



# RoomRes

Semestrální práce NSS

Matouš Najman  
Vojtěch Luňák  
Pavel Sušický  
Lenka Hornová

Letní semestr B211

## Popis aplikace

Rezervační systém místností. S místnostmi si mohou uživatelé rezervovat místnosti i další zdroje. Aplikace generuje reporty podle filtrů. Automatická notifikace. (OAuth2 přihlašování).

	Vytvořit / create	Číst / read	Změnit / update	Smazat / delete
USER	✓	✓	✓ (owner)	✓ (owner)
ADMIN	✓	✓	✓	✓
GUEST	X	✓	X	X

*Práva realizace CRUD operací podle typu uživatele.*

## Motivace

Chceme si v rámci předmětu NSS nasimulovat situaci, kdy si nás zaplatila nějaká společnost, abychom jim vytvořili rezervační systém podle jejich potřeb.

## Strategický záměr

Rezervační systém je určen pro libovolnou firmu, která si přeje zefektivnit a zrychlit správu rezervací místností za účelem šetření času, lidských zdrojů a z toho plynoucích finančních prostředků.

## SWOT analýza

<b>Strengths</b> Jednoduchost, přehlednost Rezervace bez přihlášení Webová aplikace Automatická upozornění	<b>S</b>	<b>W</b>	<b>Weaknesses</b> None :D
<b>Opportunities</b> Škálovatelnost na více poboček Integrace s účetním systémem	<b>O</b>	<b>T</b>	<b>Threats</b> Odmitání používání systému ze strany zaměstnanců

## Analýza 5F (Porter's Five Forces)

### Stávající konkurenti:

- Reenio
- Reservio
- A další

První 2 služby umožňují vytvoření a následnou správu rezervačních systémů. Ovšem fiktivní firma od nás chce vytvořit jejich vlastní rezervační systém a nechce využívat řešení podobného rázu.

**Potenciální konkurence:**

V tuto chvíli neevidujeme žádnou nově vznikající konkurenci.

**Dodavatelé:**

Naše řešení dodavatele neřeší.

**Uživatelé:**

Rezervační systém je určen pro libovolnou firmu, která si přeje zefektivnit a zrychlit správu rezervací místností za účelem šetření času, lidských zdrojů a z toho plynoucích finančních prostředků.

**Substituty:**

Substituty jsou již zmíněné mezi konkurenty. Lze využít externí službu na vytvoření rezervačního systému a následnou správu (viz. Reenio).

## Analýza PEST

**Politicko-legislativní vlivy**

Náš systém musí ručit za bezpečnost osobních dat uživatelů (GDPR).

**Ekonomické faktory**

Náš projekt je simulací, takže žádné ekonomické faktory neuvažujeme. Kdybychom ale měli nahlédnout k realitě, tak by se řešila reklama a marketing. Zároveň bychom zavedli sazby za zprovoznění systému v pobočkách.

**Sociálně-kulturní faktory**

Náš systém umožňuje pouze rezervaci místností a správu rezervačního systému. Není zde žádné místo na projevení názoru nebo přidávání vlastního obsahu. Vyhnutí se nevhodnému obsahu si obstarává zadavatel.

**Technologické vlivy**

Systém je postavený na future-proof technologiích.

## Funkční požadavky

**FRQ1 – Registrace**

Uživatel musí být schopen vytvořit si účet.

**FRQ2 – Přihlášení**

Uživatel je schopen se do systému přihlásit.

**FRQ3 – Vytvoření události**

Systém umožní uživatelům vytvořit událost.

**FRQ4 – Úprava události**

Systém umožní uživateli, který má na to práva, změnit již vytvořenou událost.

**FRQ5 – Smazání události**

Systém umožní uživateli, který má na to práva, smazat již vytvořenou událost.

**FRQ6 – Pozvání uživatelů k události**

Systém umožní uživateli, který je součástí události a má na to práva, pozvat další uživatele k události.

**FRQ7 – Přidání uživatelů k události**

Systém umožní uživateli, který je součástí události a má na to práva, rovnou přidat další uživatele k události.

**FRQ8 - Vytvoření rezervace**

Systém umožní uživatelům, která na to mají práva, vytvořit rezervaci patřící k dané události.

**FRQ9 – Úprava rezervace**

Systém umožní uživatelům, kteří na to mají práva, upravit již vytvořenou rezervaci.

**FRQ10 – Smazání rezervace**

Systém umožní uživatelům, kteří na to mají práva, smazat již vytvořenou rezervaci.

