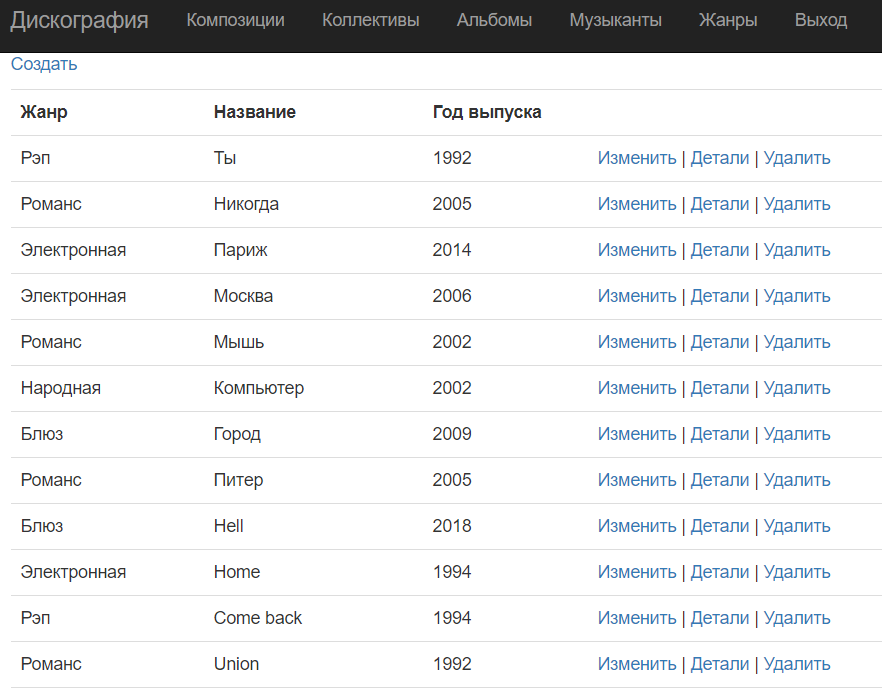


Рисунок 9 – Страница «Композиции»

При нажатии «Изменить» переходим на страницу, где можем изменить данные (Рисунок 10).

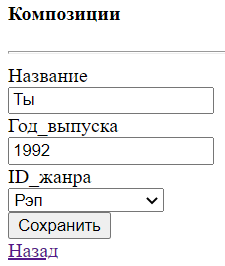


Рисунок 10 – Страница «Изменить» в композициях

При нажатии «Детали» переходим на страницу, где можем смотреть данные. При нажатии на определенного композитора или автора, web – приложение перебросит на личную страницу композитора или автора. (Рисунок 11).

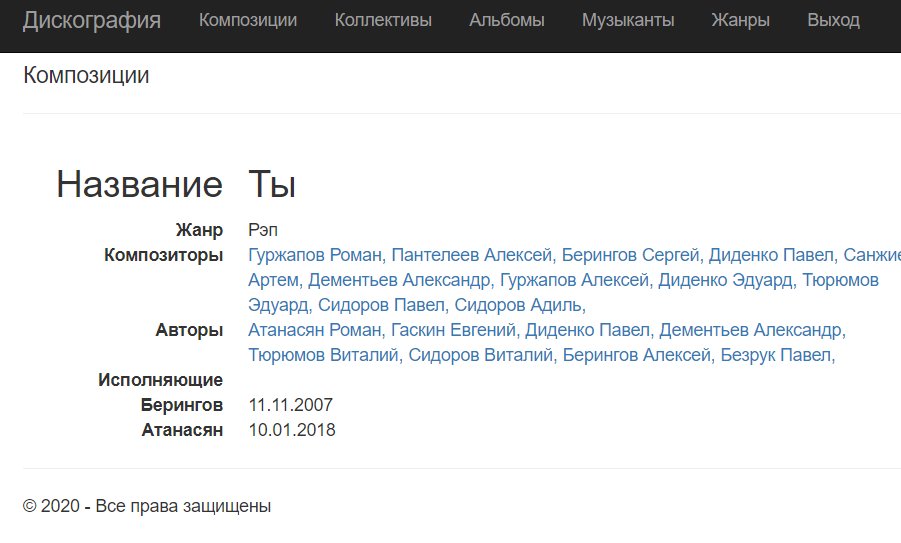


Рисунок 11 – Страница «Детали» в композициях

При нажатии «Удалить» переходим на страницу, где можем удалить данные (Рисунок 12).

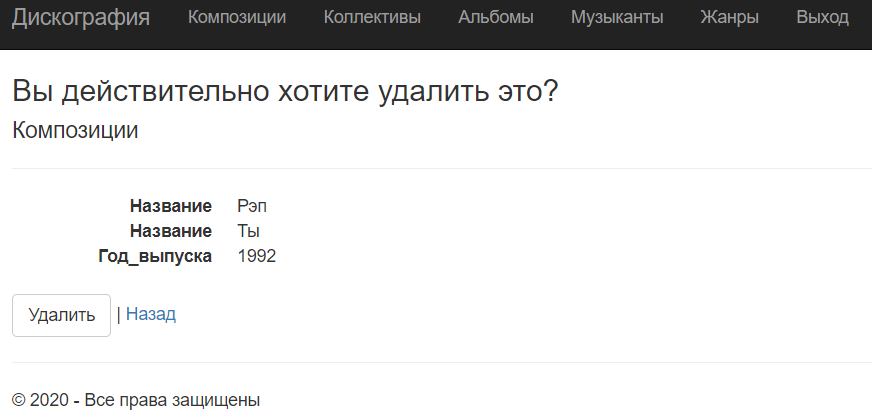


Рисунок 12 – Страница «Удалить» в композициях

При нажатии «Создать» переходим на страницу, где можем создать композицию (Рисунок 13).

