



FAKULTA
ELEKTROTECHNICKÁ
ČVUT V PRAZE

B6B36NSS - Návrh softwarových systémů

Semestrální práce

Studentský portál - BERLOGA

Obsah

Obsah	2
Popis projektu	4
Obchodní přínos	4
Stav „AS IS“	4
Analýza SWOT	5
Interní faktory	5
Silné stránky (S)	5
Slabé stránky (W)	5
Externí faktory	5
Příležitosti (O)	5
Hrozby (T)	5
Analýza 5F	6
Nová konkurence	6
Stávající konkurence	6
Vliv dodavatelů	6
Vliv zákazníků	6
Substituční produkty	6
Analýza PEST(E)	7
Economical	7
Technological	7
Environment	8
Funkční požadavky	9
FR-1: Zakládání projektu	9
FR-2: Invite učitelů a žáků	9
FR-3: Management uživatelů	9
FR-4: Operace s předměty (CRUDL)	9
FR-5: Vytvoření a psaní zpráv soukromých a skupinových	9
FR-6: Aktualizace a novinky (nice to have)	9
FR-7: Založení diskuze/fóra	10
FR-8: Správa kalendáře	10
FR-9: Odevzdávání prací a domácích úkolů (nice to have)	10
Nefunkční požadavky	11
NFR-1: Zálohování předmětových stránek	11
NFR-2: Rozšiřitelnost	11
NFR-3: Bezpečnost portálu	11
NFR-4: Podpora všech prohlížečů	11
NFR-5: Podpora škálovatelnosti	11
NFR-6: Otevřená RESTful api	11

NFR-7: Lokalizace	11
NFR-8: Jen OpenSource, nelicencované komponenty a služby	12
Seznam uživatelů	13
Případy užití	14
Role	14
Správa uživatelů	15
N-N Zprávy	16
1-1 Zprávy	17
Kalendář	18
Správa fóra	19
Správa předmětových stránek	20
Správa projektu	21
Správa novinek a aktuality	22
UML diagramy	23
Class diagram	23
Sequence diagram	24
Architektura aplikace	25
Diagram nasazení	26
Popis architektury nasazení:	26
Diagram komponent	27
Architektura serverové části	28
Architektura frontendové části	29
WBS	29
Rozbor a výběr alternativ návrhu řešení	30
Vývoj vlastního produktu	30
Využití existujícího open-source produktu	30
Zdroje	30
Lidské zdroje	31
Nástroje potřebné pro vývoj	31
Normy a standardy	32
Matice odpovědnosti (RACI)	33
Harmonogram GANTT	34
Analýza rizik FMEA	35
Znovupoužitelnost	36
Metriky	36
Plán odbavení	37
Plán podpory	37
Vyhodnocení	38

1. Popis projektu

Hlavní podstatou projektu BERLOGA je studentský portál, který poskytuje svým uživatelům různé funkce užitečné pro studium a pohodlné způsoby komunikace. Hlavními vlastnostmi projektu jsou group messaging, sdílení souborů, kalendář, rozvrh hodin a diskuzní fórum.

Studentský portál poskytuje možnost vytváření stránek ke každému předmětu, které obsahují informace o samotném předmětu a záznamy z přednášek či seminářů. Diskuzní fórum umožní pokládat dotazy a dostávat odpovědi jak od žáků, tak od učitelů. Každý student má vlastní sekci s kalendářem obsahujícím studijní rozvrh a možnosti vytváření dalších aktivit a připomínek.

Systém má několik rolí pro přihlášeného uživatele: moderátor, učitel, student a některé stránky bude moci prohlížet i nepřihlášený uživatel. V závislosti na roli uživatele (ne)budou k dispozici některé funkce portálu. Uživatelé mohou mít několik rolí zároveň (např. učitel může být zároveň moderátorem).

2. Obchodní přínos

Naše aplikace pomůže propojit všechny věci týkající se studia na jednom místě. V současné době, v souvislosti s novým omezením se studenti učí výhradně na dálku, a proto u nich objevilo mnoho různých služeb a aplikací. Pro usnadnění studia jsme vymysleli jeden studentský portál, na kterém je vše, co potřebujete student.

Pomocí vlastního kalendáře portál umožňuje snadné a intuitivní vytvoření osobního rozvrhu s rozšířenými funkcemi pro unifikaci vlastních studentských aktivit. Spokojený student - spokojená univerzita.

3. Stav „AS IS“

V současné době existuje mnoho různých aplikací pro studenta, jako jsou BRUTE, MOODLE, CW a mnoho různých samostatných stránek některých předmětů. Neexistuje místo, kde by studenti stejného kurzu mohli komunikovat a komunikovat výhradně studiem, vytvářet skupiny a chatovací místnosti, aniž by se uchýlili k třetím stranám.

4. Strategický záměr (Stav „TO BE“)

Naše strategie je taková- chceme, aby studenti měli moderní komfortní prostředí, kde mají vše, co potřebují ke studiu. Naším cílem je vytvořit studentskou sociální síť, kde existuje možnost vytvářet skupinové chaty, klást otázky týkající se učení na fóru, učitelé budou mít možnost přidávat domácí úkoly a na stejném místě kontrolovat po odevzdání žákem. Studentský portál je láska.

5. Analýza SWOT

Interní faktory

Silné stránky (S)

- Semknutý kolektiv vývojářů.
- Partnerství s univerzitou ČVUT.
- Tým vývojářů se skládá z potenciálních uživatelů.

Slabé stránky (W)

- Špatně finanční zázemí.
- Slabá časová podpora projektu.
- Nedokonalost znalosti vývojářského týmu.

