Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий
институт
Кафедра «Информатика»
кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №6

Конфигурация Spring Security
тема

Преподаватель _____ А.С. Черниговский
подпись, дата инициалы, фамилия

Студент КИ18-17/1б 031830504 _____ Е.В. Железкин
номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2020

1 Цель работы

Познакомиться с настройкой безопасности в Spring.

2 Задача работы

Взять практическое задание №5 и добавить следующий функционал:

1) Добавить простейшую страницу регистрации. Пользователь вводит свои логин и пароль, и данная информация вносится в базу данных, пользователю присваивается роль пользователя (User) приложения.

2) Добавить простейшую форму аутентификации. Форма должна быть создана студентом, а не автоматически сгенерированной Spring.

3) В приложении должен быть предусмотрен пользователь — администратор, с, отличной от роли User, ролью администратора (Admin).

4) Разграничить уровни доступа к страницам приложения. Пользователь (User) имеет доступ только к страницам просмотра всех записей и запросов.

Администратор (Admin) имеет возможность добавлять, редактировать и удалять записи.

5) Информация о пользователях и их ролях должна храниться в базе данных. Способ хранения — на усмотрение студента.

6) Предусмотреть возможность пользователю выйти из приложения (logout).

7) Продемонстрировать умение настраивать безопасность на уровне представлений. Для этого реализуйте приветствие пользователя после его входа и отображение элемента на основе его роли.

Вариант 9 - Книга

3 Ход работы

1) Официальный мануал, взятый за основу данной работы https://spring.io/guides/gs/securing-web/#run_the_app Добавлена конфигурация безопасности проекта (модуль безопасности, рисунок 1):

