

**Střední průmyslová škola elektrotechnická, Praha 2, Ječná 30**

**Informační technologie**

**Ječná 30, Praha 2**

**Library Management**

**Maksym Osetsymskyi**

**Informační a komunikační technologie**

**2025**

## *Obsah*

<i>1 Cíl práce .....</i>	<i>3</i>
<i>2 Popis aplikace .....</i>	<i>3</i>
<i>2.1 Popis – algoritmus .....</i>	<i>3</i>
<i>2.2 Postavy .....</i>	<i>4</i>
<i>2.3 Mechaniky .....</i>	<i>4</i>
<i>3 Systémové požadavky .....</i>	<i>5</i>
<i>4 Základní struktura .....</i>	<i>5</i>
<i>5 Testovací data .....</i>	<i>5</i>
<i>6 Uživatelská příručka .....</i>	<i>7</i>
<i>7 Závěr .....</i>	<i>7</i>
<i>8 Zdroje .....</i>	<i>8</i>

## 1 Cíl práce

Cílem projektu bylo vytvořit webovou aplikaci pro správu knihovny, která umožňuje kompletní správu knih, autorů, rezervací a uživatelů pomocí REST API a přehledného webového rozhraní. Projekt je založen na frameworku Spring Boot, využívá databázi PostgreSQL a je spravován pomocí Maven. Součástí zadání bylo také vytvoření jednoduchého frontendového rozhraní v HTML s využitím CSS a Bootstrapu.

Aplikace umožňuje:

- Vyhledávání knih a autorů.
- Registraci uživatelů a správu jejich profilu.
- Rezervaci knih pro uživatele.
- Kompletní CRUD operace (vytváření, čtení, úprava, mazání) nad knihami, autory, uživateli a rezervacemi.
- Správu uživatelů a rozšířené funkce přes admin rozhraní.
- Možnost dalšího rozšiřování a přidávání nových funkcí.

## 2 Popis aplikace

### 2.1 Popis – algoritmus

Library Management je webová aplikace postavená na architektuře klient-server, kde backend běží na Spring Boot s připojením na PostgreSQL databázi a frontend je realizován pomocí HTML/CSS a Bootstrapu.

- Uživatelský workflow: Uživatel se nejprve zaregistruje, poté může vyhledávat knihy a autory, rezervovat knihy, upravovat svůj profil, měnit heslo nebo uživatelské jméno.

- Pro administrátora je k dispozici admin panel, kde může spravovat knihy, autory, rezervace i uživatele (vytvářet, upravovat, mazat a zobrazovat jejich seznam).
- REST API poskytuje CRUD operace pro všechny entity, komunikuje ve formátu JSON.
- Frontend zobrazuje data uživatelům v přehledné podobě, umožňuje také interaktivní správu dat.

## 2.2 Postavy

V aplikaci nejsou klasické postavy, ale existují dvě základní uživatelské role:

- Uživatel – může prohlížet, vyhledávat a rezervovat knihy, upravovat svůj profil a heslo, vidět své rezervace.
- Administrátor – má přístup do administračního rozhraní, kde může spravovat všechny entity (knihy, autory, rezervace i uživatele).

## 2.3 Mechaniky

- CRUD operace: Přidávání, editace, mazání a zobrazení knih, autorů, rezervací a uživatelů.
- Vyhledávání: Uživatel může vyhledávat knihy a autory podle jména/názvu.
- Rezervace: Uživatelé mohou rezervovat knihy a spravovat své rezervace.
- Správa profilu: Uživatel může upravit své údaje a heslo.
- Administrace: Administrátor má přehled a správu nad všemi daty v systému, včetně správy uživatelů.
- Zobrazování detailů: Detailní popis knihy/autora, včetně odkazu na Wikipedii pro autora.
- Bezpečnost: Autentikace a autorizace uživatelů (Session based).
- Přehlednost: Přehled všech entit s možností filtrovat a vyhledávat.

### 3 Systémové požadavky

Aplikace je vyvíjena v Java 21 (OpenJDK 21), využívá Spring Boot, Maven a PostgreSQL.