Externí faktory

Příležitosti (O)

- Zájem o projekt investorů.
- Poměrně malá konkurence.
- Spolupráce s velkou firmou.
- Zájem o projekt dalších univerzit.
- Prodloužení doby mimořádných opatření.
- Ukončení činnosti jednoho z vysokoškolských portálu.

Hrozby (T)

- Zdražení podporujících servisů.
- Nová konkurence ve zkoumaných oblastech.

6. Analýza 5F

Nová konkurence

V dnešní době není na českém trhu žádný dominantní konkurent mezi studentskými portály, který by byl zaměřen na poskytování služeb pro studium a pohodlnou komunikaci ohledně nových způsobů výuky během karanténní situace.

Stávající konkurence

Kvůli nedostatku nové a skutečné konkurence není obtížné je pro nového konkurenta vstoupit na trh, jediná komplikace, která může nastat je slabá propagace portálu. Mezi klíčové bariéry vstupu na trh patří všechny body ze slabých stránek SWOT analýzy a to jsou: špatně finanční zázemí, slabá časová podpora projektu a nedokonalost znalosti vývojářského týmu.

Vliv dodavatelů

Každé softwarové projekty vyžaduje k produkci zdroje - pracovní sílu, podporující servis, již hotové komponenty a jiné zásoby. V našem případě dodavatele nejsou silní, protože patří mezi pracovními silami jen síly vývojářského týmu a nepoužívá projekt žádné unikátní servis, vzácné komponenty či drahé služby.

Vliv zákazníků

V případě zákazníku či odběratelů našeho systému je situace opačná. Celý portál je postaven na zájmu uživatelů, který ten portál ovlivňuje co nejvíc. Cílová skupina uživatelů jsou studenti. Pokud nebudou studenti mít o daný portál zájem, nebudou existovat podmínky pro unifikace a sjednocení všech studijních materiálů, nahrazování chybějící komunikace a včasná aktualizace článků funkce odběru studijních novinek.

Substituční produkty

Produkty se všemi jeho službami může být úspěšně nahrazen jiným produktem jen v případě, když podobný portál bude vyvíjen jednou z velkých programátorských společností, protože tím pádem.

7. Analýza PEST(E)

Síla vlivu může být tři typy (1, 2, 3), kde 1 je nejmenší vliv a 3 je největší.

Economical

Množství peněz, které stát může dát univerzitám

Množství peněz přidělených státem přímo ovlivňuje státní vysoké školy, které by podle našeho plánu měly službu Berloga připojit za poplatek. Přidělení peněz na projekty na pomoc studentům při výuce soukromých vysokých škol bude mít také přímý dopad na rozvoj našeho projektu.

Chtěl bych poznamenat, že připojení univerzity k Berloze nebude příliš drahé a každá univerzita si to bude moci dovolit. Takže to náš projekt rozhodně neláká.

Síla vlivu – 1

Počet studentů v zemi

Česko je poměrně vyspělá země, kde se klade velký důraz na vzdělání a zemi každoročně navštěvuje obrovské množství zahraničních studentů. V současné době neexistuje tendence k poklesu počtu studentů a strach z tohoto faktoru také nestojí za to.

Síla vlivu – 1

Konkurence

Konkurence je pro nás nejdůležitějším faktorem, protože pokud na trh uvedeme dostatečně surový SW, snadno ho může obejít jiná vyspělejší společnost, která v krátké době vyrobí lepší produkt. V Česku jsou v tuto chvíli velké firmy, které se toho mohou chopit a máme se čeho bát v tomto faktoru.

Síla vlivu – 3

Technological

Vývoj zabezpečení

Pokud se přijde na vývoj algoritmu na hashování a kryptografii, či když někdo prolomí algoritmus, který my používáme (sha-512), budeme muset nahradit zabezpečení novějším algoritmem či složitější technologií.

Síla vlivu – 2

Změna fungování prohlížečů, podpory JS apod.

Někdy výrobci a poskytovatelé prohlížečů jako Chrome od Google čas od času něco změní. Proto musíme počítat, že v tomto případě budeme muset flexibilně reagovat na novinky a dopředu zjišťovat, co plánují, abychom měli dostatek času aplikovat změny.

Síla vlivu – 3

Změny cen

Příkladem může být dnešní ceny grafických karet, kdy jejich relativní nedostupnost zvedla jejich cenu na dvojnásobek. Změny cen dodavatelů elektrotechniky a připojení internetu, vyvodí k zlevnění/zdražení portálu, což může změnit finanční analýzu.

Síla vlivu – 1

Zastarávání technologie

Technologie a hardware, který používáme, postupem času zastarává, jeho výkon se relativně zmenšuje, na trhu jsou novější a rychlejší modely. Také čas od času odejde starší kus a musí se nahradit novým.

Síla vlivu – 2

Environment

Environmentálně je naše aplikace úplně nezávislá a neexistuje žádný faktor, kromě uragánu, zemětřesení a meteoritu, který by mohl mít vliv na fungování našeho systému.

8. Funkční požadavky

8.1. FR-1: Zakládání projektu

- Každá škola má jeden
- Zakládá ji správce/zástupce ředitele/ředitel a stane se správcem/moderátorem

8.2. FR-2: Invite učitelů a žáků

- Moderátor může vytvářet invite (účty) učitelům a žákům
- Učitelé mohou vytvářet účty žákům (?)

