

Разработка веб - приложения с использованием Spring Framework и jQuery

О. Д. Глод, А. А. Чекулаев

Южный федеральный университет, Ростов-на-Дону

Аннотация: В работе рассмотрен процесс разработки веб-приложения, содержащего аудиоконтент. При проектировании системы использовалась методология ООАП (Объектно-ориентированный анализ и проектирование) и MVC (Model View Controller) паттерн проектирования. С помощью umbrello построены канонические UML (Unified Modeling Language) диаграммы, описывающие различные аспекты моделируемой системы. На основании созданных диаграмм была выполнена реализация приложения средствами языка программирования java. Также выполнен интерфейс пользователя, позволяющий продемонстрировать возможности приложения.

Ключевые слова: веб-приложение; аудио контент; методология объектно-ориентированного анализа и проектирования; UML – диаграмма; язык программирования java.

Актуальность данной задачи обусловлена программными средствами, которые используются для её реализации. Язык программирования Java был создан в 1995 году, но обновляется и в настоящее время, что позволяет ему сохранить актуальность для решения текущих задач. Последняя версия Java – 9 [1]. Аналогична ситуация и с Spring. Spring - сравнительно новая технология, первая стабильная версия была выпущена в 2004 году, но он постоянно дорабатывается, не теряя актуальности.

Новизна текущего веб-приложения заключается в предоставлении пользователю возможности собственноручно добавлять свой, интересующий его контент, расширяя базу аудиоконтента. Также важной особенностью приложения является язык программирования, на котором он написан. Большинство веб-ресурсов создаются с помощью языка программирования PHP [2]. Он прост в освоении и позволяет писать код прямо в html-странице, что импонирует многим разработчикам. Но такой подход многократно усложняет поддержку и модернизацию программного продукта. Чтобы избежать этих недостатков, для создания настоящего клиент-серверного веб-

приложения используется язык программирования Java с использованием MVC (Model-View-Controller) паттерна [3]. Он позволяет разбить код на слои, такие как контроллер, сервис и слой доступа к данным. За счет этого многократно повышаются читаемость кода. Также, с Java используется шаблон Spring Framework, имеющий модуль Spring MVC, который упрощает создание MVC структуры, а также модуль Spring Security, предоставляющий API (Application Programming Interface) для обеспечения безопасности приложения [4, 5, 6].

Для обеспечения взаимодействия пользователя с html-страницей используется JavaScript с jQuery фреймворком [7]. Этот фреймворк обеспечивает полное взаимодействие со структурой DOM (Document Object Model), позволяя взаимодействовать со структурой html-документа.

Особенностью предметной области является то, что браузеры могут проигрывать не любой аудиофайл, а только тот, на который у них есть лицензия. Например, браузер Mozilla Firefox проигрывает файлы формата *.ogg и не проигрывает формат *.mp3 [8]. Противоположная ситуация у браузера Google Chrome. Для проигрывания аудиофайлов для всех браузеров необходимо либо сохранять один аудиофайл во всех возможных типах, либо использовать свой проигрыватель для аудиофайлов и добавлять только в одном типе, например, *.mp3.

Помимо аудиофайла, в аудиоконтенте хранится метаданные, такая как: название файла, информацию об авторе, дата добавления.

В данной задаче рассматривается определенное число объектов:

Пользователи - информация о пользователях;

Роли - уровни доступа пользователя;

Ресурсы - музыкальный контент или контент любого другого образца;

Плейлисты - сборник ресурсов пользователя.

Каждый из данных объектов имеет проекцию в базе данных и создается пользовательский интерфейс, который позволит организовать CRUD (create, read, update, delete) операции с описанными выше сущностями.

Декомпозиция задачи. Все вышеизложенные задачи разрабатываются по следующему примерному плану:

- 1) Разработка сущностей;
- 2) Определение и создание основных классов;
- 3) Разработка алгоритмов обработки информации;
- 4) Разработка бизнес-логики;
- 5) Обеспечение взаимодействия между сервером и клиентом;
- 6) Разработка HTML шаблона;
- 7) Создание графического интерфейса:
 - a) Создание главной страницы веб-приложения, включающей в себя аудиоконтент и информацию о нём;
 - b) Создание страницы веб-приложения, включающей в себя плейлисты и информацию о них;
 - c) Создание страницы веб-приложения, включающей в себя аудиоконтент, использующийся в выбранном плейлисте;
 - d) Создание страницы веб-приложения, включающей в себя информацию о пользователях веб-приложения.
- 8) Создание возможности пользователю взаимодействовать с веб-приложением:
 - a) Создание возможности регистрации пользователя;
 - b) Создание возможности авторизации пользователя;
 - c) Создание возможности создать плейлист;
 - d) Создание возможности добавить аудиоконтент в плейлист.
- 9) Запуск бета-версии;

- 10) Добавление возможности поиска в базе данных:
 - a) Добавление возможности поиска аудиоконтента;
 - b) Добавление возможности поиска плейлиста.
- 11) Запуск конечной версии продукта.

