

# **Zřízení systému pro plánování dovolených pro společnost NSS, s.r.o.**



Semestrální práce

Letní semestr 2021

**Autoři:** Šesták Vít, Samatova Alina,  
Momot Arina, Buzek Alan

# Obsah

|   |    |
|---|----|
| Popis aplikace                            | 3  |
| Aktuální stav                             | 3  |
| Budoucí stav                              | 3  |
| Obchodní přínos                           | 4  |
| SWOT analýza                              | 4  |
| Analýza PEST                              | 5  |
| Analýza 5F                                | 6  |
| Funkční požadavky                         | 6  |
| Nefunkční požadavky                       | 8  |
| Seznam uživatelů                          | 8  |
| Případy užití                             | 9  |
| Class diagram                             | 11 |
| Sekvenční diagramy                        | 12 |
| Architektura                              | 16 |
| Diagram nasazení                          | 17 |
| Rozbor a výběr alternativ návrhu řešení   | 17 |
| Zdroje                                    | 19 |
| Normy a standardy                         | 20 |
| Matice zodpovědnosti (RACI matice)        | 20 |
| Harmonogram GANTT                         | 22 |
| WBS – rozdělení projektu na dílčí procesy | 22 |
| Analýza rizik FMEA                        | 24 |
| Znovupoužitelnost                         | 25 |
| Metriky                                   | 25 |
| Plán odbavení                             | 25 |
| Plán podpory                              | 25 |
| Vyhodnocení                               | 26 |

## Popis aplikace

Aplikace vznikla za účelem zjednodušení a zpřehlednění procesu čerpání dovolených v zejména středních a velkých firmách. Po registraci má uživatel možnost vytvořit tým či podat žádost o zařazení do některého z již existujících týmů. Tým se skládá z libovolného počtu pracovních skupin (oddělení), z nichž v jedné je uživatel evidován. Takto vedený uživatel může požádat o dovolenou v konkrétním termínu. Systém nejprve zkontroluje, zdali má žadatel k dispozici dostatečný počet dnů pro požadovanou dovolenou. Pokud ano, zjistí, zdali je v termínu dovolené v práci dostatečné množství zaměstnanců pokud odečteme žadatele. Pokud je i tato podmínka splněna, požadavek je potvrzen a dovolená je se všemi souvisejícími informacemi (uživatel, termín, povaha důvodu žádosti) zanesena do databáze.

## Aktuální stav

V momentálním stavu je situace nastavená tím způsobem, že se dovolené zaznamenávají do sdílené MS Excel tabulky, ke které mají přístup všichni zaměstnanci společnosti. Tento způsob však má svá značná omezení. Jedním z těch největších je skutečnost, že jelikož má firma řádově stovky zaměstnanců, přestávají být záznamy čitelné a vyhledávání v nich je značně komplikované a časově náročné. Další funkcionalitou, kterou z principu nemůže MS Excel tabulka plně poskytnout, je možnost určit, zdali si konkrétní zaměstnanec smí v žádaný termín vzít dovolenou či nikoli. Neberou se totiž v potaz okolnosti jako např. minimální počet zaměstnanců, který musí být daný den k dispozici, či závažnost a důvod dovolené.

## Budoucí stav

Systém je zakotven a využíván všemi zaměstnanci společnosti. Zaměstnanci jsou vedeni v jednotlivých pracovních skupinách, které korespondují s odděleními, jejichž jsou součástí (vývojářské, marketingové, HR, obchodní apod.). Toto dělení značně usnadňuje vyhledávání zaměstnanců příslušných určitému oddělení a umožňuje pracovat s jeho výsledky. Každé oddělení má pro každý pracovní den definovaný minimální počet zaměstnanců, které je třeba mít k dispozici. Dále systém obsahuje informace o celkovém počtu dnů, které konkrétní zaměstnanec dle smlouvy může vyčerpat. Pokud zaměstnanec požádá o dovolenou, zkontroluje systém platnost výše zmíněných podmínek a rozhodne, i s ohledem na povahu důvodu dovolené, o přijetí či zamítnutí žádosti.

## Obchodní přínos

Systém předchází mnoha nedorozuměním a nepříjemným zjištěním, které s problematikou čerpání dovolených mnohdy nastávají. Zvyšují se také schopnosti plánování a řízení jak na úrovni jednotlivých oddělení, tak i celé společnosti.

# SWOT analýza

| SILNÉ STRÁNKY (S)   | SLABÉ STRÁNKY (W)   |
|---|---|
| <ul style="list-style-type: none"> <li>• Malá konkurence, po podrobném průzkumu trhu, bylo odhaleno, že podobných aplikací je velmi málo</li> <li>• Existence velkého počtu organizací, které by mohly používat aplikace</li> <li>• Možnost rychlého rozvoje systému</li> <li>• Aplikace může být použita v organizacích libovolné velikosti</li> <li>• Usnadnění a zrychlení plánování dovolené</li> </ul> | <ul style="list-style-type: none"> <li>• Úzké zaměření aplikace</li> <li>• Omezené cílové publikum</li> <li>• Poprvé je obtížné pochopit použití aplikace</li> <li>• Systém umožňuje přiřazení uživatele pouze do 1 týmu a 1 pracovní skupiny</li> </ul>  |
| PŘÍLEŽITOSTI (O)  | HROZBY (T)  |
| <ul style="list-style-type: none"> <li>• Možnost rozvoje na globální úrovni</li> <li>• Rozvoj aplikace - přidání dalších funkcí</li> <li>• Vytvoření mobilní aplikace</li> <li>• Získání sponzorů a zvýšení kapitálu pro rozvoj aplikace</li> <li>• Neustálé zlepšování kvality aplikace</li> <li>• Přilákání nových uživatelů pomocí reklamy</li> </ul>  | <ul style="list-style-type: none"> <li>• Ztráta konkurenceschopnosti</li> <li>• Vznik nové konkurence</li> <li>• Vyšší náklady na vytvoření a propagaci systému než potenciální zisk</li> <li>• Nekvalitně vytvořená aplikace a problémy při použití</li> <li>• Nízká poptávka po aplikaci mezi uživateli</li> <li>• Kvůli nestabilní ekonomické situaci a pandemii firmy nemají peníze na naši aplikaci</li> </ul> |

**Tabulka č. 1:** SWOT Analýza

## Závěr SWOT analýzy:

V rámci návrhu řešení se zaměříme na minimalizaci slabých stránek a maximalizaci příležitostí, čímž podporujeme záměr spočívající ve vytvoření co nejpohodlnější a srozumitelné aplikace, v posílení příležitostí souvisejících s přilákáním uživatelů a rozvojem funkčnosti aplikace. Aplikujeme tedy strategii MIN-MAX.

# Analýza PEST

Je to tedy analýza politicko-právního, ekonomického, sociálně-kulturního a technologického prostředí a faktorů, které ovlivňují nebo budou ovlivňovat organizaci.

Každý z faktorů bude ohodnocen dle pravděpodobnosti, že tato situace může nastat na stupnici 1-5, kde 1 je nejnižší pravděpodobnost výskytu a 5 největší pravděpodobnost výskytu.

| Faktory vlivu             |   | Potenciální dopad na aplikaci   |
|---------------------------|---|---|
| <b>P</b><br>Politický     | <ul style="list-style-type: none"><li>• Změny v pracovním právu (3)</li><li>• Inovace v ochraně spotřebitele (4)</li></ul>  | <ul style="list-style-type: none"><li>• V současné době je politická situace stabilní.</li><li>• Změny v pracovním právu mohou přivést ke změně počtu dnů dovolené pro zaměstnance a jejich vydávání.</li><li>• Vzhledem k tomu, že aplikaci určena pro široké masy, může být ovlivněna změnou v ochranném právu spotřebitele.</li></ul>    |
| <b>E</b><br>Ekonomický    | <ul style="list-style-type: none"><li>• Inflace (3)</li><li>• Zvýšení nezaměstnanosti kvůli nestabilní ekonomické situaci, pandemii (5)</li><li>• Horší ekonomická situace (5)</li><li>• Snížení HDP (4)</li><li>• Zvýšení daní (4)</li></ul> | <ul style="list-style-type: none"><li>• V současné době je ekonomická situace nestabilní kvůli pandemii.</li><li>• Při uvedení aplikace na trh, některé funkce mohou mít komerční základ, takže aplikace bude záviset na inflaci, daních a DPH.</li><li>• Čím větší procento nezaměstnaných, tím méně možných uživatelů aplikace.</li></ul> |
| <b>S</b><br>Sociální      | <ul style="list-style-type: none"><li>• Změny v přístupu k práci a volnému času (4)</li><li>• Mobilita</li></ul>  | <ul style="list-style-type: none"><li>• V současné době je sociální situace stabilní.</li><li>• Změna v sociální oblasti může ovlivnit touhu lidí pracovat a odpočívat.</li></ul>   |
| <b>T</b><br>Technologický | <ul style="list-style-type: none"><li>• Celkový stav technologie</li><li>• Vznik nových technologií pro kvalitnější výrobu (3)</li><li>• Změny technologie (3)</li><li>• Rychlost zastarávání (3)</li></ul>                                   | <ul style="list-style-type: none"><li>• V současné době je technologická situace stabilní, ale vývoj nových technologií postupuje poměrně rychle.</li><li>• Změny v technologické oblasti mohou mít vliv na zavádění nových funkcí do aplikace a její změnu.</li></ul>  |