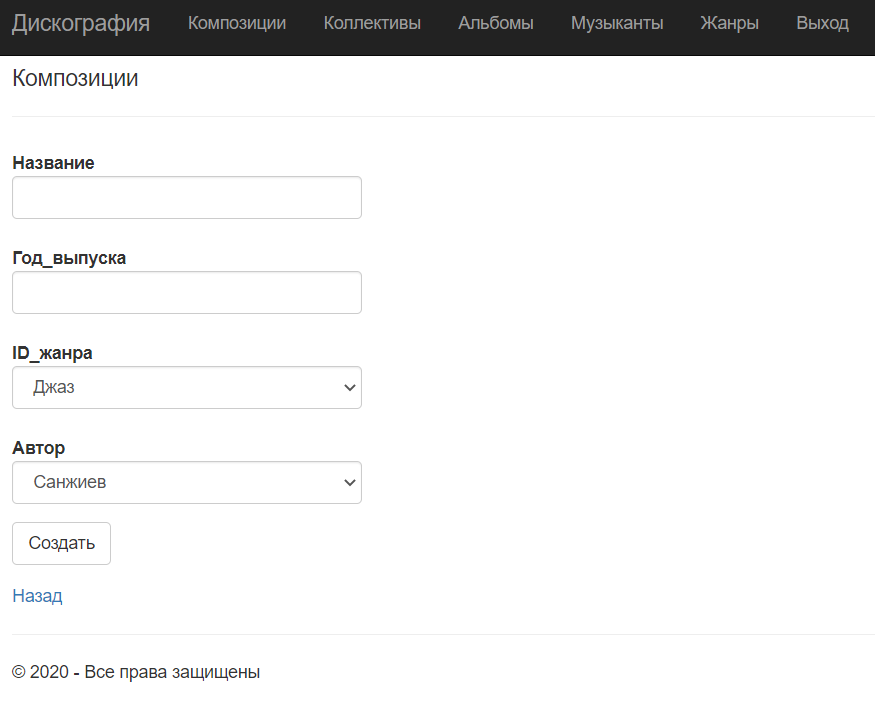


Рисунок 13 – Страница «Создать» в композициях

При нажатии «Коллективы» в шапке страницы выводится список коллективов (Рисунок 14).

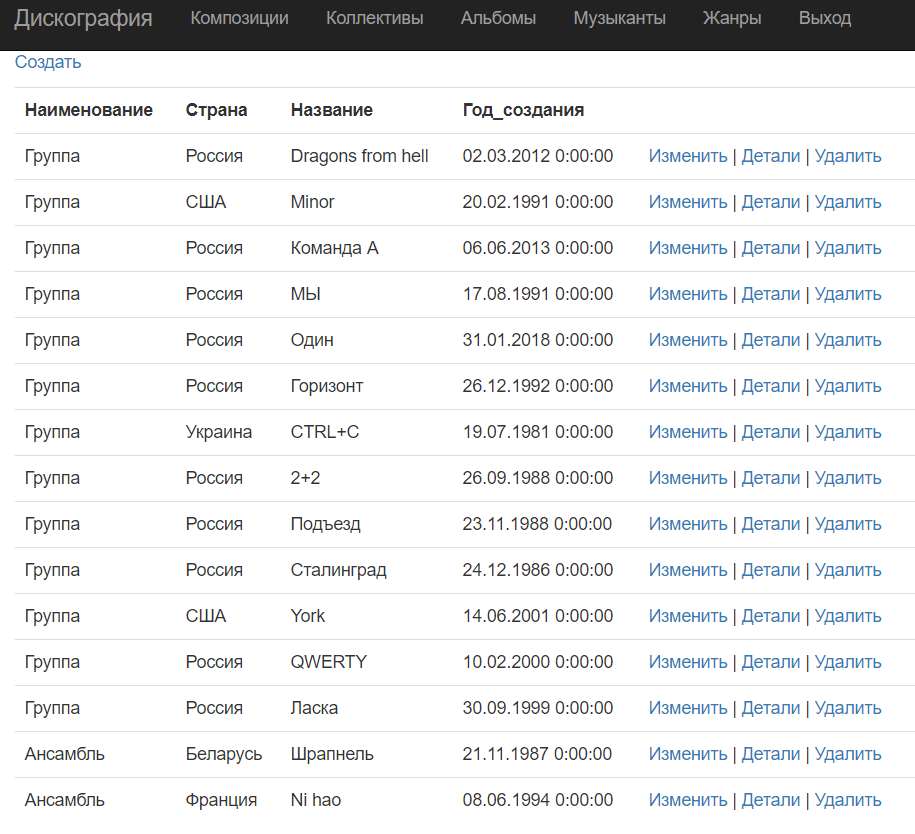


Рисунок 14 – Страница «Коллективы»

При нажатии «Изменить» переходим на страницу, где можем изменить данные (Рисунок 15).

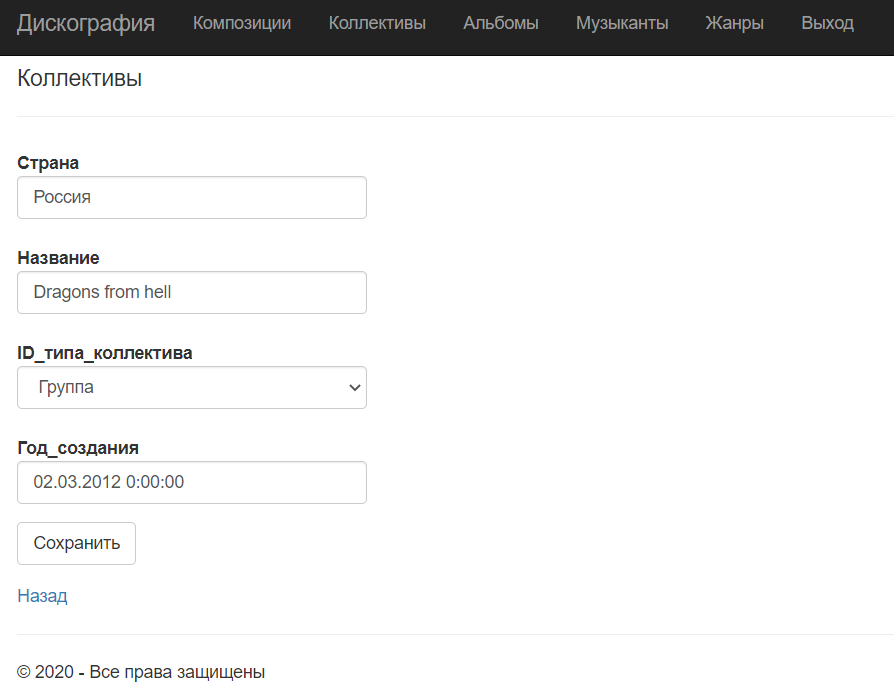


Рисунок 15 – Страница «Изменить» в коллективах

При нажатии «Детали» переходим на страницу, где можем смотреть данные. При нажатии на определенного музыканта, web – приложение перебросит на личную страницу музыканта. (Рисунок 16).