**FRQ11 – Zobrazení rezervací**

Systém je schopen zobrazit informace o rezervaci – náležitost k události, časy začátku a konce.

**FRQ12 – Zobrazení událostí**

Systém umožní zobrazit informace o události – název, pořadající, časy, dílčí rezervace.

**FRQ13 – Přidání pobočky**

Systém umožní přidat adminovi pobočku.

**FRQ14 – Úprava pobočky**

Systém umožní upravit adminovi informace o pobočce.

**FRQ15 – Smazání pobočky**

System umožní adminovi smazat již vytvořenou pobočku.

#### **FRQ16 – Vytvoření reportu**

System umožní generování reportů rezervací/eventů/poboček.

#### **FRQ17 – Notifikace uživatelů**

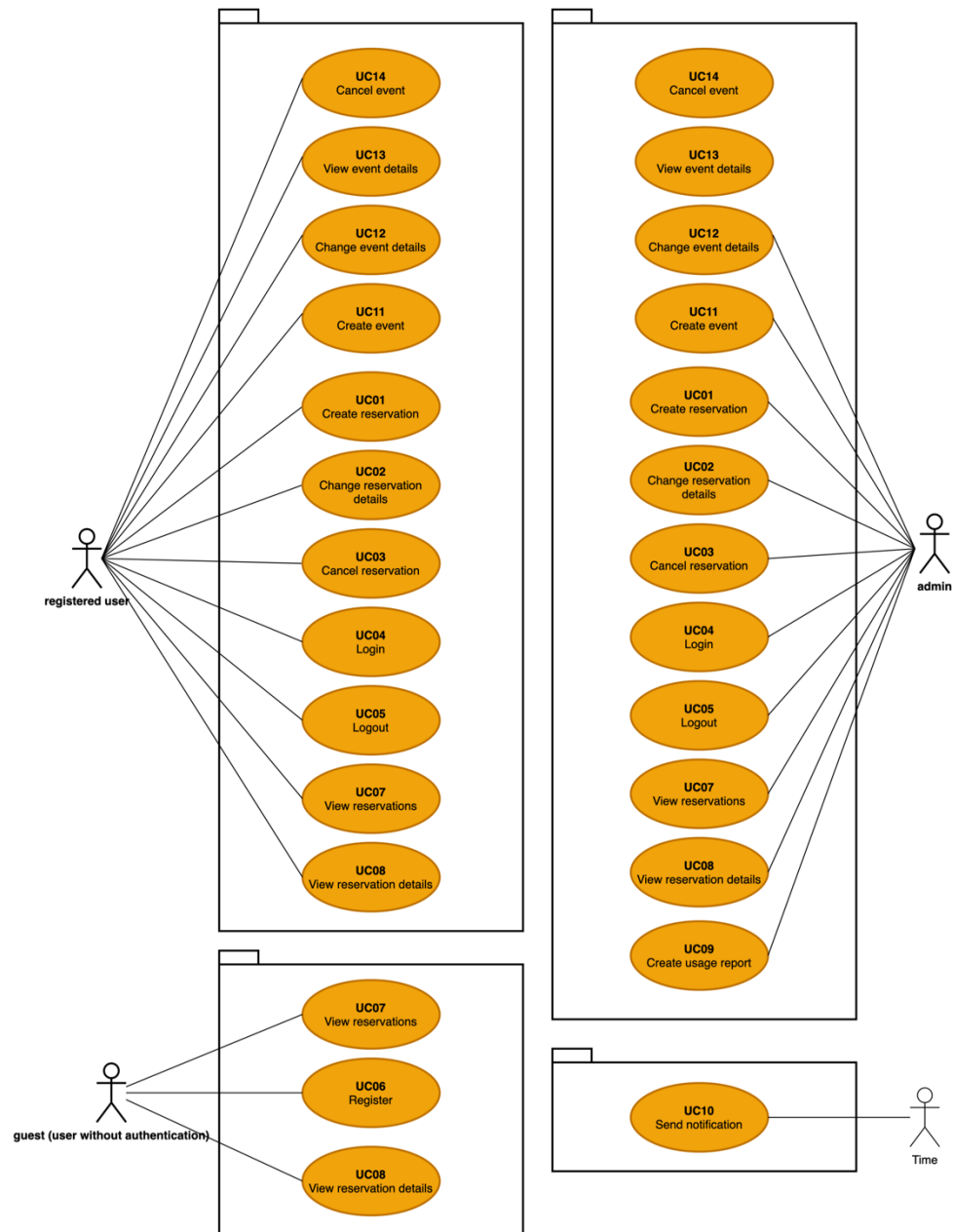
System umožní notifikovat uživatele při definovaných akcích (změna rezervace, pozvání do události, ...).