Tabulka č. 2: PEST Analýza

## Analýza 5F

|                               |   |
|-------------------------------|---|
| <b>Stávající konkurenti</b>   | Po průzkumu trhu bylo zjištěno, že k dnešnímu dni existuje velmi malý počet podobných aplikací, takže konkurence není velká.  |
| <b>Potenciální konkurenti</b> | V současné době jsou potenciálními konkurenty pouze některé nově vytvořené aplikace nebo ty, které jsou nyní ve vývoji.   |
| <b>Dodavatelé</b>             | Dodavateli jsou společnosti vyrábějící nástroje pro vytváření a podporu aplikací (např. počítače, servery, vývojové prostředí atd.). Vyjednávací síla dodavatelů je celkově nízká, jelikož je na trhu velké množství výrobců, kteří se snaží své výrobky prodat a konkurují si cenou. |
| <b>Kupující</b>               | Naší cílovou skupinou jsou firmy a organizace (přesněji zaměstnavatelé a jejich pracovníci). V současné době je na trhu velmi velký počet odběratelů. Ale vzhledem k nestabilní ekonomické situaci ve světě a rostoucí nezaměstnanosti kvůli pandemii se jejich počet může snížit.    |
| <b>Substituty</b>             | Za substituci naší aplikace lze považovat webové stránky se stejnou funkcí nebo ty aplikace, které nabízejí šablony pro snadnější plánování a lepší zobrazení (např. ve formátu excel).   |

Tabulka č. 3: 5F Analýza

## Funkční požadavky

| ID   | Název                       | Popis   |
|------|-----------------------------|---|
| FR-1 | Registrace uživatele        | Systém bude podporovat registraci nového uživatele.   |
| FR-2 | Přihlášení uživatele        | Uživatel se může přihlásit a odhlásit ze systému na základě emailu a hesla.   |
| FR-3 | Uživatelské role            | Systém bude podporovat role řadového zaměstnance, admina a systémového admina.  |
| FR-4 | Dělení uživatelů            | Uživatel budou moci být přidáváni do jednotlivých týmů a jejich podmnožin (prac. skupin).   |
| FR-5 | Zobrazení dovolených        | Systém bude umožňovat zobrazení schválených dovolených: <ul style="list-style-type: none"><li>• pro právě přihlášeného uživatele</li><li>• pro všechny podřízené admina</li></ul> Informace budou moci být zobrazeny pomocí tabulky nebo kalendáře. |
| FR-6 | Přidání žádosti o dovolenou | Uživatel bude moci přidat novou dovolenou pomocí vyplnění: <ul style="list-style-type: none"><li>• datum od - do</li><li>• důvod dovolené</li></ul> Systém schválí dovolenou pouze v případě, pokud   |

|       |  |  |
|-------|--|--|
|       |  | <p>jsou splněny podmínky:</p> <ul style="list-style-type: none"> <li>• uživatel má dostatek zbývajících dovolených v daném období</li> <li>• je splněn požadavek na minimální počet pracovníků v pracovní skupině, do které uživatel patří</li> <li>• jedná se o pracovní den (dny) pro daného pracovníka</li> </ul> |
| FR-7  | Zobrazení vlastní dovolené               | Systém zobrazí všechny žádosti o dovolené podané přihlášeným uživatelem a jejich status:   |
| FR-8  | Vytvoření týmu                           | Systém umožní uživateli vytvořit tým, uživatel se automaticky stane adminem týmu.  |
| FR-9  | Úprava týmu                              | Systém umožní adminovi upravit název týmu.   |
| FR-10 | Zobrazení týmů                           | Systém zobrazí uživateli všechny týmy, do kterých může požádat o vstup.  |
| FR-11 | Podat žádost o vstup do týmu             | Systém umožní uživateli požádat o vstup do týmu.   |
| FR-12 | Přidání žádost o vstup do týmu           | Systém umožní adminovi přijmout či zamítnout žádost o vstup do týmu nového člena.  |
| FR-13 | Odebrání člena týmu                      | Systém umožní adminovi odebrat člena z týmu  |
| FR-14 | Smazání týmu                             | Systém umožní adminovi smazat tým  |
| FR-15 | Vytvoření pracovní skupiny               | Systém umožní adminovi vytvořit pracovní skupinu.  |
| FR-16 | Přidání nového člena do pracovní skupiny | Systém umožní adminovi přidat nového člena týmu do pracovní skupiny a přiřadit mu roli.  |
| FR-17 | Odebrání člena z pracovní skupiny        | Systém umožní adminovi odebrat člena týmu z pracovní skupiny.  |
| FR-18 | Úprava skupiny                           | Systém umožní nastavit upravit název skupiny a minimální nutný počet pracovníků podle rolí za účelem kontroly žádostí o dovolenou.   |
| FR-19 | Zobrazit skupiny                         | Systém umožní adminovi zobrazit pracovní skupiny v týmu.   |
| FR-20 | Změnit roli člena týmů                   | Systém umožní adminovi změnit roli člena týmu na z běžného uživatele na admina a naopak.   |

**Tabulka č. 4:** Funkční požadavky

## Nefunkční požadavky

| ID    | Název  | Popis   |
|-------|--|---|
| NFR-1 | Intuitivní a přívětivé uživatelské webové rozhraní | Systém bude poskytovat přívětivé a intuitivní webové rozhraní splňující zásady UI/UX designu pomocí heuristických doporučení.                     |
| NFR-2 | Responzivita                                       | Webové rozhraní bude responzivní, tedy plnohodnotně zobrazitelné na telefonech, tabletech či PC.  |
| NFR-3 | Podpora prohlížečů                                 | Aplikace bude plně spustitelná v posledních verzích Google Chrome.  |
| NFR-4 | SPA  | Webové uživatelské rozhraní bude implementováno na principu Single Page Application pro zajištění plynulých přechodů mezi jednotlivými stránkami. |
| NFR-5 | Persistence dat                                    | Veškerá aplikační data budou ukládána do databáze PostgreSQL a denně zálohována.  |
| NFR-6 | Dostupnost   | Systém bude dostupný v 99,9% času (mimo plánovanou údržbu).   |

Tabulka č. 5: Nefunkční požadavky

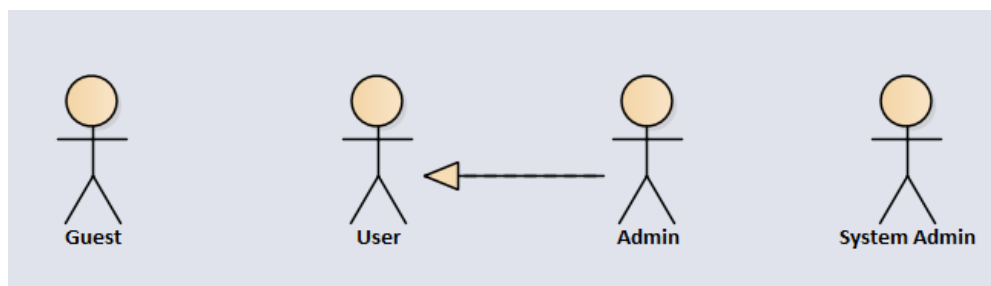
## Seznam uživatelů

Guest - anonymní uživatel.

User - běžný uživatel systému, má práva pouze v rámci svého profilu.

Admin - běžný uživatel systému, který je však správcem 1 týmu, má plná práva v rámci svého týmu.