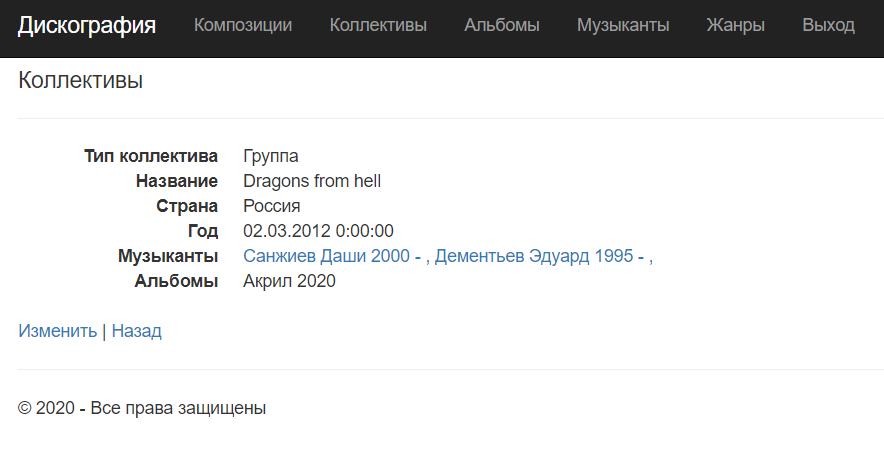


Рисунок 16 – Страница «Детали» в коллективах

При нажатии «Удалить» переходим на страницу, где можем удалить данные (Рисунок 17).

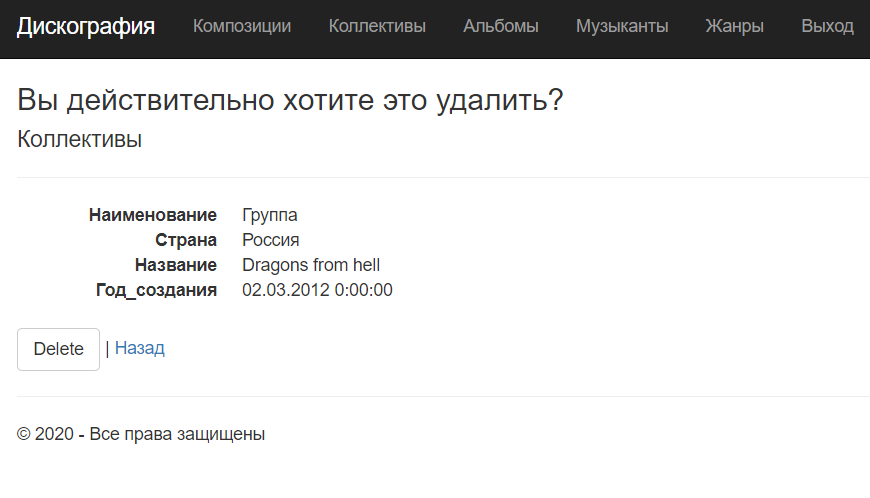


Рисунок 17 – Страница «Удалить» в коллективах

При нажатии «Создать» переходим на страницу, где можем создать коллектив (Рисунок 18).

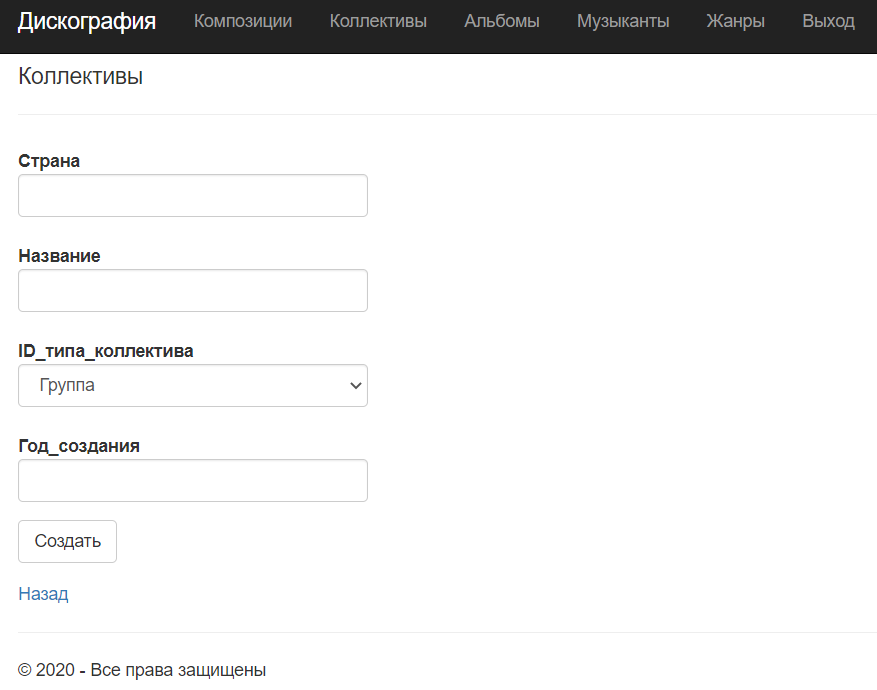


Рисунок 18 – Страница «Создать» в коллективах

При нажатии «Альбомы» в шапке страницы выводится список альбомов (Рисунок 19).

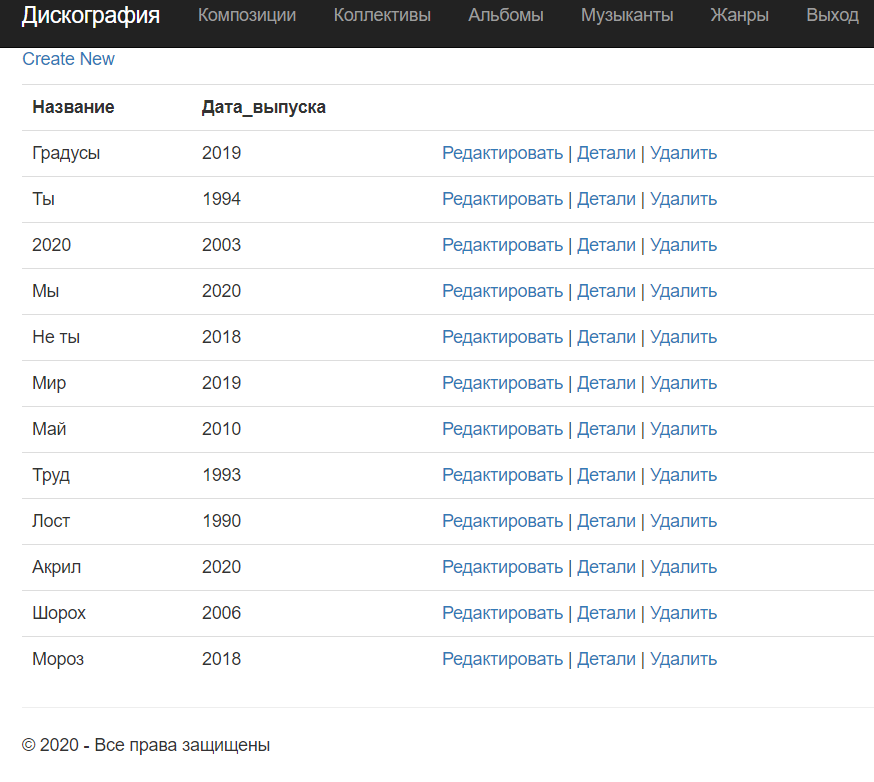


Рисунок 19 – Страница «Альбомы»

При нажатии «Изменить» переходим на страницу, где можем изменить данные (Рисунок 20).