8.3. FR-3: Management uživatelů

- Přidávání a úprava uživatelů, přidávání oprávnění
- Učitel (když učí předmět) nebo moderátor mohou přidávat/odstranit žáka do předmětu
- Uživatel si bude moci změnit heslo a dodatečné údaje

8.4. FR-4: Operace s předměty (CRUDL)

- Učitel nebo moderátor budou moci přidávat, měnit a odstraňovat předměty
- Každý předmět má 1-N učitelů
- Každý předmět má automaticky po založení svou stránku, a učitel, který je v předmětu, nebo moderátor, jí může upravovat

8.5. FR-5: Vytvoření a psaní zpráv soukromých a skupinových

- Uživatel může napsat jinému, který je v systému (přes jméno nebo email)
- Uživatel může vytvořit skupinu pro více lidí, bude její hlavní člen
- Uživatelé mohou poslat soubor do chatu
- Uživatelé mohou zprávu upravit nebo odstranit (mark deleted)

8.6. FR-6: Aktualizace a novinky (nice to have)

- Podobné z FELu
- Moderátor přidávají novinky, a vybírají, komu se pošlou, nebo jsou globální, ke škole či předmětu

8.7. FR-7: Založení diskuze/fóra

- Uživatel může vytvořit a odpovídat na diskuze
- Uživatel nebo učitel může uzavřít diskusi

8.8. FR-8: Správa kalendáře

- Každý uživatel má 1 svůj kalendář
- Uživatel může do kalendáře přidávat události, přednášky a cvičení, vlastní aktivity a další
- Možnosti přidat jiného uživatele k události v kalendáři (skupinové aktivity)
- Úpravy a odstraňování...

8.9. FR-9: Odevzdávání prací a domácích úkolů (nice to have)

- Každý předmět bude mít čas pro odevzdávání domácích prací s časovačem

9. Nefunkční požadavky

9.1. NFR-1: Zálohování předmětových stránek

- Každý semestr (nebo ručně) může učitel předmětu udělat kopii (zálohu) pro předešlé ročníky

9.2. NFR-2: Rozšiřitelnost

- Použití nových funkcí, které nejsou předvídány v době návrhu, vyžaduje rozšiřitelné nástroje a metody pro vývoj aplikace.

9.3. NFR-3: Bezpečnost portálu

- Hashování hesel
- Autentizovaný přístup

9.4. NFR-4: Podpora všech prohlížečů

- Systém musí být kompatibilní s Chrome, Firefox, Opera, Safari.

9.5. NFR-5: Podpora škálovatelnosti

- Systém bude umožňovat jednoduchou rozšiřitelnost v podobě škálování aplikace, a to virtuálně nebo přímo na více fyzických serverech skrze load-balancer.
- Zároveň systém by mohl, jelikož hodně operací je čtení, umožňovat vertikální nebo horizontální škálování i pro databáze. Mohl by být použit architekturou Master-Slave nebo Multi-Master-Slave. Shardování zde je nepoužitelné, protože data nám zde tvoří nerozdělitelný celek.

9.6. NFR-6: Otevřená RESTful api

- Pro otevření dalších možností a vývoj externích aplikací třetími stranami, naše aplikace bude obsahovat api s dokumentací s autorizovaným přístupem.

9.7. NFR-7: Lokalizace

- Aplikace bude obsahovat možnost výběru lokalizace, ze základu bude anglická a česká verze

9.8. NFR-8: Jen OpenSource, nelicencované komponenty a služby

- Jazyky, prostředí, frameworky (kromě operačního systému, vstupních nákladů jako serveru a zabezpečení, například firewall apod.) budou bez licencování, nebo s platbou bez opakování. Cenu a rozhodování se prodiskutuje předem se zákazníkem.

10. Seznam uživatelů

Systém bude podporovat 4 typy uživatelů:

Host

Host se může pouze registrovat, přihlásit se a prohlížet jen omezený počet stránek.

Student

Přihlášený (jako student) uživatel již může využívat všechny vlastnosti portálu, jaké jsou group messaging, sdílení souborů, kalendář, rozvrh hodin a diskuzní fórum. Nemůže ale student provádět žádné operace s předměty (FR-4).

