

# Semestrální práce z předmětu NSS

## Pocket Lingo

Projekt [repository](#)

Zpracovali - Anna Kachmasheva, Aiya Rakhimova, Mukan Atazhanov

# Obsah

	<b>2</b>
1. Popis aplikace, motivace	4
2. Strategický záměr projektu	4
3. Obchodní přínos	4
4. Analýza SWOT	5
5. Analýza pěti sil (5F)	5
6. PEST analýza	6
7. Funkční požadavky	7
8. Nefunkční požadavky	8
9. Seznam uživatelů	9
10. UML diagramy	11
10.1 Class diagram	11
10.2 Sekvenční diagram	12
11. Případy užití	14
12. Výběr vhodné architektury	20

- Popis aplikace, motivace

Projekt pro zapamatování cizích slovíček na základě flashkaret metodou pereodického opakování slov. Flashcard je abstraktní oboustranná karta, slovo v rodném jazyce uživatele je napsáno na skryté straně a slovo v zvoleném jazyce je napsáno na druhé straně. Uživatel vytvoří flashkarty sám nebo stahuje karty jiných uživatelů z repository. Poté uživatel zadá režimu učení a prohlíží si karty. Pokud uživatel zná překlad daného slova, může si jej klikne na tlačítko "Vím" nebo v opačném případě na "Nevím". Periodické algoritmus opakování učení zobrazuje karty se slovíčky, která uživatel neví častěji než ta, která ví.

- Strategický záměr projektu

Strategickým záměrem je ulehčit učení nových jazyků metodou pereodického opakování. Podle světovou statistiky v Evropě skoro 90% studentů se učí cizí, nový jazyky. Na základě toho naše free-to-use, user-friendly aplikace by měla mít úspěch nejen mezi studentů, ale i mezi starší generace. Vytvořit aplikace vzhledem do budoucna, což znamená aplikace musí být rozšiřovatelná a škálovatelná.

- Obchodní přínos

Planujeme vytvořit free-to-use web aplikace, ale vzhledem na to se dá pochopit, že nemám žádný peníze na placení serveru a podobných věc. Na to máme řešení - reklama. Diki reklamě budeme mít možnost existovat na trhu. Určitě bude možnost na vypnutí reklamy, což je Premium verze webu.

## • Analýza SWOT

Strength	Weaknesses
<ul style="list-style-type: none"> <li>Jednoduchá aplikace</li> <li>Mladý, motivovaný tým s hořícíma očima.</li> </ul>	<ul style="list-style-type: none"> <li>Tým skládající ze studentů</li> <li>Šíleně vysoká konkurence</li> </ul>
Opportunities	Treats
<ul style="list-style-type: none"> <li>Spolupráce se školami.</li> <li>Rozšíření popularity v ostatním světě.</li> </ul>	<ul style="list-style-type: none"> <li>Nedůvěra ze strany uživatele</li> <li>Plánujeme velkou funkcionalitu, co může přinést problémy s deadline.</li> <li>Malý zisk z reklam</li> </ul>

## • Analýza pěti sil (5F)

Parameter	Význam	Popis	Řešení
Hrozby vzniku substitutů	Střední	Na trhu existuje několik analogů projektu	Podporovat a zlepšovat funkčnost projektu
Stávající konkurenti	Vysoký	Trh v této oblasti je vysoce konkurenční a perspektivní	Provádět neustálé sledování nabídek konkurentů. Rozvíjet jedinečnost projektu. Zvýšit úroveň znalostí o projektu.
Hrozby vstupu nových konkurentů na trh	Vysoký	Riziko vzniku takových projektů je poměrně vysoké.	

Parameter	Význam	Popis	Řešení
Hrozby vzniku substitutů	Střední	Na trhu existuje několik analogů projektu	Podporovat a zlepšovat funkčnost projektu
Vyjednávací síla zákazníků	Střední	Existují aplikace, které na trhu existují již dlouho, a tak mají dobré hodnocení. Nedostatek současné funkčnosti projektu	Vytvořit nové unikátní funkce aplikace. Zlepšit funkčnost podle zaostávajících ukazatelů.
Vyjednávací síla dodavatelů	Nízký	Stabilní práce ze strany vývojářů. Zachování funkčnosti projektu	

## • PEST analýza

Politické faktory	Politická strana nemá na náš projekt žádný vliv
Ekonomické faktory	K dispozici budou placené a bezplatné verze.
Sociálně-kulturní faktory	Vzdělání je vždy relevantní a existuje mnoho lidí, kteří mají touhu učit se cizí jazyky  Pro většinu je tento formát učení mnohem pohodlnější a levnější než výuka v jazykových kurzech
Technologické faktory	Vývoj umělé inteligence Nástup nových technologií

## • Funkční požadavky

Funkční požadavky se týkají funkce, které aplikace bude poskytovat.

- Systém neumožní pracovat s webovou aplikací nepřihlášenému uživateli.
- Pro autentizaci uživatele program bude používat kontrolu vstupních řetězců uživatelského loginu a hesla.
- Systém neumožní uložit data formuláře do DB, aniž by nebyla vyplněná všechna povinná pole formuláře.
- Pokud bude na formuláři nějaká hodnota povinná (bez ohledu na to, zda se jedná o zadávací pole, rozbalovací seznam či jiný prvek), bude tato skutečnost uživateli jasně prezentována.
- Systém bude vypisovat chybové hlášky.
- V případě, že uživatel provede akci, o výsledku které má smysl informovat uživatele, systém tuto akci provede a po úspěšném dokončení, zobrazí uživateli informační hlášení.
- Všechny záznamy v DB musí mít přidělené jedinečné identifikační číslo (ID). Toto číslo automaticky přiděluje systém.
- Systém provádí logování :
  - Každé změny záznamu v DB.
  - Chyby včetně svých podrobných informací.
  - Všechny přístupy uživatelů k citlivým datům včetně operace čtení.
- Přístupová práva budou každému uživateli přidělovány na základě jeho rolí v systému. Admin systému bude moci změnit roli.
- Evidence a správa uživatelů Adminem
- Po úspěšné registraci uživatele systém automaticky vytvoří repozitář a statistiky uživatele. Repozitář bude mít flash karty a balíčky karet.
- Aplikace umožní přihlášenému uživateli:
  - změnit své osobní údaje
  - vytvářet a spravovat flash karty
  - přidávat popis, obrázky, překlad do karet
  - vytvářet a spravovat balíčky karet
  - přidávat otevřené balíčky karet jiných uživatelů do svého repozitáře
  - studovat karty a zároveň měnit svůj osobní pokrok v učení
  - sledovat svůj pokrok v učení flash karet a v případě potřeby jej smazat
  - získávat úspěchy (například že se naučil 10 slov)
  - přidávat tagy do karet
  - hledat karty podle tagů. Uživatel má možnost vyhledávat karty pouze mezi svými vlastními kartami, nebo pokud byla karta vytvořena jiným uživatelem, pak tato karta musí být v otevřeném balíčku.