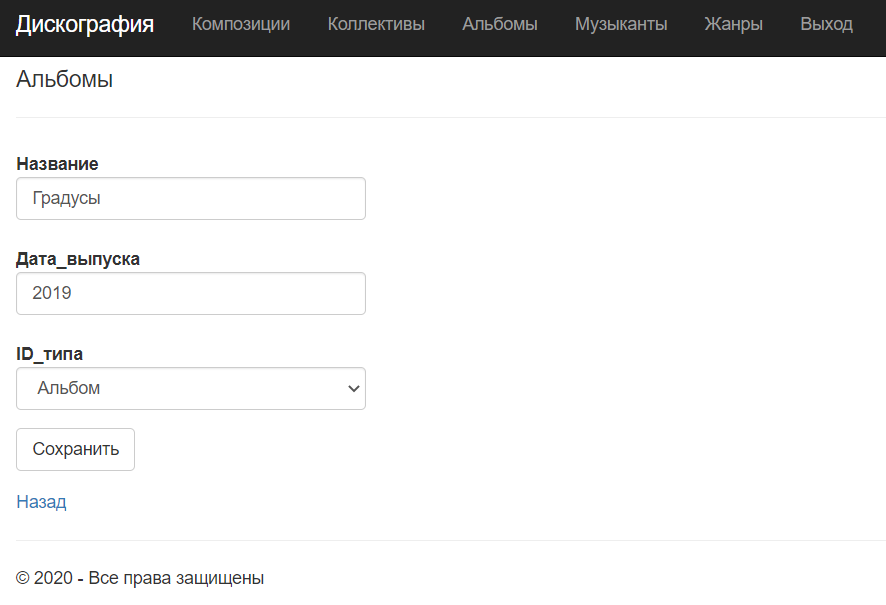


Рисунок 20 – Страница «Изменить» в альбомах

При нажатии «Детали» переходим на страницу, где можем смотреть данные. При нажатии на определенную композицию, web – приложение перебросит на личную страницу композиции. (Рисунок 21).

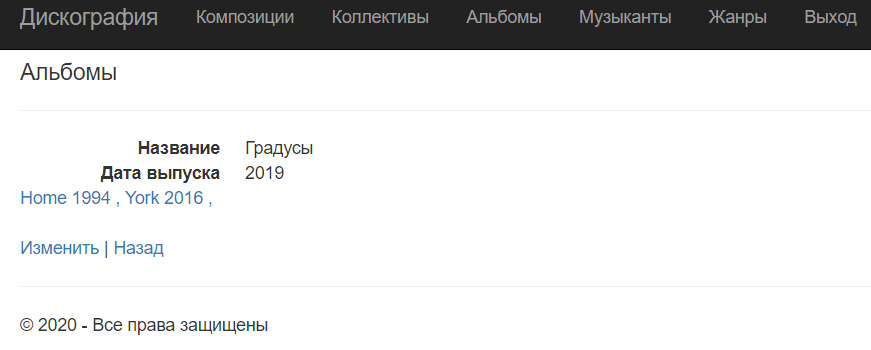


Рисунок 21 – Страница «Детали» в альбомах

При нажатии «Удалить» переходим на страницу, где можем удалить данные (Рисунок 22).

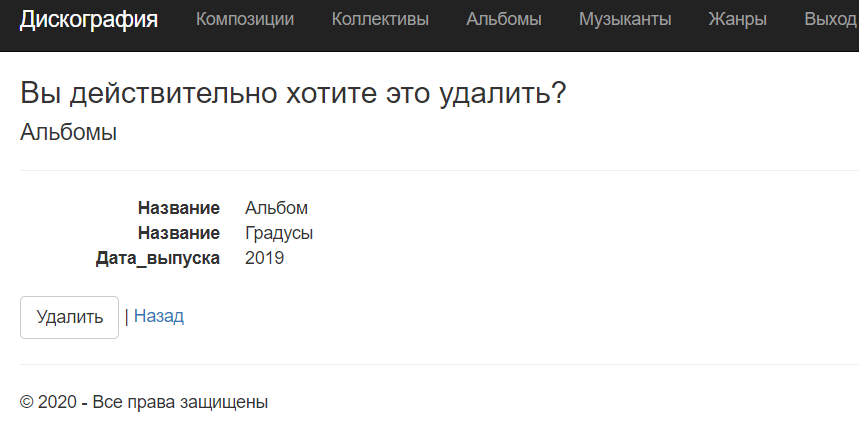


Рисунок 22 – Страница «Удалить» в альбомах

При нажатии «Создать» переходим на страницу, где можем создать альбом (Рисунок 23).

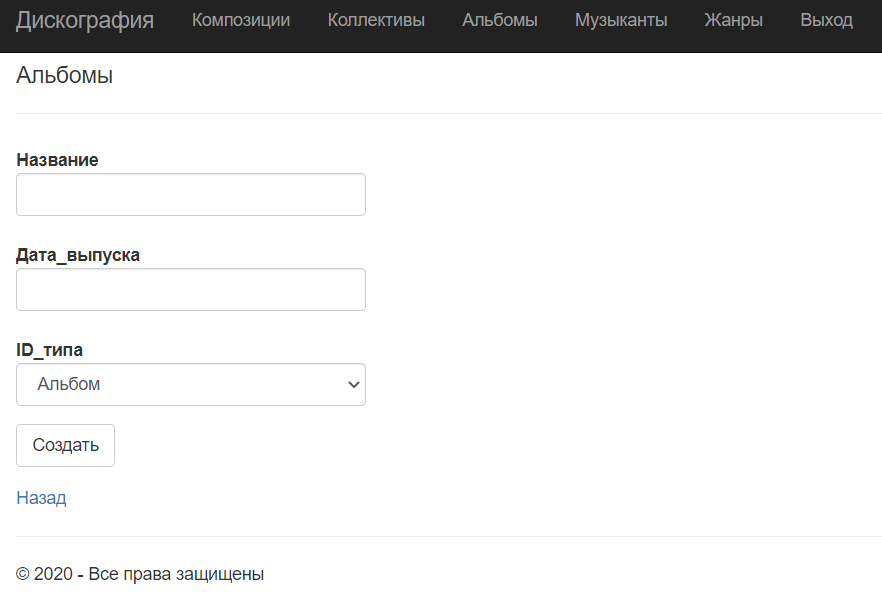


Рисунок 23 – Страница «Создать» в альбомах

При нажатии «Музыканты» в шапке страницы выводится список музыкантов (Рисунок 24).

