

Checkpoint 2

Meeting Room Reservation System

Verze dokumentu: 2.2.0

Datum odevzdání: 15.01.2023

Název projektu: Meeting Room Reservation System

Odkaz na stránky projektu: [B221_B6B36EAR/rastvdmy](https://b221_b6b36ear.rastvdmy.cz/)

Řešitele: Dmytro Rastvorov

Termín cvičení: ST – 18:15

Jméno cvičícího: Miroslav Holeček

Obsah

- Téma
- Krátký popis očekávané funkcionality
- Druhy uživatelů
- Funkce pro jednotlivé typy uživatelů
- Systémová omezení
- Objektový model (UML class diagram)
- Databáze
- API
- Popis aplikace a její struktury
- Poznámky k použití projektu
- Docker
- Instalace aplikaci
- Odkazy
- Zpětná vazba

- **Téma:**

Meeting Room Reservation System

- Systém, který podporová správu jednacích místností a jejich rezervaci.

- **Krátký popis očekávané funkcionality:**

1. Seznam hlavních funkcí aplikace:

- Systém správy se skládá z uživatelské a administrativní části:

- **Uživatel**

- **Administrator**

2. Komu bude výsledný systém určen:

- Tento systém je určený pro skupinu osob, které plánují schůzky v konferenčních místnostech.

- **Druhy uživatelů:**

❖ Systém je určen pro 2 typy uživatelů: **Uživatel** a **Administrátor**.

- **Uživatel** – je uživatel, který komunikuje s aplikací pro rezervaci konferenčních místností.
- **Administrator** – je uživatel, který spravuje rezervace konferenčních místností.

Jaké funkce bude plnit

❖ Systém obsahuje následující funkce pro uživatele:

- **Uživatel** – má možnost rezervovat místnost, zrušit rezervaci, uvidí stav místnosti a čas, kdy bude rezervována.
- **Administrator** – má možnost přidávat místnost, odstranit místnost, změnit vlastnosti místnosti a má stejné možnosti, které má Uživatel.

● Systémová omezení

❖ Systém má pro uživatele omezení.

➤ Uživatel si nemůže:

- Rezervovat 1 pokoj 2 krát (stejně jako i administrátor)
- Vytvořit a odstranit místnost.
- Upravit vlastnosti místnosti.
- Zobrazit uživatele, kteří mají rezervované pokoje kromě sebe

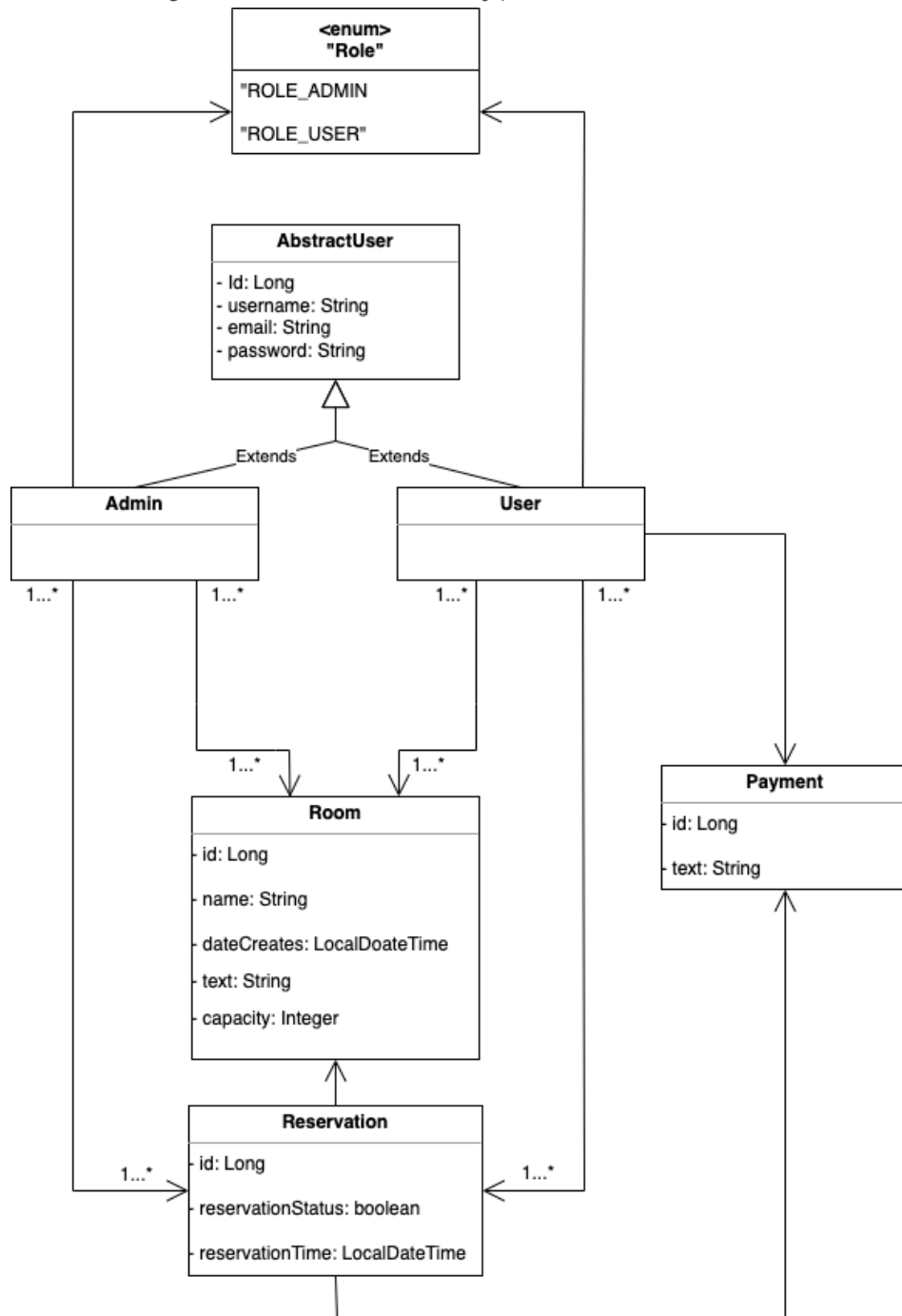
❖ Taky máme omezení pro aplikaci:

- Máme omezení v čase podle reálného času. To jest my nemůžeme udělat rezervaci na příští rok nebo měsíc, ale můžeme jen v tuto dobu v tomto měsíci.
- Nemůžeme zadávat chybná data. Budeme dostávat chybu.

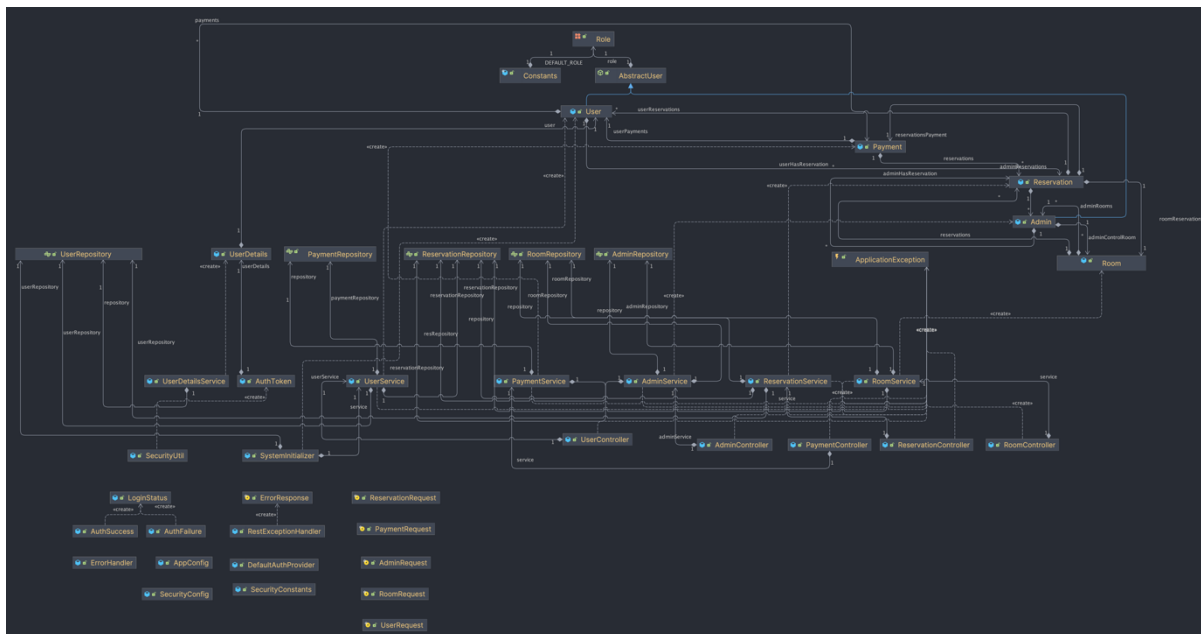
- **Objektový model (UML class diagram)**

- UML diagram (vytvořen pomocí draw.io)

(Demonstruje vazbu mezi entity)



- UML diagram vytvořen pomocí **IntelliJ IDEA**
(Demonstruje plnou vazbu mezi třídami)



- **Databáze**

Projekt využívá databázi typu PostgreSQL.

- **API**

Projekt využívá rozhraní REST API pomocí aplikace Postman. V dokumentaci je možný najít všechna API, které pokrývají logiku projektu.