- Filtrovat a seřadit seznam karet
- Aplikace musí podporovat registraci Google.

## • Nefunkční požadavky

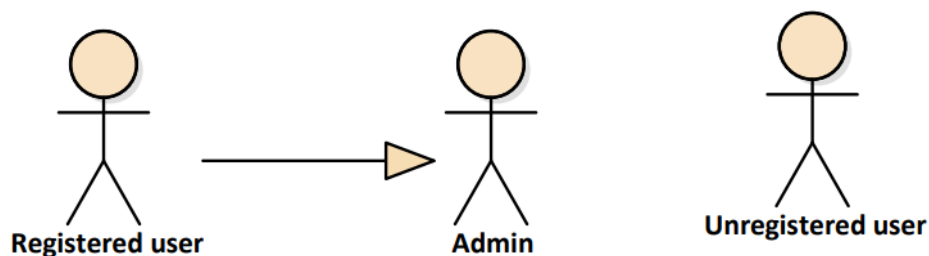
Nefunkční požadavky se týkají především návrhu a implementace webové aplikace a přímo se nedotýkají funkce na rozdíl od funkčních požadavků.

- Klient webové aplikace tvoří libovolný webový prohlížeč, zajištěna funkčnost alespoň na prohlížečích Google Chrome a Mozilla Firefox.
- Webová aplikace musí být psána v Javě a s použitím frameworku Spring Boot.
- Aplikace bude napojena na relační DB PostgreSQL a bude s ní možné komunikovat pomocí HTTP dotazů nebo přes webový prohlížeč.
- Frontend bude implementován pomocí ReactJS.
- “User-friendly” GUI.
- Možnost rozšíření aplikace v budoucnu
- Aplikace by měla chránit svá data před nahráním nesprávných dat a to jak na straně aplikace tak na straně DB
- Byznys logika aplikace bude umístěna do odděleného modulu s definovaným rozhraním, aby bylo možné přes toto rozhraní otestovat správnou funkčnost jednotlivých metod.
- Operace bude probíhat v rámci transakce, aby se v případě chyby byla aplikace schopna vrátit zpět do konzistentního stavu.
- Ke každé komponentě bude napsán popis definující jasně její chování a funkcionalitu.
- Dokumentace API požadovaná pro programovací jazyk Java ve formátu JavaDoc.
- V případě, že po určité době nečinnosti dojde k vypršení tzv. session uživatel se bude muset znovu přihlásit.
- Systém validuje vstupní data uživatele na správnost zadaného typu dat bez nutnosti kontaktovat server.
  - Pro textové pole zvaliduje maximální délku zadaného řetězce
  - Do číselného pole neumožní zadat pro číslo nerelevantní znaky
  - Do rozbalovacího seznamu neumožní zadat jinou hodnotu, než je v seznamu hodnot

Splněním tohoto požadavku nejsou dotčeny další validace, které vyplývají z business logiky. Pokud je nějaká validace bude porušena, systém upozorní uživatele konkrétním chybovým hlášením.

- Všechny prvky aplikace, které provádějí nějakou akci (tlačítka, hypertextové odkazy, významové ikony), budou po najetí kurzoru nad daný ovládací prvek:
- demonstrovat uživateli, že se jedná o ovládací prvek změnou kurzoru myši, zobrazí nápovědu k danému ovládacímu prvku formou tooltipu
- V případě, že bude nutné na nějakém místě aplikace zadat uživatelem heslo, pak systém místo reálných znaků zobrazí v tomto poli znaky zástupné

## • Seznam uživatelů



Obrázek č.1 – Uživatelé aplikace

Systém bude podporovat 2 typy uživatelů:

- Neregistrovaný/Registrovaný uživatel
- Admin

### Neregistrovaný uživatel

Neregistrovaný uživatel se může registrovat.

### Registrovaný uživatel

Autorizovanému uživateli jsou k dispozici následující možnosti:

- Zobrazení a aktualizace osobních údajů (e-mail, heslo a login)
- Odstranění účtu
- Vytvoření / čtení / aktualizace / odstranění flash-karty/ balíčku flash-karet a jeho obsah ve vlastním repozitáři
- Přidání a odebrání obsahu flash-kartičky. Obsahem může být obrázek (ImageContent), audio (AudioContent), mnemotechnická pomůcka (MnemonicContent), věta s tímto slovem (ContextSentenceContent)
- Prohlížet, třídit, filtrovat balíčky karet ve svém repozitáři



- Přidávat a odebírat veřejné balíčky karet vytvořené jinými uživateli z svého vlastního repozitáře
- Vytvořit / přečíst tag
- Najít tag podle jména a zobrazit podobné tagy
- Přidat tag na flash-kartu
- Hledat kartičky podle tagu / názvu
- Třídít flash-kartičky
- Zobrazit a vymazat své statistiky. Statistika zahrnuje průběh každé flash-karty v jeho repozitáři
- Zobrazit své úspěchy (př: naučit se 10 nových slov)
- Může používat tréninkový režim

Pro výše uvedený režim může registrovaný uživatel změnit nastavení: číslo karet / sadu karet. Kromě toho bude uživatel schopen po dokončení režim, vidět jeho výsledky a celkové hodnocení všech uživatelů.

