

При синтезе аудиокниг несомненным преимуществом пользуется компилиативная модель синтеза, поскольку она дает более выразительно звучание при условии наличия качественных образцов отдельных звуков, и при этом требует меньше вычислительных ресурсов нежели формантно-голосовая.

## 1.2 Аналоги, их недостатки и достоинства

Веб-приложение для синтеза, хранения и распространения аудиокниг – это программное средство, хранящее аудио версии литературных произведений, и способное их синтезировать на основе печатного текста. С его помощью можно озвучивать книги различными голосами, а также распространять созданные аудиокниги среди интернет-сообщества. Кроме того, веб-приложение позволяет скачивать хранимые и распространяемые аудиокниги на устройство. Для использования приложения не требуется устанавливать дополнительно программное обеспечение на устройство, достаточно любого браузера.

Также важным достоинством является возможность сохранения аудиокниг на сервере приложения, с привязкой к аккаунту пользователя. Очень удобным является наличие функционала, позволяющего проводить поиск и фильтрацию распространяемых на сайте аудиокниг.

Так как полных аналогов приложения для синтеза, хранения и распространения аудиокниг не существует, в данном разделе будут рассмотрены два вида частичных аналогов: приложения для хранения и распространения аудиокниг и приложения для озвучивания текста.

Одним из самых популярных приложений хранения и распространения аудиокниг, является «Audio-knigki.com» (рисунок 1.1) [6].

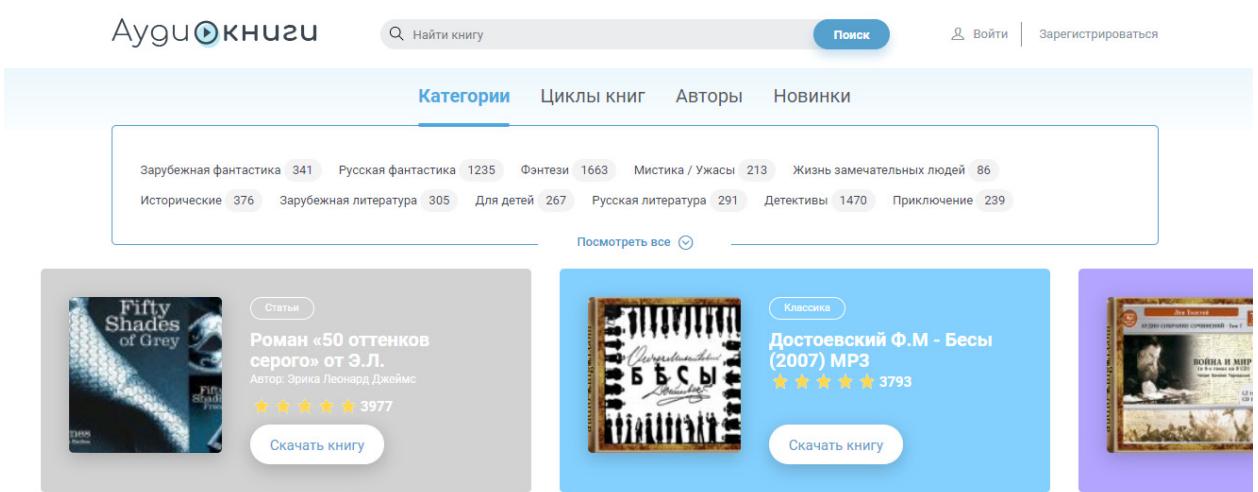


Рисунок 1.1 – Приложение для хранения и распространения аудиокниг «Audio-knigki.com»

Это бесплатное приложение, в котором можно прослушивать и скачивать

вать различные аудиокниги. Здесь можно найти литературу на любой вкус: детективы, нон-фикшн, фантастика, сказки, любовные романы, аудиоспектакли и другое. Есть возможность выбора аудиокниг по жанру, автору или циклам.

Для зарегистрированных пользователей имеется возможно добавить на сайт собственную версию аудиокниги. Или заказать добавление книги у администрации сайта.

В приложении предусмотрены возможность оценивания аудиокниг и добавления комментариев. Так же имеется возможность сортировки аудиокниг по рейтингу и времени добавления.

У данного приложения можно выделить следующие преимущества:

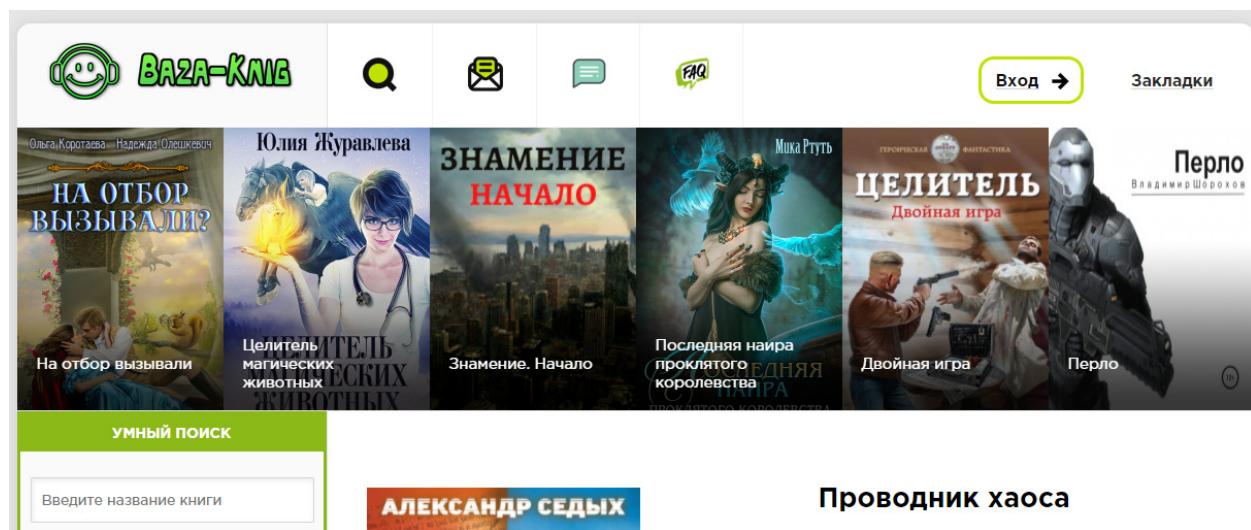
- возможность прослушать отрывок аудиокниги прямо на сайте;
- возможность добавлять на сайт собственные аудиокниги;
- возможность сортировки и фильтрации аудиокниг;
- возможность заказать добавление аудиокниги у администрации.

Недостатки:

- нельзя загрузить аудиофайл напрямую, на сайте размещены только ссылки на скачивание со сторонних ресурсов;
- часть аудиокниг доступна только на платной основе.

В целом, «Audio-knigki.com» – это одно из лучших приложений для хранения и распространения аудиокниг.

Ещё одним популярным приложением для хранения и распространения аудиокниг является «Baza-Knig» (рисунок 1.2) [7].



В приложении на данный момент доступно более 20 000 аудиокниг. Классика, фантастика, юмористические и детские книги – все это и многое другое в mp3 формата и бесплатно. Кроме аудиокниг разных жанров на сайте

есть аудиоспектакли и курсы по изучению иностранных языков. Найти интересующую литературу можно через поиск или по категориям. Для скачивания аудиокниг не требуется регистрация.

Также имеется возможность прослушивать аудиокниги прямо на сайте. Зарегистрированный пользователь может сохранить понравившуюся аудиокнигу в закладки. В приложении присутствует гибкая система оценивания аудиокниг на основе нескольких параметров.

Преимущества:

- возможность полностью прослушать аудиокнигу без скачивания;
- гибкая система оценок;
- возможность скачивания аудиокниги прямо с сайта;
- все аудиокниги абсолютно бесплатны;
- возможность сортировки и фильтрации аудиокниг.

Недостатки:

- нельзя добавить на сайт собственную версию аудиокниги;
- отсутствует возможность заказать добавление книги у администрации.

«Baza-Knig» несомненно одно из лучших приложений для хранения и распространения аудиокниг как в плане количества аудиокниг, так и удобства использования. Для тех, кто хочет не только скачать аудиокнигу, но и прослушать произведение прямо на сайте, это прекрасный выбор.

Также стоит обратить внимание на такое приложение как «Au-books» (рисунок 1.3) [8].

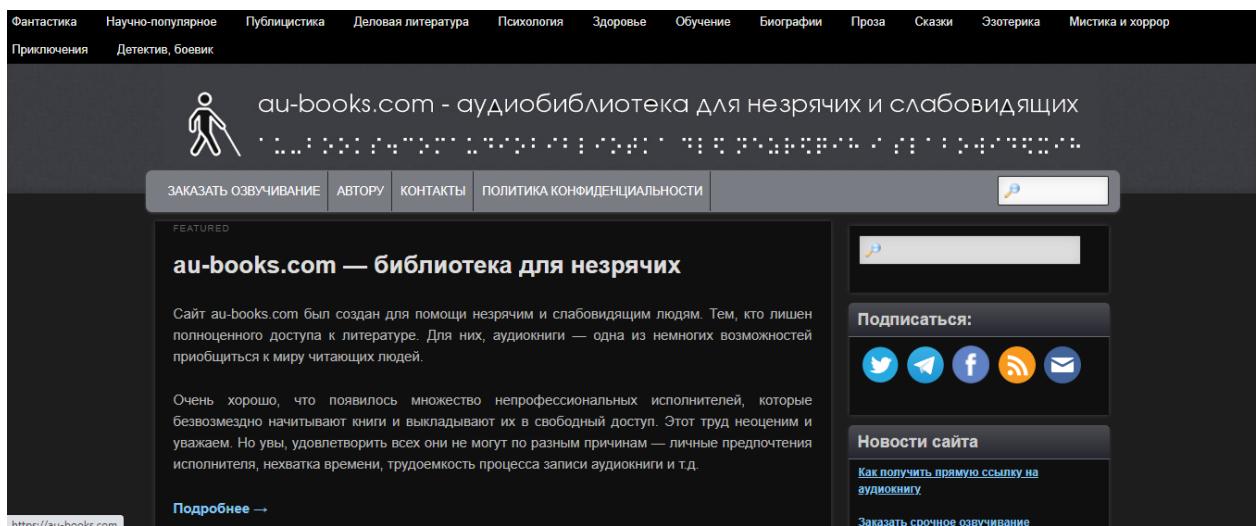


Рисунок 1.3 – Приложение для хранения и распространения аудиокниг «Au-books»

Этот сервис предназначен в первую очередь для незрячих и слабовидящих людей. Для них аудиокниги — это возможность приобщиться к миру читающих людей. Сегодня на сайте есть множество книг: фантастика, научно-популярная, детская, приключение, биографии, обучение и многое

другое. Найти интересующую литературу можно через поиск по названию.