### **Nefunkční požadavky**

#### **NFRQ1 – Uživatelské rozhraní v angličtině**

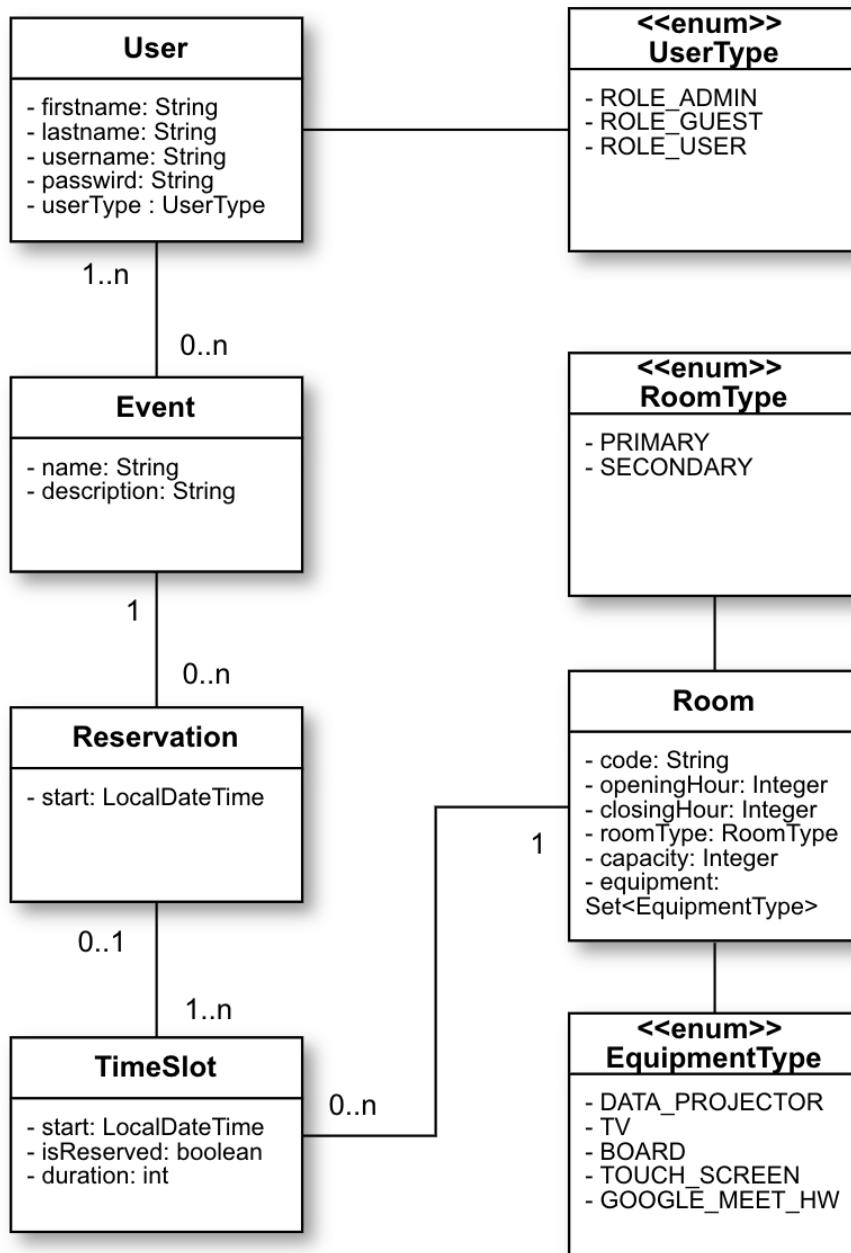
#### **NFRQ2 – Zabezpečení uživatelských informací**