## Admin

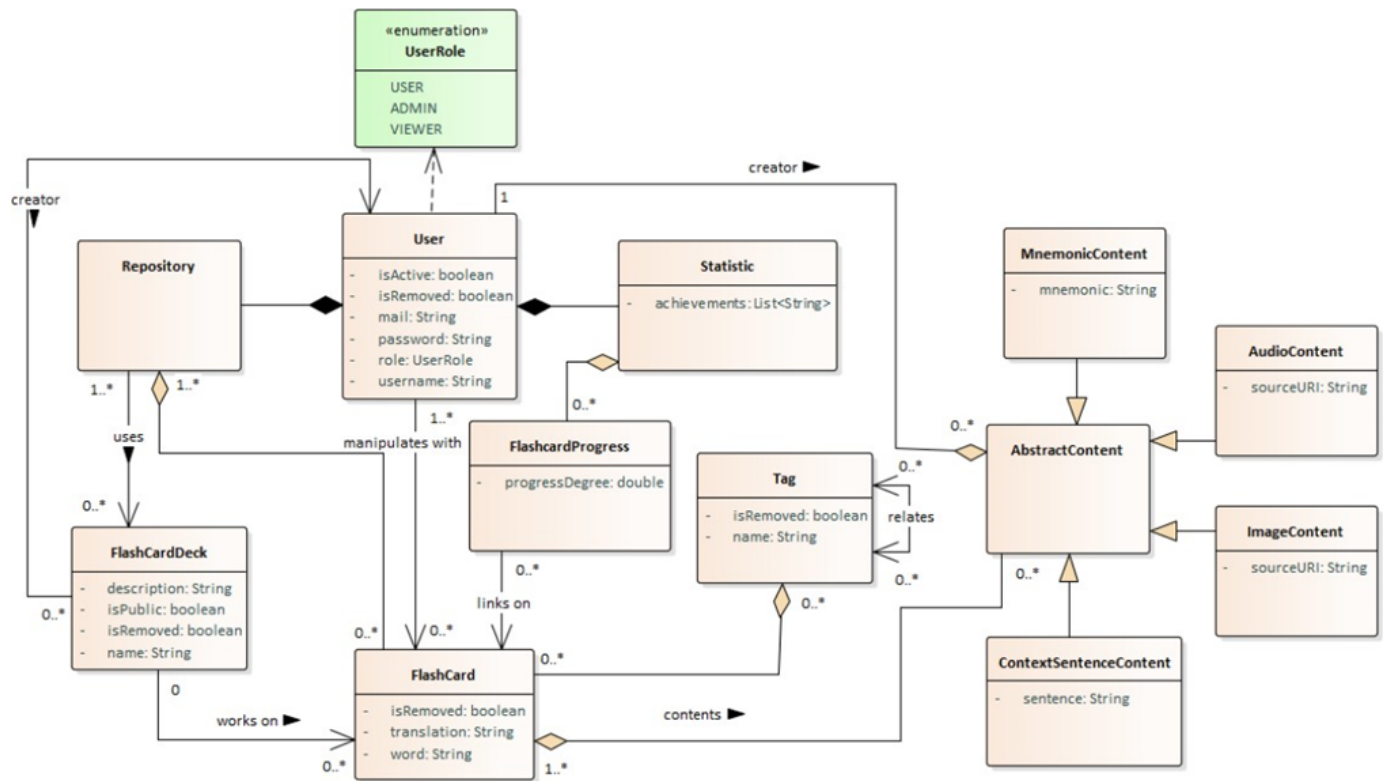
Možnosti administrátoru:

Administrátor může dělat vše, co registrovaný uživatel a navíc:

- Blokovat / Odblokovat uživatele
- Číst seznam všech uživatelů / repozitáře se všemi kartičkami / statistikami
- Číst seznam všech balíčků karet, i když jsou soukromé
- Přidat / odebrat flash-kartu / balíček flash-karet z jednoho repozitáře do druhého
- Zrušení statistiky pro všechny registrované uživatele
- Číst / aktualizovat / odebrat flashcardProgress / úspěchy pro všechny registrované uživatele