Скачивание аудиокниг бесплатно и не требует регистрации. Имеется возможность заказать синтез аудиокниги, с использованием одного из 7 голосов, представленных на сайте. Синтез книги возможен на русском и украинском языках. Среднее время озвучивания одной книги – сутки.

На данный момент на сайте размещено около 100 000 аудиокниг.

Преимущества:

- возможность заказать озвучивание аудиокниги;
- огромное количество аудиокниг, размещенных на сайте;
- возможность сортировки и фильтрации аудиокниг;
- все аудиокниги абсолютно бесплатны.

Недостатки:

- нельзя загрузить аудиофайл напрямую, на сайте размещены только ссылки на скачивание со сторонних ресурсов;
- нельзя добавить на сайт собственную версию аудиокниги;
- непонятный интерфейс.

«Au-books» обладает наибольшим объемом фонда аудиокниг среди аналогов, и в целом является одним из лучших приложений для хранения и распространения аудиокниг. Однако некоторые проблемы вызывает пользовательский интерфейс, в частности непонятен принцип регистрации на сайте.

И так, рассмотрим приложения для озвучивания текста. Одним из наиболее доступных среди них является «VoxWorker» (рисунок 1.4) [9].

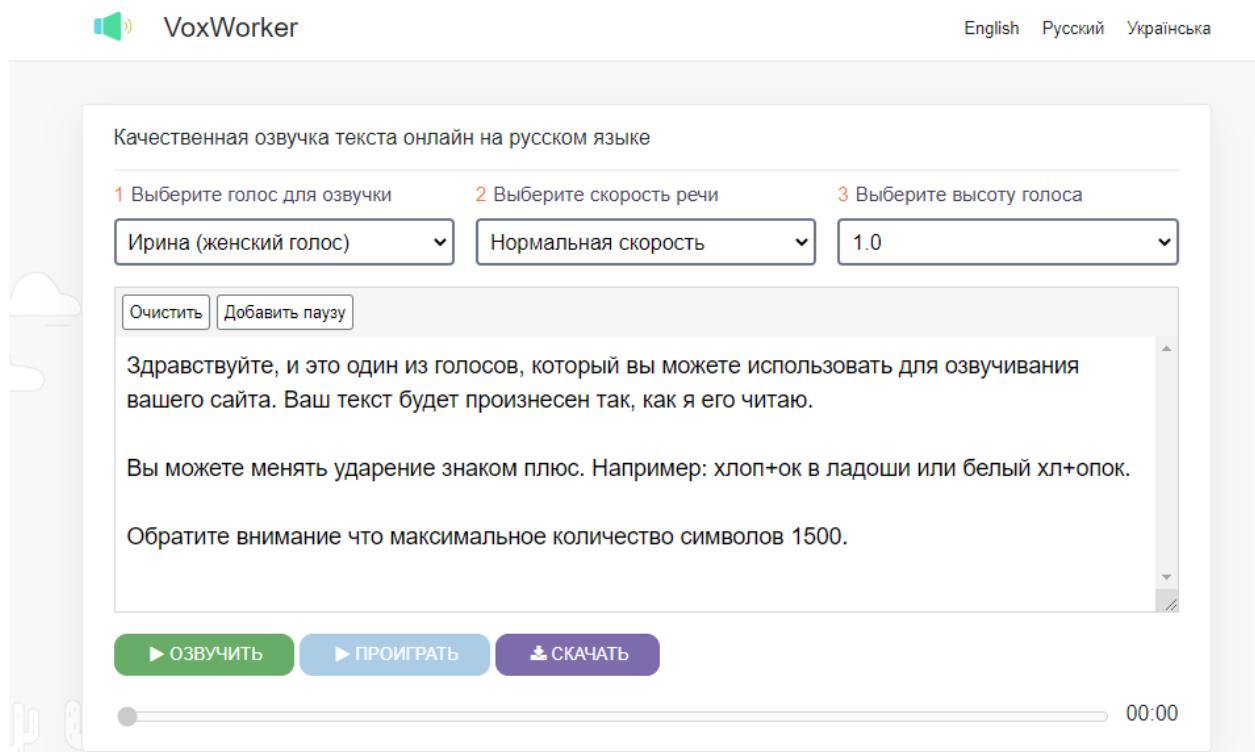


Рисунок 1.4 – Приложения для озвучивания текста «VoxWorker»

«VoxWorker» – это онлайн сервис для озвучки текста, который может переводить текст в аудиозапись. Сайт адаптирован для любых гаджетов, поэтому удобно работать и с мобильной версией. Функционал приложения подразумевает озвучивание небольших фрагментов текста (до 1500 символов).

Для синтеза речи можно выбрать мужской или женский голоса с разным тембром или акцентом. Можно изменять скорость и высоту речи. Озвучивать можно тексты на русском и английском языках.

Результат озвучки можно скачать как файл формата mp3, самого популярного формата для аудиозаписей. Или прослушать прямо в приложении. Сервис не сохраняет тексты для озвучивания. Все голосовые файлы удаляются с сервера через один час.

Преимущества:

- 11 различных голосов для озвучивания;
- возможность настраивать скорость речи и высоту голоса;
- возможность скачать полученный аудио файл;
- возможность прослушать результат прямо на сайте.

Недостатки:

- малый объём озвучиваемого текста;
- нельзя сохранить результат на сервере приложения.

В целом, «VoxWorker» удобный синтезатор речи с минималистичным дизайном и широким функционалом. Однако объем текста, озвучиваемого приложением, за один раз, очень мал.

Так же стоит обратить внимание на такое приложение как «UniTools» (рисунок 1.5) [10].

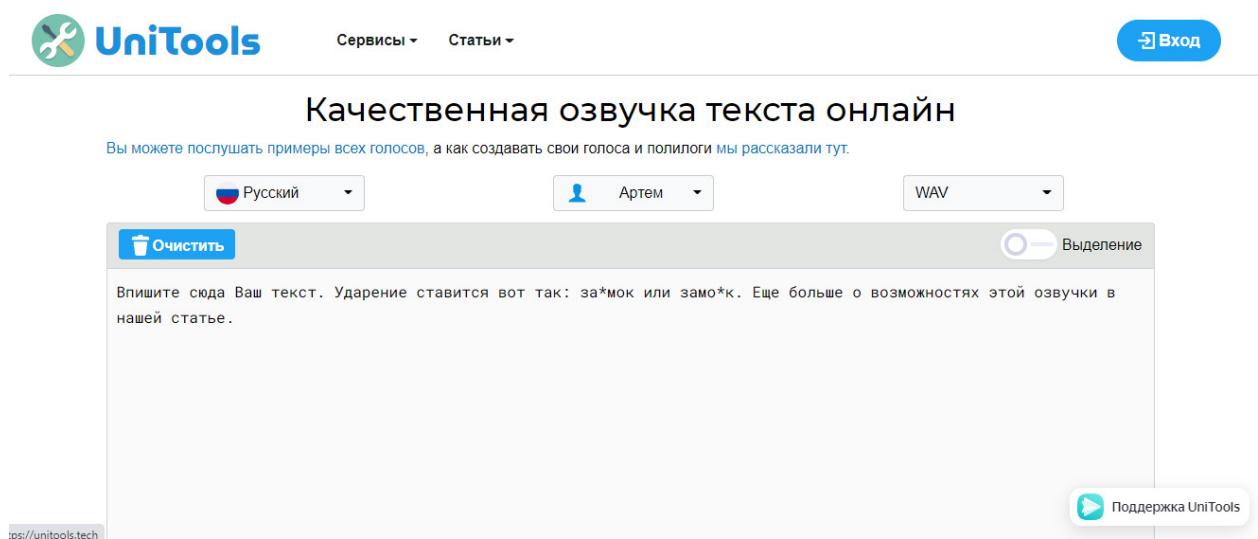


Рисунок 1.5 – Приложения для озвучивания текста «UniTools»

Пожалуй, «Unitools» является одним из наиболее интересных проектов для озвучки. Приложение интуитивно понятно и не вызывает затруднений при использовании даже у начинающих.

Использовать сервис удобно. Озвучка осуществляется голосами, которые легко можно спутать со звучанием речи живого человека. Имеется возможность сохранить результат озвучивания в форматы MP3 и WAV.

Система предусматривает работу в бесплатной и платной версии. Первая отлично подходит для начинающих, а также для тех, кому нужно перевести в аудио формат небольшой фрагмент текста. В день возможно воспроизведение до 750 символов на безвозмездной основе.

Использовать платные возможности сервиса могут зарегистрированные пользователи после пополнения баланса. Стоимость озвучки весьма демократична. Перевод текста в звук обычным голосом обойдется в 1 рубль за 1000 символов. Цена на премиум голоса – 3 рубля за 1000 символов.

Пополнение баланса осуществляется удобным способом и занимает пару минут. Количество доступных для озвучивания символов определяется только балансом на счете.

Преимущества:

- более сорока голосов для озвучивания на семи различных языках;
- возможность скачать полученный аудио файл;
- возможность прослушать результат прямо на сайте;

Недостатки:

- объем озвучиваемого текста в бесплатной версии;
- нельзя сохранить результат на сервере приложения.

«Unitools» хорошее приложение для озвучивания текста с удобным интерфейсом и большим количеством голосов. Однако, при бесплатном использовании, объема озвучиваемого текста хватит лишь на очень короткие тексты.

Ещё одним прекрасным приложением для озвучивания текста является «Zvukogram» (рисунок 1.6) [11].