Systém Admin - správce celého systému, má přístup k seznamu týmů a seznamu uživatelů



Obrázek č. 1: Seznam uživatelů



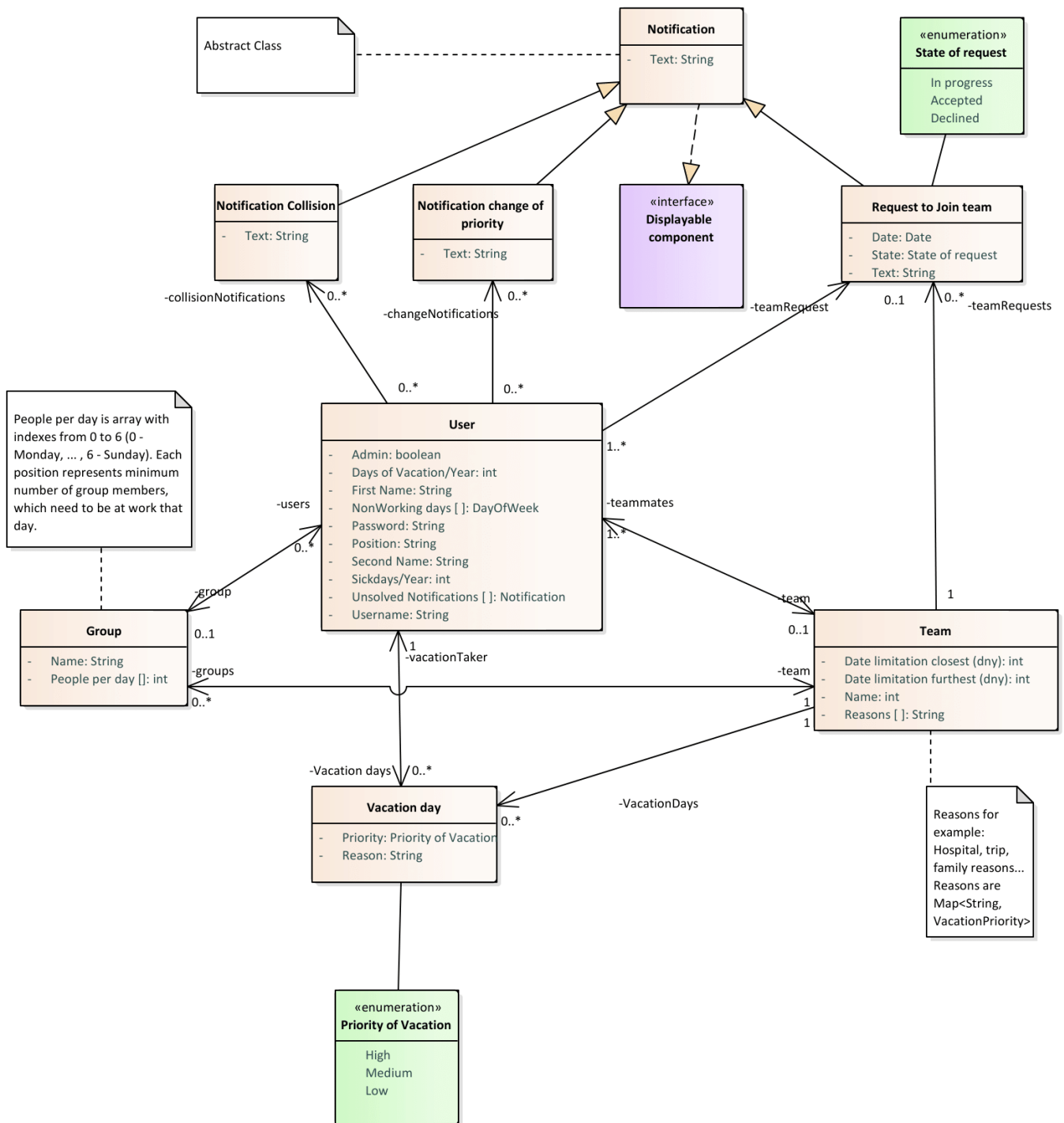
[illegible]

**Obrázek č. 2:** Diagram případů užití

- Create Vacation:
  1. Uživatel vyplní formulář "Create Vacation" (uživatel vybere datum od do z kalendáře, vybere důvod z listu důvodů, systém automaticky vyplní políčko priority)
  2. Uživatel může změnit prioritu (v tom případě systém pošle "priority notification" adminovi)
  3. Systém zvaliduje data
    - jestli datum není v minulosti
    - jestli uživatel již nemá dovolenou ten den
    - jestli uživateli zůstalo ještě dostatek dnů na dovolenou tento rok
    - jestli je to pracovní den pro uživatele
    - jestli je dostatek pracovníků v pracovní skupině
  4. IF data jsou nevalidní THEN zvýrazní chybně políčka AND GO TO STEP 1 ELSE systém přidá dovolenou uživateli
- Delete Vacation:
  1. Uživatel vybere dovolenou ze seznamu dovolených
  2. Systém kontroluje zda dovolená není v minulosti IF nevalidní THEN GO TO STEP 1

3. Systém se zeptá na potvrzení
4. Uživatel potvrdí smazání
5. Systém smaže dovolenou ze seznamu dovolených
- Show Vacation Days
  1. Systém zobrazí kalendář se všemi dovolenými uživatele
- Create team
  1. Uživatel vyplní formulář "Create Team" (uživatel vyplní jméno týmu a limitů pro dovolené)
  2. Systém zvaliduje data
    - jestli jméno již není obsaženo
    - jestli uživatel není již admin
  3. IF data jsou nevalidní THEN zdůrazní chybně políčka AND GO TO STEP 1 ELSE systém vytvoří nový tým a změní roli uživatele na admina
- Send request to join team
  1. Uživatel vybere tým ze seznamu týmů
  2. Systém se zeptá na potvrzení
  3. Uživatel potvrdí žádost
  4. Systém pošle žádost adminovi tohoto týmu
- Accept request to join team
  1. Admin vybere žádost ze seznamu žádostí
  2. Admin zmáčkne tlačítko "přijmout žádost"
  3. Systém se zeptá na potvrzení
  4. Admin potvrdí přijetí
  5. Systém přidá uživatele do týmu
- Change role of user
  1. Admin vybere uživatele ze seznamu uživatelů v týmu
  2. Admin vybere novou roli (admin, not admin) pro uživatele
  3. Admin zmáčkne tlačítko uložit
  4. Systém se zeptá na potvrzení
  5. Admin potvrdí změnu
  6. Systém změní roli uživatele
- Create group
  1. Admin vyplní formulář "Create Group" (admin vyplní jméno skupiny a vybere pro každý den v týdnu odpovídající počet pracovníků, kteří mají být přítomni v práci)
  2. Systém zvaliduje data
    - jestli jméno již není obsaženo
    - jestli počty jsou vybrané pro každý den v týdnu
  3. IF data jsou nevalidní THEN zdůrazní chybně políčka AND GO TO STEP 1 ELSE systém vytvoří novou pracovní skupinu