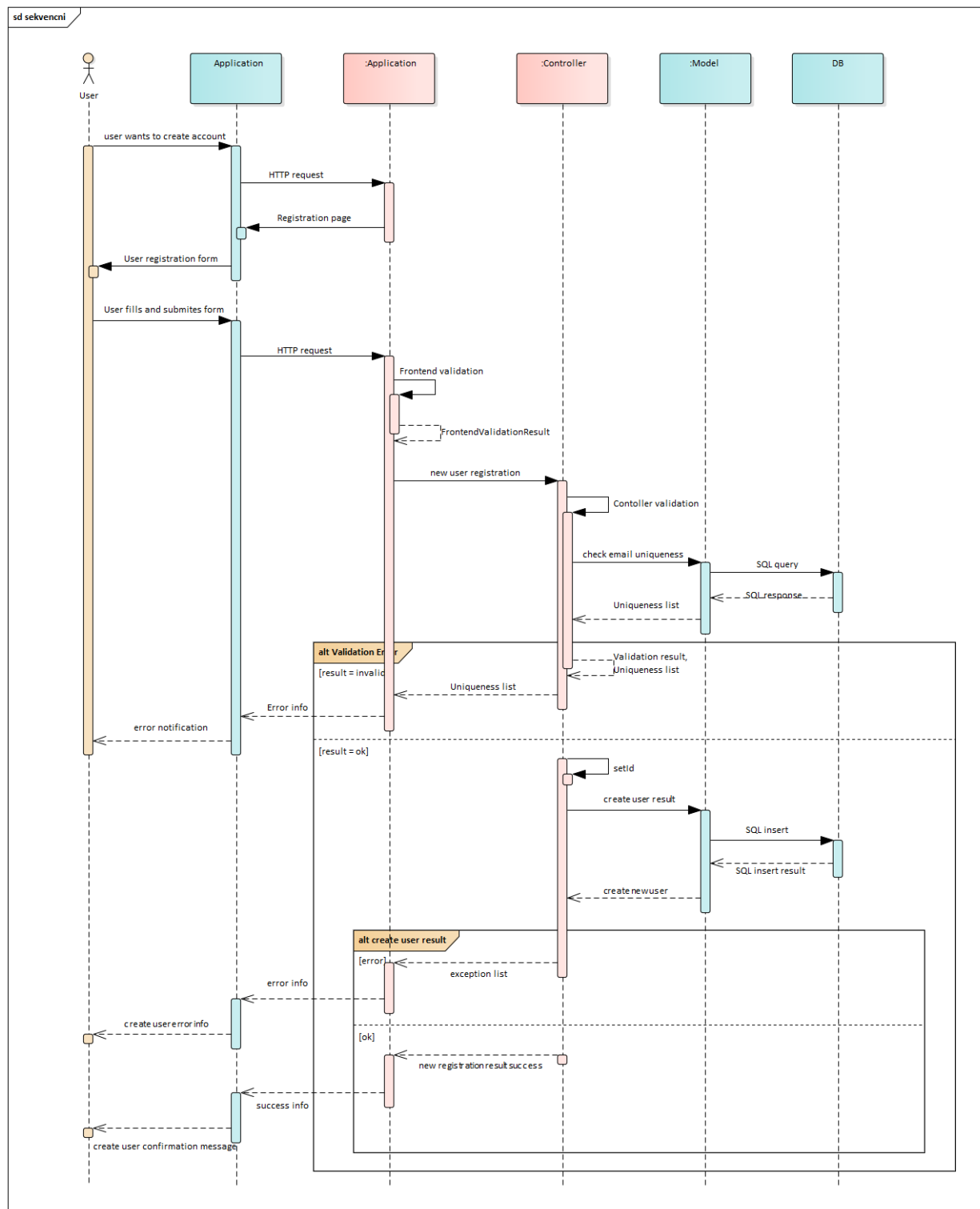
- UML diagramy

## 10.1 Class diagram

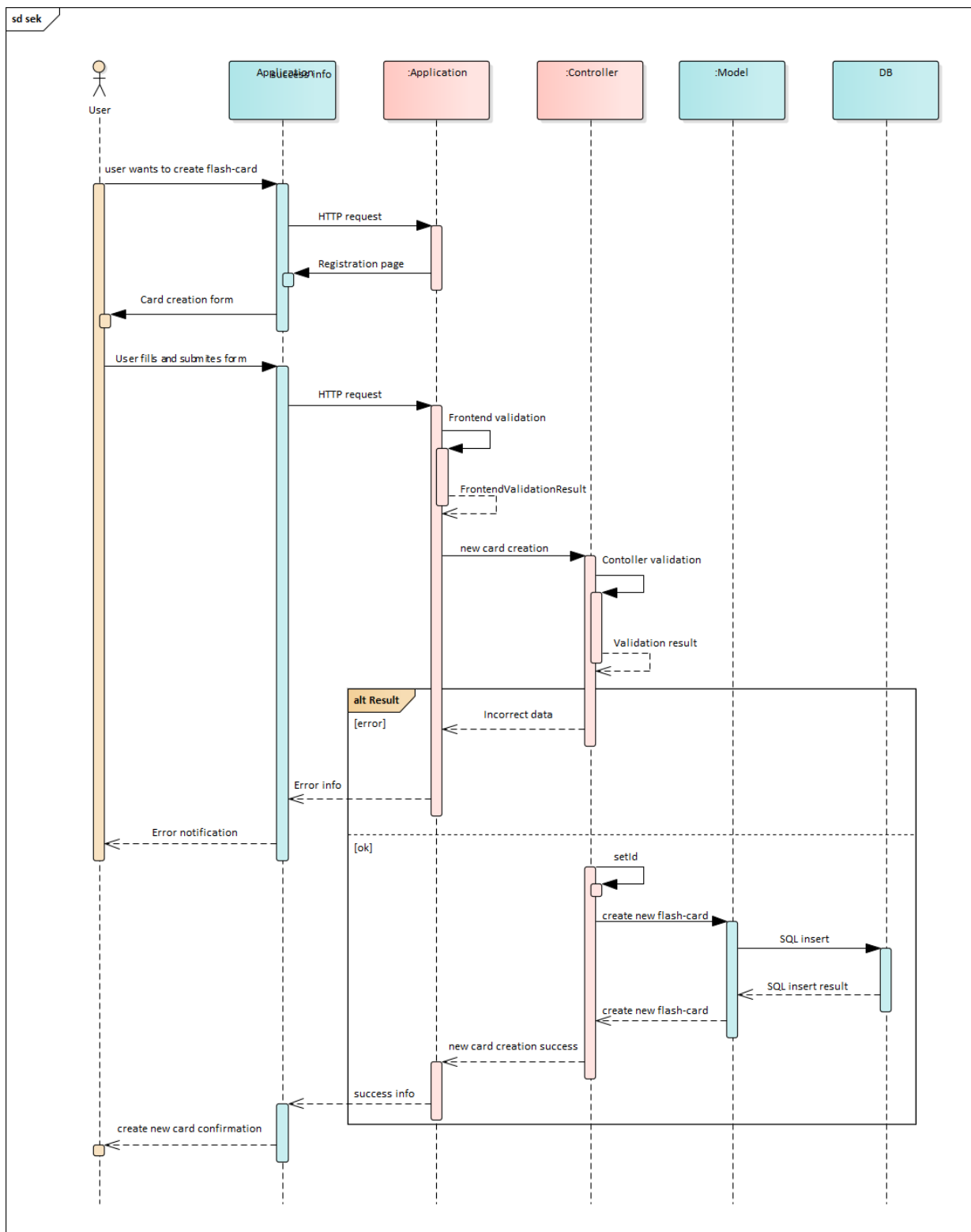


Obrázek č.2 – Class diagram

## 10.2 Sekvenční diagram

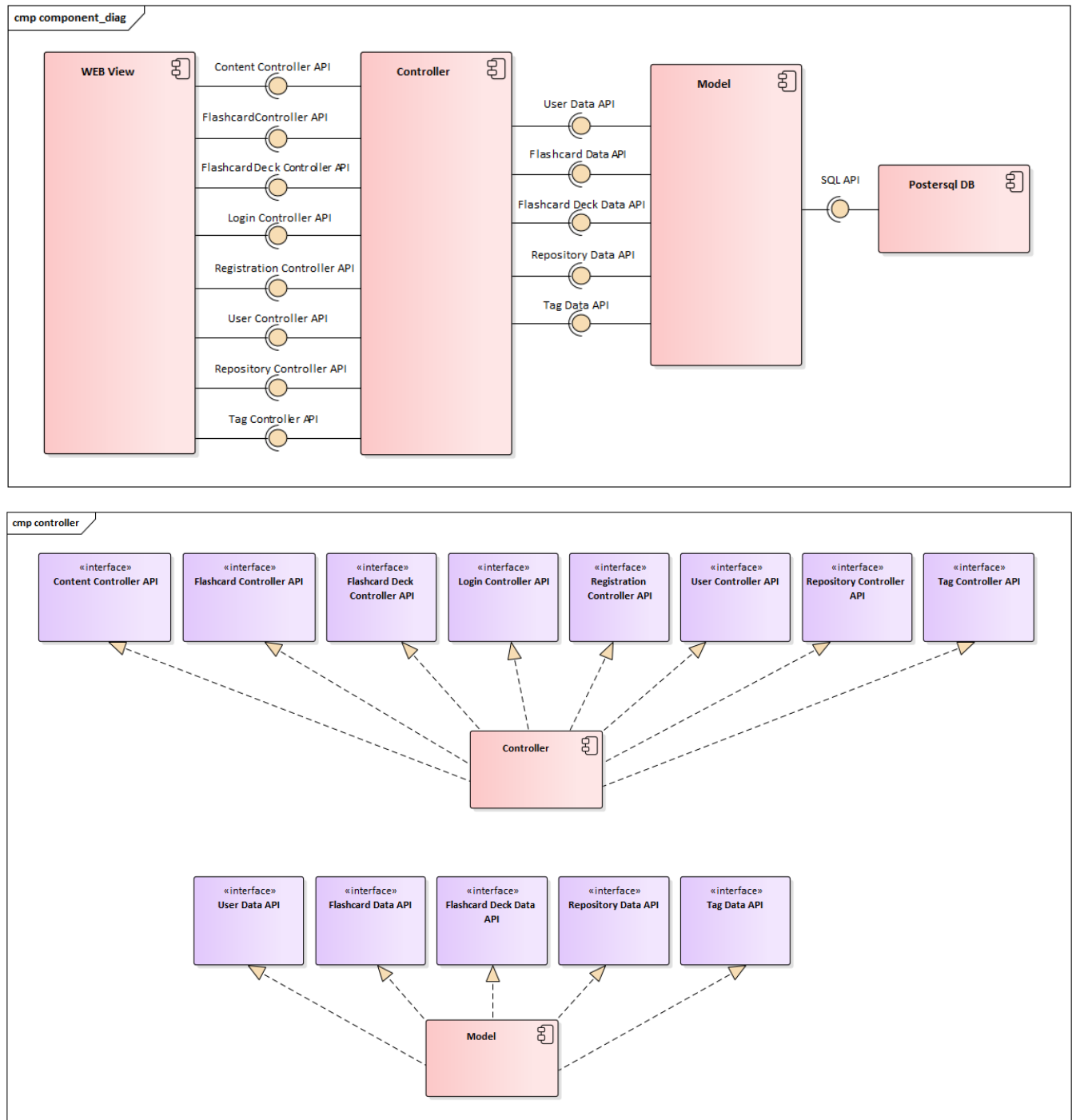


Obrázek č.3 – Sekvenční diagram (registrace)



Obrázek č.4 – Vytvoření flash-kartičky

## 10.3 Component diagram



Obrázek č.5 – Component diagram

- Případy užití



Obrázek č.6 – UC

## 11.1 Registrace

### Registrace

Zaregistruje se buď pomocí běžné registrace nebo použitím registraci Google. Při běžné registraci je nutné zadat uživatelský login a heslo, kterým se bude do aplikace přihlašovat.

SCENARIOS
<p>Basic Path:</p> <ol style="list-style-type: none"><li>1. Systém zobrazí uživateli registrační formulář.</li><li>2. Uživatel vyplní formulář.</li><li>3. IF systém zvaliduje (Bus. rule – BR: Pole je povinné) THEN GO TO STEP 4 ELSE GO TO STEP 2, kde systém upozorní uživatele na nevalidní komentáře a validní komentář zůstanou vyplněné.</li><li>4. Systém uloží formulář a zobrazí upozornění</li><li>5. Uživatel přihlásí do svého účtu</li><li>6. Systém zobrazí úvodní stránku</li></ol>

### Registrace nového uživatele Adminem

Zaregistruje nového uživatele pomocí běžné registrace. Při vytváření nového uživatele je nutné zadat uživatelský login a heslo, kterým se bude do aplikace přihlašovat.

## 11.2 Správa účtu

### Přihlášení

Umožní uživateli, po zadání hesla a loginu, se přihlásit do aplikace v jedné z následujících rolí: Registrovaný uživatel, Admin.