Teacher

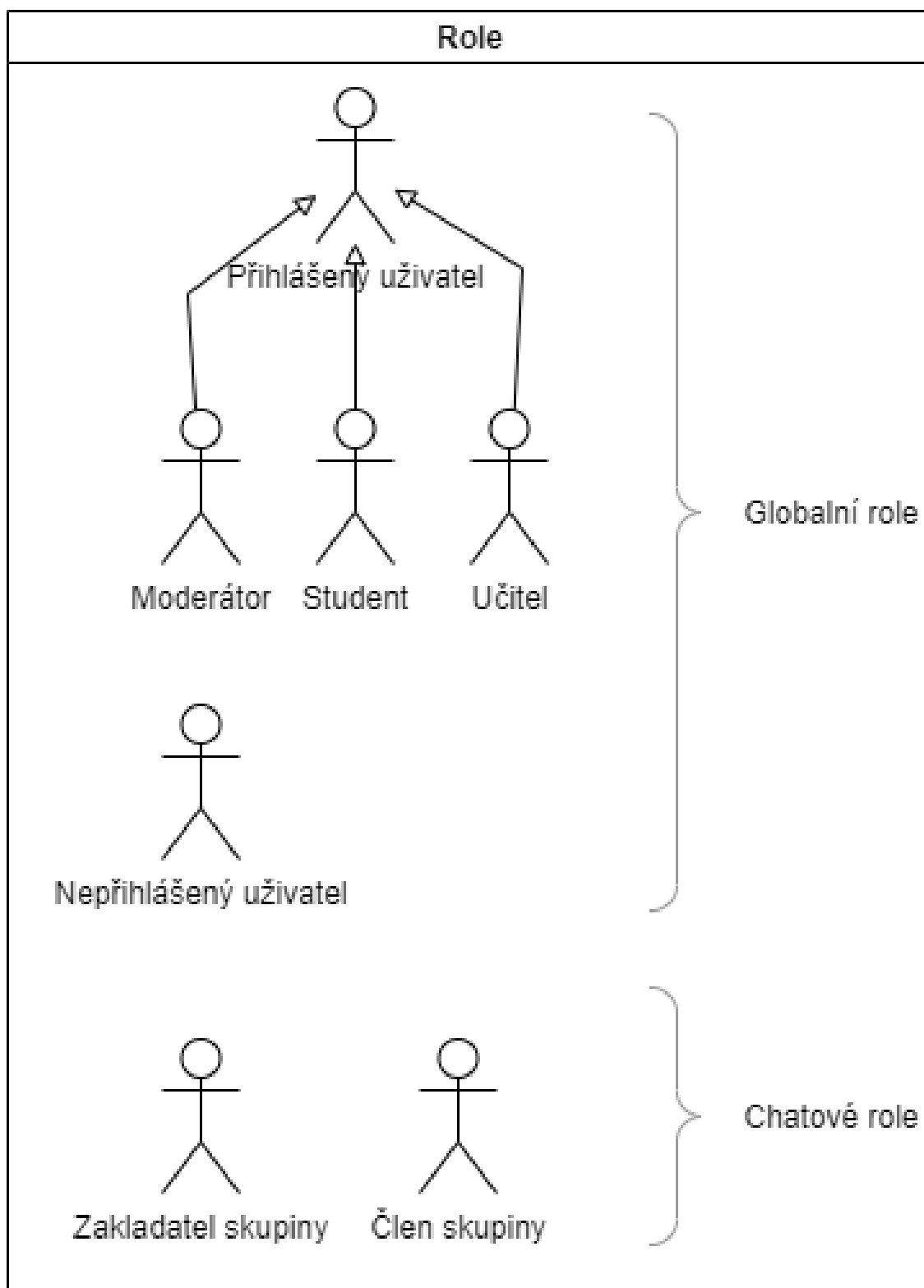
Učitel může dělat totéž co student, ale kompletně využívat všechno spojené s předměty: přidávat, měnit a odstraňovat.

Moderator

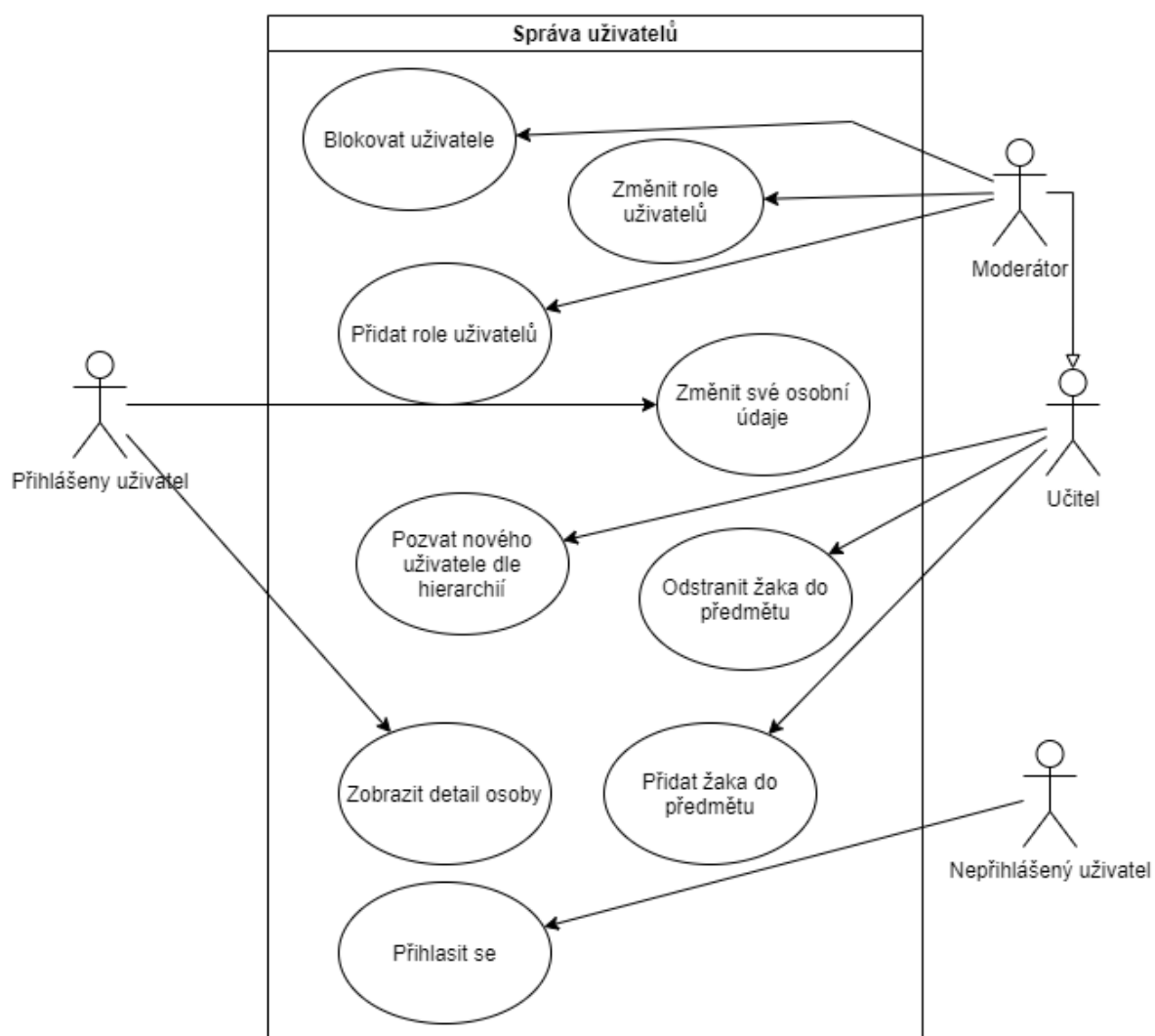
Moderator (admin) může dělat totéž co všechny dotyčné uživatele, ale navíc má možnost nastavovat role uživatelů, vytvářet nové uživatele a odstraňovat je.