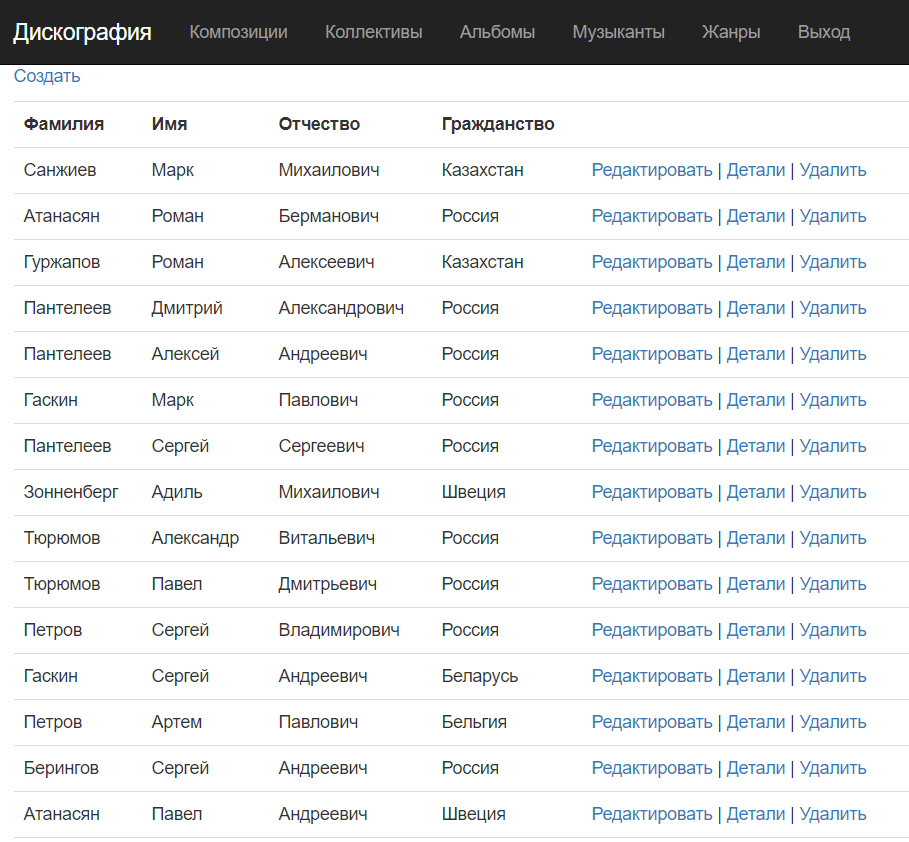


Рисунок 24– Страница «Музыканты»

При нажатии «Изменить» переходим на страницу, где можем изменить данные (Рисунок 25).

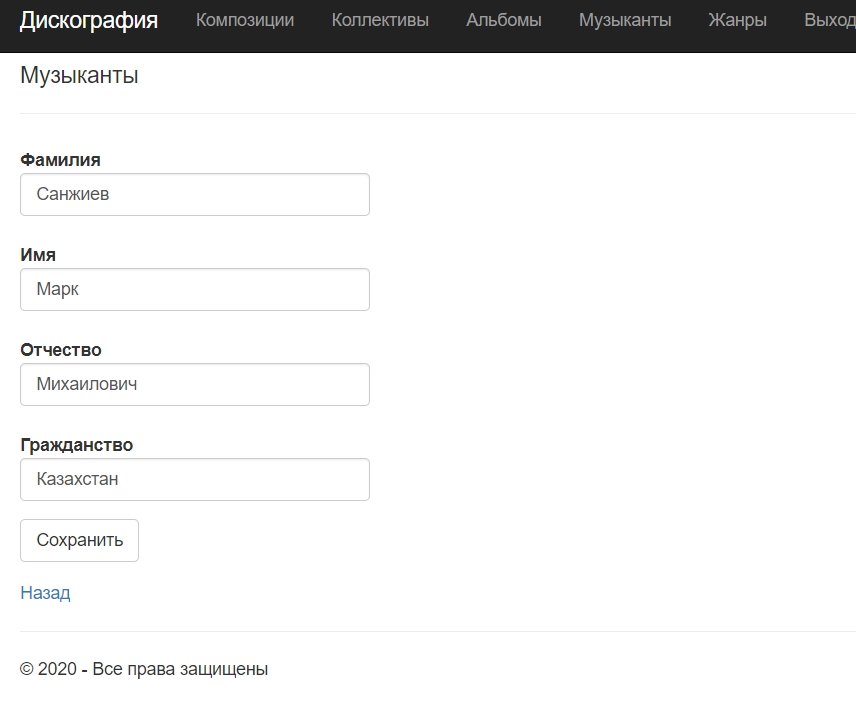


Рисунок 25 – Страница «Изменить» в музыкантах

При нажатии «Детали» переходим на страницу, где можем смотреть данные. При нажатии на определенного коллектива, web – приложение перебросит на личную страницу коллектива. (Рисунок 26).

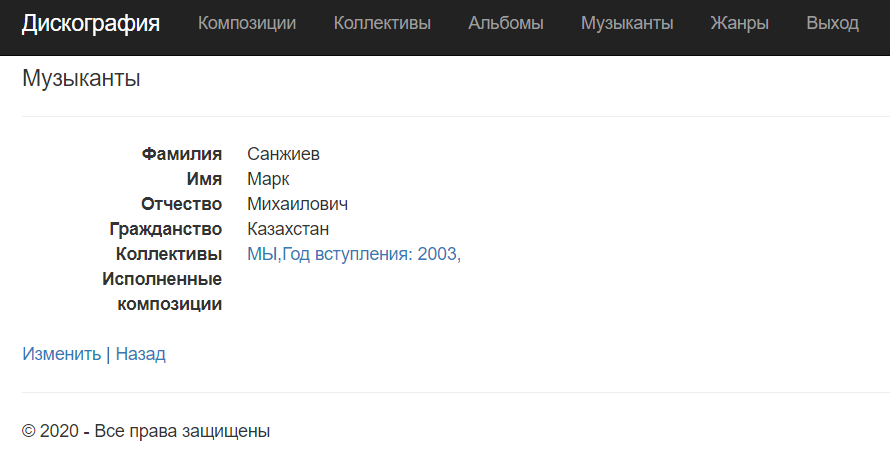


Рисунок 26 – Страница «Детали» в музыкантах

При нажатии «Удалить» переходим на страницу, где можем удалить данные (Рисунок 27).