# Use Cases



## Class diagram

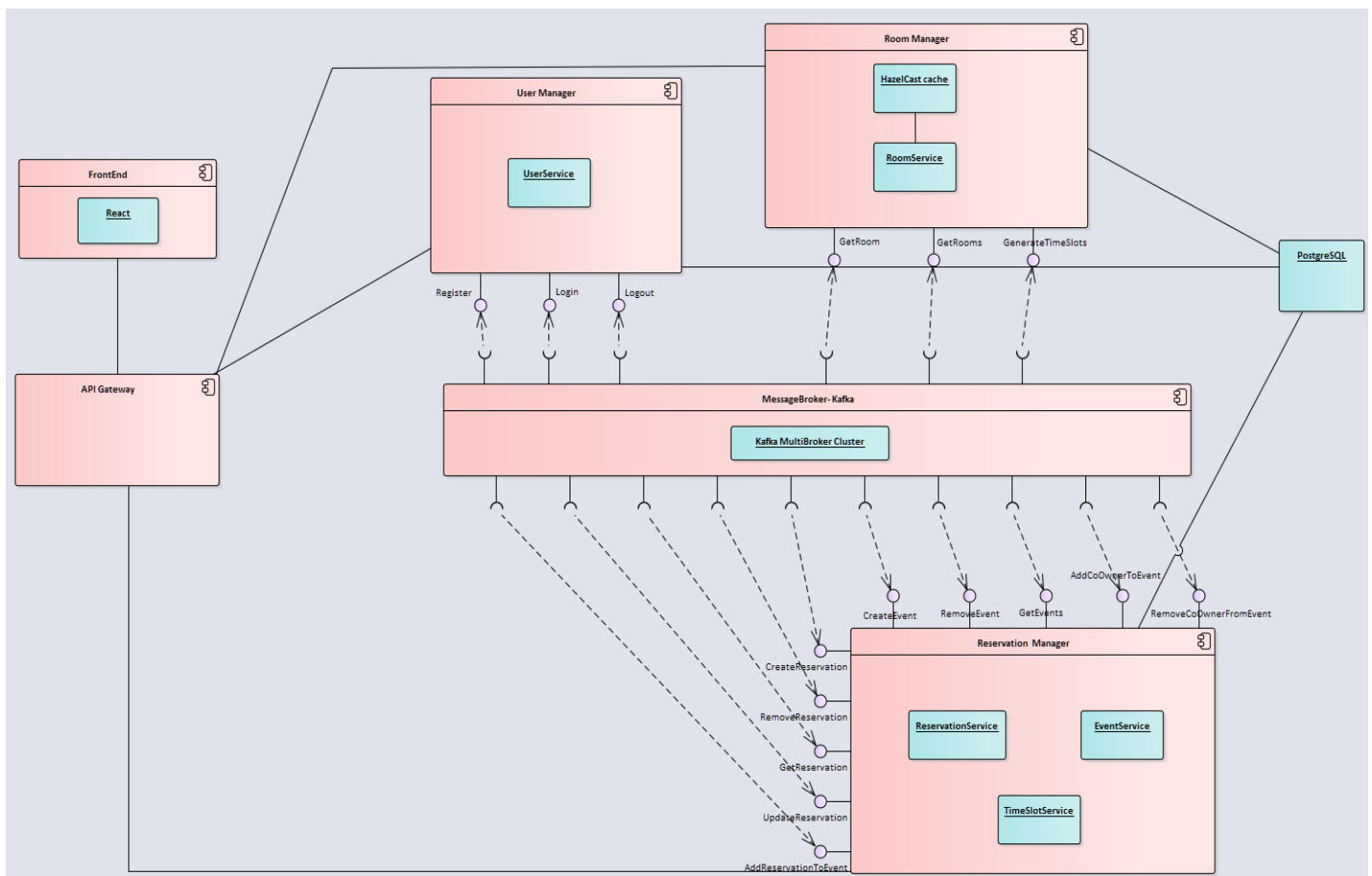
Reservation má start jen z důvodu jednoduchosti práce s rezervacemi.