11. Případy užití

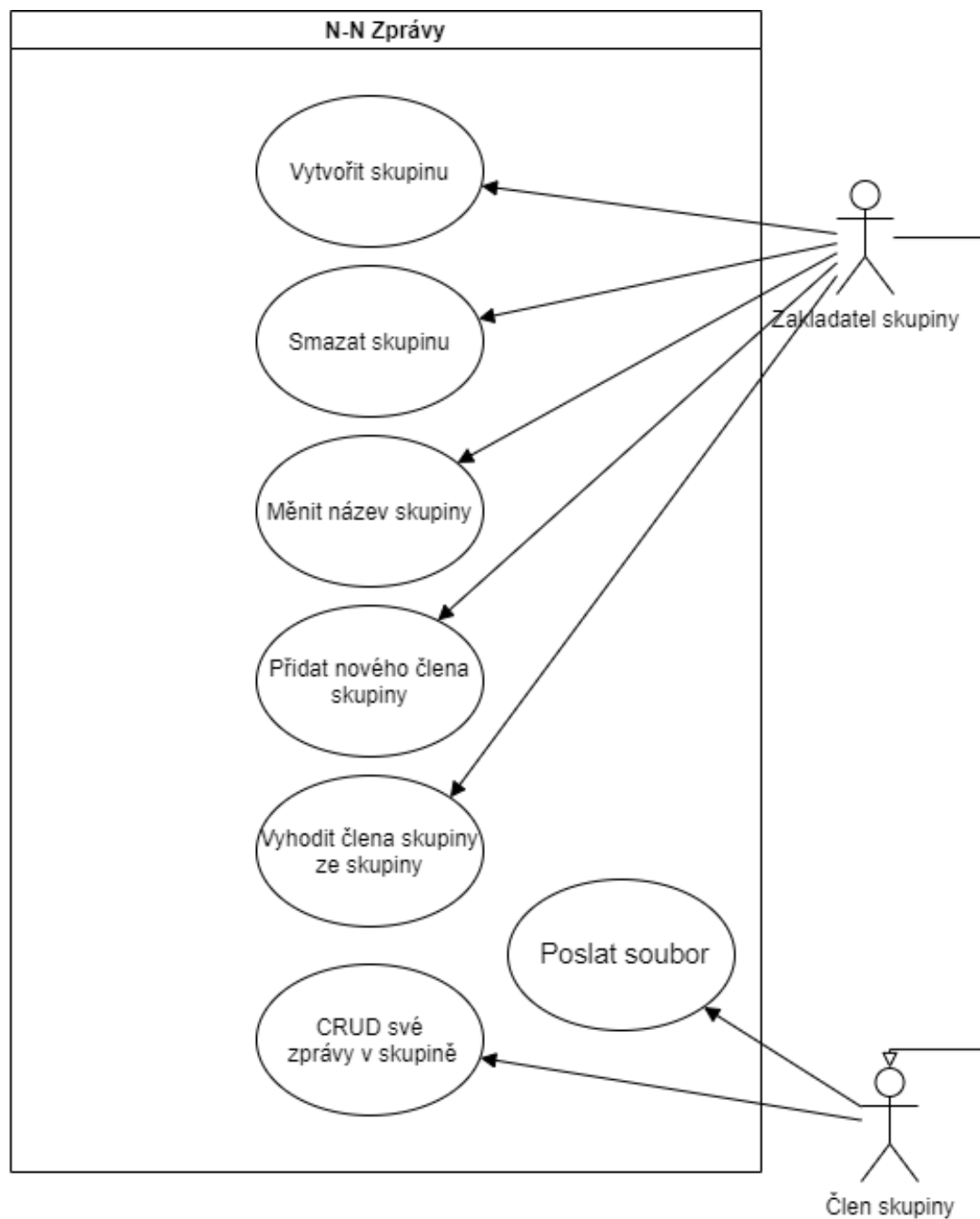
11.1. Role



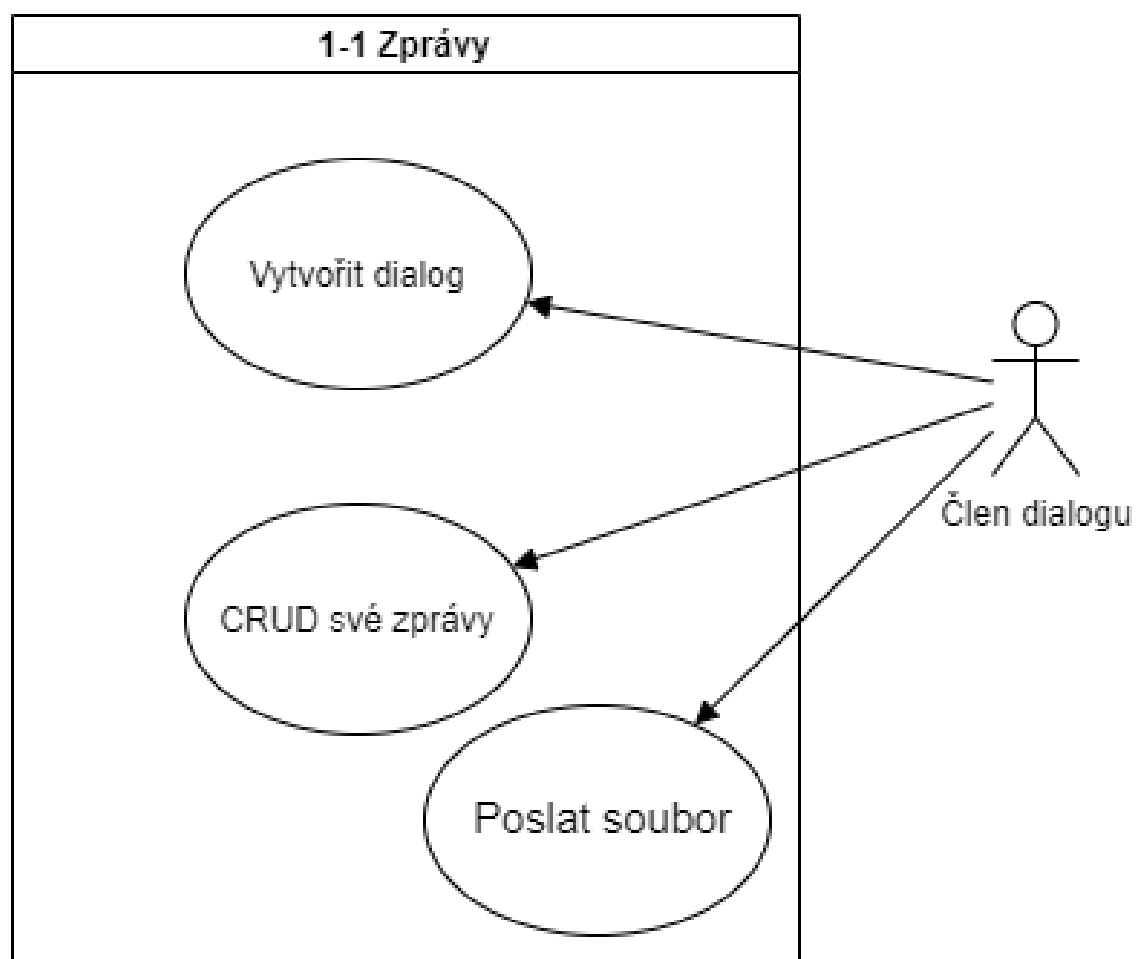