Možnosti spuštění:

#### 1. Přes Docker:

- Nutné mít nainstalovaný Docker a IDE podporující Javu 21.
- Projekt je potřeba nejprve zbuildovat pomocí mvn clean package.
- Spuštění kontejnerů přes příkaz docker-compose up --build.
- Po spuštění je aplikace dostupná na <http://localhost:9000/>
- Pro administrátora použít login a heslo: admin / admin
- Pro běžného uživatele je nutná registrace.

#### 2. Lokální spuštění bez Dockeru:

- Mít nainstalované IDE, Java OpenJDK 21, Maven a PostgreSQL.
- Vytvořit databázi library (lze použít výchozí port 5432, případně změnit v application.properties).
- V souboru application.properties (adresář src/main/resources) nastavit skutečné hodnoty pro databázového uživatele:

spring.datasource.username=...

spring.datasource.password=...

- Projekt spustit v IDE nebo přes Maven (mvn spring-boot:run).

### 4 Základní struktura

Projekt je rozdělen do několika balíčků (packages):

## **- Controllers**

- Rozdělené na REST a View.

## **- *REST*:**

- AuthController – správa autentikace, registrace, přihlášení.
- AuthorController, BookController, BorrowController – CRUD operace, komunikace ve formátu JSON.

## **- *View*:**

- AuthorViewController, BookViewController, BorrowViewController, UserViewController, MainViewController – zpracování požadavků z webového rozhraní, vrací HTML stránky, redirekty.

## **- DTOs**

- Obsahuje datové třídy pro přenos dat mezi frontendem, backendem a databází.
- Pro každou entitu (Author, Book, Borrow, User) existují DTO pro uložení, úpravu i zobrazení.

## **- Entities**

- Datové modely pro entity Author, Book, Borrow (rezervace), User.

## **- Repositories**

- Rozhraní pro komunikaci s databází (pro každou entitu vlastní repository).

## **- Service**

- Obsahuje byznys logiku pro každou entitu.
- Pro User rozděleno na UserManagementService (správa uživatelů) a UserServ (načítání uživatelů podle jména).

## **- Security**

- Konfigurace bezpečnosti (SecurityConfig), vlastní implementace uživatelských detailů (UserDetailsImpl).

## **5 Testovací data**

Testování probíhalo jak na úrovni uživatelského rozhraní, tak pomocí REST API (např. přes PostMan).

Testovací scénáře:

- Administrátor:
  - Přihlášení do admin rozhraní, vytváření, editace a mazání knih, autorů, uživatelů a rezervací.
  - Kontrola, zda jsou CRUD operace správně reflektovány v databázi a zobrazují se v přehledech.
- Uživatel:
  - Registrace a přihlášení.
  - Vyhledávání knih a autorů, detailní zobrazení.
  - Vytváření rezervací, kontrola zobrazení vlastních rezervací.
  - Úprava vlastního profilu, změna hesla.
  - Ověření, že uživatel nemůže provádět administrátorské operace.

## 6 Uživatelská příručka

- Uživatel navštíví webovou aplikaci a provede registraci.
- Po přihlášení může vyhledávat knihy a autory, rezervovat knihy, prohlížet si své rezervace.
- V sekci profilu může upravit své osobní údaje a změnit heslo.
- Administrátor má navíc možnost přepnout do administračního panelu, kde může spravovat knihy, autory, rezervace a uživatele.
- Všechny operace jsou přehledně dostupné z hlavního menu.
- U každé knihy je možné zobrazit detailní popis, u autora odkaz na jeho stránku na Wikipedii.

## 7 Závěr

Projekt Library Management mě fakt bavil, protože jsem se díky němu naučil spoustu nových věcí – hlavně jak pracovat se Spring Bootem, PostgreSQL databází, dělat REST API a něco málo i s frontendem přes HTML a Bootstrap. Také jsem si poprvé zkusil testování přes PostMan a nasazení přes Docker, což mi přišlo zajímavé.

Pro mě osobně bylo asi nejtěžší udělat správně autorizaci a autentikaci. Původně jsem to chtěl vyřešit pomocí JWT tokenů, ale měl jsem s tím dost problémů a zabralo mi to hodně času. Nakonec jsem to změnil na Session based autentikaci a to už fungovalo dobře.

Celkově jsem rád, že se mi podařilo aplikaci rozchodit a je připravená na případné rozšiřování o nové funkce.

## 8 Zdroje

**ChatGPT** – pomohl s vytvářením funkce vyhledávání knih podle názvu.