## Architektura

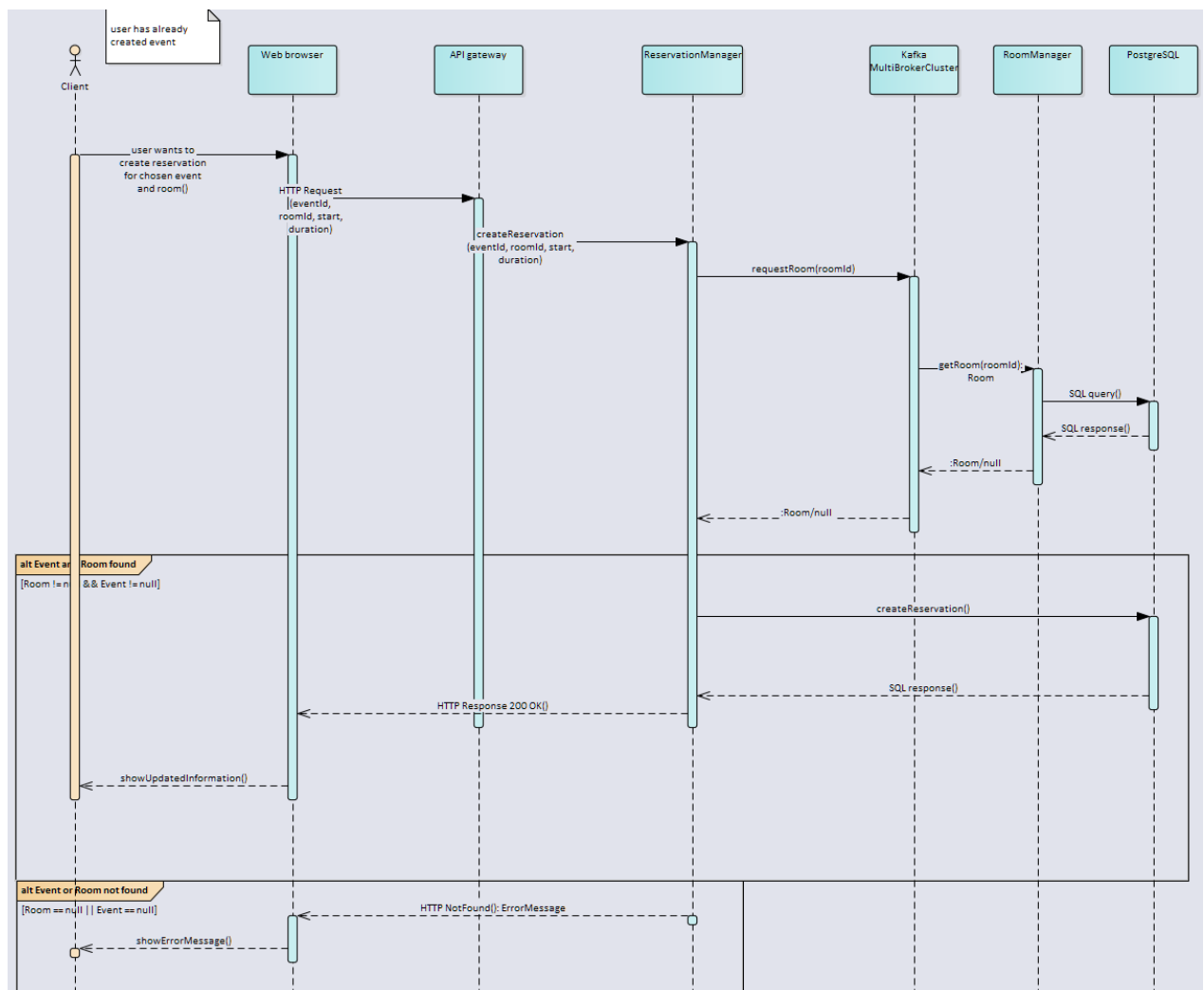
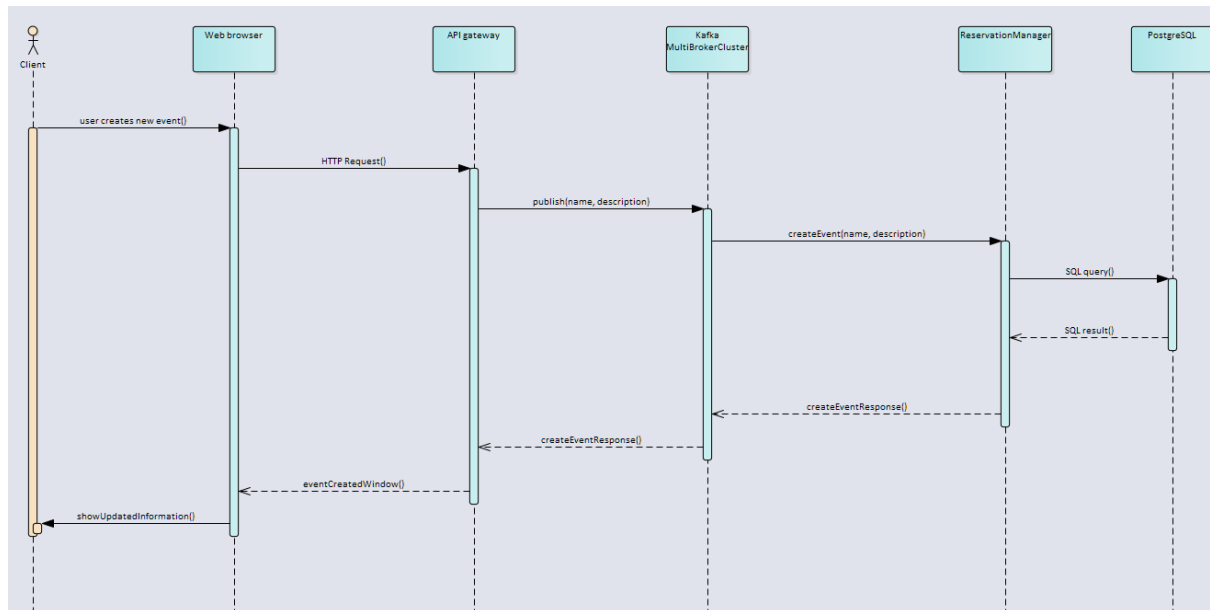
Architekturu jsme, na základě konzultace s vámi, zvolili viz. Obrázek - mikroservisní architektura. Využití Microservices, Kafka MultiBroker Cluster a API Gateway.