SCENARIOS
<p>Basic Path:</p> <ol style="list-style-type: none"><li>1. Systém zobrazí uživateli přihlašovací formulář.</li><li>2. Uživatel vyplní formulář.</li><li>3. IF systém zvaliduje (Bus. rule – BR: Pole je povinné) THEN GO TO STEP 4 ELSE GO TO STEP 2</li><li>4. Systém přesměruje registrovaného uživatele do úložiště</li></ol>

### Změna osobní údajů

Umožňuje upravit osobní údaje uživatele. Všechny osobní údaje lze změnit. Pro změnu hesla jej lze zadat dvakrát. Uživatel může měnit pouze své údaje. Admin může měnit údaje libovolného registrovaného uživatele včetně jeho roli, pokud uživatel není blokován.

### Smazání účtu

Odstraní z aplikace zvoleného uživatele. Odstraněný uživatel již nebude moci využívat webovou aplikaci. Admin může odstranit libovolného registrovaného uživatele, registrovaný uživatel má možnost smazat pouze svůj osobní účet.

### **Blokování/odblokování uživatele Adminem**

Zablokuje/odblokuje uživatele v aplikaci. Zablokovaný uživatel již nebude moci využívat webovou aplikaci pokud nebude odblokován.

### **Zobrazení seznamu všech uživatelů**

Zobrazí seznam všech uživatelů, kteří mohou aplikaci využívat.

### **Zobrazení profilu uživatele**

### **Filtrování/řazení seznamu všech uživatelů Adminem**

## **11.3 Správa flash karet**

### **Vytvoření kartičky**

Umožňuje uživateli vytvořit novou flash kartu ve svém repozitáři.