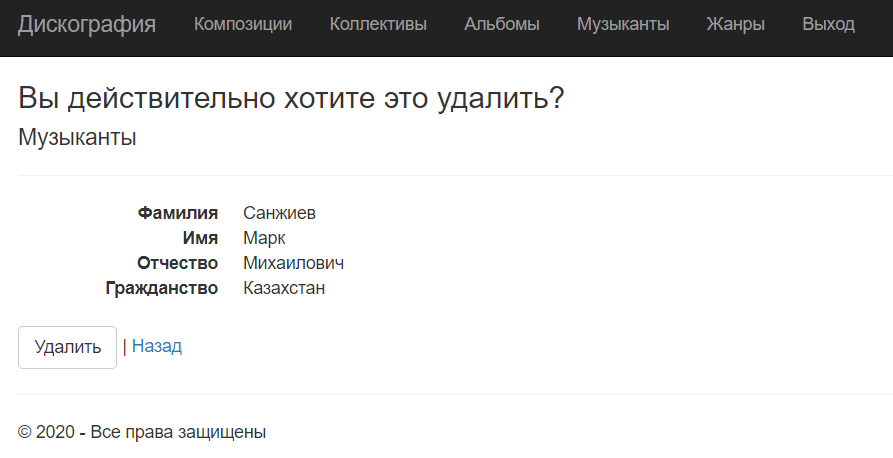


Рисунок 27 – Страница «Удалить» в музыкантах

При нажатии «Создать» переходим на страницу, где можем создать музыканта (Рисунок 28).

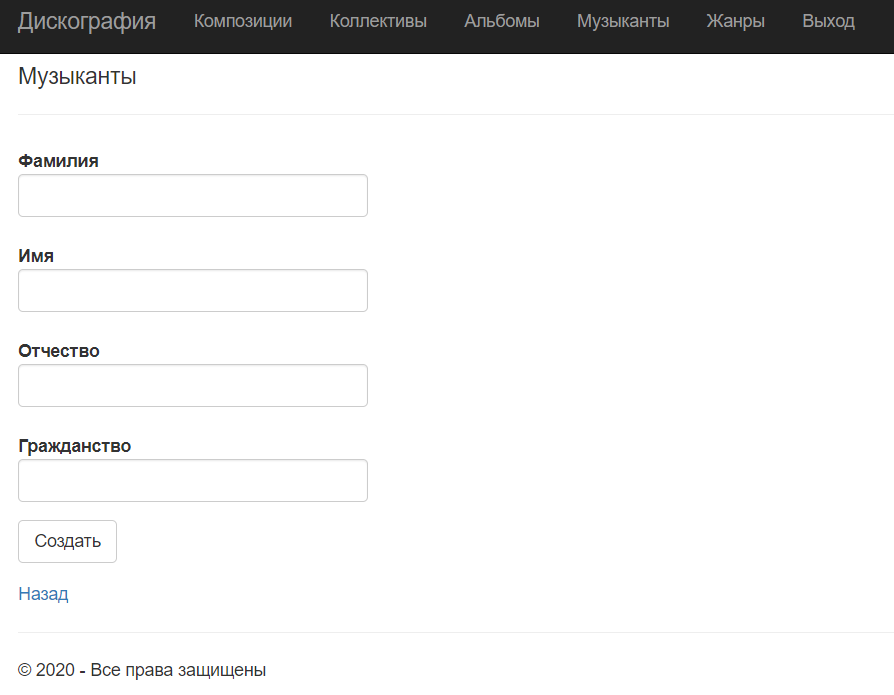


Рисунок 28 – Страница «Создать» в музыкантах

При нажатии «Жанры» в шапке страницы выводится список жанров(Рисунок 29).

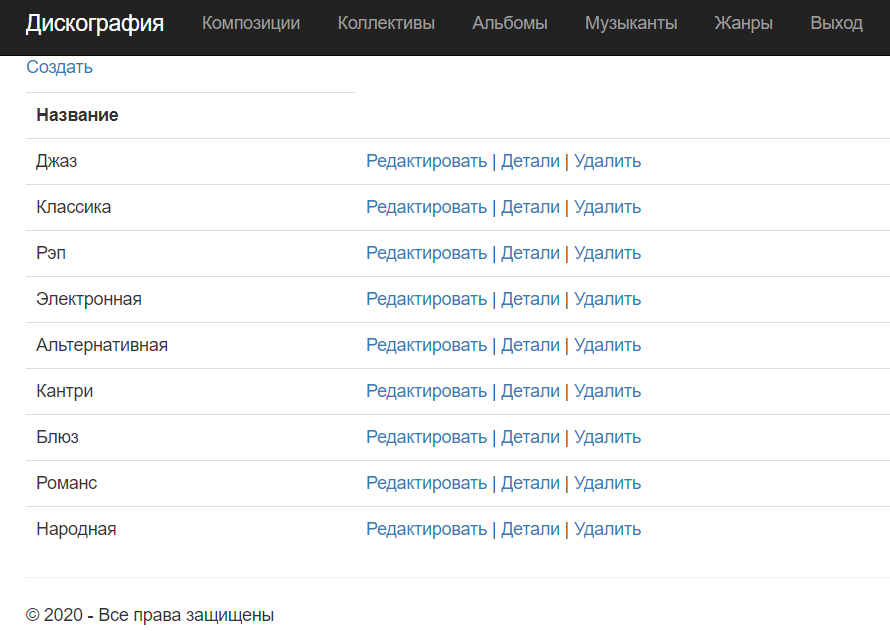


Рисунок 29 – Страница «Жанры»

При нажатии «Изменить» переходим на страницу, где можем изменить данные (Рисунок 30).