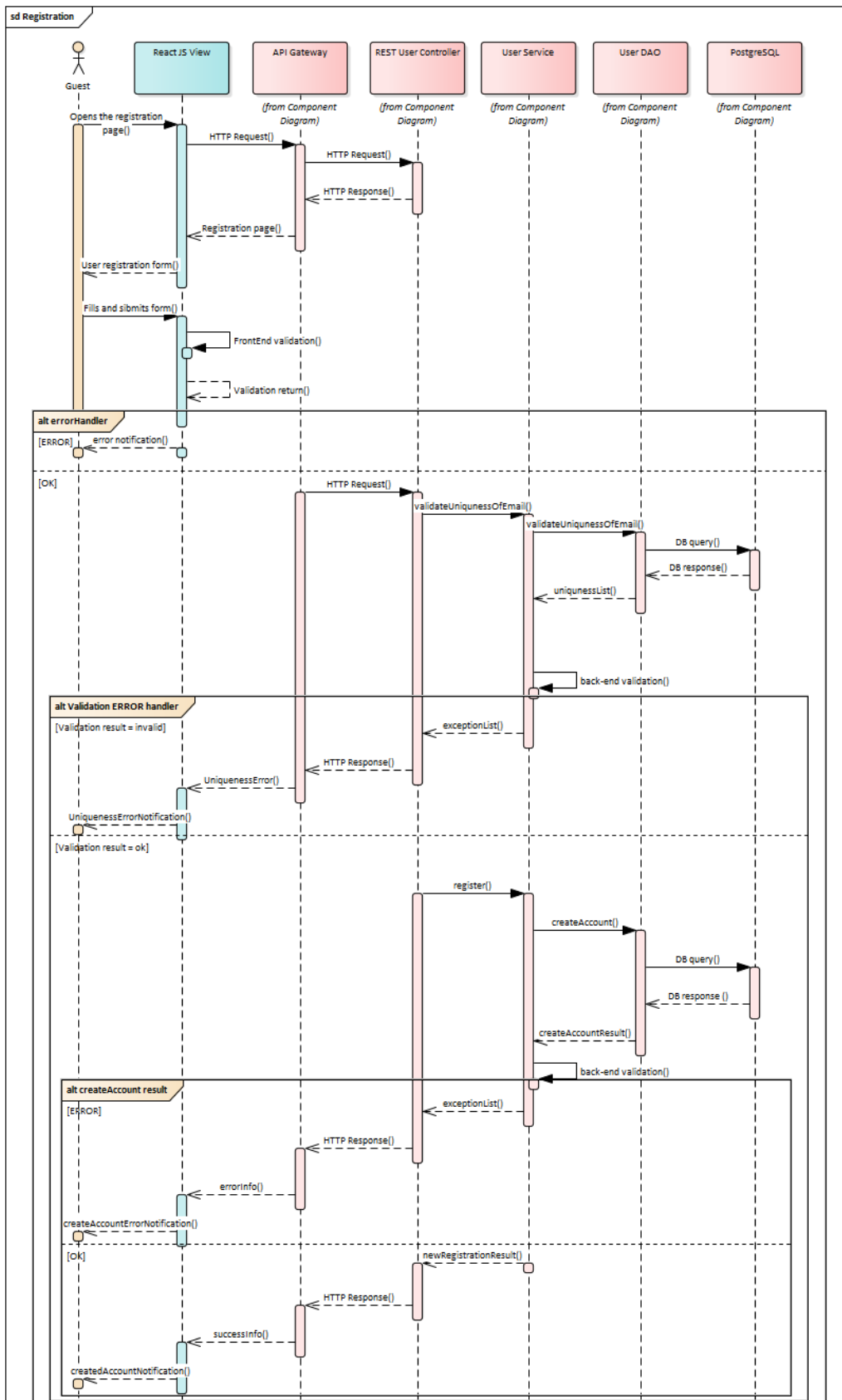
# Class diagram



Obrázek č. 3: Class diagram

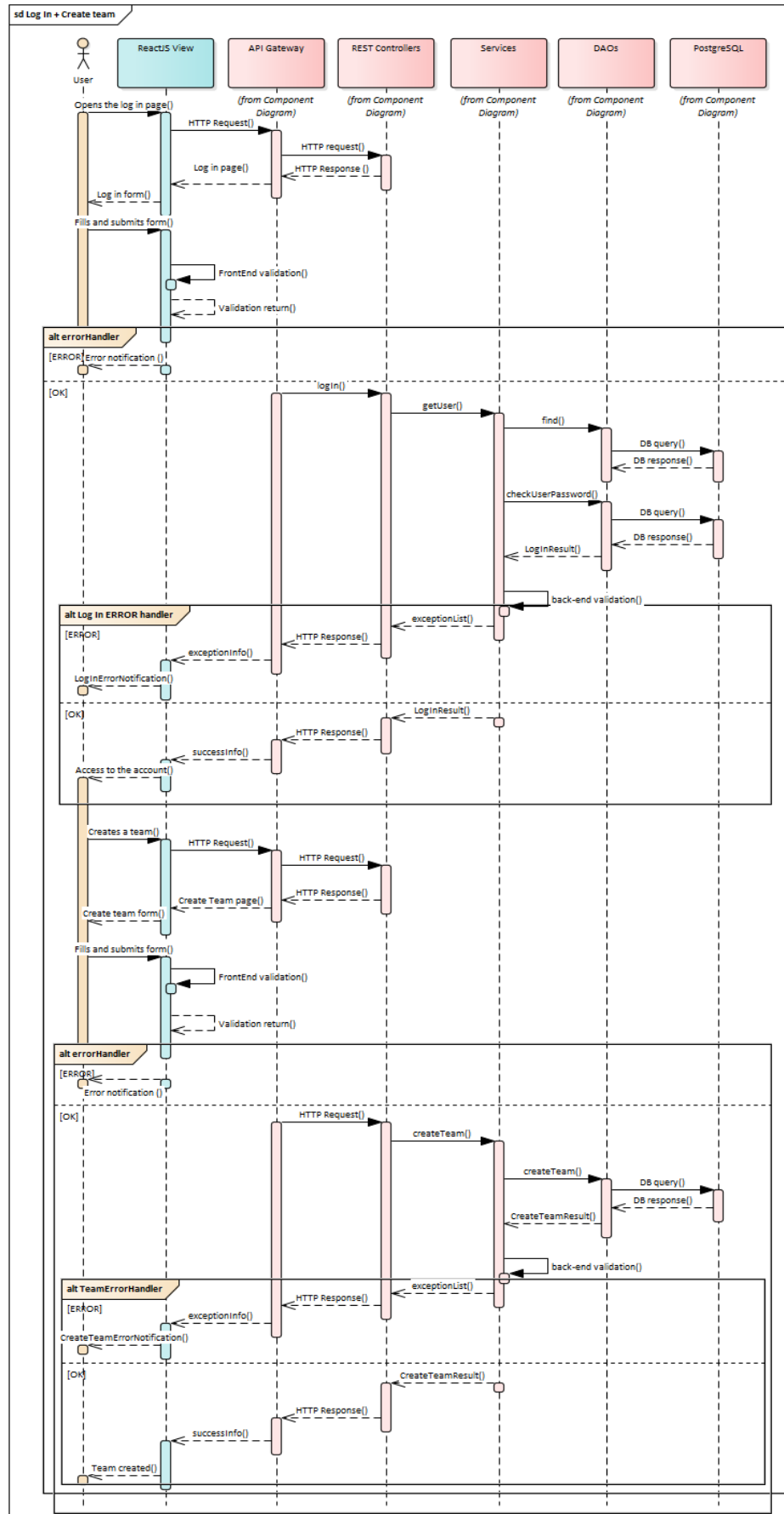
# Sekvenční diagramy

- User registration



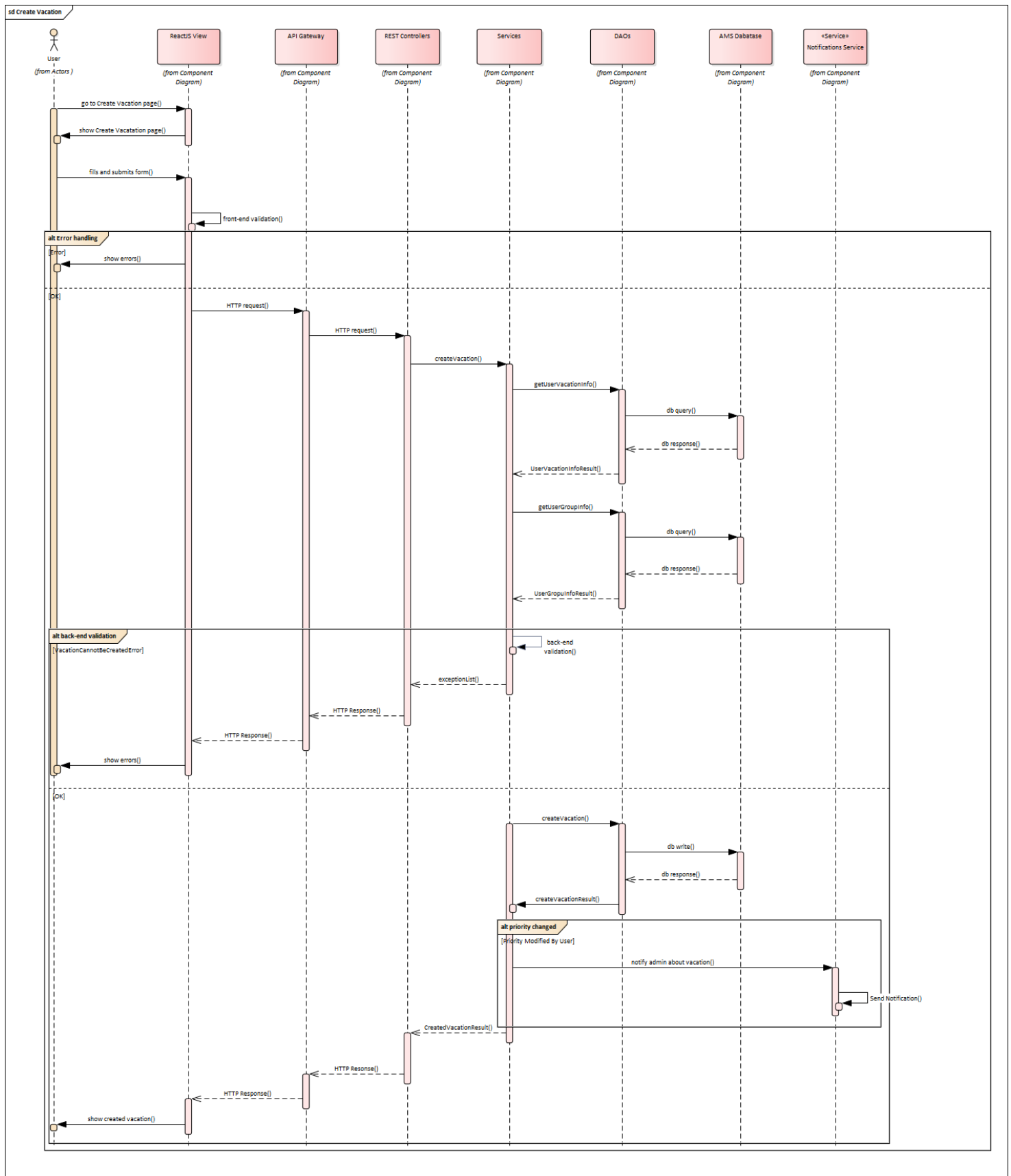
Obrázek č. 4: Sekvenční diagram pro registraci uživatele