SCENARIOS
Basic Path: <ol style="list-style-type: none"><li>1. Systém registrovanému uživateli zobrazí formulář pro vytvoření nové karty.</li><li>2. Registrovaný uživatel vyplní formulář.</li><li>3. IF systém zvaliduje (Bus. rule – BR: Pole je povinné) THEN GO TO STEP 4 ELSE GO TO STEP 2, kde systém upozorní uživatele na nevalidní komentáře a validní komentář zůstanou vyplněné.</li><li>4. Systém uloží formulář a přesměruje registrovaného uživatele do úložiště, kde zobrazí všechny jeho karty.</li></ol>

### **Aktualizace kartičky Adminem**

Umožňuje upravit údaje o karte. Kartička bude aktualizována pro všechny uživatele, kteří ji mají ve svém repozitáři.

Aktualizace kartiček ADMINEM slouží pro moderace web aplikace.

SCENARIOS
Basic path: <ol style="list-style-type: none"><li>1. Uživatelům s roli ADMIN, zobrazí kartičky nevyhovující pravidlem web aplikace.</li><li>2. ADMIN, zmáčkne na kartičku.</li><li>3. Systém přesměruje do kartičky.</li><li>4. ADMIN, udělá změny kartičky.</li><li>5. Systém uloží změny.</li></ol>

### **Zobrazení kartičky**

Admin může zobrazit libovolné karty, běžný uživatel může zobrazit pouze karty, které jsou ve veřejných balíčcích.



### Odebrání kartičky z repozitáře

Odstraní kartičku ze zvoleného repozitáře pro Admina. Pro běžného uživatele umožňuje odstraní kartičku z vlastního repozitáře.

### Zobrazení seznamu všech kartiček v repozitáře

#### Zobrazení seznamu všech kartiček pro Admina

Zobrazí seznam všech karet ve všech repozitářích.

### Řazení/filtrování kartiček v repozitáře

#### Vyhledávání kartičky podle slova pro Admina

Umožní Adminu zobrazit vše karticky

SCENARIOS
Basic path: <ol style="list-style-type: none"><li>1. ADMIN, zmáčkne tlačítko zobrazení všech kartiček.</li><li>2. Systém, ukáže vše kartičky.</li></ol>

## 11.4 Správa balíčků karet

### Vytvoření balíčků karet

Umožní přidat do repozitáře uživatele nový balíček karet. Při vytváření je nutné zadat název. Je možnost také přidat popis.

SCENARIOS
Basic path: <ol style="list-style-type: none"><li>1. Registrovaný uživatel zmáčkne tlačítko pro vytvoření nového balíčku.</li><li>2. Systém zobrazí formulář.</li><li>3. IF uživatelské vstupní data validní (Bus. rule - BR: Pole je povinné) THEN GO TO STEP 4 ELSE GO TO STEP 2, kde systém ukáže nevalidní inputy.</li><li>4. Systém vytvořit balíček podle dat a pak přesměruje uživatele do úložiště.</li></ol>

### Aktualizace balíčků karet Adminem

Aktualizace balíčků ADMINEM slouží pro moderace web aplikace.

SCENARIOS
Basic path: <ol style="list-style-type: none"><li>6. Uživatelům s roli ADMIN, zobrazí balíčky nevyhovující pravidlem web aplikace.</li><li>7. ADMIN, zmáčkne na balík.</li><li>8. Systém přesměruje do balíčku.</li><li>9. ADMIN, zvolí buď deaktivovat balíček nebo skryt balíček.</li><li>10. Systém uloží změny.</li></ol>

### Zobrazení balíčků karet

Pokud uživatel není Admin, může si prohlížet pouze veřejné balíčky.

SCENARIOS
Basic path: <ol style="list-style-type: none"><li>1. Registrovaný uživatel, stlačí na sekce prohlížení.</li><li>2. System, ukaže input pole.</li><li>3. Uživatel napíše název balíčku.</li><li>4. IF balíček existuje, systém ukáže potřebný balíček ELSE ukáže prázdný container.</li></ol>

### Smazání balíčků karet

Odstraní z aplikace zvolený balíček. Odstraněný balíček již nebude moci nikdo používat.

SCENARIOS
Basic path: <ol style="list-style-type: none"><li>1. Registrovaný uživatel, zmáčkne na potřebující balíček.</li><li>2. Systém, zobrazí možnosti akce s balíčkem.</li><li>3. Uživatel vybere možnost smazání.</li><li>4. Systém uloží změny a smaže balíček.</li></ol>

### Odstranění balíčku karet z repozitáře uživatele

Umožní registrovanému uživateli odstranění balíčku ze svého vlastního repozitáře a Adminu umožňuje jeho odstranění z libovolného repozitáře.

SCENARIOS
Basic path: <ol style="list-style-type: none"><li>1. IF uživatel není ADMIN THEN GO TO <b>Smazání balíčků karet</b> ELSE GO TO <b>Aktualizace balíčků karet Adminem</b></li><li>2. Systém, uloží změny.</li></ol>

### Zobrazení seznamu všech balíčků karet

Umožní Adminu zobrazit vše balíčky a registrovanému uživateli pouze veřejně

SCENARIOS
Basic path: <ol style="list-style-type: none"><li>3. Uživatel, zmáčkne tlačítko zobrazení všech balíčků.</li><li>4. IF uživatel není ADMIN THEN GO TO STEP 3 ELSE GO TO STEP 4.</li><li>5. Systém, ukáže veřejné balíčky.</li><li>6. Systém, ukáže vše balíčky.</li></ol>

## 11.5 Správa tagů

### Vytvoření tagu

Umožní uživateli přidat nový tag.

### Aktualizace tagu

Umožní Adminu aktualizovat tag.

### Zobrazení tagů

Zobrazí vše tagy.

### Zobrazení podobných tagů

Zobrazí propojené tagy s zadaným tagem, pokud existují. Tagy jsou propojeny, pokud patří ke stejné flash kartě.

### Odstranění tagu

Umožní Adminu odstranit tag z aplikace.

### Vyhledávání tagu podle názvu

### Přidání tagu na kartičku

Umožní uživateli přidat tag na kartičku, která je v jeho vlastním repozitáře.