Dokumentace REST API (Postman) je možný najít zde:

<https://documenter.getpostman.com/view/22903223/2s8ZDU4Nzj>

- **Popis aplikace a její struktury**

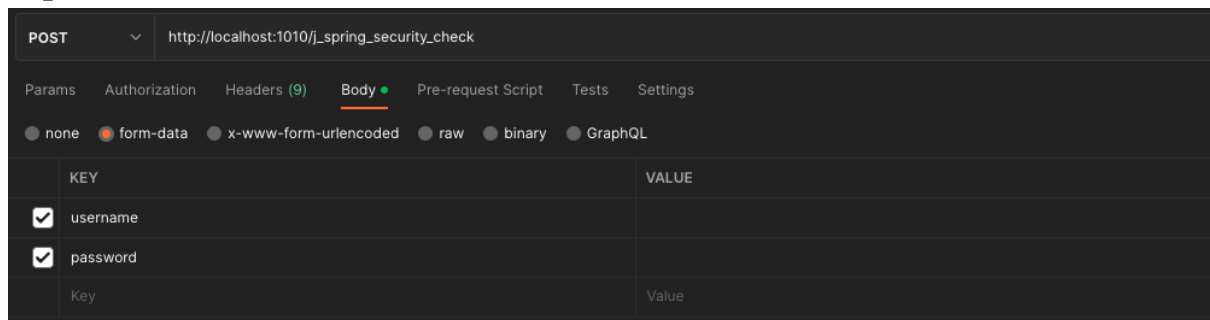
Tato aplikace umožňuje dělat rezervaci místnosti, kde user může vytvořit rezervaci, zobrazit jeho hodinovou cenu a zjistit informace o něm.

Na druhou stranu má admin mnohem více možností, než user. Kromě toho, co může dělat user, má admin plnou volnost při vytváření místností, rezervací, userů a dalších adminů a může je také měnit a mazat.

● Poznámky k použití projektu

Projekt využívá Security podporu, proto aby uživatel mohl program používat, musí si vytvořit vlastní profil.

Při spuštění v aplikaci Postman ve složce **REST API/Authorization/POST** můžete přihlášení potvrdit a dole se zobrazí output (viz níže obrázek) který říká, že přihlášení se dopadlo úspěšně.



● Docker

Jako bonus byl přidán **Docker**, takže se můžeme připojit k databázi pomocí aplikace a také vygenerovat soubor Docker. Pro propojení s databází používáme **docker-compose.yml**

● Instalace aplikaci

!! DŮLEŽITÉ: Projekt funguje ve spojení s JDK 18. Před instalací a spuštěním projektu se ujistěte, že je vaše verze aktuální.

- 1) Zkopírujte SSH projektu ze stránky [Git](#).
- 2) Otevřete na svém počítači terminál a napište **git clone** ([odkaz SSH](#))
- 3) Soubor se objeví ve složce, ke které jste zadali cestu v terminálu.

- 4) Otevřete soubor projektu v prostředí IDE.
(Doporučuji [IntelliJ IDEA](#))
- 5) Poté je třeba nakonfigurovat databázi v souboru **src/main/resources/application.yaml** a taky když chceme využívat Docker, musíme nakonfigurovat v souboru **docker-compose.yml**.
- 6) Otevřete **Maven**, přejděte na položku Lifecycle, rozbalte ji a klikněte na položku package
(poté, když budete něco změnit, tak musíte na začátku vybrat clean a pak package). Tím se projekt vygeneruje, projde všemi potřebnými operacemi a bude připraven ke spuštění.
- 7) Po dokončení tohoto procesu přejděte do složky **src/main/java/reservation/room/meeting/sem/ear** a vyberte soubor **Application.java**.
- 8) Poté klikněte na šipku spuštění a projekt se začne kompilovat.
- 9) Pak spustíme [Postman](#), otevřeme složku **Authorization/POST**, napíšeme nakonfigurované přihlašovací údaje a můžete projekt využívat.

● Odkazy

Odkaz na projekt: https://gitlab.fel.cvut.cz/B221_B6B36EAR/rastvdmj

Odkaz na dokumentaci v Postmanu:

<https://documenter.getpostman.com/view/22903223/2s8ZDU4Nzi>

● Zpětná vazba

Pro mě byl tento předmět něčím novým a vzrušujícím. Dozvěděl jsem se, jak funguje Java Spring Boot, k čemu slouží a jaké má funkce.

Při psaní semestrálního projektu jsem se mohl naučit o psaní Spring Boot aplikaci, propojení k databázi, použití Dockeru a Postmanu.

Semestrální projekt mě také naučil pracovat samostatně a řešit chyby, které se mi zdály obtížné.

Rád bych vyjádřil zvláštní poděkování přednášejícím a také cvičícímu Miroslavu Holečkovi za příjemný zážitek z tohoto předmětu a vklad do našich znalostí!

Děkuji vám za pozornost!