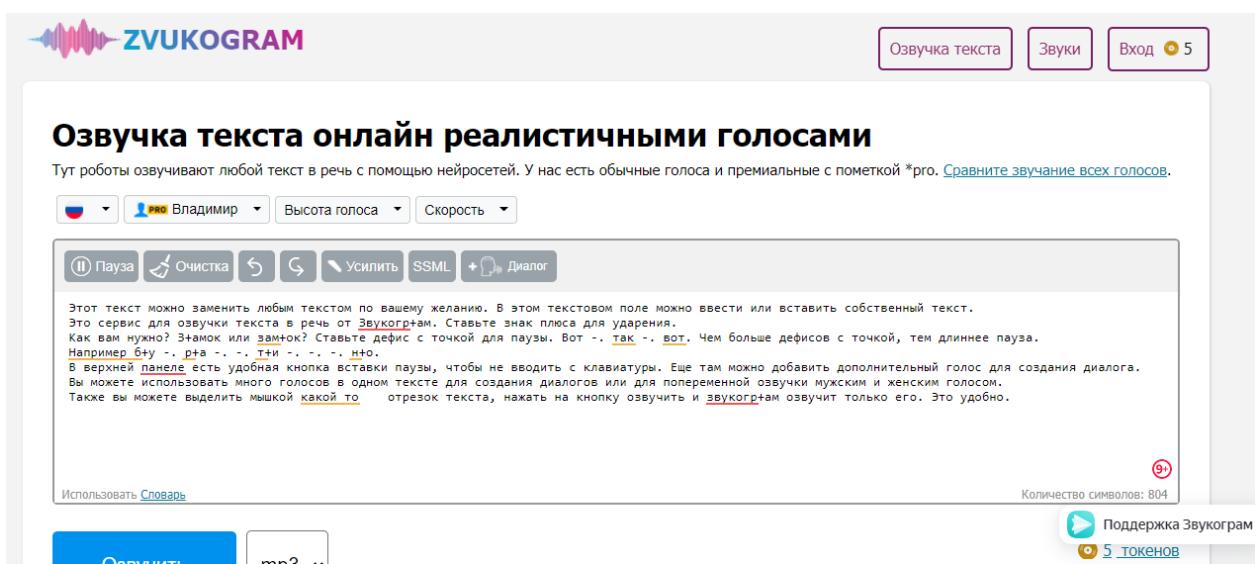


Рисунок 1.6 – Приложения для озвучивания текста «Zvukogram»

## 2 МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

### 2.1 Функциональная модель программного средства

Построение приложения подразумевает проектирование его функциональной модели. Функциональная модель приложения представлена в виде диаграмм вариантов использования и информационной модели предметной области.

#### 2.1.1 Варианты использования приложения

Варианты использования данного приложения различаются в зависимости от статуса пользователя. А статус, в свою очередь, зависит от того была ли пройдена авторизация, и от роли авторизированного пользователя. Все возможные статусы отображены на рисунке 2.1.

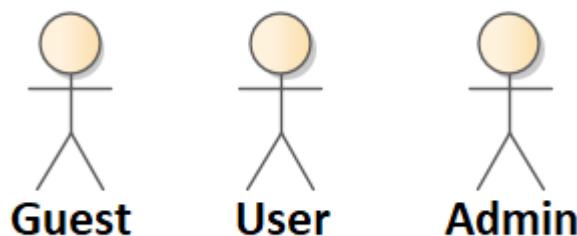


Рисунок 2.1 – Все возможные статусы пользователя

Чтобы пользователи могли зарегистрироваться и авторизоваться, необходимо предусмотреть наличие в базе данных специальных сущностей, предназначенных для хранения информации о пользователе. Диаграмма сущностей, участвующих в процессах регистрации и авторизации пользователя, отображена на рисунке 2.2.

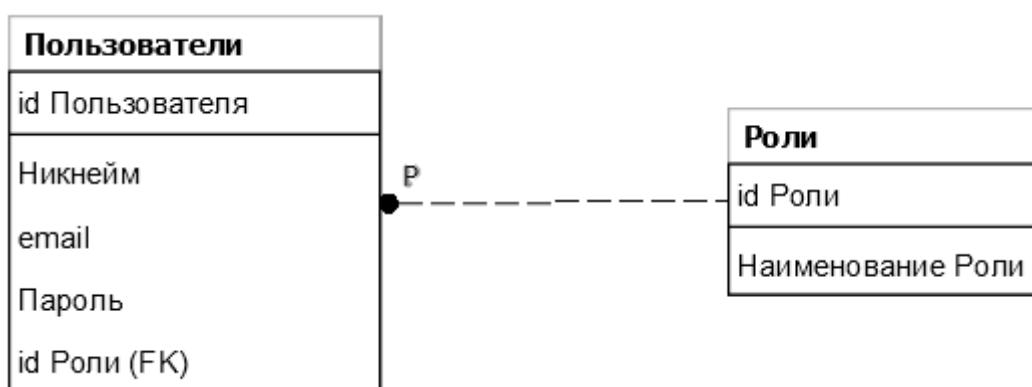


Рисунок 2.2 – Диаграмма сущностей хранящих информацию о пользователе

На данной диаграмме отображены две сущности. Сущность Пользователи представляет собой каждого пользователя, зарегистрировавшегося в приложении, а сущность Роли представляет все возможные роли пользователя.

Любой пользователь, использующий приложение, изначально имеет статус «Guest». Такой пользователь не имеет доступа к синтезу аудиокниг, не может оценивать аудиокниги и не способен добавить аудиокнигу в приложение. Для смены статуса требуется пройти авторизацию. Все действия, которые может выполнять пользователь со статусом «Guest» указаны на рисунке 2.3.



Рисунок 2.3 – Диаграмма вариантов использования приложения пользователем со статусом «Guest»

После прохождения авторизации статус пользователя меняется на «User» или «Admin», это зависит от роли пользователя хранимой в базе данных. Все действия, которые может выполнять пользователь с ролью «User» указаны на рисунке 2.4.

В приложении должна быть предусмотрена административная часть. К ней имеют доступ только те пользователи, которые обладают статусом «Admin». Функциональность доступная пользователям со статусом «Admin» представлена на рисунке 2.5.

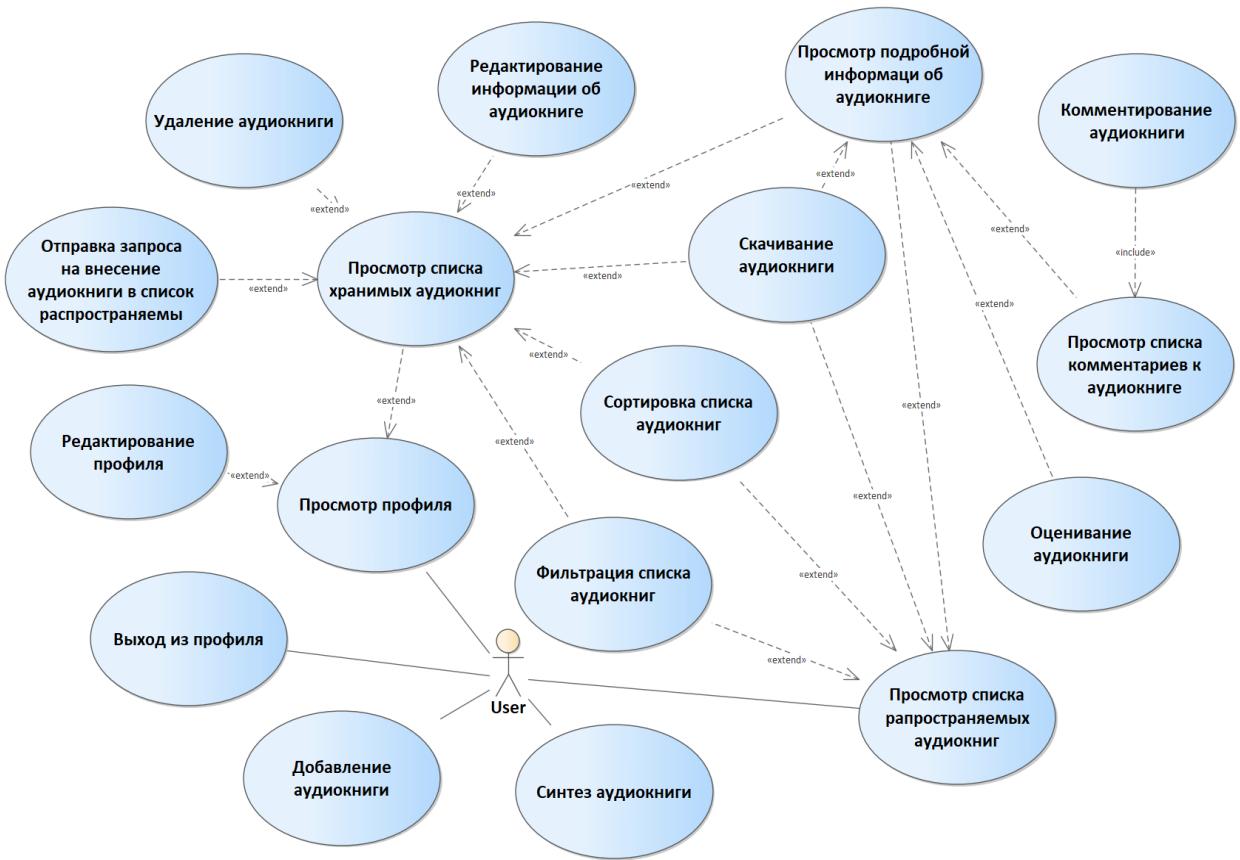


Рисунок 2.4 – Диаграмма вариантов использования приложения пользователем со статусом «User»