- Log In and Create team



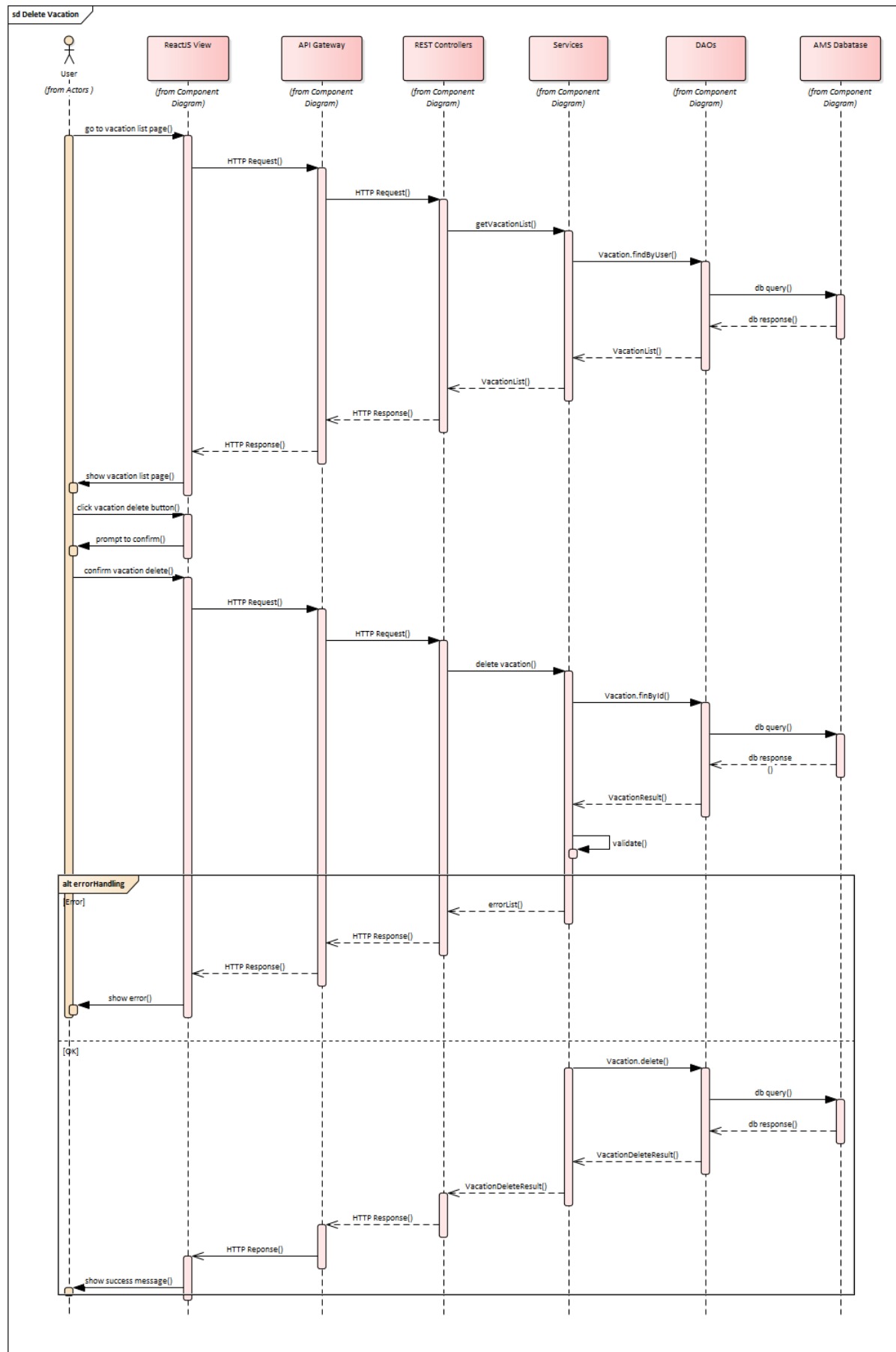
Obrázek č. 5: Sekvenční diagram pro přihlášení uživatele a vytvoření týmu

- Create Vacation



Obrázek č. 6: Sekvenční diagram pro vytvoření dovolené

- Delete Vacation



Obrázek č. 7: Sekvenční diagram pro smazání dovolené

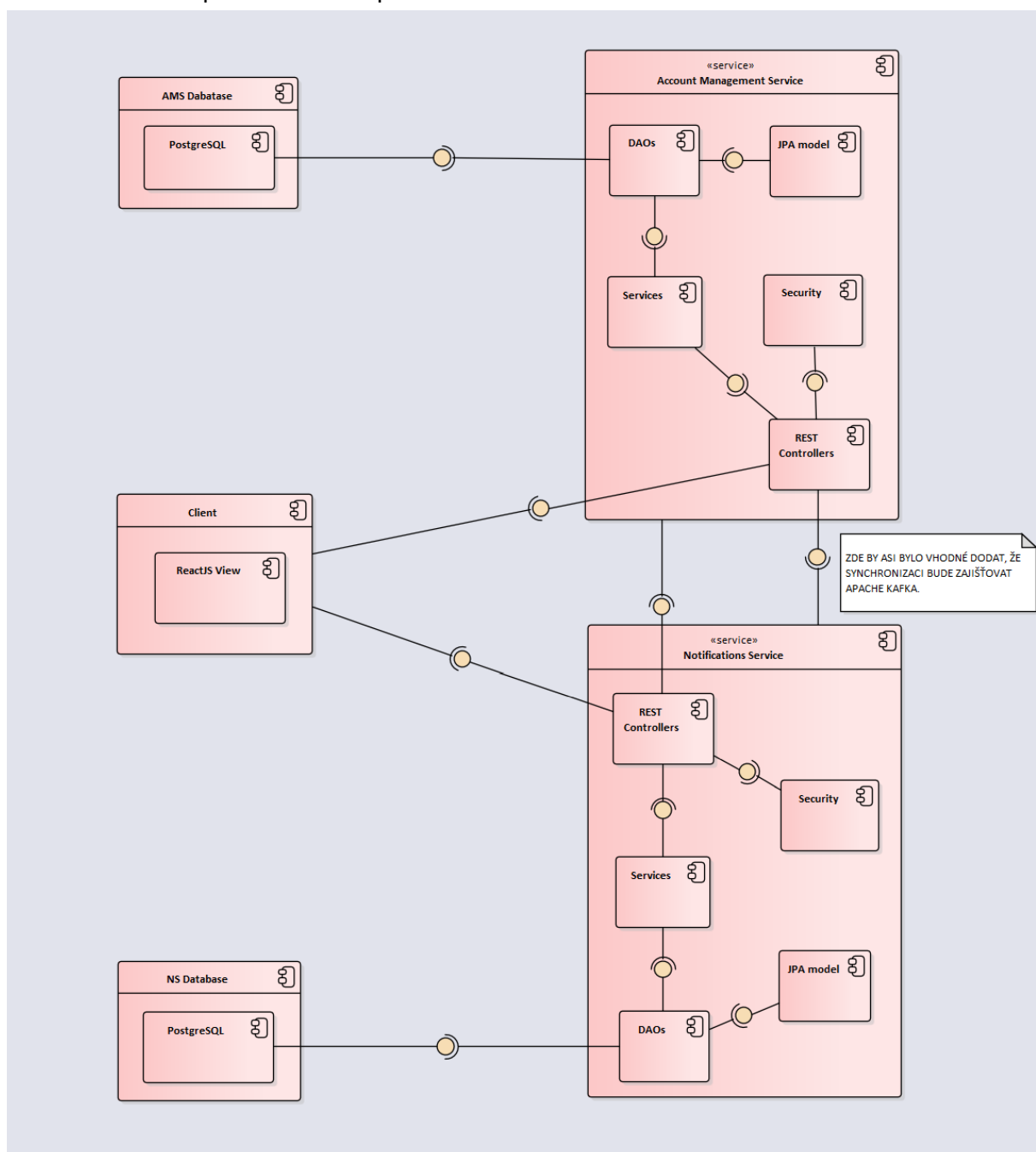
# Architektura

Aplikace je koncipována jako microservisní aplikace sestávající se z 2 servisních komponent - Account Management service a Notification service. Tyto komponenty jsou volané z ReactJS frontendové komponenty.

