

УДК 004.415.2.031.42

Ю. П. Быковский

САЙТ ДЛЯ ФИТНЕС-ЦЕНТРА НА ЯЗЫКЕ JAVA С ИСПОЛЬЗОВАНИЕМ ФРЕЙМВОРКА SPRING

Статья посвящена комплексной разработке и исследованию веб-приложения, предназначенного для автоматизации процессов управления современным фитнес-центром. Проект реализован с использованием языка программирования Java и фреймворка Spring, что позволило создать отказоустойчивое, масштабируемое и функциональное решение, удовлетворяющее требованиям цифрового обслуживания клиентов и администрирования спортивных учреждений.

Основной задачей системы является цифровизация ключевых бизнес-процессов: регистрация и авторизация пользователей, просмотр расписания тренировок с возможностью фильтрации по направлениям (йога, кардио, силовые тренировки и др.), онлайн-запись на групповые и персональные занятия, управление абонеентами, интеграция с системой уведомлений, контроль посещаемости, а также ведение внутренней аналитики и статистики. В рамках разработки акцент сделан на повышении удобства взаимодействия с системой как для клиентов, так и для персонала центра – администраторов и тренеров.

Приложение поддерживает несколько ролей: клиент, тренер, администратор, – каждая из которых имеет собственный набор прав и функциональных возможностей. Так, тренеры могут вносить изменения в своё расписание, просматривать списки записавшихся, оценивать посещаемость и получать обратную связь. Администратор имеет доступ к панели управления залами, тренерами, типами тренировок и тарифными планами, а также к административной панели для управления пользователями и статистическими отчётами. Пользователь может записаться на тренировку, просмотреть информацию об абонеентах, а также посмотреть свою историю посещений и количество оставшихся занятий по абонеенту в профиле.

Архитектурно приложение построено по многоуровневой модели с четким разделением слоев: представления (View), логики (Service), доступа к данным (Repository) и модели (Model). Серверная часть написана на Spring Boot с использованием REST API, что обеспечивает простоту взаимодействия с фронтендом и мобильными приложениями. Используется база данных PostgreSQL с продуманной схемой хранения информации: таблицы пользователей, расписаний, тренировок, абонеентов, платежей, уведомлений и обратной связи связаны отношениями «один ко многим» и «многие ко многим». Дополнительно используется Liquibase для управления миграциями и версионностью базы данных.

Серверная часть системы разработана на языке Java с использованием фреймворка Spring Boot [1]. Выбор данного фреймворка обоснован его модульностью, широким сообществом и поддержкой встроенных решений для безопасности, работы с базами данных, создания REST API и интеграции с другими сервисами. Для хранения информации применяется реляционная база данных PostgreSQL. Используется нормализованная схема с поддержкой транзакций, ограничений целостности и связей между сущностями, реализованных через JPA (Hibernate). Основные таблицы включают: users, trainer_info, trainer_schedule, gym_info, subscription, user_subscription, visit_history.

Безопасность веб-приложения реализована средствами Spring Security с использованием токенов JWT (JSON Web Token), обеспечивающих безопасную аутентификацию и авторизацию. Также внедрены механизмы защиты от CSRF-атак, шифрования паролей с помощью BCrypt и ограничения по IP-адресам для административной панели.

Клиентская часть построена на HTML5, CSS3 и JavaScript с применением шаблонизатора Thymeleaf. Внедрена адаптивная верстка для корректного отображения на мобильных и десктопных устройствах. Используется Bootstrap для унификации внешнего вида, а также подключаются кастомные JS-скрипты для динамической работы интерфейса. Форма обратной связи, календарь тренировок и фильтрация данных работают асинхронно с применением AJAX.

Проект предусматривает расширение за счёт дополнительных модулей. К примеру, модуль «Онлайн-оплата» позволяет интегрировать сервисы приема платежей, таких как Stripe, YooMoney или СберPay. В модуле «Видеотренировки» реализована возможность просмотра записанных занятий в личном кабинете клиента, а также проведения прямых трансляций через подключаемые видео-сервисы (YouTube, Zoom API). В будущем планируется разработка мобильных приложений для Android и iOS с использованием фреймворка Flutter или Kotlin Multiplatform и подключением через существующий REST API.

Особое внимание было уделено процессу тестирования: реализованы юнит-тесты с использованием JUnit и Mockito, а также интеграционные тесты для проверки корректности взаимодействия между слоями. UI-тесты выполнялись с помощью Selenium и Postman для ручной отладки REST-запросов. При разработке использовалась среда IntelliJ IDEA и система управления версиями Git с размещением репозитория на GitHub. Было проведено тестирование приложения для проверки некоторых параметров корректности: запуска приложения, добавления задач, редактирования записи, отображения приложения на различных устройствах; а также правильности сохранения данных при закрытии приложения. В ходе тестирования не было выявлено никаких ошибок, опечаток, сбоев. Приложение корректно отображается в телефоне, кнопки выполняют строго отведённую им роль.