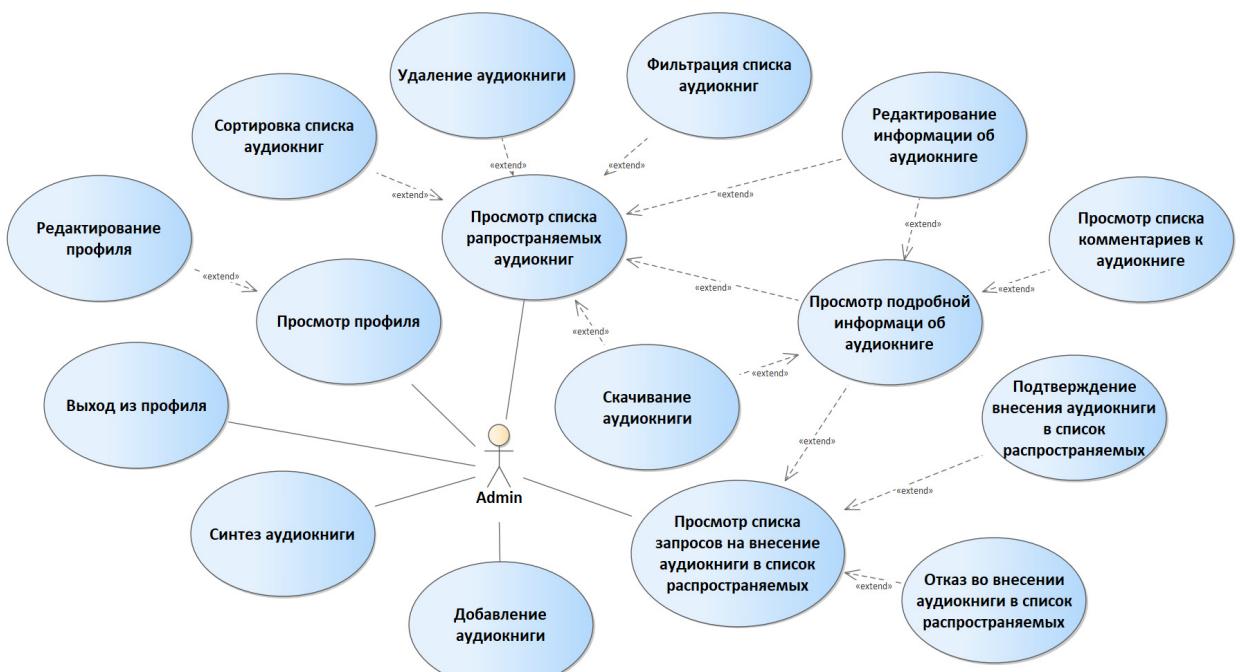


Рисунок 2.5 – Диаграмма вариантов использования приложения пользователем со статусом «Admin»

Поскольку в качестве основной платформы функционирования приложения был выбран веб, основной архитектурой приложения будет являться клиент-серверный подход, реализующий концепцию «клиент-сервер» с двумя независимыми друг от друга точками входа, совмещенный с архитектурным пятерном MVC (рисунок 3.1).

Клиент-серверная архитектура – вид архитектуры приложений, при которой нагрузка распределена между двумя объектами: сервером и клиентом. Связь в паре клиент-сервер осуществляется посредством сетевых протоколов, каждый в этой паре является по сути программным обеспечением. К достоинствам такого подхода относят:

- невысокие технические и системные требования к машинам-клиента;
- безопасность хранения данных, т.к. базы данных расположены на сервере, который, как правило, защищен гораздо лучше, чем машины-клиенты;
- отсутствие необходимости дублирования исходного программного кода на машине-клиенте;
- возможность использования нескольких физических компьютеров, объединенных в одну сеть, в качестве сервера;
- быстрое обновление ПО на машинах-клиентах.

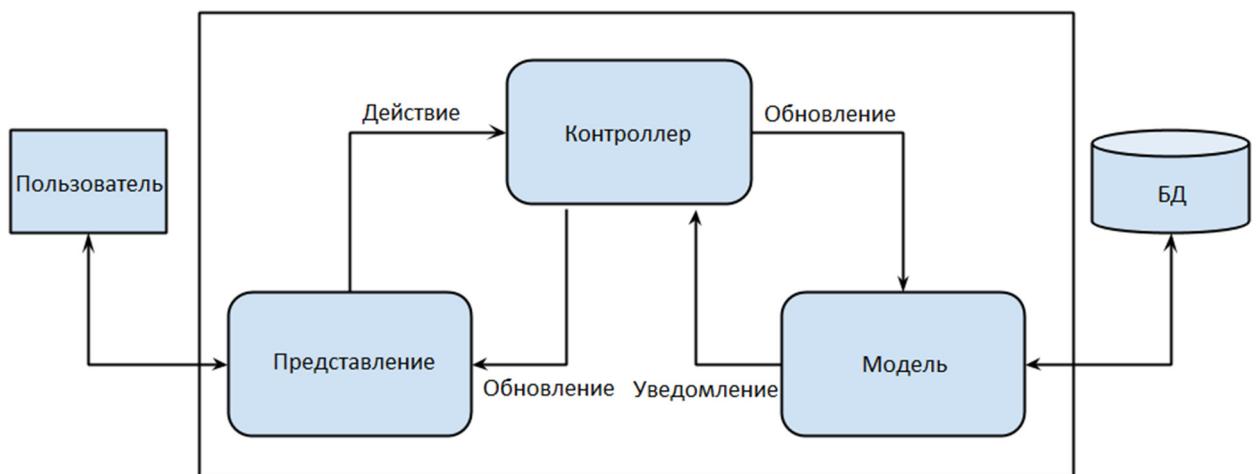


Рисунок 3.1 – Схема работы MVC

MVC – это схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер. Изначально это был всего лишь шаблон проектирования, но по прошествии некоторого периода времени данный подход эволюционировал до архитектурного шаблона. Этот архитектурный паттерн является одним из самых популярных в среде веб-приложений. Приложения, основанные на данной архитектуре, легко сопровождать и достраивать, они гибкие и легко масштабируемые.

## 4.2 Взаимодействие с базой данных

В качестве системы управления базами данных (СУБД) была выбрана система MySQL версии 4.0.

Данная СУБД предоставляет достаточно широкий перечень возможностей для взаимодействия с базой данных: формирование запросов, осуществление поиска необходимых данных, синхронизация информации, а также выполнение аналитической обработки данных и получение разнообразных отчетов. MySql — это реляционная система управления базами данных. То есть данные в ее базах хранятся в виде логически связанных между собой таблиц, доступ к которым осуществляется с помощью языка запросов SQL. MySql — это достаточно быстрая, надежная и, главное, простая в использовании СУБД.

Работать с MySql можно не только в текстовом режиме, но и в графическом. Существует очень популярный визуальный интерфейс для работы с этой СУБД — MySql Workbench. Окно данной программы представлено на рисунке 3.3. Этот интерфейс позволяет значительно упростить работу с базами данных в MySql.