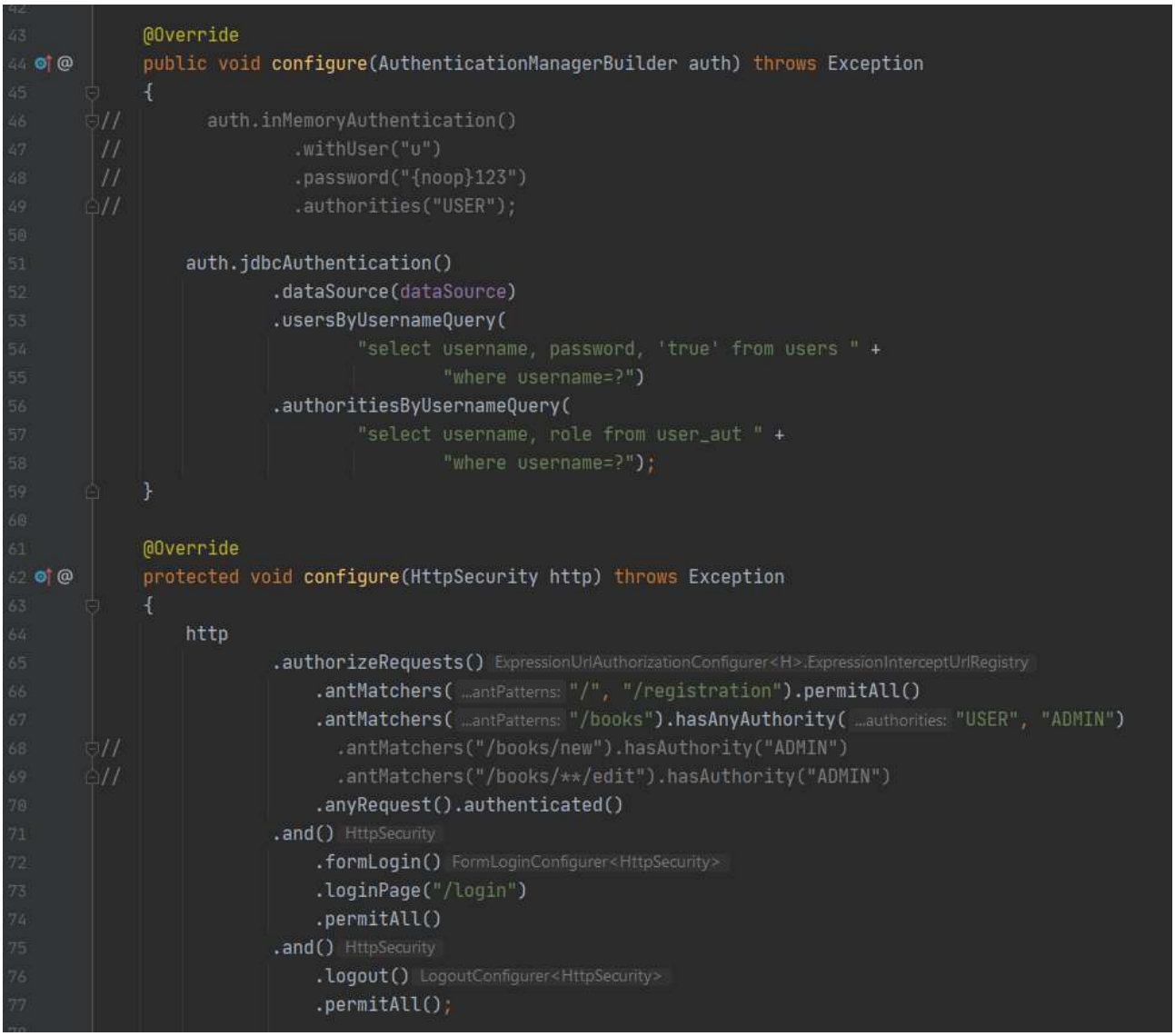

Рисунок 1 – Часть конфигурации (модуль security), SpringSecurityConfig.java

2) Реализован контроллер пользователя(доступа) (рисунок 2):

```java
@Controller
@RequestMapping
public class UserController
{
    private final UserServiceImpl userService;
    private final BookServiceImpl bookService;

    public UserController(@Qualifier("userService") UserServiceImpl UserService, @Qualifier("bookService") BookService
    {
        this.userService = UserService;
        this.bookService = bookService;
    }

    @GetMapping("/profile")
    public String getUserProfile(@AuthenticationPrincipal org.springframework.security.core.userdetails.User user,
                                 Model model)
    {
        model.addAttribute( s: "user", userService.findByUsername(user.getUsername()).orElse(new User()));
        model.addAttribute( s: "springUser", user);
        boolean isAdmin = Arrays.asList(user.getAuthorities().toArray())
                .contains(new SimpleGrantedAuthority( role: "ADMIN"));
        model.addAttribute( s: "isAdmin", isAdmin);
        model.addAttribute( s: "count", bookService.count());
        return "security/profile";
    }
```

Рисунок 2 – Часть контроллера пользователя, UserController.java

3) Созданы и настроены формы аутентификации (рисунок 3):

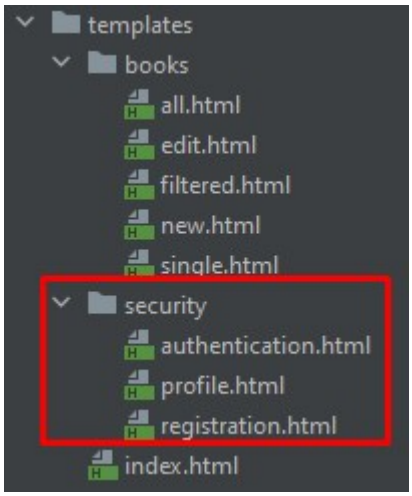

Рисунок 3 – Часть структуры проекта, отвечающей за представление (модуль view)

4) Интегрированы модули для работы с сущностью(книгой) из предыдущих работ (рисунок 4):