Backendové komponenty aplikace budou používat vrstvenou architekturu složenou z vrstev:

- databázová
- perzistentní - DAOs
- doménová - models
- aplikační/business - services - obsahují hlavní funkce aplikace
- prezentační - REST API controllers

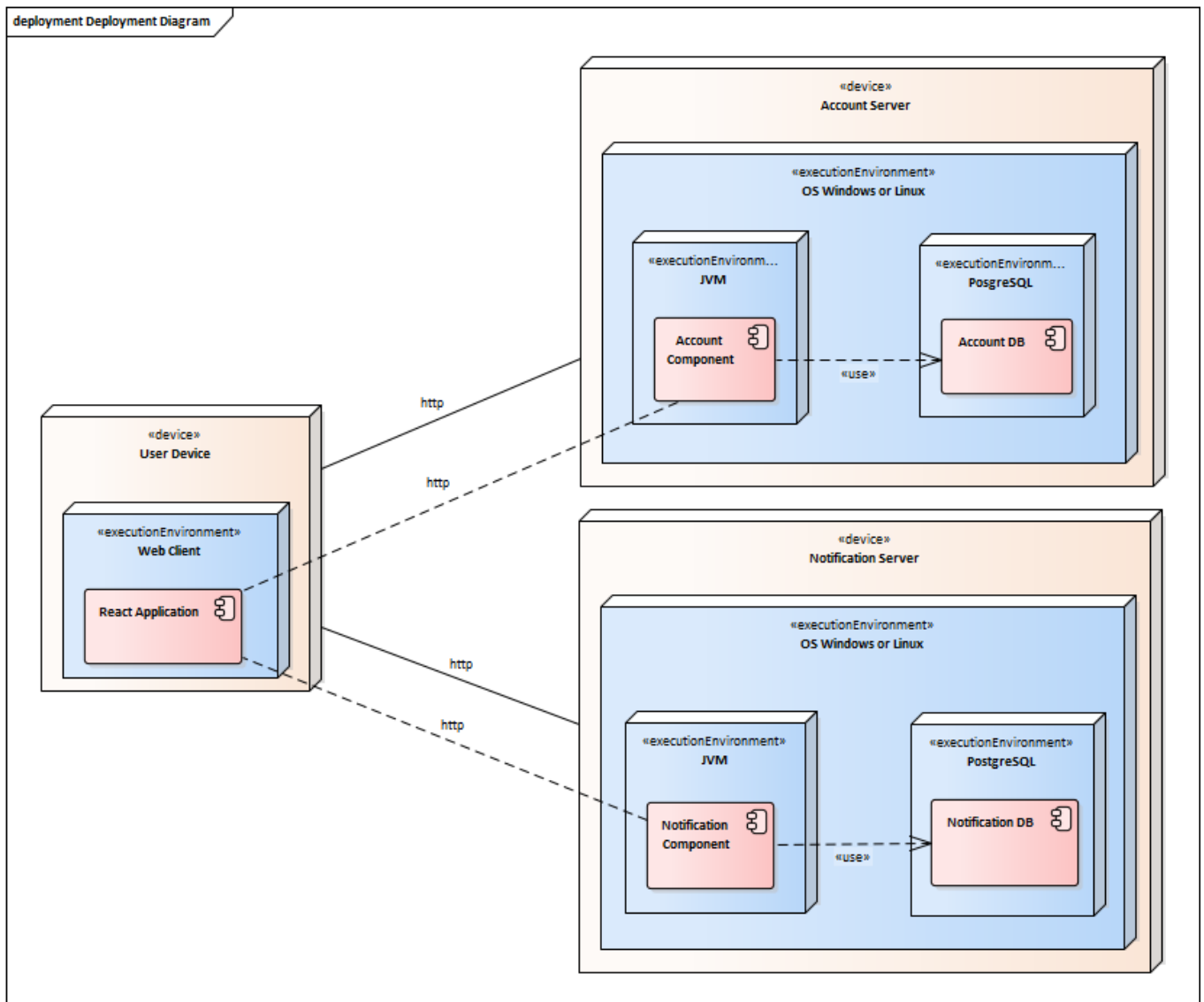
Frontendová část aplikace bude implementována JS knihovnou React.





Obrázek č. 8: Component diagram

## Diagram nasazení



Obrázek č. 9: Diagram nasazení

## Rozbor a výběr alternativ návrhu řešení

### Způsob vývoje:

#### Vlastní vývoj

Naše společnost se zabývá vývojem aplikací již mnoho let, takže máme k dispozici všechny potřebné znalosti pro vývoj této aplikace. Protože myšlenka vytvoření této aplikace patří našim zaměstnancům, vědí přesně, čeho chtějí dosáhnout a rozumějí všem funkcím, které by v aplikaci měly být. Jediná věc, která je nevýhodou v této variantě řešení je, že v současné době naši zaměstnanci jsou velmi pracovně vytížení a pracují na dalších projektech, takže doba vývoje aplikace bude delší a není

možné přesně určit dobu dokončení. Pokud však vezmeme v úvahu možnosti řešení z finanční stránky, je tato možnost nejméně nákladná.

## Objednání u jiné firmy

Tato možnost má své výhody, například to, že nebudeme muset trávit čas vyvíjením aplikace a doba vývoje aplikace by měla trvat kratší dobu, než ve variantě vlastního vývoje. Také nám bude poskytnuta IT podpora aplikace ze strany dodavatele. Ale tato možnost má rizika, že výsledek bude neuspokojivý a nedostaneme to, co jsme chtěli. Bude nutné podrobněji definovat požadavky a podmínky pro výkon práce. Také náklady na tuto variantu budou výrazně vyšší než v první variantě.

## Delegování na jinou firmu

Můžete také vzít v úvahu možnost delegování provedení části zpracování aplikace na jinou firmu (například vývoj nebo testování). Tato možnost může pomoci naší firmě usnadnit a urychlit proces vývoje aplikace. Ale také s sebou nese rizika, protože tým jiné firmy bude muset jasně a správně pochopit koncept a funkcionalitu naší aplikace. Tato možnost nebude tak drahá jako objednání u jiné firmy, ale bude stále nákladnější než vlastní vývoj.

## Alternativní způsoby vytvoření aplikace:

### Architektura

Místo mikroservicesní architektury bychom mohli použít monolitickou architekturu. V této architektuře celá aplikace funguje jako jedna velká komponenta. Monolitické architektury jsou často vhodným řešením pro malé aplikace, ale při zvětšování aplikace se můžou stát těžkopádnými, a proto nám to nevyhovuje. Má také tyto nevýhody: obtížné škálování služeb nezávisle na sobě, složitost vývoje a správy nasazení při nárůstu základu kódu, což zpomaluje vydávání verzí a implementaci nových funkcí, aktualizace schémat dat můžou být stále obtížnější. Tyto problémy se dají řešit pomocí alternativních architektur, jako je například mikroservicesní architektura.

### Backend

Pro vývoj backendu bychom mohli použít knihovnu Node.js, která se používá pro generování JS aplikací na straně serveru. Node.js se používá k vytváření aplikací v JavaScriptu, navíc je to bezplatné, open-source a využívá různé platformy. Node.js nabízí rychlejší psaní skriptů a škálovatelnost. Nevýhodou Node.js je jednovláknovost, což znamená, že s jedním vláknem lze vyřídit jeden požadavek. Spring-boot Java je naproti tomu vícevláknová, což znamená, že lze současně provádět několik úkolů a multi-threading pomáhá aplikacím fungovat lépe. Node.js je v současné době stejně jako Spring-boot velmi oblíbená technologie a používá se na velkých webových aplikacích.