11.2. Správa uživatelů



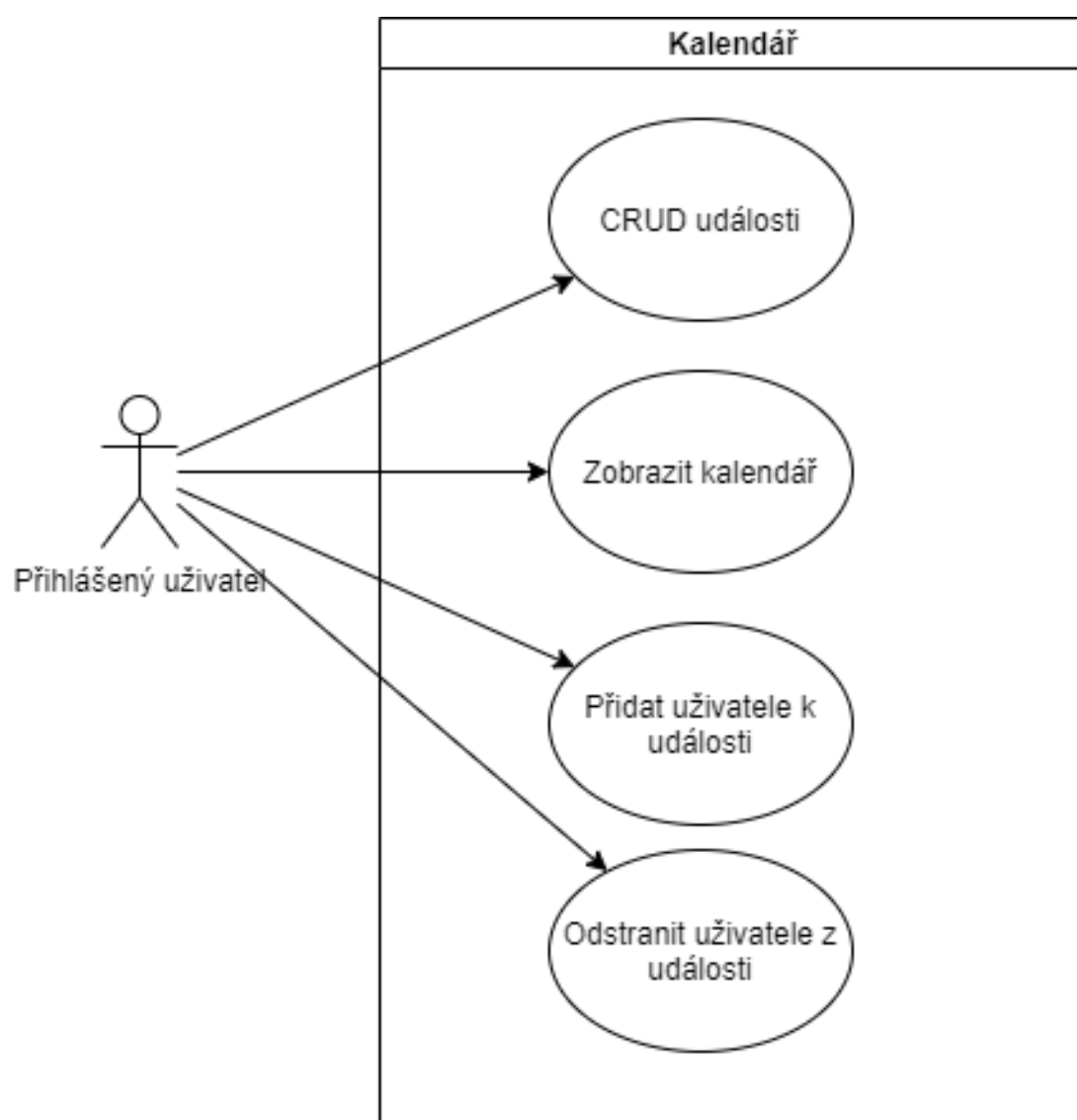