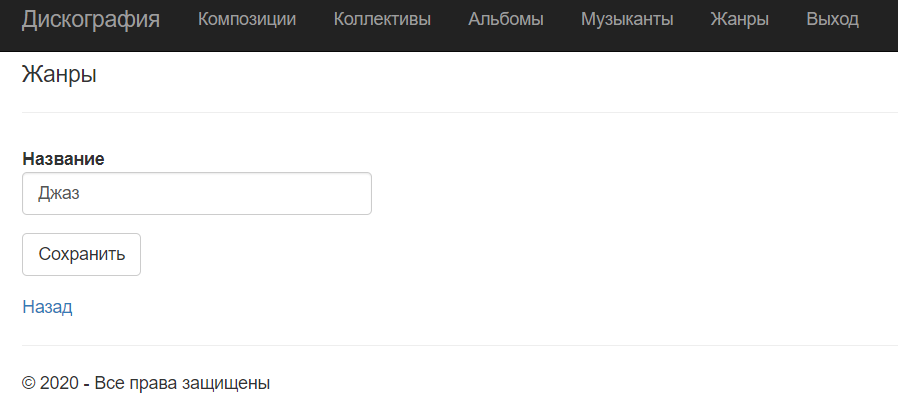


Рисунок 30 – Страница «Изменить» в жанрах

При нажатии «Детали» переходим на страницу, где можем смотреть данные. При нажатии на определенную композицию, web – приложение перебросит на личную страницу композиции. (Рисунок 31).

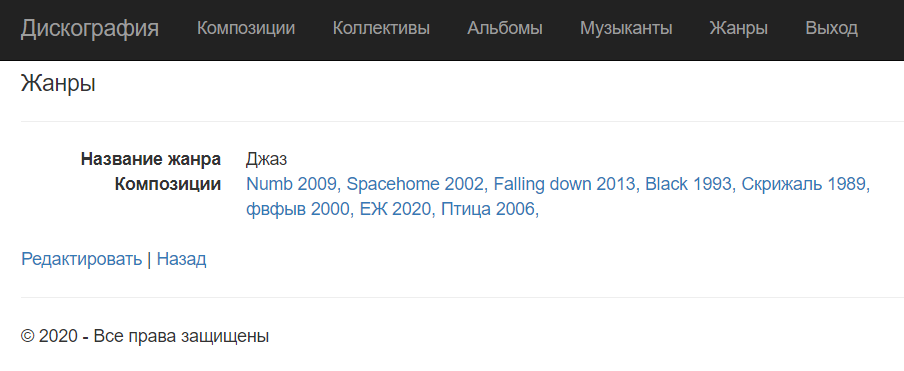


Рисунок 31 – Страница «Детали» в жанрах

При нажатии «Удалить» переходим на страницу, где можем удалить данные (Рисунок 32).

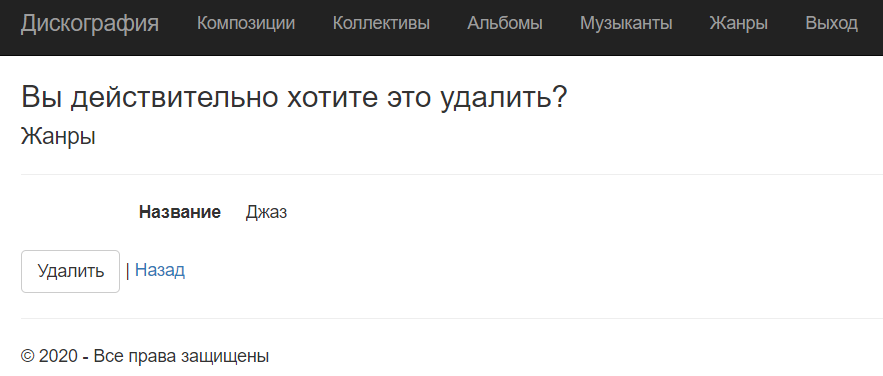


Рисунок 32 – Страница «Удалить» в жанрах

При нажатии «Создать» переходим на страницу, где можем создать жанр (Рисунок 33).

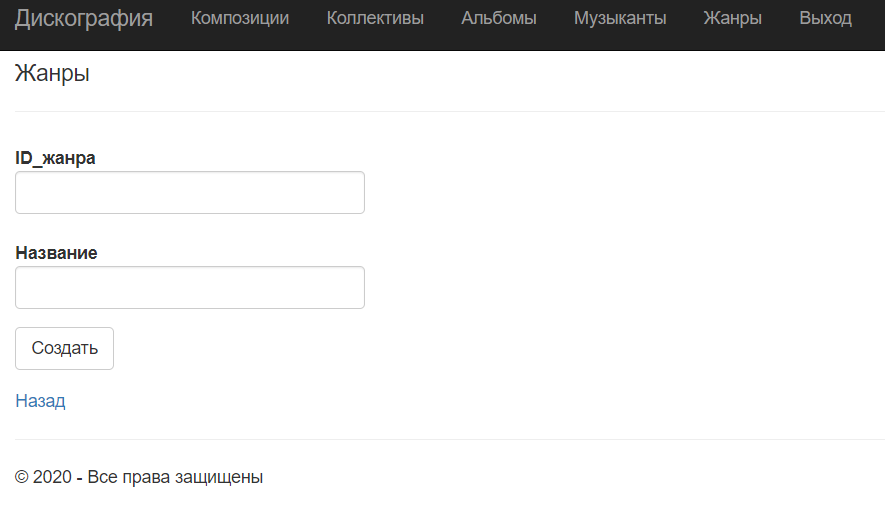


Рисунок 23 – Страница «Создать» в жанрах

При нажатии «Выход» в шапке страницы, вы выходить из аккаунта и переходим к домашней странице (Рисунок 24)