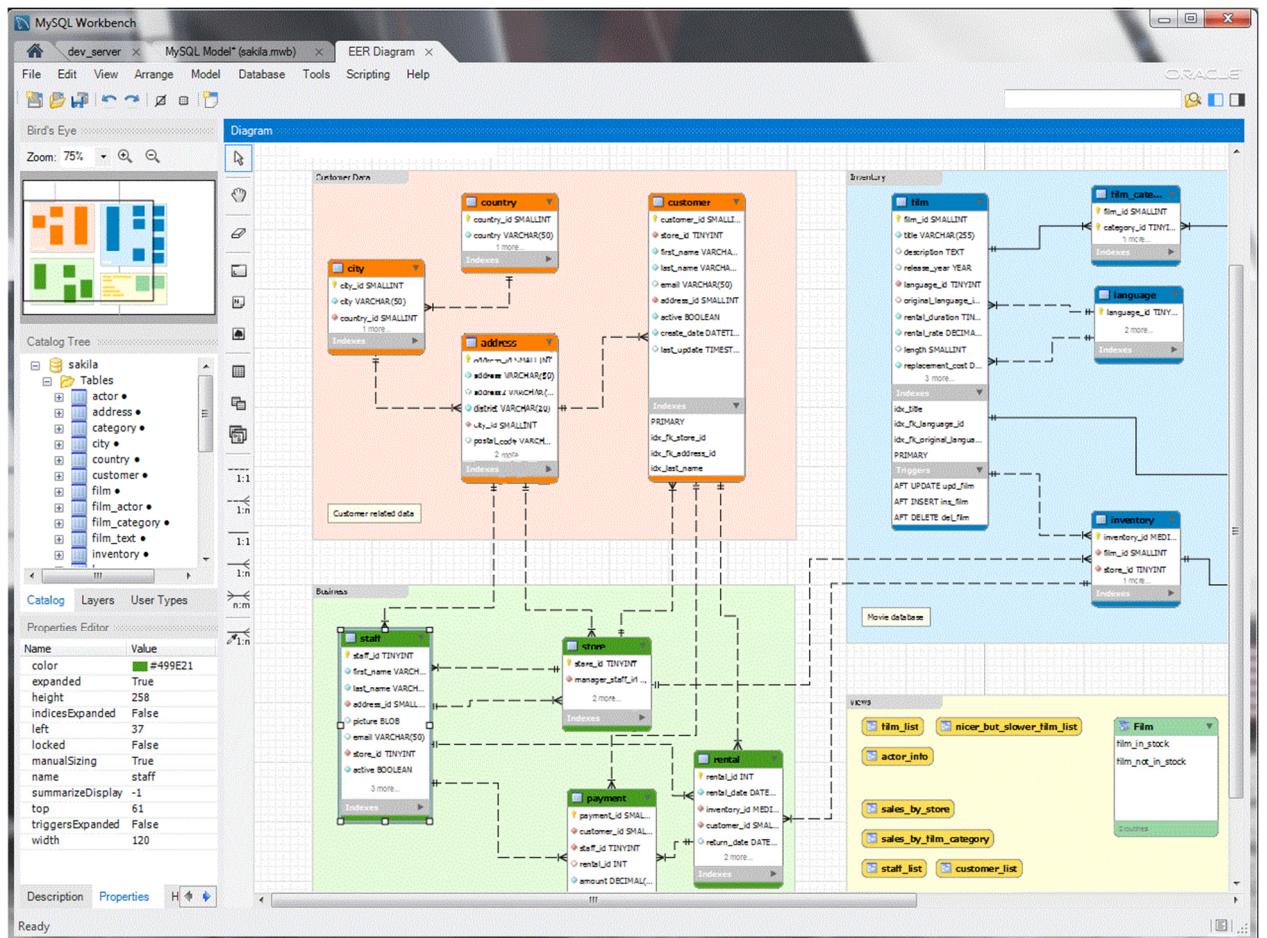


Рисунок 4.1 – Окно приложения MySql Workbench

```

@Controller
@RequestMapping("user")
public class UserController2 {
    @Autowired
    private UserService userService;
    @RequestMapping(value = "/me", method = RequestMethod.GET)
    public String getMyUserPage(Model model, @AuthenticationPrincipal UserDetails currentUser) {
        User = userService.getNickname(currentUser.getUsername());
        model.addAttribute("userInfo", user);
        return "userPage";
    }
    @RequestMapping(value = "/{id}", method = RequestMethod.GET)
    public String getUserPage(Model model,
        @PathVariable("id") Long id) {
        User = userService.getById(id);
        model.addAttribute("userInfo", user);
        return "userPage";
    }
    @RequestMapping(value = "/add", method = RequestMethod.POST)
    public String addUser(@ModelAttribute("userForm") @Valid User userForm,
        BindingResult result, Model model) {
        if (bindingResult.hasErrors()) {
            return "redirect:/registration";
        }
        if (!userForm.getPassword().equals(userForm.getPasswordConfirm())){
            model.addAttribute("errorMessage", "Пароли не совпадают");
            return "registrationPage";
        }
        if (!userService.add(userForm)){
            model.addAttribute("errorMessage", "Пользователь с таким именем уже существует");
            return "registrationPage";
        }
        return "redirect:/login";
    }
    @RequestMapping(value = "/edit", method = RequestMethod.PUT)
    public String editUser(@RequestBody User userForm) {
        if (!userForm.getPassword().equals(userForm.getPasswordConfirm())){
            return "redirect:/user/me/edit?error=1";
        }
        if (!userService.edit(userForm)){
            return "redirect:/user/me/edit?error=2";
        }
        return "redirect:/user/me";
    }
    @DeleteMapping(value = "/{id}")
    public String deleteUser(Model model, @AuthenticationPrincipal UserDetails currentUser,
        @PathVariable("id") Long id) {
        User = userService.getById(id);
        userService.delete(user);
        model.addAttribute("userInfo", user);
        return "userPage";
    }
}

```

Рисунок 4.2 – Пример контроллера

Хранить логику обработки запросов в контроллере было бы громоздко и неудобно. С целью избегания этого приложение имеет 3 слоя:

- веб;
- бизнес логика;
- доступ к данным.

Контроллеры хранятся в веб слое и обращаются к сервисам, которые хранятся в слое бизнес логики. Сервисы содержат в себе всю логику изменения данных, пришедших в запросе. Для выполнения своей задачи сервисы могут обращаться к другим сервисам. На рисунке 4.3 приведён пример сервиса, отвечающего за работу с комментариями.

```

@Service
@Transactional
public class CommentServiceImpl implements CommentService {

    @Autowired
    private CommentRepository commentRepository;

    @Override
    public Iterable<Comment> getAllByAudiobook(Audiobook audiobook) {
        return commentRepository.findAllByAudiobook(audiobook);
    }

    @Override
    public void add(Comment comment) {
        commentRepository.save(comment);
    }

    @Override
    public void edit(Comment comment) { commentRepository.save(comment); }

    @Override
    public void delete(Comment comment) { commentRepository.delete(comment); }

}

```

Рисунок 4.3 – Пример сервиса

Для повышения независимости слоёв друг от друга используется принцип инверсии зависимостей. Согласно нему компоненты не знают друг о друге ничего кроме интерфейсов. Это позволяет в любой момент подменять компоненты другими с таким же интерфейсом. Он является частью большего принципа написания кода, распространяющегося не только на веб-приложения – SOLID. Этот принцип является очень важным при написании качественного и поддерживаемого кода. Каждая буква в названии отвечает за отдельный принцип, несущий свои правила написания кода:

- принцип одной ответственности (single responsibility);
- принцип открытости/закрытости (open-closed);
- принцип подстановки Барбары Лисков (Liskov substitution);
- принцип разделения интерфейса (interface segregation);
- принцип инверсии зависимостей (dependency inversion).

Spring Framework позволяет настраивать зависимости автоматически, с помощью аннотации `@Autowired`. На рисунке 4.4 приведён пример настройки зависимостей для контроллера аудиокниг.

```

private final AudiobookService audiobookService;
private final CommentService commentService;
private final AudiobookFileService audiobookFileService;
private final UserService userService;
private final CreatorService creatorService;

@Autowired
public AudiobookController(AudiobookService audiobookService, CommentService commentService,
AudiobookFileService audiobookFileService, UserService userService, CreatorService creatorService) {
    this.audiobookService = audiobookService;
    this.commentService = commentService;
    this.audiobookFileService = audiobookFileService;
    this.userService = userService;
    this.creatorService = creatorService;
}

```

Рисунок 4.4 – Пример настройки зависимостей

После применения всех преобразований к данным сервисы обращаются к репозиториям для внесения изменений в базу данных. В приложении реализован репозиторий для выполнения базовых операций с любой моделью. При необходимости выполнения нестандартных действий создаётся дополнительный репозиторий, наследуемый от базового, и в нём реализуются все необходимые операции. На рисунке 4.5 приведён пример базового репозитория.

```

@Repository
public interface CrudRepository<T, ID> extends Repository<T, ID> {
    <S extends T> S save(S var1);

    <S extends T> Iterable<S> saveAll(Iterable<S> var1);

    Optional<T> findById(ID var1);

    boolean existsById(ID var1);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> var1);

    long count();

    void deleteById(ID var1);

    void delete(T var1);

    void deleteAll(Iterable<? extends T> var1);

    void deleteAll();
}

```

Рисунок 4.5 – Пример базового репозитория

Репозитории обращаются к базе данных за данными, которые в приложении хранятся в моделях – второй части архитектуры MVC. Модели содержат только данные. На рисунке 4.6 приведён пример модели.

```
@Entity  
@Table(name = "user")  
public class User {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id", nullable = false)  
    private Long id;  
  
    @NotNull  
    @Column(name = "nickname", nullable = false)  
    private String nickname;  
  
    @Column(name = "email", nullable = true)  
    private String email;  
  
    @NotNull  
    @Column(name = "password", length = 70, nullable = false)  
    private String password;  
  
    @ManyToOne(fetch = FetchType.EAGER)  
    @JoinColumn(name = "role_id", nullable = false)  
    private Role role;  
  
    public User() {}  
    public Long getId() { return id; }  
    public String getNickname() { return nickname; }  
    public void setNickname(String nickname) { this.nickname = nickname; }  
    public String getEmail() { return email; }  
    public void setEmail(String email) { this.email = email; }  
    public String getPassword() { return password; }  
    public void setPassword(String password) { this.password = password; }  
    public Role getRole() { return role; }  
    public void setRole(Role role) { this.role = role; }  
}
```

Рисунок 4.6 – Пример модели

Для хранения данных в приложении используется база данных. Для удобной работы с ней используется Spring Data JPA. Он позволяет работать с базой данных как будто сущности в ней – это объекты, к которым можно обратиться напрямую из кода приложения.

Последняя часть архитектуры MVC – представление – реализована клиентской частью приложения в виде шаблонов html страниц, с использованием Thymeleaf и JavaScript.

Thymeleaf – это современный серверный механизм Java-шаблонов для веб- и автономных сред, способный обрабатывать HTML, XML, JavaScript, CSS и даже простой текст. Основной целью Thymeleaf является создание элегантного и удобного способа шаблонизации. Thymeleaf также был разработан с самого начала с учетом стандартов Web, особенно HTML5, что позволяет создавать полностью соответствующие стандарту шаблоны страниц. На рисунке 4.7 приведён пример шаблона HTML страницы.

```

<!DOCTYPE html>
<html lang="ru"
      xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org"
      xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity4">
<head>
    <title>Spring Security Example </title>
    <div th:replace="fragments/header :: header-css"/>
</head>
<body>
    <div th:replace="fragments/header :: header"/>
    <div id="content" class="container mt-4">
        <div class="row justify-content-center" >
            <div class="card col-lg-5 col-md-7 col-sm-10" th:object="${userInfo}">
                <div class="card-body" >
                    <div class="row justify-content-center" >
                        <div class="col-9 mb-1" th:if="#{userInfo.nickname.equals(#authorization.authentication.name)}" >
                            <h2 class="card-title" style="text-align: center">Мой профиль</h2>
                        </div>
                        <div class="col-9 mb-1" th:unless="#{userInfo.nickname.equals(#authorization.authentication.name)}" >
                            <h2 class="card-title" style="text-align: center">Профиль пользователя <p
th:text="${userInfo.nickname}"></p></h2>
                        </div>
                    </div>
                    <input type="hidden" required name="id" th:field="*{id}"><b>
                    <div class="row justify-content-center" >
                        <div class="mb-2 col-9" >
                            <span class="" name="nickname" id="nickname">
                                <p th:text="Никнейм: +'${userInfo.nickname}'"></p></span>
                            </div>
                        </div>
                    <div class="row justify-content-center" >
                        <div class="mb-2 col-9" id="email" name="email" >
                            <div class="col" th:unless="#{strings.isEmpty(userInfo.email)}" >
                                <p th:text="Email: +'${userInfo.email}'"></p>
                            </div>
                            <p class="col" th:if="#{strings.isEmpty(userInfo.email)}" >Email: не указан</p>
                        </div>
                    </div>
                    <div class="row justify-content-center" >
                        <div class="mb-2 col-9" th:switch="${userInfo.role.id}">
                            <p th:case="2">Статус: администратор сайта</p>
                            <p th:case="3">Статус: Администратор сайта</p>
                            <p th:case="1">Статус: пользователь сайта</p>
                        </div>
                    </div>
                    <div class="row justify-content-center mt-2" th:if="#{userInfo.nickname.equals(#authorization.authentication.name)}" >
                        <a id="edit-profile-btn" class="col-lg-7 col-md-8 col-sm-9 btn btn-primary btn-block" th:href="@{/user/me/edit}">
                            Редактировать профиль
                        </a>
                    </div>
                </div>
            </div>
        </div>
        <div th:replace="fragments/footer :: footer"/>
    </body>
</html>

```

Рисунок 4.7 – Пример шаблона Html страницы

– перейти на страницу с подробной информацией об аудиокниге, по нажатии на название одной из аудиокниг.

– скачать аудиокнигу, нажав на кнопку «скачать», расположенную в области описания выбранной книги.