### Frontend

Pro vývoj frontendu bychom mohli použít framework AngularJS či JS.Vue pro rendering obsahu na DOM používá šablony HTML a JSX, zatímco React používá pouze JSX. Další variantou pro frontend by byla technologie Thymeleaf,

kteřá je jakýmsi moderním ztělesněním starších JSP a kteřá umí velmi dobře spolupracovat se Spring-boot backendem.

## Zdroje

### Lidské zdroje:

Náš tým se skládá celkem ze 4 členů, takže jeden člověk může zastávat více funkcí.

- **Projektový manažer (1)** - Šesták Vít

Klíčový člen vývojového týmu, který je zodpovědný za vedení, plánování a organizaci projektu, koordinaci spolupráce, řízení a kontrolu realizace projektu tak, aby bylo dosaženo projektových cílů ve stanoveném termínu a kvalitě.

- **Analytik (4)** - Šesták Vít, Samatova Alina, Momot Arina, Buzek Alan

Představuje v projektovém týmu roli, která je zodpovědná za sběr požadavků od všech zainteresovaných stran, jejich porozumění, zachycení a určení jejich priorit. Úkolem analytika je analyzovat projekt a nabízet kvalitní návrhy řešení. Představuje „spojovací článek“ mezi vývojovým týmem a jeho budoucími uživateli.

- **Vývojář (4)** - Buzek Alan, Šesták Vít, Samatova Alina, Momot Arina

Zodpovědný za vývoj jednotlivých komponent systému a jejich sjednocení do integrovaného řešení, včetně zajištění jeho souladu s navrženou architekturou. Může se podílet i na návrhu uživatelského rozhraní a na jednotkovém testování (unit-testing).

- **Tester (4)** - Buzek Alan, Šesták Vít, Momot Arina, Samatova Alina

Zodpovědný za provedení testování a identifikaci nedostatků a složitostí používání aplikaci, aby došlo k jeho zlepšení a přepracování do pohodlnějšího, srozumitelnějšího a snadno použitelného systému.

### Nástroje potřebné pro vývoj:

Manažerské:

- **Google Docs** - Pro vytváření, ukládání a úpravu dokumentů a snadnou spolupráci na projektové dokumentaci
- **Microsoft Teams** - Pro videohovory a spolupracování na projektu
- **Facebook Messenger** - Pro interní komunikaci v týmu
- **Toggle** - sledování stráveného času

Vývojářské:

- **Enterprise Architect** - Modelovací nástroj použili jsme k vytvoření use cases, class diagramu, sekvenčních diagramů, component diagramu a diagramu nasazení
- **IntelliJ IDEA** - Vývojové prostředí pro jazyk JAVA
- **PostgreSQL** - Používáme standardní relační databázi
- **REST API** - Tato architektura rozhraní se nám osvědčila již dříve, umožňuje přehledně mapovat http metody, přiřazovat k nim obslužné rutiny a snadno přistupovat ke zdrojům
- **Postman** - Intuitivní tool, který umožňuje zasílat http požadavky a s jejich pomocí pak testovat REST API i na localhostu, aniž by musel běžet funkční frontend
- **Vývojářské nástroje v prohlížeči** - vývoj a ladění UI
- **Heroku** - nasazení aplikace na produkční server, sledování chování aplikace v produkčním prostředí
- **Kibana** - sledování logů

## Normy a standardy

V tomto projektu jsou zahrnuty následující normy a standardy:

- Java, Maven, SQL, JQuery, CSS, React, Hibernate - standard W3C
- ISO/IEC 12207 (standard, který se zabývá životním cyklem softwaru)

## Matice zodpovědnosti (RACI matice)

- R (responsible) – osoba/osoby, které na úkolu pracují
- A (accountable) – osoba odpovědná za úkol jako celek
- C (consulted) – osoba/osoby, které mohou podpořit úkol konzultací
- I – (informed) osoba/osoby informované o výsledku, nebo i postupu plnění

Při použití vyjádření kompetencí pomocí RACI platí, že pro každý úkol má existovat právě jedna osoba, která je k úkolu ve vztahu A (accountable). <sup>[3]</sup> Ostatní vztahy mohou být 0 až n-krát.

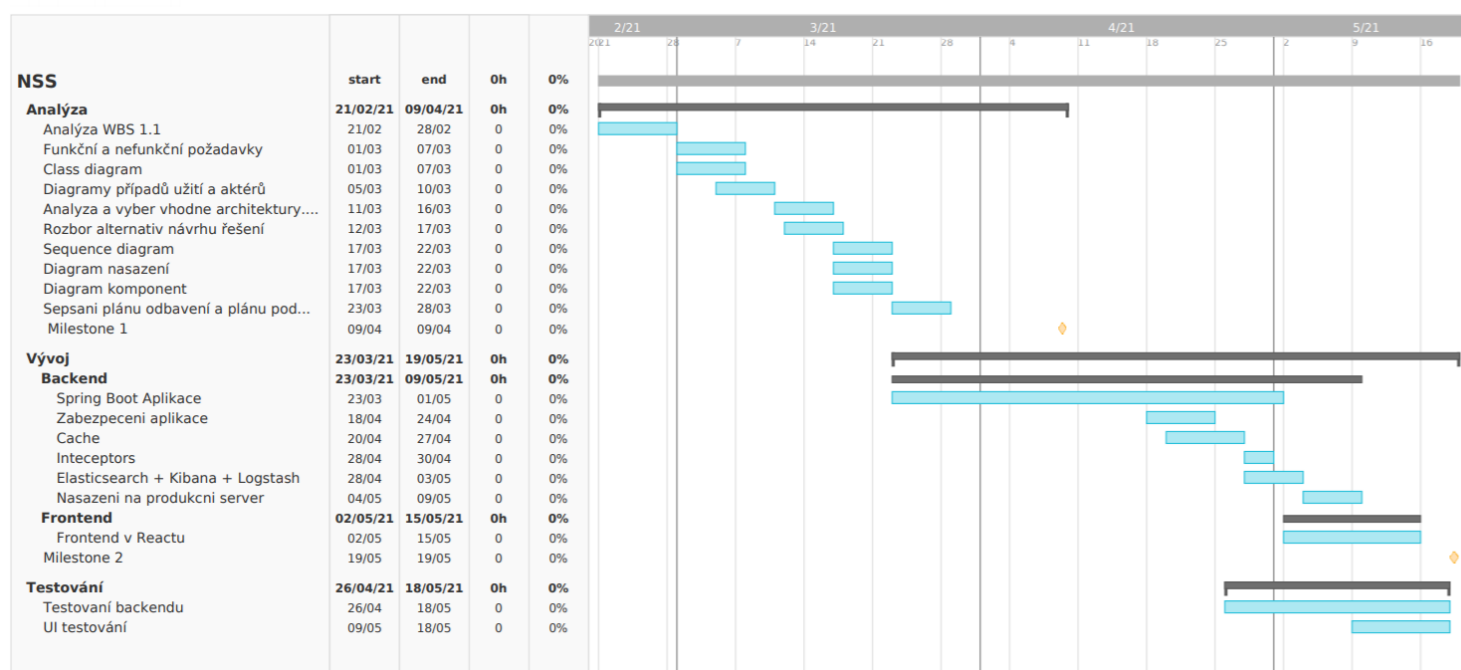
|                           | Vít Šesták | Alan Buzek | Arina Momot | Alina Samatova |
|---------------------------|------------|------------|-------------|----------------|
| <b>Analýza</b>            |            |            |             |                |
| Popis aplikace            | A          | I          | I           | I              |
| Analýza                   | I          | I          | A, R        | I              |
| Funkční požadavky         | I          | A, R       | I           | I, C           |
| Nefunkční požadavky       | I          | A, R       | I           | I              |
| Případy užití (Use cases) | I          | I          | I           | A, R           |