Приложение имеет следующие функциональные возможности:

1. Предоставление аудиоконтента;
2. Регистрация и авторизация пользователя;
3. Добавление нового контента пользователем;
4. Добавления плейлиста пользователем.

Проанализировав возможности программных продуктов Java+Spring, PHP, Python [9], JavaScript, TypeScript, CoffeeScript [10, 11] и приняв во внимание данные, приведенные в [12, 13, 14], выбраны следующие инструментальные средства для разработки программного продукта: серверная часть выполняется с помощью языка программирования Java 8 и Spring Framework; клиентская часть выполняется с помощью JavaScript и jQuery; в качестве реляционной СУБД выберем H2 [15]. H2 - открытый исходный код базы данных облегченного Java. Он может быть встроен в Java-приложение или работать в режиме клиент-сервер. В качестве сервлет-контейнера используется Apache Tomcat [16, 17].

На рис.1 приведена созданная диаграмма использования. Детализация вариантов использования представлена в табл.1.

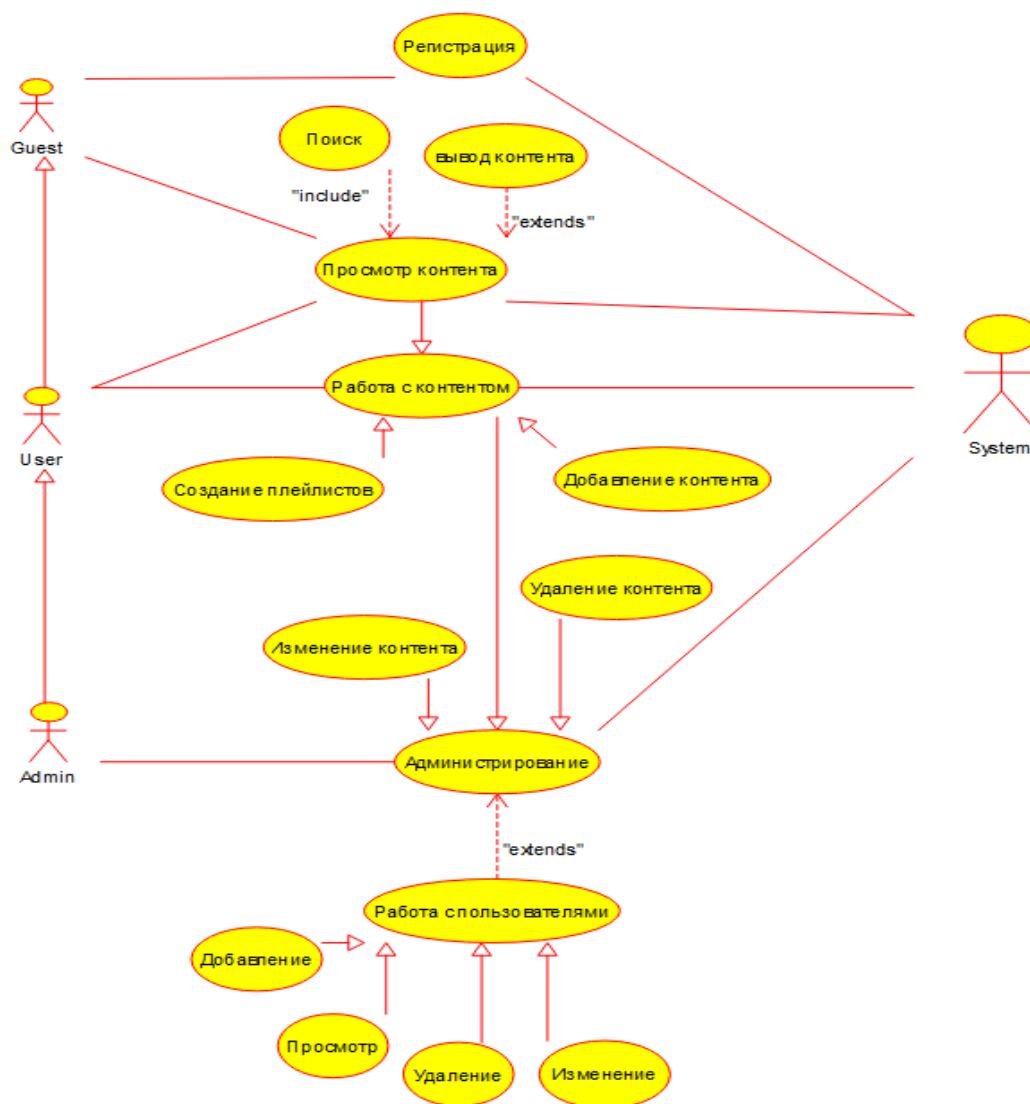


Рис. 1. – Диаграмма вариантов использования

Далее построена диаграмма классов (Static Structure diagram) - диаграмма, которая демонстрирует классы системы, их атрибуты, методы и взаимосвязи между ними. Для обеспечения взаимодействия программы с базой данных использовалось объектно-реляционное отображение.

ORM (Object Role Model) - это технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая “виртуальную объектную базу данных” [2, 3].

Таблица № 1

Детализация вариантов использования

Вариант использования	Описание
Просмотр контента	Вывод музыки или видео на экран
Работа с контентом	Просмотр контента и добавление нового контента (добавление соответствующей строки в базу данных, загрузка на сервер контента)
Администрирование	Предоставляется полный доступ к контенту, CRUD операции с ним и с пользователями

Эта задача легко выполняется с помощью такого инструмента, как Hibernate. Для того, чтобы его использовать, в систему автоматической сборки проекта (Gradle) устанавливается драйвер, позволяющий взаимодействовать выбранной базой данных. После необходимо создать конфигурационный файл и указать в нем параметры подключения к базе данных. Hibernate осуществляет маппинг Java-классов с таблицами базы данных с помощью Java-аннотаций:

- `@Table(name="name")` - указывает имя таблицы в базе данных;
- `@Id` - указывается id поле;
- `@GeneratedValue(strategy = GenerationType.AUTO)` - указывает, что поле id будет автоматически заполнено;
- `@Column(name="name")` - указывает, какое имя столбца в базе данных.

Также существуют аннотации `@OneToMany`, `@ManyToOne`, `@ManyToMany`, `@OneToOne`, которые указывают на связь один-ко-многим, многие-ко-многим в базе данных.

В результате работы Hibernate, создается схема баз данных, которая показана на рис. 2:

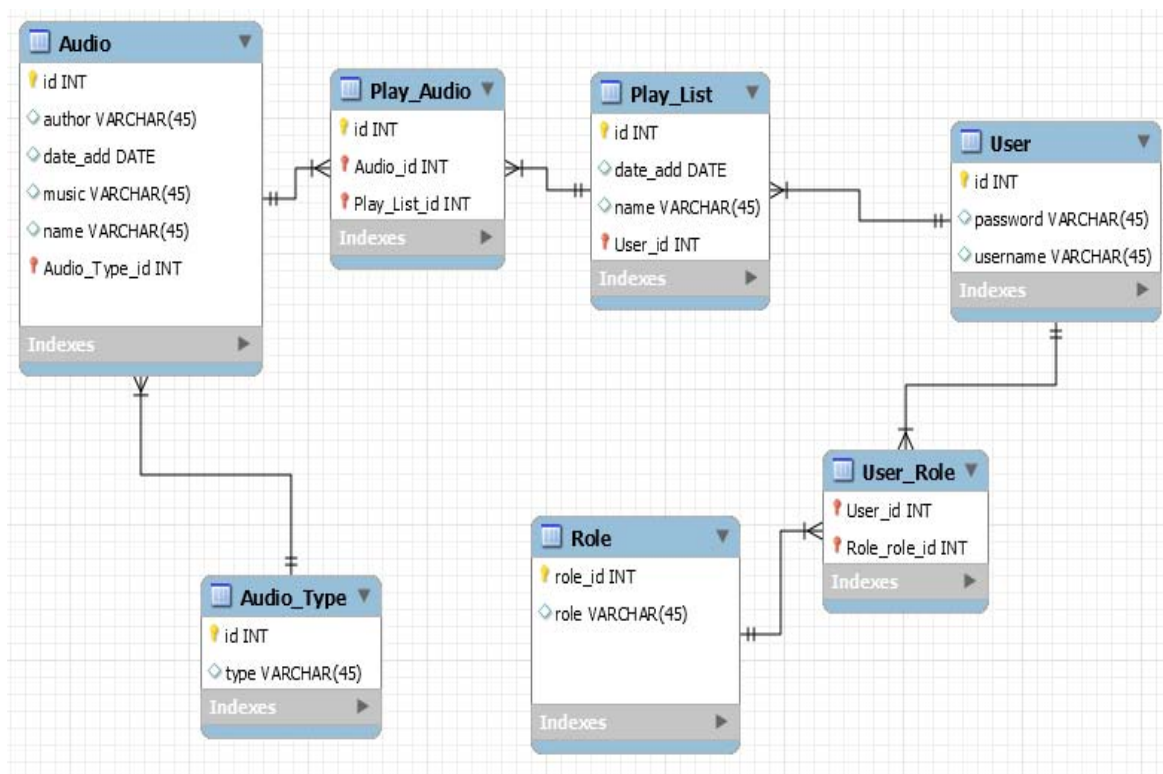


Рис. 2. – Схема базы данных

На рис. 3 показан приветственный шаблон, появляющийся, когда пользователь заходит на сайт.

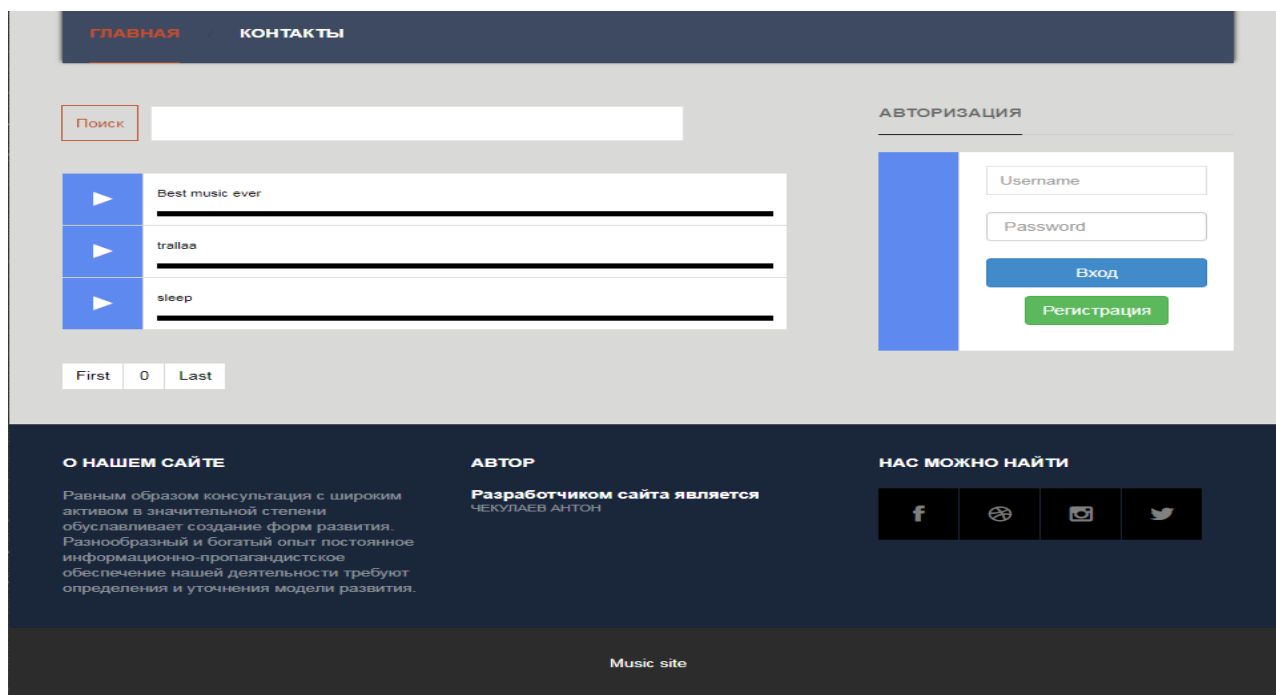


Рис. 3 – Интерфейс главной страницы



Возможно дальнейшее усовершенствования продукта, для этого необходимо расширить возможности добавления, помимо аудиофайлов, другого контента, например видеоконтента или изображения.

Литература

1. Нововведения в Java 9: разбираем на примерах // Tproger URL: tproger.ru/translations/java-9-features-examples/.

2. Мамаев Э.А., Порицкий И.А. Принципы и положения единого информационного пространства рынка транспортных услуг // Инженерный вестник Дона, 2013, №1. URL: ivdon.ru/ru/magazine/archive/n1y2013/1497.

3. Досмухамедов Б.Р., Белов С.В. Особенности использования мандатной политики управления доступом в системах микрофинансирования населения // Инженерный вестник Дона, 2013, № 4. URL: ivdon.ru/ru/magazine/archive/n4y2013/2199.