Стоит отметить, что на каждой странице разработанного приложения имеется заголовок, с помощью которого можно в любой момент перейти на нужную пользователю страницу. Для неавторизованного пользователя данный заголовок содержит:

– пункт «ListenToBook», по нажатии на который осуществляется переход на главную страницу приложения;

– пункт «Распространяемые аудиокниги», по нажатии на который осуществляется переход на главную страницу приложения;

– пункт «Регистрация», по нажатии на который происходит перенаправление на страницу регистрации пользователя;

– пункт «Авторизация», по нажатии на который происходит перенаправление на страницу авторизации пользователя.

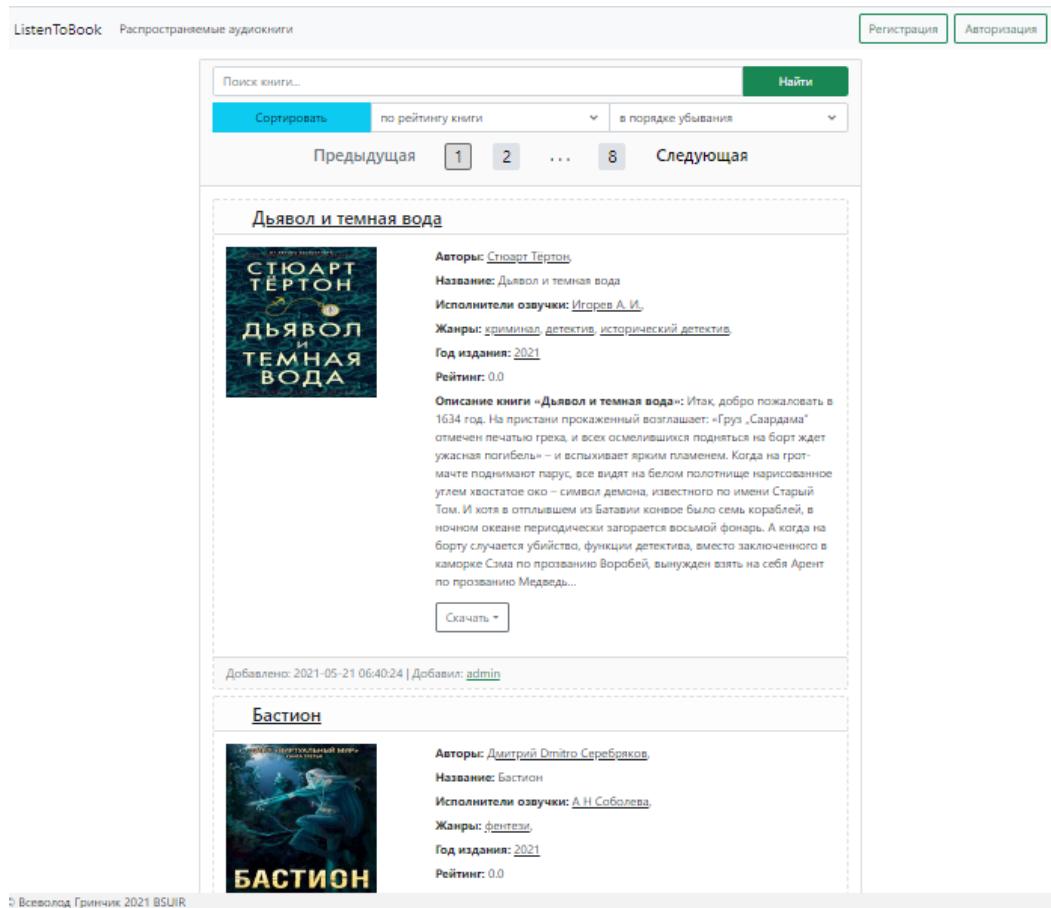


Рисунок 6.1 – Интерфейс главной страницы приложения

При нажатии на название одной из аудиокниг в списке, происходит пе-

ренаправление на страницу с подробной информацией о выбранной аудиокниге. Интерфейс страницы с подробной информацией об аудиокниге, представлен на рисунке 6.2. Здесь пользователь может узнать подробнее о выбранной аудиокниге, а также просмотреть список комментариев к аудиокниге. На этой странице также присутствует возможность скачать аудиокнигу.

## Дьявол и темная вода



Автор: Стюарт Тёртон  
Название: Дьявол и темная вода  
Озвучивает: Игорев А. И.  
Жанры: криминал, детектив, исторический детектив  
Год издания: 2021  
Рейтинг: 0.0

Описание книги «Дьявол и темная вода»: Итак, добро пожаловать в 1634 год. На пристани прокаженный возглашает: «Груз „Саардама“ отмечен печатью греха, и всех осмелившихся подняться на борт ждет ужасная гибель» – и вспыхивает ярким пламенем. Когда на грот-мачте поднимают парус, все видят на белом полотнище нарисованное углем хвостатое око – символ демона, известного по имени Старый Том. И хотя в отплывшем из Батавии конвое было семь кораблей, в ночном океане периодически загорается восьмой фонарь. А когда на борту случается убийство, функции детектива, вместо заключенного в каморке Сэма по прозванию Воробей, вынужден взять на себя Арент по прозванию Медведь...

[Скачать ▾](#)

Добавлено: 2021-05-21 06:40:24 | Добавил: [admin](#)

[user](#) 2021-05-21 07:13:50

Сейчас много книг в стиле попаданцев, но хочу отметить цикл Андрея Стоева «За последним порогом», и первую книгу цикла «Начало». Автор придумал интересный, логичный мир со своими законами и особенностями.

Рисунок 6.2 – Интерфейс страницы подробной информации об аудиокниге

При нажатии на наименование автора или исполнителя озвучки, происходит перенаправление на страницу со списком аудиокниг, в создании которых он принимал участие. При нажатии на наименование жанра, происходит перенаправление на страницу со списком аудиокниг, отмеченных этим жанром. При нажатии на год издания книги, происходит перенаправление на страницу со списком аудиокниг, с совпадающим годом издания.

Для увеличения функциональности программного средства относительно неавторизованного пользователя требуется выполнить регистрацию для создания нового профиля или же авторизацию для входа в уже существующий.

При нажатии на пункт «Регистрация» в заголовке страницы пользователь будет перенаправлен на страницу регистрации, интерфейс которой пред-

ставлен на рисунке 6.3. Здесь пользователь должен ввести корректные никнейм и пароль (который требуется повторить в поле «Повторите пароль»). Для подтверждения регистрации требуется нажать на кнопку «Зарегистрироваться». Если введённые данные окажутся некорректными, то отобразится сообщение с указанием конкретной ошибки. Интерфейс подобного случая представлен на рисунке 6.4.

## Регистрация

Никнейм

Пароль

Повтор пароля

**Зарегистрироваться**

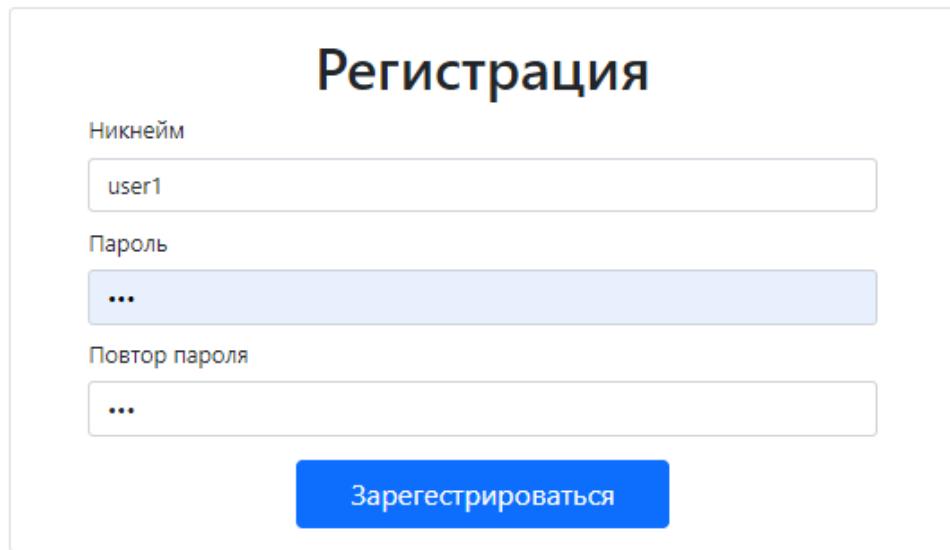
A screenshot of a user registration interface. The title is 'Регистрация'. There are three input fields: 'Никнейм' with value 'user1', 'Пароль' with placeholder '...', and 'Повтор пароля' with placeholder '...'. A blue button at the bottom right says 'Зарегистрироваться'.

Рисунок 6.3 – Интерфейс страницы регистрации пользователя

## Регистрация

Пароли не совпадают

Никнейм

Пароль

Повтор пароля

**Зарегистрироваться**

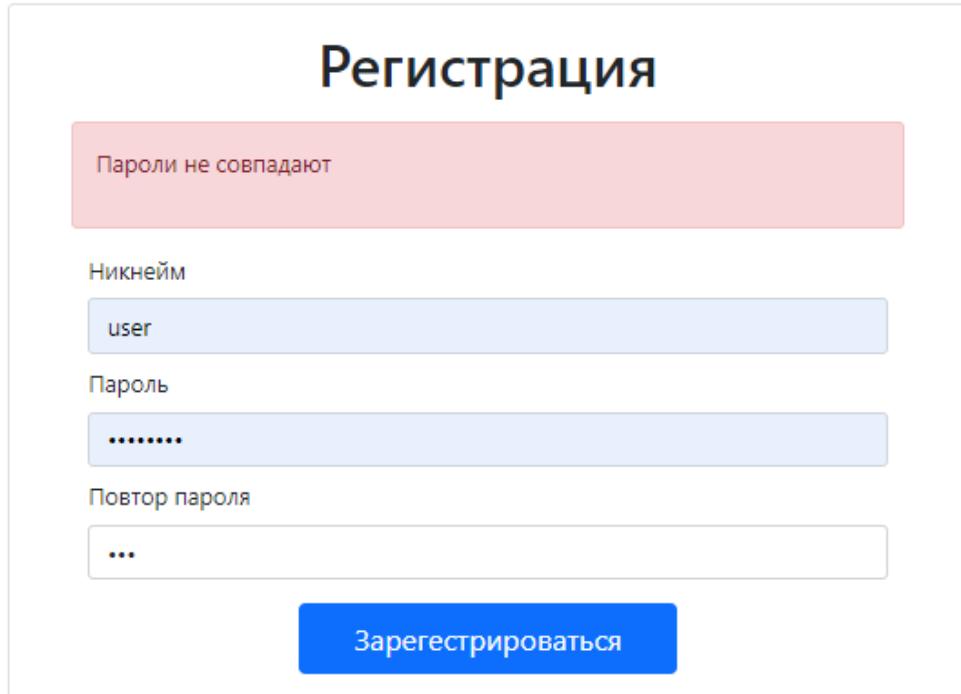
A screenshot of a user registration interface similar to Figure 6.3, but with an error message 'Пароли не совпадают' (Passwords do not match) displayed in a red box above the input fields. The input fields show 'user' in the nickname field, '.....' in the password field, and '...' in the repeat password field. The 'Зарегистрироваться' button is at the bottom.