|  |         |      |      |      |
|--|---------|------|------|------|
| Class diagram  | R       | I    | I    | A, R |
| Sekvenční diagramy   | I       | A, R | A, R | I    |
| Návrh architektury   | R, C    | A, R | R, C | R, C |
| Component diagram  | A, R    | I, C | I    | I    |
| Diagram nasazení   | A, R    | I    | I    | I    |
| Rozbor a výběr alternativ návrhu řešení                        | I       | I    | A, R | I    |
| WBS  | I       | I    | I    | A, R |
| Matice zodpovědnosti   | I       | I    | A, R | I    |
| Zdroje   | I       | I    | A, R | I    |
| Harmonogram GANTT  | I       | I    | I    | A, R |
| Analýza rizik FMEA   | I       | I    | A, R | I    |
| Metriky  | I       | I    | A, R | I    |
| Plán odbavení a podpory  | I       | I    | A, R | I    |
| Pokročilá analýza  | I       | I    | A, R | I    |
| <b>Vývoj - Backend (Spring)</b>                                |         |      |      |      |
| Rozdělení na mikroservisy                                      | A, R, C | R, C | R    | R    |
| Implementace Cache   | A, R    | R, C | I    | I    |
| Implementace logování (Elasticsearch, Log4J, Kibana, Logstash) | R, C    | R, C | I    | A, R |
| Implementace dalších částí aplikace                            | R       | A, R | R    | R    |
| Nasazení na server Heroku                                      | I       | C    | A, R | I    |
| <b>Vývoj - Frontend (REST)</b>                                 | I, C    | A, R | I    | I    |
| <b>Testování</b>   |         |      |      |      |
| Testovací scénáře  | R       | R    | R    | A, R |
| Testování backendu   | A, R    | R    | R    | R    |

|                                |   |      |      |   |
|--------------------------------|---|------|------|---|
| UI testování                   | R | R    | A, R | R |
| Testování zabezpečení aplikace | R | A, R | R    | R |
| Dokončení aplikace             | R | A, R | R    | R |

Tabulka č. 6: RACI matice zodpovědnosti

## Harmonogram GANTT



Obrázek č. 10: Harmonogram GANTT

## WBS – rozdělení projektu na dílčí procesy

### 1. Analýza

- 1.1. Sběr požadavků (3h)
- 1.2. Analýza (strategický záměr, obchodní přínos, zdroje, normy a standardy, matice zodpovědnosti, WBS, harmonogram GANTT, analýza rizik FMEA) (8h)
- 1.3. Funkční a nefunkční požadavky (2.5 h)
- 1.4. Diagramy případů užití a aktérů (Use case diagrams) (4 h)
- 1.5. Class diagram (3 h)
- 1.6. Analýza a vyber vhodne architektury. Návrh řešení (2h)

- 1.7. Rozbor alternativ návrhu řešení (1.5h)
- 1.8. Sequence diagram (4h)
- 1.9. Diagram nasazení (3h)
- 1.10. Diagram komponent (2h)
- 1.11. Sepsání plánu odbavení a plánu podpory (1h)

Celkem analýza: 34 h

---

## 2. Vývoj

- 2.1. Backend
  - 2.1.1. Spring Boot Aplikace (150h)
  - 2.1.2. Cache (6h)
  - 2.1.3. Zabezpečení aplikace (5h)
  - 2.1.4. Interceptors, Facade (3h)
  - 2.1.5. Nasazení na produkční server (10h)
  - 2.1.6. Elasticsearch + Kibana + Logstash (6h)
- 2.2. Frontend
  - 2.2.1. Frontend v Reactu (30h)

Celkem vývoj: 210 h

---

## 3. Testování

- 3.1. Testování backendu (20 h)
- 3.2. UI testování (5 h)

Celkem testování: 25 h

## Ostatní náklady

- Týmové schůze: 11 (týdnů) \* 1 h = 11 h
- 

**Celkové odhadované náklady na vytvoření softwaru: 280 h.**

# Analýza rizik FMEA

| Prvek        | Možná chyba                             | Možný následok chyby  | V ý z n a m | Možná příčina vzniku   | V ý s k y t | O d h a l i t e l n o s t | R P N | Doporučené opatření   | Odpovědná osoba    |
|--------------|---|---|-------------|--|-------------|---------------------------|-------|---|--------------------|
| Analýza      | Špatně provedená analýza                | Ztráta času na přepracování analýzy, porušení časového plánu, vznik problémů v dalších částech projektu(impleme ntace, testování) | 10          | Nekompetentnost analytiků, nepochopení klíčové myšlenky aplikace                           | 6           | 4                         | 240   | Kompetentní a pozorný přístup k analýze, zohlednění všech faktorů ovlivňujících projekt | Projektový manažer |
| Vývoj        | Nesprávně zvolená architektura aplikace | Porušení časového plánu a přepracování aplikace   | 9           | Nekompetentnost analytiků a vývojářů, nepřemýšlejte o všech procesech aplikace             | 6           | 4                         | 216   | Podrobné studium všech architektur pro vytváření aplikací, správně provedená analýza    | Vývojář            |
| Vývoj        | Chyba v používaných technologiích       | Porušení časového plánu a přepracování částí aplikace   | 7           | Nekompetentnost vývojářů, nepochopení funkčnosti aplikace                                  | 7           | 4                         | 196   | Kompetentní a pozorný přístup k vývoji, podrobné studium potřebných technologií         | Vývojář            |
| Testování    | Aplikace neprojde akceptačními testy    | Porušení časového plánu a přepracování aplikace   | 6           | Nekompetentnost vývojářů a testerů, nesprávně napsané testy nebo chyby při vývoji aplikace | 6           | 3                         | 108   | Kompetentní a pozorný přístup k vývoji a testování                                      | Tester             |
| Dokumen tace | Neúplná / nesprávná dokumentace         | Porušení časového plánu a přepracování dokumentace  | 3           | Nekompetentnost/ nepozornost pracovníků  | 2           | 1                         | 6     | Pečlivé vytváření dokumentace   | Projektový manažer |
| Celý projekt | Odchod člena týmu z projektu / firmy    | Zvýšení času na vytvoření aplikace a porušení časového plánu  | 6           | Choroba / Nespokojenost pracovníků s pracovními podmínkami/ lepší pracovní příležitosti    | 4           | 3                         | 72    | Dodržovat dobré pracovní podmínky   | Projektový manažer |

**Tabulka č. 7:** Analýza rizik FMEA



# Znovupoužitelnost

Při vývoji aplikace bylo cílem dosáhnout co možná nejvyšší kvality aplikace ve smyslu použitelnosti, spolehlivosti a znovupoužitelnosti kódu.

Výše zmíněných vlastností bylo v aplikaci dosaženo použitím mikroservisní architektury, frameworků a knihoven:

- Backend: Microservice-based Java web framework Spring-Boot (umožňuje opětovné použití komponent - services (i těch, které jsou již nasazeny) lze sdílet v různých projektech), Eclipse a design patterns.

Z backendu taky je možné znovu používat část pečující o zabezpečení aplikace, která je přizpůsobena na 4 uživatelské role a přihlašování prostřednictvím emailu a hesla.

- Frontend: knihovny React JS(použití komponent), Axios, Tailwind CSS

Využívání těchto technologií umožňuje opětovné použití kódu a usnadňuje testování a údržbu projektu.

## Metriky

V projektu jsou sledovány tyto metriky:

- Počet případů užití: **30**
- Počet unit testů: **30**
- Celkový čas strávený na projektu (1MD = 5h): **35 MD**
- Počet lidí potřebných k projektu: **4 lidé**
- Množství chyb zjištěných po dokončení aplikace: **6**

## Plán odbavení

- Nasazení na produkční server (Heroku).
- Školení uživatelů, při kterém jim bude detailně popsáno, jak aplikaci používat.

## Plán podpory

- Oprava problémů, které byly zjištěny při pilotním provozu.
- Naše firma je také zodpovědná za školení pracovníků a poskytnutí pomoci při používání systému.
- Vytváření a vypracování nových funkcí potřebných pro provoz systému.

Naše firma zákazníkovi zaručí kompletní podporu tak, aby se zákazník o ICT problémy vůbec nemusel starat. Zákazník se bude moci obrátit s jakýmkoliv problémem souvisejícím s vytvořeným systémem.

# Vyhodnocení

Cílem naší semestrální práce bylo vytvořit aplikace, která usnadní a urychlí uživatelům plánování dovolené. Tento projekt byl pro nás velmi užitečný a zajímavý. V průběhu prací se nám podařilo poznat a osvojit si nové technologie a zlepšit své týmové dovednosti. Také jsme se naučili, jak kvalitativně analyzovat projekt a dělat jasnou dokumentaci.

Semestrální projekt nám přinesl opravdu hodně zkušeností. Naučili jsme se propojovat frontend a backend, správně používat návrhové vzory, vyzkoušeli jsme nasazení projektu na platformě Heroku. Všichni v týmu pracovali aktivně a zodpovědně a dělali vše, co měli udělat ve správný čas. Myslíme si, že náš projekt je dobře promyšlený a dobře zpracovaný.