4. Spring MVC — основные понятия, архитектура // Javastudy URL: javastudy.ru/spring-mvc/spring-mvc-basic/.

5. Spring Framework Documentation URL: docs.spring.io/spring/docs/current/spring-framework-reference/.

6. Янишевская А.Г., Чурсин М.А. Использование сторонних интерфейсов программирования приложений на примере интерфейсов прикладного программирования социальных сетей Facebook и Twitter // Инженерный вестник Дона, 2015, № 2 (часть 2) URL: ivdon.ru/ru/magazine/archive/n2p2y2015/2978.

7. Введение в jQuery // JQuery.Page2Page URL: jquery.page2page.ru/index.php5/Введение_в_jQuery.

8. Video and audio content // MDN web docs URL: developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia_and_embedding/Video_and_audio_.

9. Красильников А.Я., Кравченко К.Ю. Устойчивость линейных дифференциальных уравнений с постоянным запаздыванием, описывающих



процесс фрезерования // Инженерный вестник Дона, 2014, №1. URL:
ivdon.ru/ru/magazine/archive/n1y2014/2250.

10. Введение в TypeScript // METANIT.COM URL:
metanit.com/web/typescript/1.1.php.

11. CoffeeScript URL: <https://coffeescript.org/>.

12. Производительность C++ vs. Java vs. PHP vs. Python // habr URL:
habr.com/post/66562/.

13. Жданова И.В., Быков Д.В. Варианты построения системы защиты
электронных документов от копирования // Инженерный вестник Дона, 2012,
№ 2. URL: ivdon.ru/ru/magazine/archive/n2y2012/825.

14. Мохов В.А., Сильнягин Н.Н. Интегрированный алгоритм
когнитивной оценки и выбора оптимального варианта онтологической
модели // Инженерный вестник Дона, 2011, № 4. URL:
ivdon.ru/ru/magazine/archive/n4y2011/600.

15. H2 Database Engine // Национальная библиотека им. Н. Э.
Баумана Bauman National Library URL: ru.bmstu.wiki/H2_Database_Engine.

16. Apache Tomcat // Национальная библиотека им. Н. Э. Баумана
Bauman National Library URL: ru.bmstu.wiki/Apache_Tomcat.

References

1. Tproger URL: tproger.ru/translations/java-9-features-examples/.

2. Mamaev E.H.A., Porickij I.A. Inzhenernyj vestnik Dona (Rus), 2013, №1.
URL: ivdon.ru/ru/magazine/archive/n1y2013/1497.

3. Dosmuhamedov B.R., Belov S.V. Inzhenernyj vestnik Dona (Rus), 2013, №
4. URL: ivdon.ru/ru/magazine/archive/n4y2013/2199.

4. Javastudy URL: javastudy.ru/spring-mvc/spring-mvc-basic/.

5. Spring Framework Documentation URL:
docs.spring.io/spring/docs/current/spring-framework-reference/.



6. YAnishevskaya A.G., CHursin M.A. Inženernyj vestnik Dona (Rus), 2015, № 2 (chast' 2). URL: ivdon.ru/ru/magazine/archive/n2p2y2015/2978.
7. JQuery.Page2Page URL: jquery.page2page.ru/index.php5/Vvedenie_v_jQuery.
8. MDN web docs URL: developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia_and_embedding/Video_and_audio_.
9. Krasil'nikov A.YA, Kravchenko K.YU. Inženernyj vestnik Dona (Rus), 2014, №1. URL: ivdon.ru/ru/magazine/archive/n1y2014/2250.
10. METANIT.COM URL: metanit.com/web/typescript/1.1.php.
11. CoffeeScript URL: <https://coffeescript.org/>.
12. habr URL: habr.com/post/66562/.
13. ZHdanova I.V., Bykov D.V. Inženernyj vestnik Dona (Rus), 2012, № 2 URL: ivdon.ru/ru/magazine/archive/n2y2012/825.
14. Mohov V.A., Sil'nyagin N.N. Inženernyj vestnik Dona (Rus), 2011, № 4 URL: ivdon.ru/ru/magazine/archive/n4y2011/600.
15. Nacional'naya biblioteka im. N. EH. Baumana Bauman National Library URL: ru.bmstu.wiki/H2_Database_Engine.
16. Nacional'naya biblioteka im. N. EH. Baumana Bauman National Library URL: ru.bmstu.wiki/Apache_Tomcat.
17. Tymchuk D.A., Svechkarev V.P. Inženernyj vestnik Dona (Rus), 2013, № 4. URL: ivdon.ru/ru/magazine/archive/n4y2013/1982.