11.3. N-N Zprávy



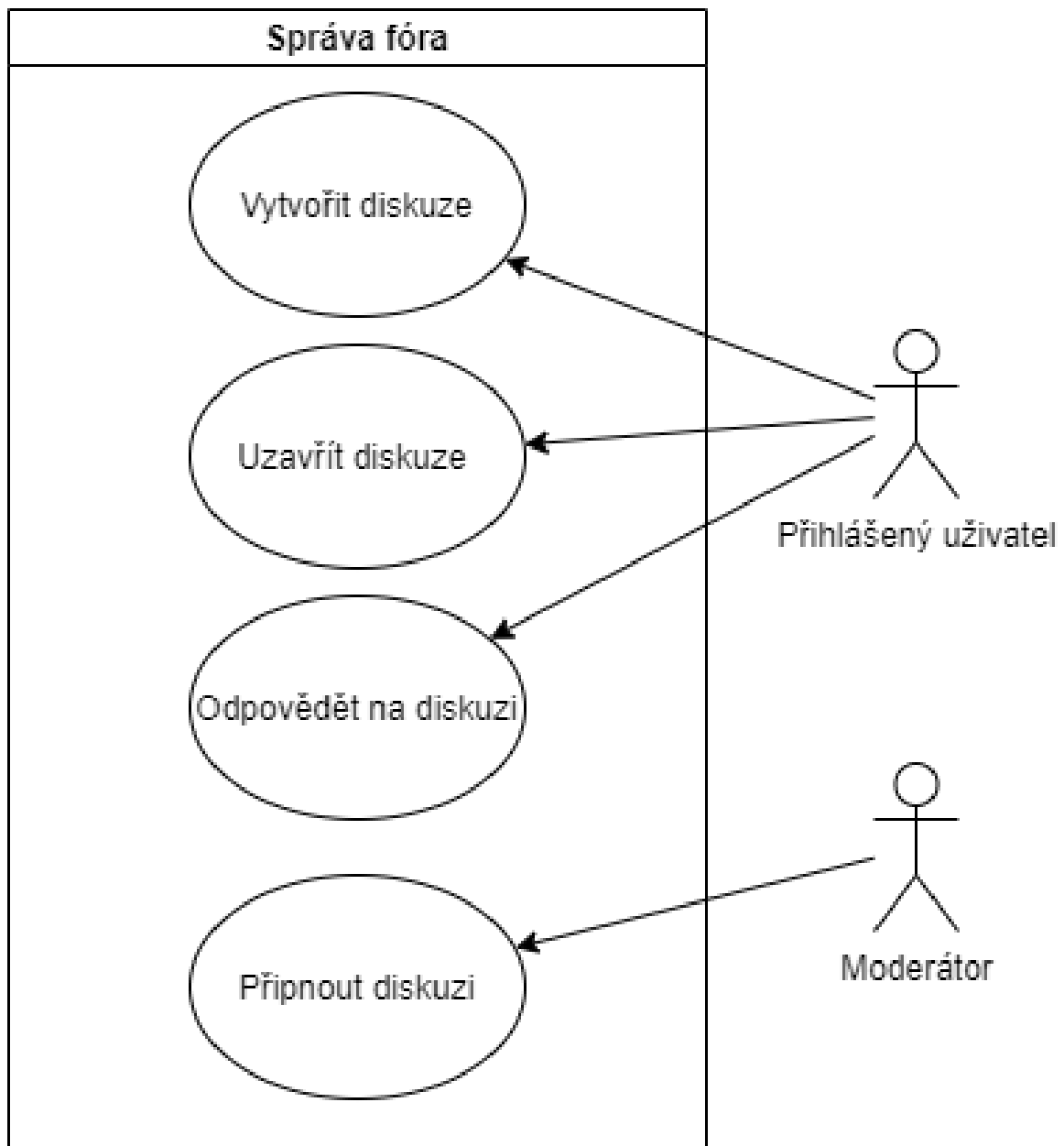
11.4. 1-1 Zprávy



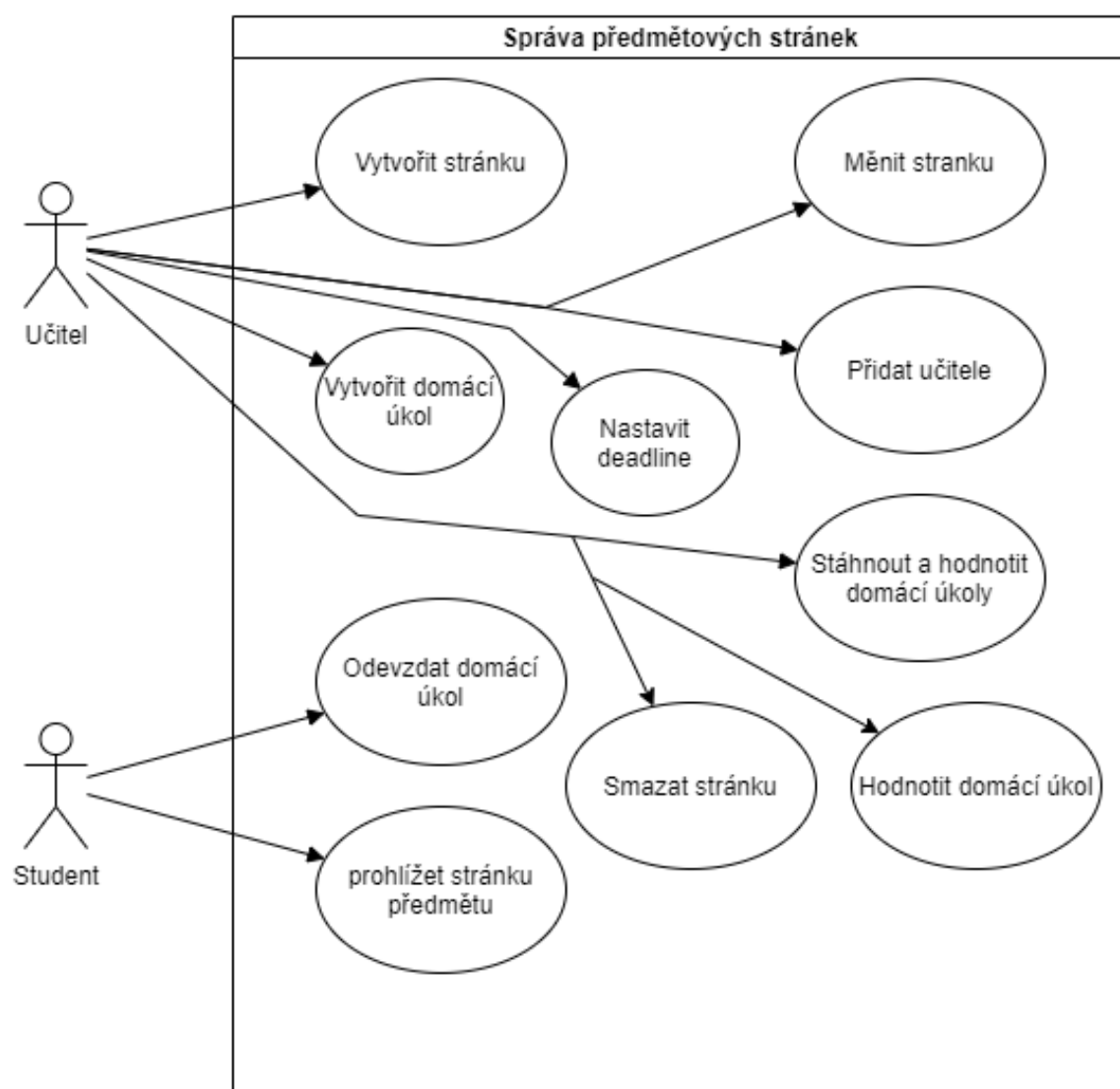
