Geekbrains

**Исследование особенностей создания программного продукта на языке Java в условиях отсутствия технического задания на примере создания сервиса для онлайн-записи в салон красоты**

IT-специалист:

Программист Java

Шевелев А.А.

Санкт-Петербург

2024

Содержание

1. Введение 3
2. Основная часть4

2.1. Используемые технологии4

2.2. Структура проекта6

2.3. Система безопасности7

1. Заключение8
2. Список использованной литературы9
3. Приложения10
4. **Введение**

Целью данного проекта является демонстрация знаний и навыков, полученных при обучении в компании Geekbrains по специальности Цифровые Профессии. Программист Java.

В качестве объекта для разработки был выбран сервис онлайн-бронирования для салона красоты. Однако при необходимости данный проект может быть использован и в других областях. Например, с незначительными изменениями, данный сервис можно использовать для бронирования переговорных комнат в организации и т.п.

1. **Основная часть**
   1. **Используемые технологии**

В основе разработанного сервиса лежит фреймворк Spring. Согласно Wikipedia, Spring - это универсальный [фреймворк](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) [с открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) для [Java](https://ru.wikipedia.org/wiki/Java)-платформы. Spring обеспечивает решения многих задач, с которыми сталкиваются Java-разработчики и организации, которые хотят создать информационную систему, основанную на платформе Java. Этот фреймворк предлагает последовательную модель и делает её применимой к большинству типов приложений, которые уже созданы на основе платформы Java. Считается, что Spring реализует модель разработки, основанную на лучших стандартах индустрии, и делает её доступной во многих областях Java.

По сути, Spring может быть рассмотрен как коллекция меньших фреймворков. Большинство этих фреймворков может работать независимо друг от друга, однако они обеспечивают большую функциональность при совместном их использовании.

Для создания «каркаса» проекта использовался *Spring Boot* - популярный фреймворк для создания веб-приложений с использованием Java. Это часть фреймворка Spring, которая представляет собой набор инструментов и библиотек для создания приложений корпоративного уровня. Spring Boot упрощает создание приложений, которые можно легко развернуть и запустить с минимальной конфигурацией.

Центральной частью Spring является контейнер *Inversion of Control*, который предоставляет средства конфигурирования и управления объектами Java с помощью рефлексии. Контейнер отвечает за управление жизненным циклом объекта: создание объектов, вызов методов инициализации и конфигурирование объектов путём связывания их между собой.

В качестве фреймворка для доступа к данным в данном проекте выбран *Spring Data JPA* (Java Persistence API), который позволяет радикально ускорить процесс разработки базового функционала по доступу к базе данных и улучшить читаемость кода.

В качестве фреймворка аутентификации и авторизации в данном проекте используется *Spring Security*: конфигурируемый инструментарий процессов аутентификации и авторизации, поддерживающий много популярных и ставших индустриальными стандартами протоколов, инструментов и практик.

В качестве базы данных в данном приложении использована H2 Database. Выбор обусловлен исключительно демонстрационными целями. В случае реального использования данного сервиса в качестве базы данных предполагается использовать PostgreSQL, для подключения к которой достаточно небольших изменений в конфигурационном файле.

Кроме того, в проекте использована библиотека *Lombok* – основанная на аннотациях библиотека Java, позволяющая сократить шаблонный код. В Lombok предлагаются различные аннотации, цель которых – заменить ненужный повторяющийся код.

И, наконец, для создания пользовательского интерфейса использован *Thymeleaf* - современный серверный механизм Java-шаблонов, способный обрабатывать HTML, XML, JavaScript, CSS и даже простой текст. Основной целью Thymeleaf является создание элегантного и удобного способа шаблонизации.

* 1. **Структура проекта**

При разработке данного проекта использовался шаблон (или паттерн) MVC (Model-View-Controller) - схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер.

Основная цель применения этой концепции состоит в отделении бизнес-логики (модели) от её визуализации (представления, вида). За счёт такого разделения модификация каждого компонента может осуществляться независимо, а также повышается возможность повторного использования кода.

Конечно, в рамках данного проекта использование шаблона MVC носит скорее демонстрационный характер, однако при повышении сложности программного продукта его использование безусловно оправдано.

В частности, могут решаться следующие задачи:  
- к одной модели можно присоединить несколько видов, при этом не затрагивая реализацию модели;  
- не затрагивая реализацию видов, можно изменить реакции на действия пользователя используя другой контроллер;  
- разделение «зон ответственности» разработчиков.

* 1. **Система безопасности**

Система безопасности сервиса онлайн-бронирования для салона красоты построена на фреймворке Spring Security.

Предусмотрены 3 роли пользователей:

- Администратор (ADMIN)

- Менеджер (MANAGER)

- Клиент (CLIENT)

Часть ресурсов системы доступна всем пользователям (в том числе незарегистрированным). Например, просмотр списка салонов, списка мастеров и прейскуранта на услуги.

Отдельные ресурсы доступны только пользователям с определенной ролью. Например, онлайн-бронирование доступно только клиентам, а просмотр и редактирование всех бронирований - только менеджерам.

# Заключение

В процессе разработки сервиса онлайн-бронирования для салона красоты подтвердилось удобство использования фреймворка Spring и, в частности, Spring Boot для быстрого старта проекта и постепенного наращивания его функциональности.

Использованные «дочерние» фреймворки, такие как Spring Security, Spring Data JPA, а также библиотеки Thymeleaf и Lombok позволяют избежать рутинного кодирования, ускорить процесс разработки, стандартизировать и структурировать программный код, улучшить его читаемость.

К сожалению, ограниченное время для разработки не позволило реализовать все задуманные функции. В частности, предполагалось реализовать механизм напоминания клиентам о ближайшей записи по электронной почте с использованием JavaMail API.

1. **Список использованной литературы**

1. Статья: «Spring Framework» ([Spring Framework — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Spring_Framework))

2. Статья: «Model-View-Controller» ([Model-View-Controller — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Model-View-Controller))

3. Книга: Герберт Шилдт. «Java. Полное руководство». Издательство «Диалектика-Вильямс», 2018 год , 1488 с.

4. Книга: Крейг Уоллс. «Spring в действии». Издательство «ДМК-Пресс», 2022 год , 544 с.

1. **Приложения**

Приложение 1. ER-диаграмма базы данных