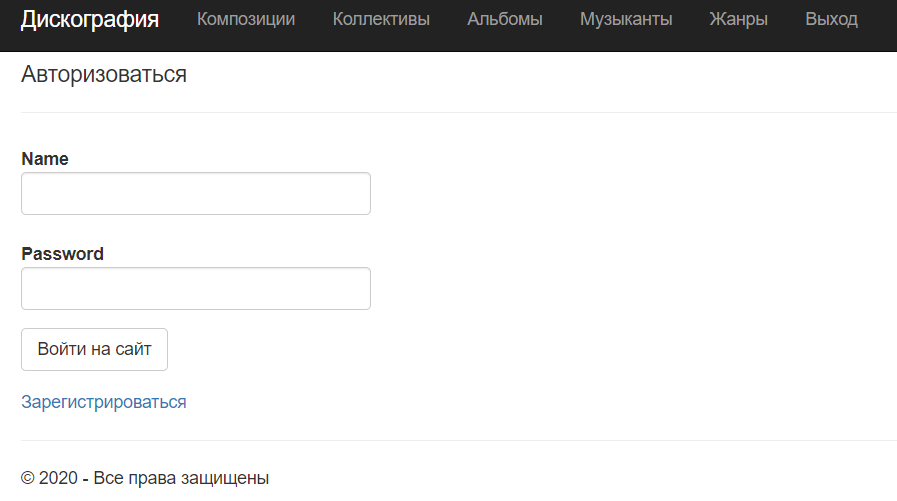


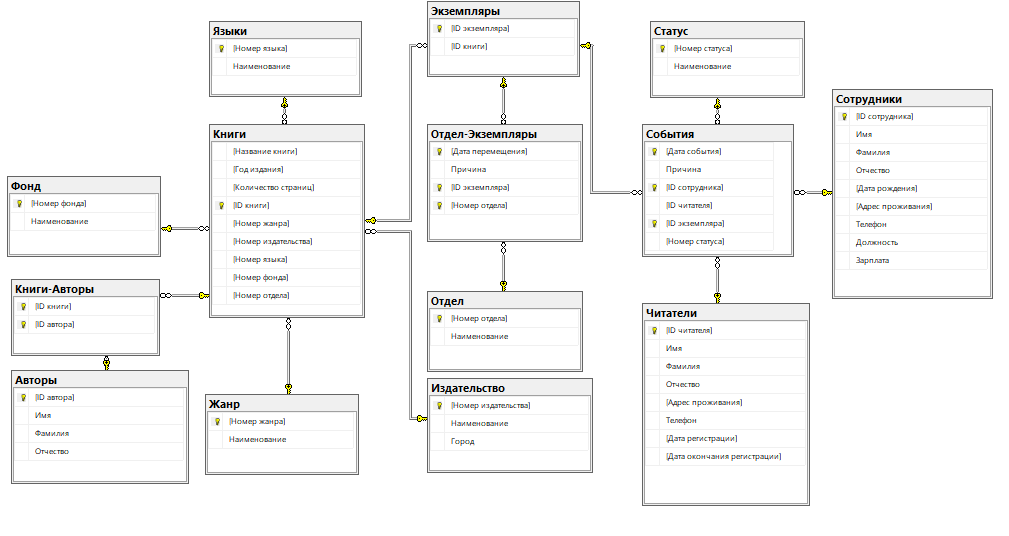
Рисунок 24 – Домашняя страница при нажатии «Выход»

Вывод

В ходе данной курсовой работы были получены основные навыки для работы с web-приложением. На основе созданной ранее базе данных и языках программирования C#, HTML было спроектировано web-приложение. Были спроектированы нужные модели, отражающие сущности и их связи в базе данных, контроллеры и представления.

Приложение А. Диаграммы

Рисунок А.2 – Логическая схема базы данных



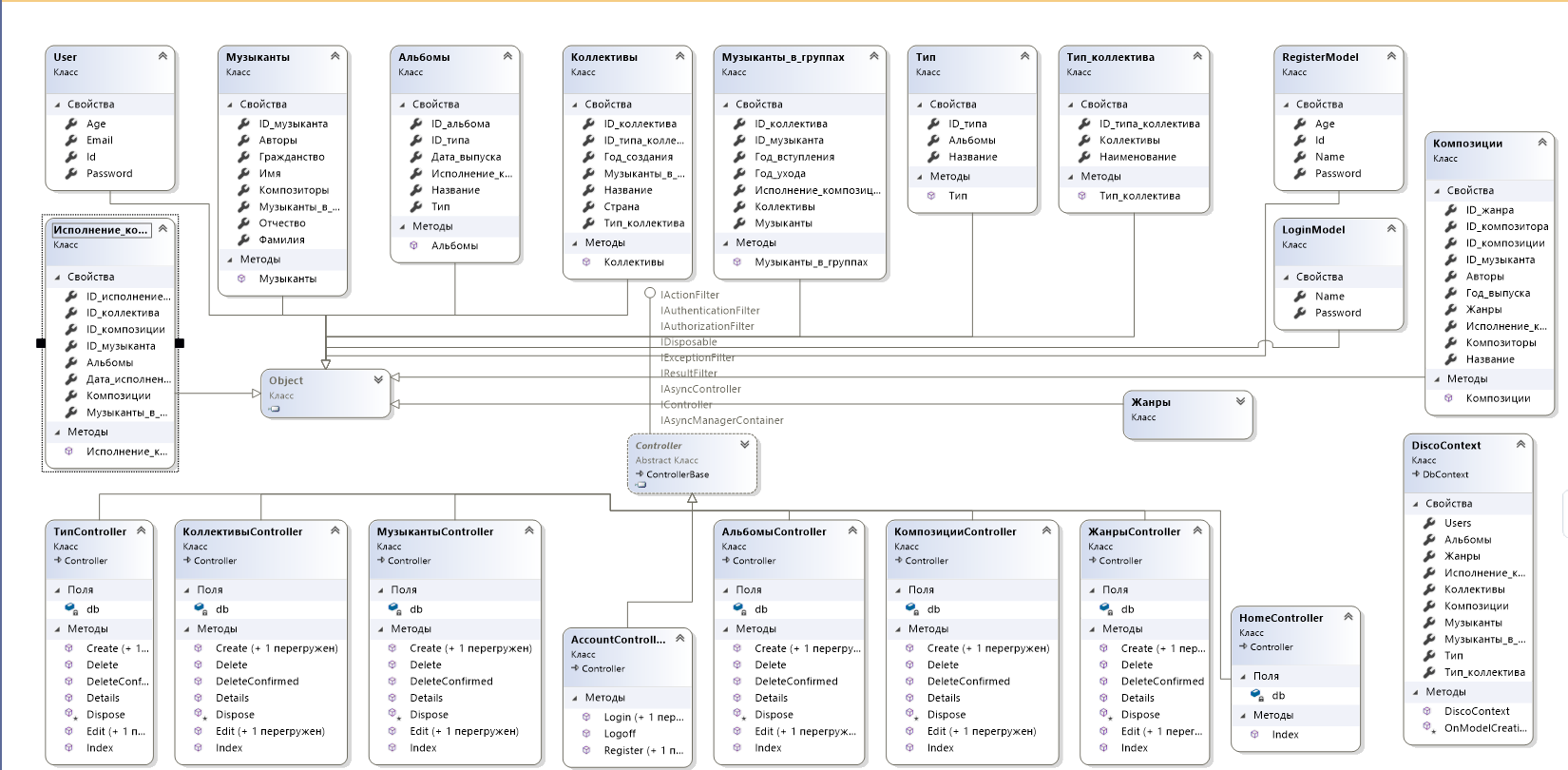


Рисунок A.1 – Диаграмма классов приложения