Komunikace mezi klientem a serverem bude zprostředkováno skrze Rest API. Posílání notifikací a REST requestů bude zajištěno pomocí Apache Kafka. Cache layer na úrovni mezi backendem a databází bude vytvořen pomocí Hazelcast.

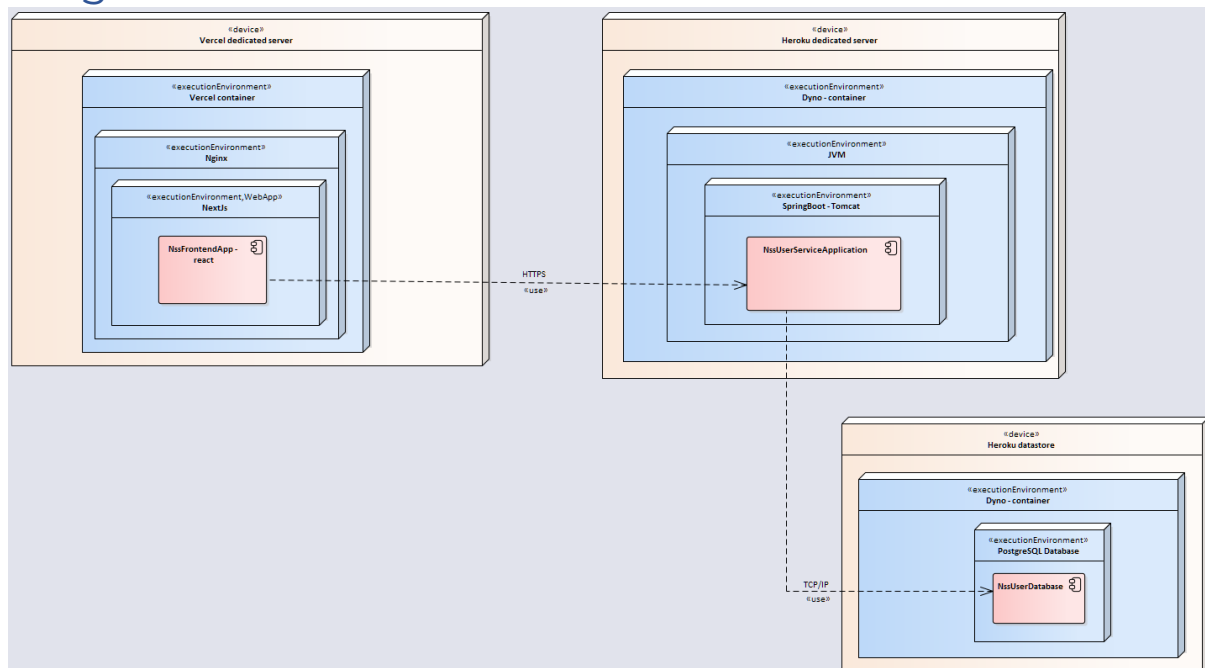




# Sekvenční diagramy



## Diagram nasazení



## Rozbor a výběr alternativ návrhu řešení

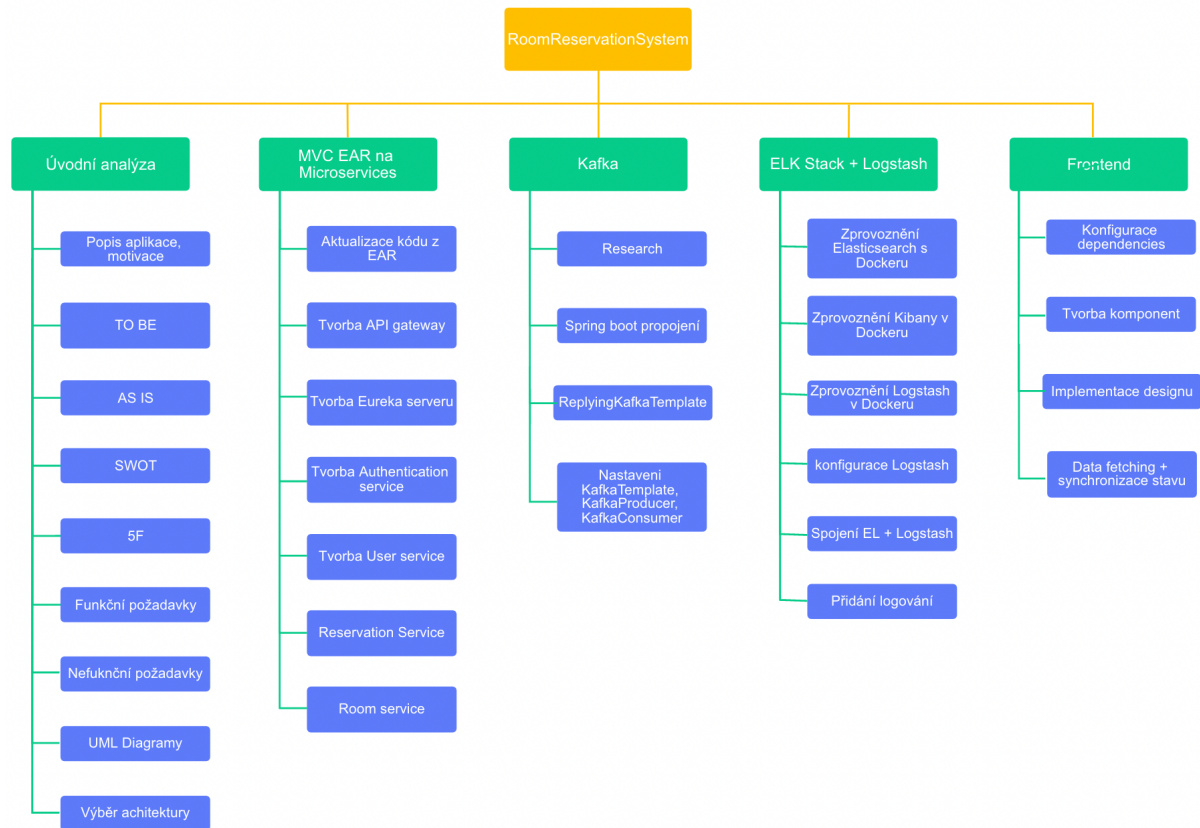
### Použité technologie

- Ideální technologie a programovací jazyk => **SpringBoot + Java**
- Readme file v gitu s popisem a instrukcemi
- Běžná database => **PostgreSQL**
- Využití Cache => **Hazelcast cluster cache** (Reservation Service)
- Messaging => **Kafka**
- Security => **JWT Authentication** (UserService)
- Interceptory => **GlobalInterceptor** (ApiGateway), **AuthTokenFilter** (Reservation and Room Services v package security – JWT check)
- **REST API**
- Nasazení produkce na **Heroku** – standalone User microservice na <https://nss-reservation-system.herokuapp.com> (branch heroku\_deploy)
- Struktura Microservices
- Použití ElasticSearch => **ELK Stack**
- 6 design pattern:
  - **Facade** – ReservationController má více service, aby agregoval data efektivně.
  - **Interceptors** – popsáno výše
  - **Builder** – Více Spring configů je realizováno pomocí builder pattern
  - **Factory** – Kafka Consumer a Producer definují Factory na vytváření specifických producer a consumer
  - **Dependency Injection** – pomocí anotace @Autowired