SCENARIOS
Basic Path: <ol style="list-style-type: none"><li>1. INCLUDE (Zobrazení kartičky).</li><li>2. Registrovaný uživatel vybere "Přidat tag".</li><li>3. INCLUDE (Zobrazení tagů).</li><li>4. Registrovaný uživatel vybere tag a klikne na tlačítko "Přidat".</li><li>5. Systém kartičku uloží a zobrazí registrovanému uživateli.</li></ol>

### Odebrání tagu z kartičky

Umožní Adminu odstranit tag z kartičky.

## 11.6 Správa statistik

### Zobrazení statistik

Umožní Adminu zobrazit statistiky všech uživatelů a pouze jeho vlastní pro registrovaného uživatele.

### Vymazání statistik

Umožní Adminu vymazat statistiky libovolného uživatele a pouze jeho vlastní pro registrovaného uživatele.

## 11.7 Tréninkový režim

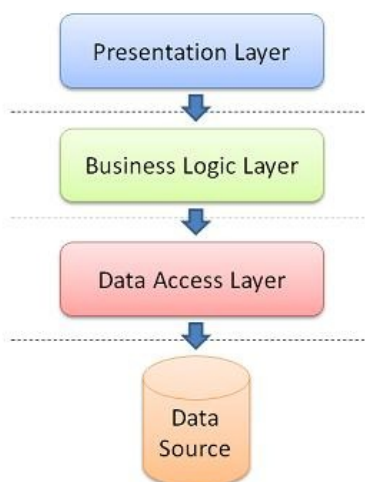
### Použití tréninkového režimu

### Vybirání flash karet z osobního repozitáře

### Zobrazení výsledků

- Výběr vhodné architektury

První z vybraných architektur je **vrstvená architektura**.



Obrázek č.7 – Třívrstvá architektura

Tři vrstvy byly zvoleny jako nejvhodnější počet vrstev pro back-endovou část aplikaci:

- 1) Prezentační — stará se o zobrazení dat uživateli a zpracování dat od uživatele.
- 2) Aplikační — provádí byznys logiku.
- 3) Datová — stará se o ukládání a načítání dat z databáze.

Uživatel komunikuje s prezentační vrstvou, která pak komunikuje s vrstvami pod ní. Tyto vrstvy pak komunikují s ostatními nižšími vrstvami, dokud jedna z vrstev nerozhodne, že požadavek byl zpracován. V každém kroku je s požadavkem provedena operace. Požadavek může ale nemusí dostat až k nejnižší vrstvě. Pokud vrstva zpracuje požadavek, může tato data vrátit. Poté procházejí vrstvami zpět k uživateli.

Pro klientskou část aplikace byla zvolena technologie React, která bude komunikovat se serverovou částí aplikace přes REST API. První a zároveň nejdůležitější podmínkou RESTful je architektonické omezení klient-server.

**Monolitická architektura** - je tradiční jednotný model pro návrh softwarového programu. Monolitický v tomto kontextu znamená složený celý v jednom kuse.

## • Design patterns

Využité design patterns v projektu.

- Interceptor
- Factory
- Chain of Responsibility
- Proxy
- Singleton
- IoC
- Template
- DAO
- Facade
- Adapter
- Builder
- DTO

## • Analýza rizik

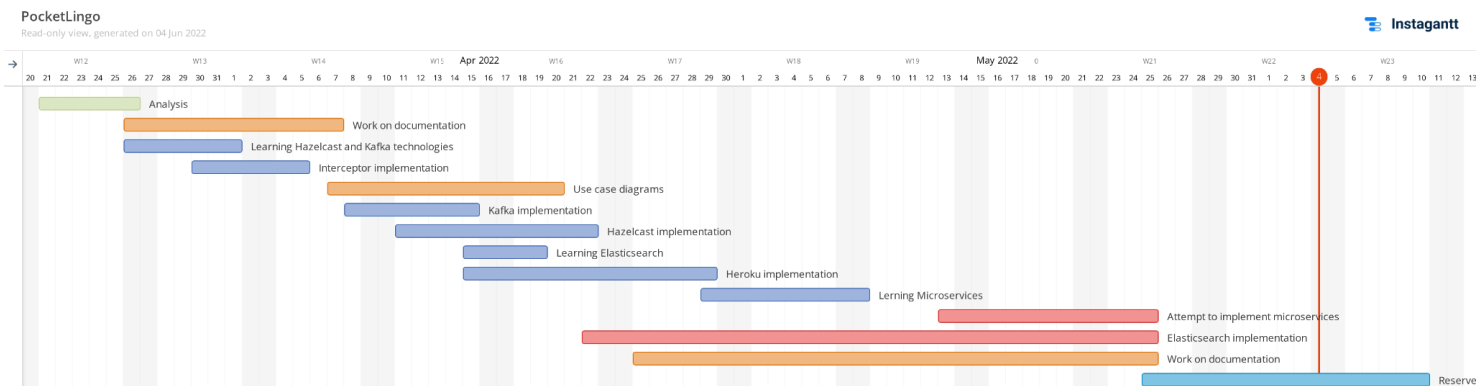
Procesní část	Selhání	Possible impact on customer	Možný dopad na zákazníka	S E V	O C C	D E T	R P N	Doporučené opatření	Možný dopad na projekt	Odpovědný subject
Implementace	Chyba v technologiích	Zpoždění dodání projektu	Chybný výběr technologií	2	5	4	40	Prostudovat dokumentaci	Fin. problémy kvůli zpoždění harmonogramu	Anna Kachmasheva
Implementace	Low-quality kod	Nespokojenost zákazníka	Nedostatek času	2	5	3	39	Přesná analýza a plánování	Nizká kvalita UI	Mukan Atazhanov
Ostatní	Člen týmu odchází	Zpoždění dodání projektu	Nespokojenost v týmu	2	2	3	12	Domluvit se předem	Fin. problémy kvůli zpoždění harmonogramu	Aiya Rakhimova

## • Zdroje

### Lidské zdroje

- Analytik(1) - odpovídá za analýzu projektu a plánování řešení problému.
  - **Aiya Rakhimova**
- Projektový manažer(1) - odpovídá za řízení projektu. Kontroluje deadliny a splnění požadavků.
  - **Mukan Atazhanov**
- Developeři(3)
  - **Aiya Rakhimova** - odpovídá za analytickou část a za service cachování Hazelcast. Vytvoření dokumentace
  - **Anna Kachmasheva** - odpovídá za back-end část, za service ElasticSearch a za další technologií.
  - **Mukan Atazhanov** - odpovídá za front-end část, za service Kafka a konečně deployment.