Рисунок 6.4 – Интерфейс страницы регистрации пользователя в случае ввода некорректных значений

Если все данные окажутся корректными, то пользователь будет зарегистрирован и перенаправлен на страницу авторизации.

При нажатии на пункт «Авторизация» в заголовке страницы пользователь будет перенаправлен на страницу авторизации, интерфейс которой представлен на рисунке 6.5. Здесь пользователь должен ввести корректные никнейм и пароль. Для подтверждения авторизации требуется нажать на кнопку «Войти». Если введённые данные окажутся некорректными, то отобразится сообщение с указанием конкретной ошибки. Интерфейс подобного случая представлен на рисунке 6.6.

The screenshot shows a user login form titled 'Авторизация' (Authorization). It contains two input fields: 'Никнейм' (Nickname) with the value 'user' and 'Пароль' (Password) with the value '\*\*\*\*\*'. Below the inputs is a blue 'Войти' (Enter) button.

Рисунок 6.5 – Интерфейс страницы авторизации пользователя

The screenshot shows a user login form titled 'Авторизация' (Authorization). A red error message box at the top contains the text 'Неверный никнейм или пароль.' (Incorrect nickname or password.). Below the message are two input fields: 'Никнейм' (Nickname) with the value 'user' and 'Пароль' (Password) with the value '\*\*\*\*\*'. Below the inputs is a blue 'Войти' (Enter) button.

Рисунок 6.6 – Интерфейс страницы авторизации пользователя в случае ввода некорректных значений

После того, как пользователь будет авторизован, у него появятся новые функциональные возможности в программном средстве. Также изменится и заголовок на каждой странице приложения, интерфейс обновленного заголовка представлен на рисунке 6.7.

Для авторизованного пользователя со статусом «User» данный заголовок содержит:

- пункт «ListenToBook», по нажатии на который осуществляется переход на главную страницу приложения;
- пункт «Распространяемые аудиокниги», по нажатии на который осуществляется переход на главную страницу приложения;
- пункт «Добавить аудиокнигу», по нажатии на который осуществляется переход на страницу добавления аудиокниги;
- пункт «Синтезировать аудиокнигу», по нажатии на который осуществляется переход на страницу синтеза аудиокниги;
- пункт «Мой профиль», по нажатии на который происходит перенаправление на страницу с информацией о профиле текущего пользователя;
- пункт «Мои аудиокниги», по нажатии на который осуществляется переход на страницу со списком книг, хранимых пользователем;
- пункт «Выход», по нажатии на который происходит выход пользователя из текущего профиля.

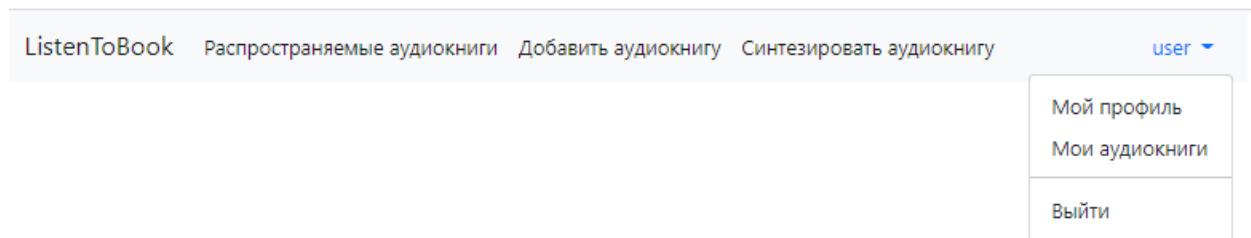


Рисунок 6.7 – Интерфейс заголовка для авторизованного пользователя со статусом «User»

При нажатии на пункт «Мой профиль» пользователь будет перенаправлен на страницу с информацией о своём профиле. Интерфейс данной страницы представлен на рисунке 6.8. Здесь пользователь может просмотреть информацию о своём профиле, а также перейти на страницу его редактирования.

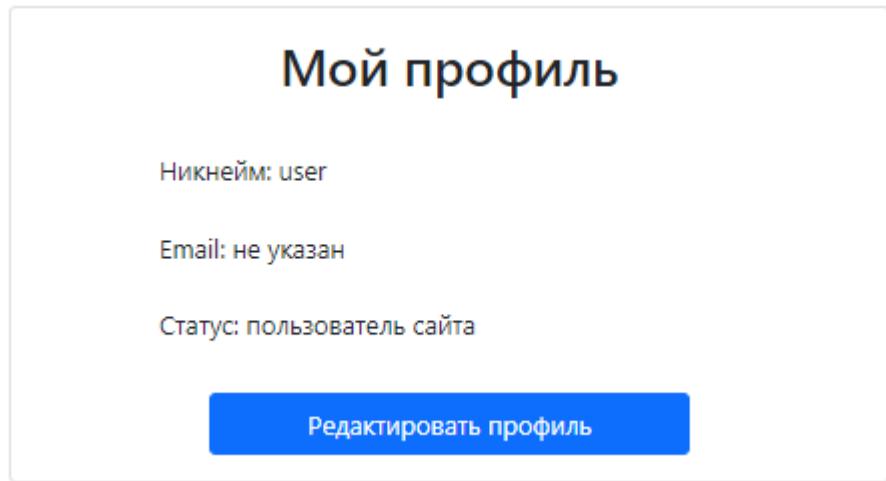


Рисунок 6.8 – Интерфейс страницы с информацией о профиле текущего пользователя

При нажатии на кнопку «Редактировать профиль» пользователь будет перенаправлен на страницу редактирования профиля. Интерфейс данной страницы представлен на рисунке 6.9. Здесь пользователь может изменить никнейм, пароль и email.

При нажатии на кнопку «Сохранить изменения» вся введённая информация будет сохранена, после чего пользователь будет перенаправлен на страницу информации о своём профиле.

Если введённые данные окажутся некорректными, то отобразится сообщение с указанием конкретной ошибки. Интерфейс подобного случая представлен на рисунке 6.10.

The screenshot shows a light gray rectangular form titled 'Редактирование данных пользователя' at the top center. It contains four input fields: 'Никнейм' with the value 'user', 'Email' with the value 'admin@meloman.mel', 'Пароль' with three dots as placeholder, and 'Повтор пароля' with three dots as placeholder. At the bottom center is a blue rectangular button with white text that reads 'Сохранить изменения'.

Рисунок 6.9 – Интерфейс страницы редактирования профиля пользователя

## Редактирование данных пользователя

Пользователь с таким именем уже существует

Никнейм

Email

Пароль

Повтор пароля

**Сохранить изменения**

Рисунок 6.10 – Интерфейс страницы редактирования профиля пользователя в случае ввода некорректных значений

При нажатии на кнопку «Мои аудиокниги» пользователь будет перенаправлен на страницу со списком аудиокниг, хранимых пользователем. Интерфейс данной страницы представлен на рисунке 6.11. По сравнению со страницей распространяемых аудиокниг, на странице хранимых аудиокниг появились новые возможности:

- отправка заявки на внесение аудиокниги в список распространяемых;
- удаление аудиокниги;
- редактирование информации об аудиокниге.

Также стоит отметить, что у авторизованного пользователя появились новые возможности на странице подробной информации об аудиокниге:

- отправка заявки на внесение аудиокниги в список распространяемых;
- оценивание аудиокниги;
- комментирование аудиокниги.

Интерфейс данной страницы для пользователя со статусом «User» представлен на рисунке 6.12.

Поиск книги...

Найти

Сортировать по дате добавления книги в порядке убывания

Предыдущая Следующая 1

**Бастион**



Авторы: Дмитрий Dmitro Серебряков,  
Название: Бастион  
Исполнители озвучки: А Н Соболева.  
Жанры: фентези,  
Год издания: 2021  
Рейтинг: 0.0  
Описание книги «Бастион»: Приключения героев продолжаются.

Скачать Оценить Распространить

Добавлено: 2021-05-21 06:48:58 | Добавил: admin Редактировать Удалить

**Дьявол и темная вода**



Авторы: Стоарт Тёртон,

Рисунок 6.11 – Интерфейс страницы отображения списка аудиокниг, хранимых пользователем

**Бастион**



Автор: Дмитрий Dmitro Серебряков  
Название: Бастион  
Озвучивает: А Н Соболева  
Жанр: фентези  
Год издания: 2021  
Рейтинг: 0.0  
Описание книги «Бастион»: Приключения героев продолжаются.

Скачать Оценить

Добавлено: 2021-05-21 06:48:58 | Добавил: admin Редактировать Удалить

admin 2021-05-21 08:43:58  
Good

admin  
Введи текст комментария

Добавить комментарий

Рисунок 6.12 – Интерфейс страницы подробной информации об аудиокниге для пользователя со статусом «User»

При нажатии на пункт «Добавить аудиокнигу» в заголовке страницы пользователь будет перенаправлен на страницу добавления аудиокниги, интерфейс которой представлен на рисунке 6.13. Здесь пользователь может указать всю возможную информацию об аудиокниге, а также установить в качестве обложки любое изображение. При нажатии на кнопку «Добавить» происходит добавление аудиокниги, после чего отображается страница подробной информации о добавленной аудиокниге.