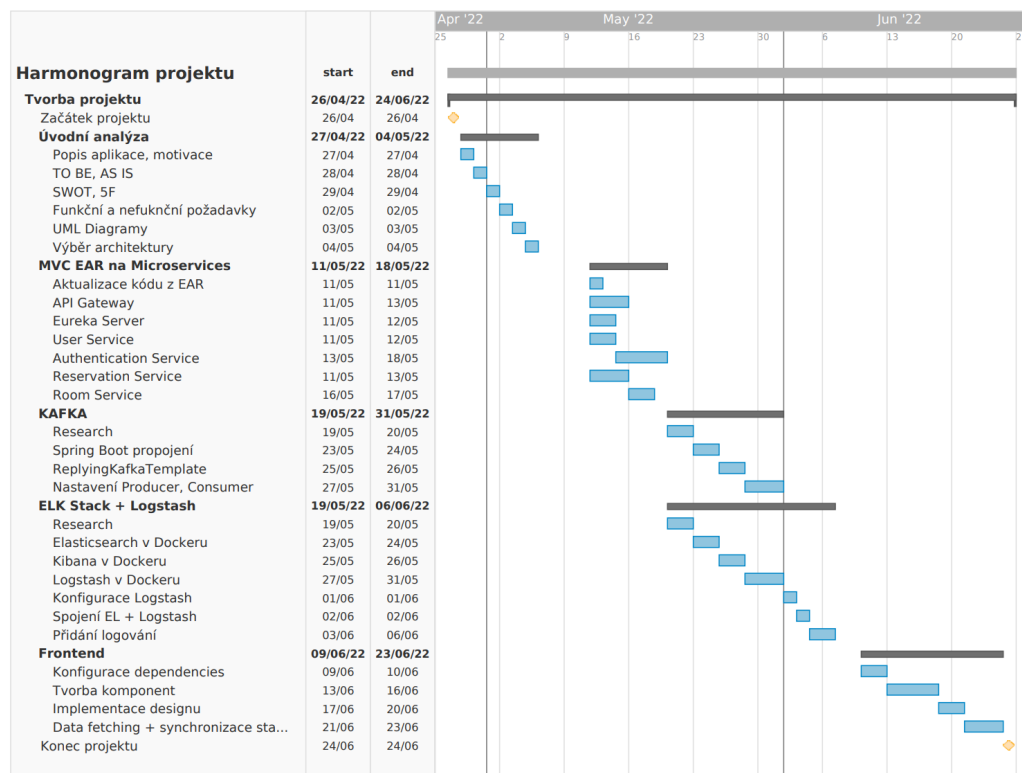
- **DAO** – Ve všech Microservices.

Alternativu návrhu řešení jsme nerozebírali, jelikož jsme se dohodnuli s cvičícím na konečném návrhu na společné konzultaci.

## Work Breakdown Structure (WBS)



## Gant



## RACI matice

	Vojtěch Luňák	Pavel Sušický	Matouš Najman	Lenka Hornová
<b>Úvodní analýza</b>				
Popis aplikace, motivace	R, C	I	R, A	I
TO BE, AS IS	R, A	R, A, C	R, A	I
SWOT, 5F	R, A	R, A, C	R, A	I
Funkční a nefunkční požadavky	R, A	R, A, C	R, A	I
UML Diagramy	R, A	C	R	I
Výběr architektury	R	R, A, C	R, A	I
<b>MVC EAR na Microservices</b>				
Aktualizace kódu z EAR	R, A	I	R, A	I
API Gateway	I	R, A	I	I
Eureka Server	I	R, A	I	I
User Service	R, A	C	R	I
Authentication Service	R, A	C	R	I
Reservation Service	R	C	R, A	I
Room Service	R	C	R	I
<b>KAFKA</b>				
Research	R	R	R	I
Spring Boot Propojení	R, A	C	I	I
ReplyingKafkaTemplate	R, A	C	I	I
Nastavení Producer, Consumer	R, A	C	I	I
<b>ELK Stack + Logstash</b>				
Research	R	R	R	I
Elasticsearch v Dockeru	I	R, A	R	I
Kibana v Dockeru		R, A	R	I
Logstash v Dockeru	I	C	R, A	I
Spojení EL + Logstash	I	C	R, A	I
Přidání logování	I	C	R, A	I
<b>Frontend</b>				
Konfigurace dependencies	I	R, A	I	I
Tvorba komponent	I	R, A	I	I
Implementace designu	I	R, A	I	I
Data fetching + synchronizace stavu	R	R, A	I	I

## Znovupoužitelnost

Backend jsme přepoužili z předmětu EAR, přepoužitelnost je tak dosažitelná. Frontend je napsaný v komponentách, které lze dále využívat.

## Plán podpory

Projekt dál nebude podporován

## Metriky

Při tvorbě projektu jsme dbali na přehlednost a srozumitelnost kódu. Dále jsme se soustředili na následující metriky:

- Funkčnost 8/10
- Znovupoužitelnost 9/10
- Přehlednost 7/10
- Testovatelnost 6/10
- Zabezpečení 6/10

## Zdroje

- Baeldung
- Youtube
- Dokumentace využitých technologií
- Přednášky z předmětů EAR, OMO

## Vyhodnocení

Projekt byl pro nás velkým přínosem. Získali jsme základní přehled o moderních technologiích využívaných hlavně na BE a některými jsme si i zkusili pracovat. Nejsložitější pro nás byl začátek, jelikož jsme doposud měli zkušenosti jen s MVC architekturou. Proto vybrat správnou architekturu a dale předělat MVC project z EAR bylo dost složité.