Разработанное веб-приложение может быть внедрено как в локальной среде (через Tomcat), так и на облачных платформах – например, Heroku, Railway или DigitalOcean. Возможность настройки окружения с помощью Docker и docker-compose делает систему удобной для развертывания и масштабирования. Приложение может быть доработано для мультифилиальной работы с разделением на разные города и филиалы.

Разработанное приложение адаптировано под различные расширения экрана (до 720 px включительно), начиная с самых больших и заканчивая экранами мобильных устройств (см. рисунок 1).

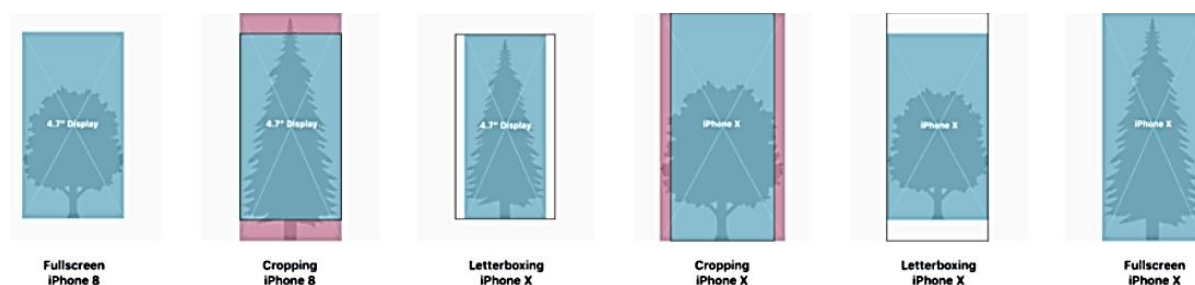


Рисунок 1 – Применение адаптации приложения

В результате был получен полноценный программный продукт, способный эффективно поддерживать деятельность фитнес-центра. Он может использоваться как готовое решение для малых и средних спортивных учреждений, а также как стартовая

платформа для построения более масштабных корпоративных систем управления в сфере спорта и оздоровления. Благодаря открытой архитектуре приложение также представляет интерес для научно-исследовательских и учебных проектов в области программной инженерии, информационных систем и бизнес-аналитики.

Литература

1 Нимейер, П. Программирование на Java. Исчерпывающее руководство для профессионалов / Патрик Нимейер, Дэниэл Леук ; [пер. с англ. М. А. Райтмана]. – М. : Эксмо, 2014. – 1216 с.

УДК 004.051

А. В. Владыковский

РАЗРАБОТКА КНИЖНОГО ИНТЕРНЕТ-МАГАЗИНА С ИСПОЛЬЗОВАНИЕМ ФРЕЙМВОРКА REACT.JS

Статья посвящена разработке книжного интернет-магазина с использованием фреймворка React.js. Рассмотрены этапы проектирования, технологический стек и ключевые функциональные возможности системы, такие как поиск, фильтрация и оформление заказов. Приводится схема базы данных и описываются особенности реализации безопасной авторизации через JWT. Результаты демонстрируют возможность создания масштабируемого и удобного в эксплуатации приложения, а перспективы развития включают интеграцию платёжных систем и персонализированных рекомендаций.

Современное развитие информационных технологий и бурное развитие электронной коммерции стимулируют создание специализированных онлайн-платформ. Интернет-магазин книг представляет собой не только удобное средство для покупки литературы, но и эффективный инструмент продвижения издательской продукции. Цель настоящей работы – описать процесс разработки веб-приложения, реализованного с использованием фреймворка React.js [1], а также обозначить его основные функциональные возможности и перспективы дальнейшего развития.

Разработка подобного приложения требует комплексного подхода. Необходим анализ бизнес-процессов, выбор оптимальных технологий, проектирование архитектуры клиентской и серверной частей, а также реализация схемы базы данных, обеспечивающей хранение информации о книгах, пользователях, заказах и отзывах. Рассмотрим основные фрагменты этих составляющих.

Поиск и фильтрация книг. Система реализует поиск по ключевым параметрам (название, автор, жанр). Пользователь может дополнительно сортировать результаты по цене, рейтингу или дате добавления.

Просмотр карточки книги. При выборе конкретного издания отображается подробная карточка с описанием, изображением обложки, информацией об авторе, а также отзывами других пользователей.

Работа с корзиной. Функционал корзины позволяет добавлять выбранные книги, изменять их количество и осуществлять оформление заказа. При оформлении заказа пользователю необходимо заполнить контактные данные, после чего заказ передаётся на обработку серверной части приложения.

Авторизация и безопасность. Реализован механизм регистрации и входа с использованием JSON Web Token (JWT) для защиты пользовательских данных.