## Добавление аудиокниги

Название  
Мотивация

Файл обложки  
Выберите файл Motivatsii.jpg

Авторы книги  
Брайан Трейси Удалить

Брайан Трейси Добавить

Исполнители озвучки  
Алексеев А. И. Удалить

Алексеев А. И. Добавить

Жанры аудиокниги  
фантастика  
фентези  
криминал  
  
фентези  
криминал  
детектив  
исторический детектив

Файл аудиокниги  
Выберите файл audiobook.mp3

Описание  
самомотивации среду, в которой каждый человек захочет и сможет работать на 100 %. Справиться с этой задачей поможет 21 принцип мотивации сотрудников, каждый из которых можно применять сразу же после прочтения. ②

Год издания  
2021

Добавить

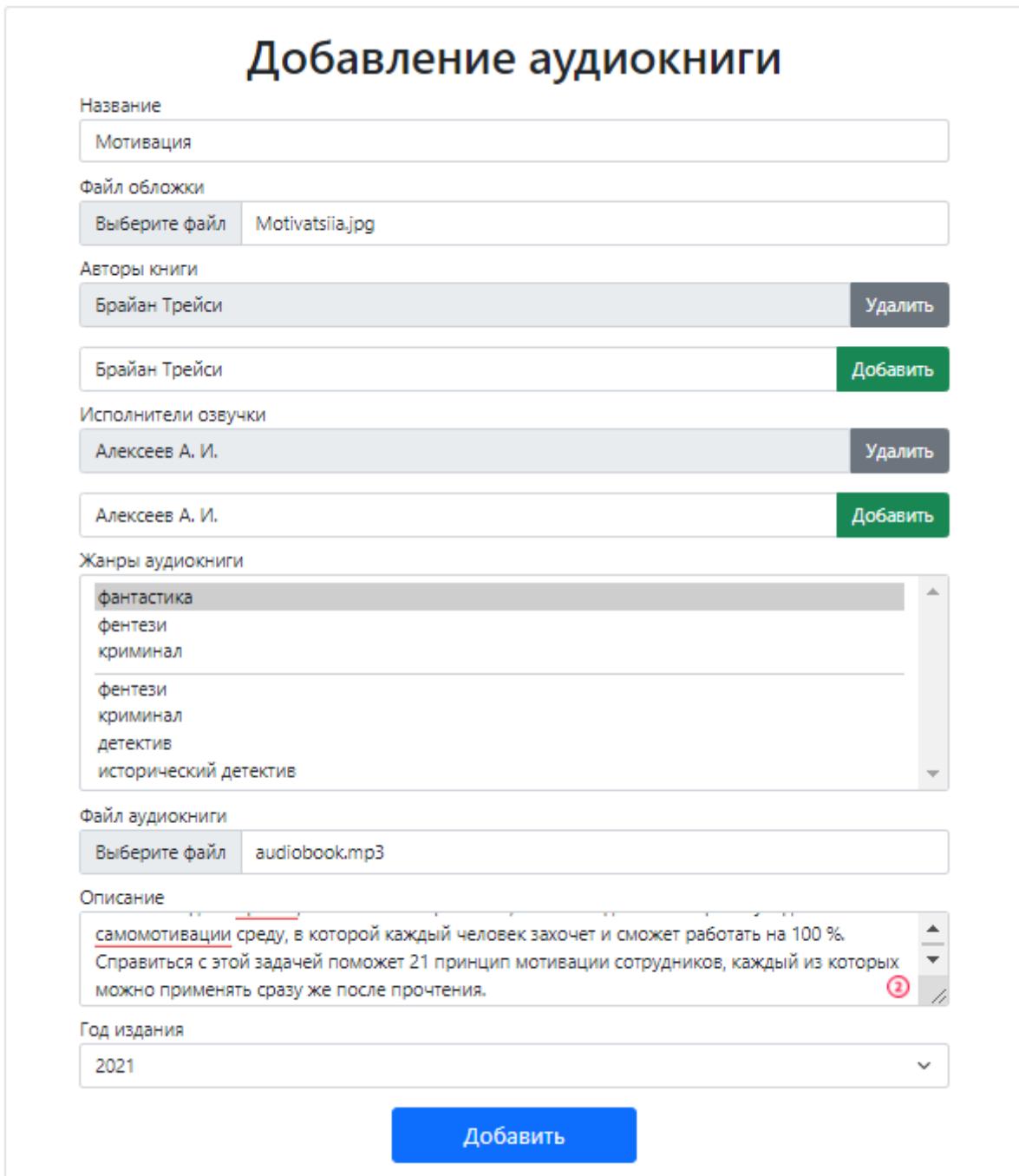


Рисунок 6.13 – Интерфейс страницы добавления аудиокниги

При нажатии на пункт «Синтезировать аудиокнигу» в заголовке страницы пользователь будет перенаправлен на страницу синтеза аудиокниги, интерфейс которой представлен на рисунке 6.14. Здесь пользователь может подобрать параметры синтеза, синтезировать и прослушать пробный фрагмент текста. А также загрузить текстовый файл и синтезировать аудиокнигу. После нажатия на кнопку «Начать синтез» начнется процесс синтеза аудиокниги, по окончании которого кнопка «Скачать аудиокнигу» станет активна.

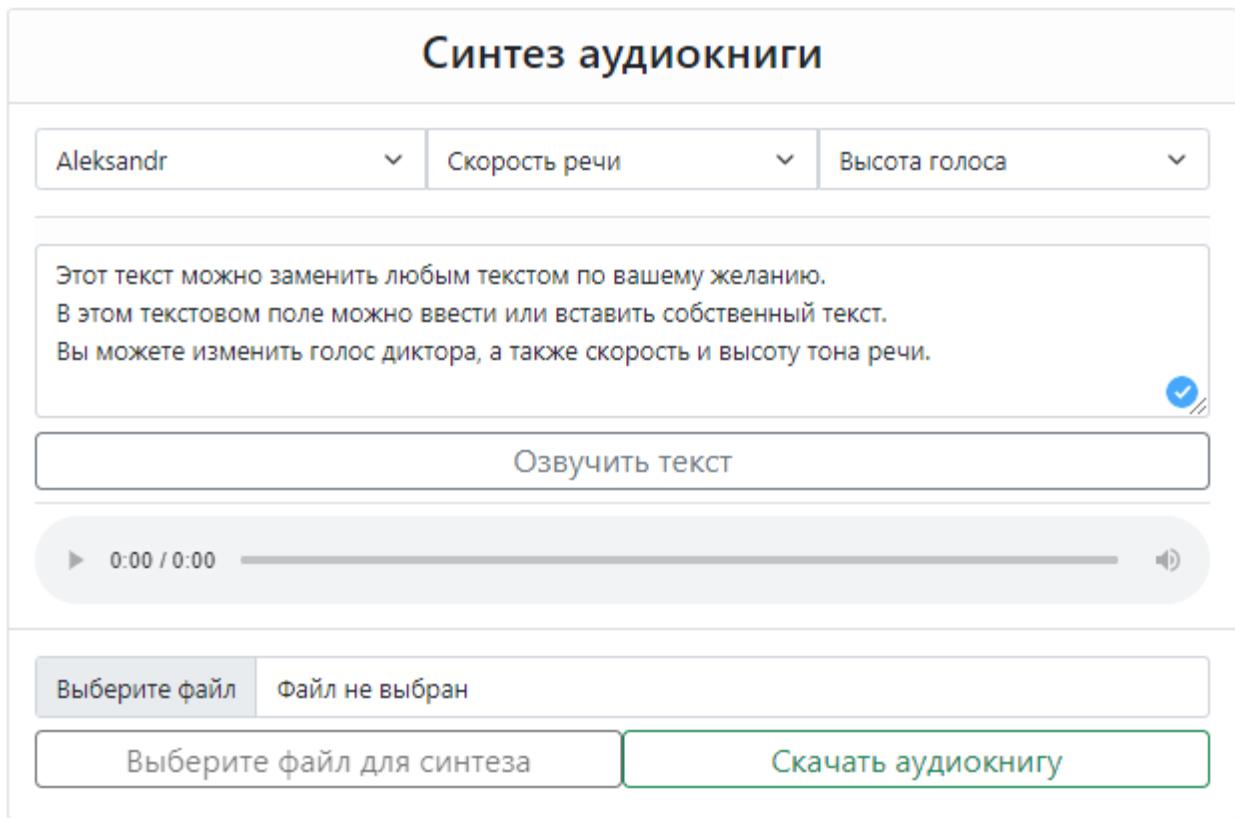


Рисунок 6.14 – Интерфейс страницы синтеза аудиокниги

Для авторизованного пользователя со статусом «Администратор» в заголовке страницы добавится пункт «Список заявок», по нажатии на который осуществляется переход на страницу со списком заявок на внесение аудиокниги в список распространяемых. Интерфейс данной страницы представлен на рисунке 6.15.

Также на главной странице пользователь со статусом «Администратор» имеет возможность удалять аудиокниги, нажав на кнопку «Удалить», на странице подробной информации об аудиокниге пользователь имеет возможность перейти на страницу редактирования данной аудиокниги, нажав на кнопку «Редактировать».

Предыдущая	1	Следующая
<b>Заявка номер 1</b>		
Наименование книги: <u>Бастион</u>		
Никнейм отправителя: <u>admin</u>		
Дата и время отправления: 2021-05-21 08:34:06.0		
<a href="#">Отказать</a>	<a href="#">Подтвердить</a>	
<b>Заявка номер 2</b>		
Наименование книги: <u>Дьявол и темная вода</u>		
Никнейм отправителя: <u>admin</u>		
Дата и время отправления: 2021-05-21 08:34:42.0		
<a href="#">Отказать</a>	<a href="#">Подтвердить</a>	

Рисунок 6.15 – Интерфейс страницы со списком заявок на внесение аудиокниги в список распространяемых

При нажатии на кнопку «Подтвердить» аудиокнига становится распространяемой, а заявка удаляется. При нажатии на кнопку «Отказать», заявка будет удалена, а статус аудиокниги останется неизменным.

В итоге программное средство работает стабильно. Интерфейс удобный и понятный. Предусмотрены различные проверки во избежание критических ситуаций и ошибок пользователя. Была учтена логика администрирования.