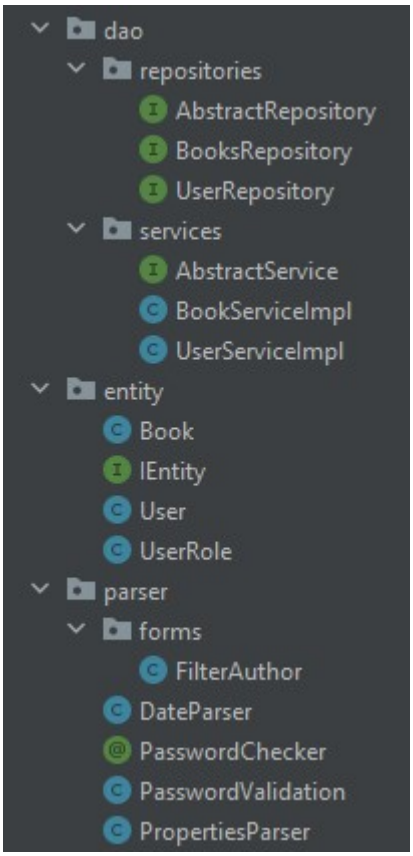

Рисунок 4 – Часть структуры проекта (модуль model) WebApp.java

5) Запущенное приложение (рисунки 5-8):

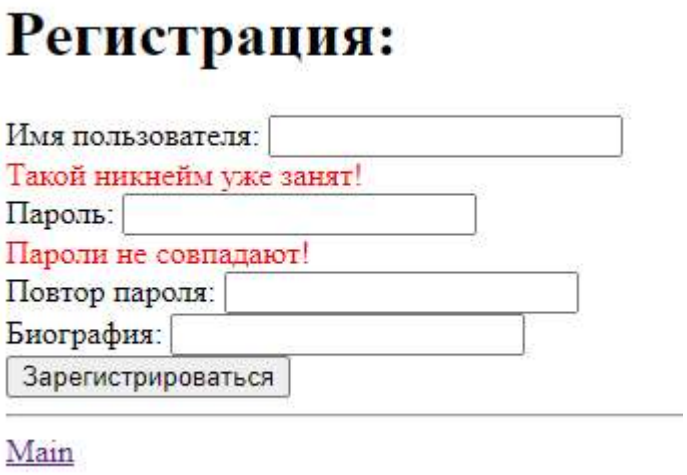

Рисунок 5 – Проверка на регистрации

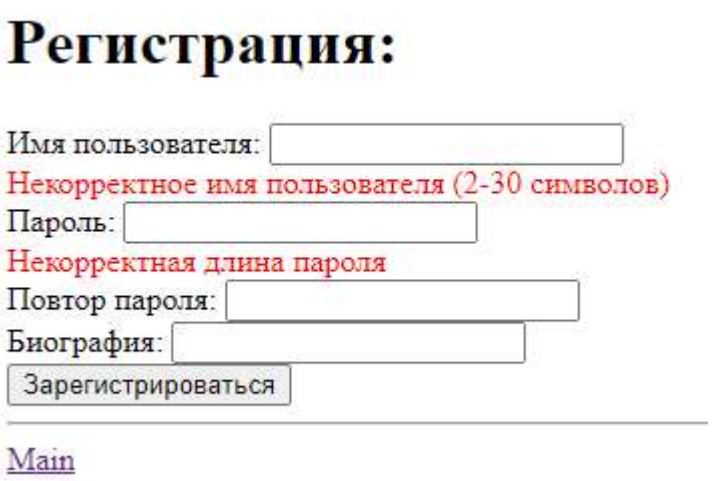

Рисунок 6 – Проверка на регистрации (2)

Рисунок 7 – Авторизация под пользователем, имеющем роль «USER»

Рисунок 8 – Авторизация под пользователемм, имеющем роль «ADMIN»

4 Вывод

В ходе данной лабораторной работы был изучены основы работы с настройкой безопасности в Spring.