## • GANTT



## • Matice zodpovědnosti

Úkoly	Mukan	Aiya	Anna
Kafka			
Heroku			
Hazelcast			

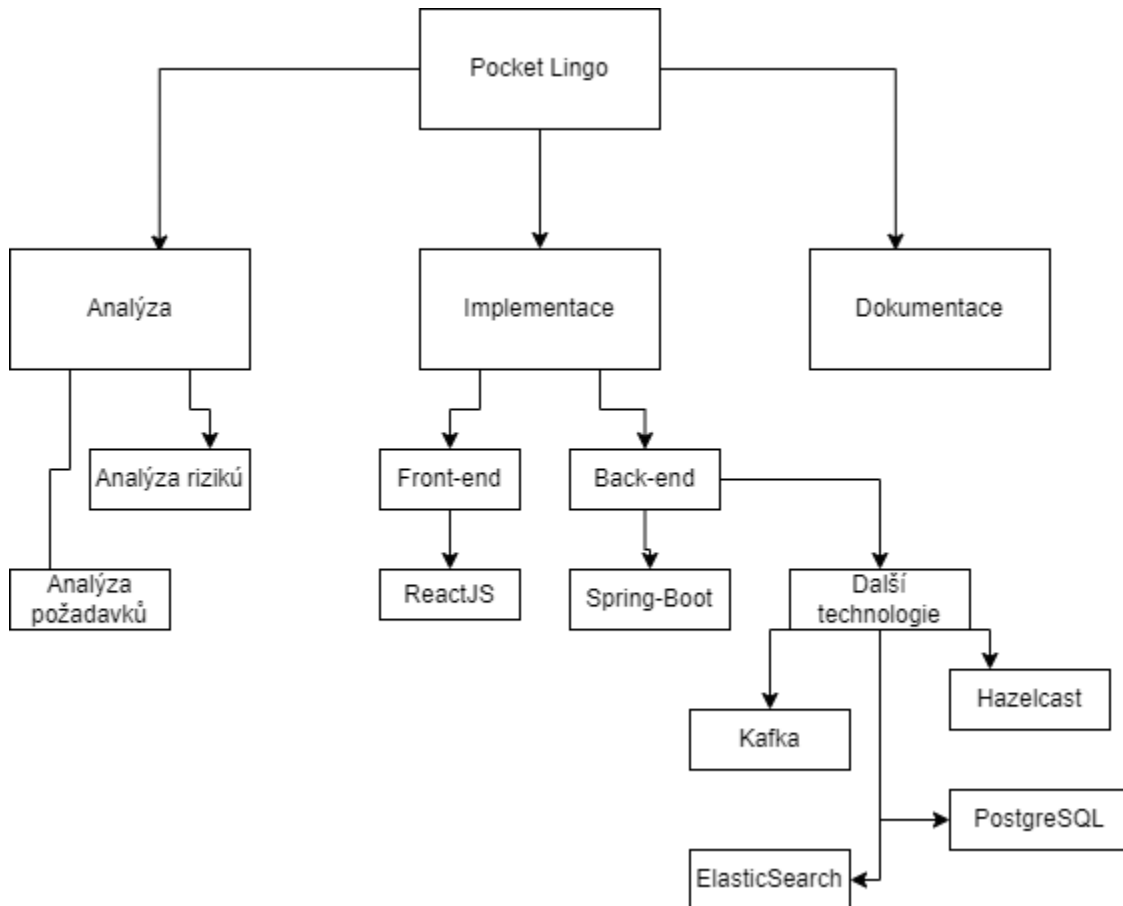
Elasticsearch			
Interceptors			
Use case			
Analyza 5F			
Analyza PEST			
Seznam uzivatelu			
Sekvencni diagram			
UML diagram			
Funkcni pozadavky			
Nefunkcni pozadavky			
Design patterns			
Pripady uziti			
Popis aplikace,motivace			
AS-IS			
TO-BE			
Analyza SWOT			
Component diagram			
Microservices			
Dokumentace			

## • Alternativ návrhu řešení

- Open-Source řešení  
Nejlevnější přístup. Nevýhoda v tom, že je obtížné najít řešení s otevřeným zdrojovým kódem, které by odpovídalo všem požadavkům.
- Vlastní vývoj  
Nejúčinnější alternativa. Tento přístup však prodlužuje dobu vývoje a je těžko určit datum dokončení projektu. Hlavní výhoda je, že můžeme přidat nové features během vývoje a nebudu to stát tak moc peněz.

- Objednat u další firmy  
Dobrá alternativa, jestli nechceme se starat o vývoju. Ale pro dobré výsledky musíme poskytnout podrobné rozepsané požadavky a specifikace. Nejdražší přístup. Je vhodná, jestli máte moc peněz a produkt vyžadován v omezeném čase.

## ● WBS - Work Breakdown Structure



## ● Normy a standardy

Při vytváření tohoto projektu vývojáři musí dodržovat následující standardy

- Java 17
- Spring 2.5.4
- PostgreSQL (W3C standard)
- ReactJS 18.0.0



- HTML (W3C standard)
- CSS (W3C standard)

## ● Metrika

V projektu jsou sledovány tyto metriky

- Počet případů užití
- Celkový čas strávený na projektu
- Počet lidí potřebných k projektu

## ● Plan podpory

1. Oprava problémů, které nebyly odhaleny provedenými testy
2. Vypracování dalších, nových požadavků, které jsou spojené s tímto projektem ale buď nebyly ve specifikace nebo nebyly realizovány.

## ● Vyhodnocení

**Aiya Rakhimova** - Přišel se mi tento předmět moc přínosný. Dozvěděla jsem o spoustě nových technologiích, o jejich existenci a využití v praxi.

**Anna Kachmasheva** - Díky této práci jsem se dozvěděla něco nového o analýze. Seznámila jsem se s něčím novým a zopakovala jsem znovu, co jsem se naučila na jiných předmětech v předchozích semestrech.

**Mukan Atazhanov** - Během tohoto semestru jsem se naučil novým technologiím jako kafka, hazelcast, heroku. Dozvěděl jsem víc o architektuře software-u, což určitě bude užitečné v budoucnosti.