11.5. Kalendář



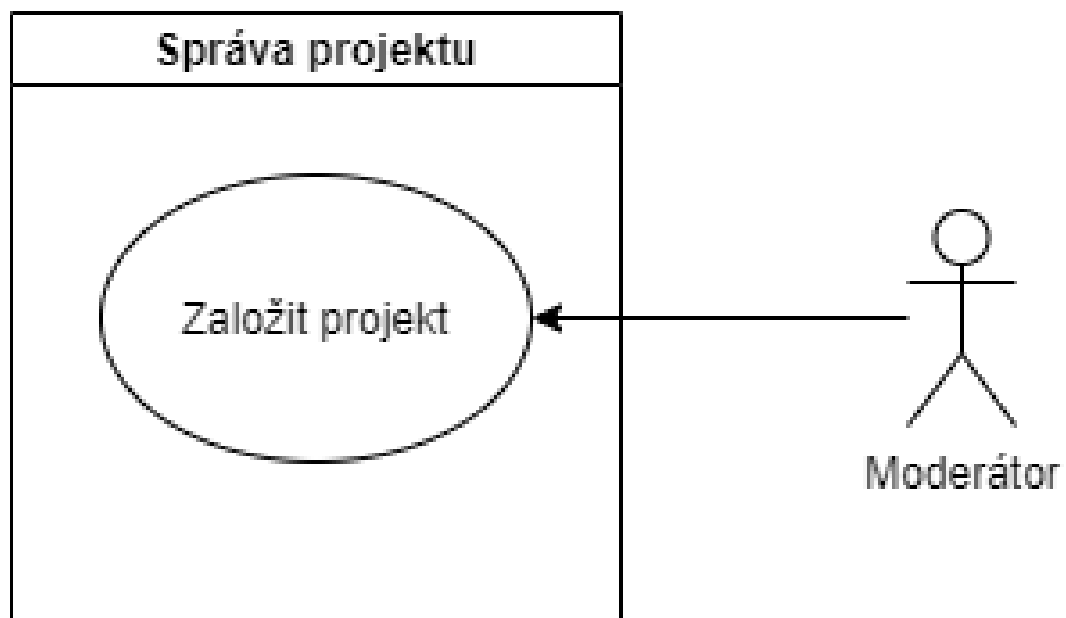
11.6. Správa fóra



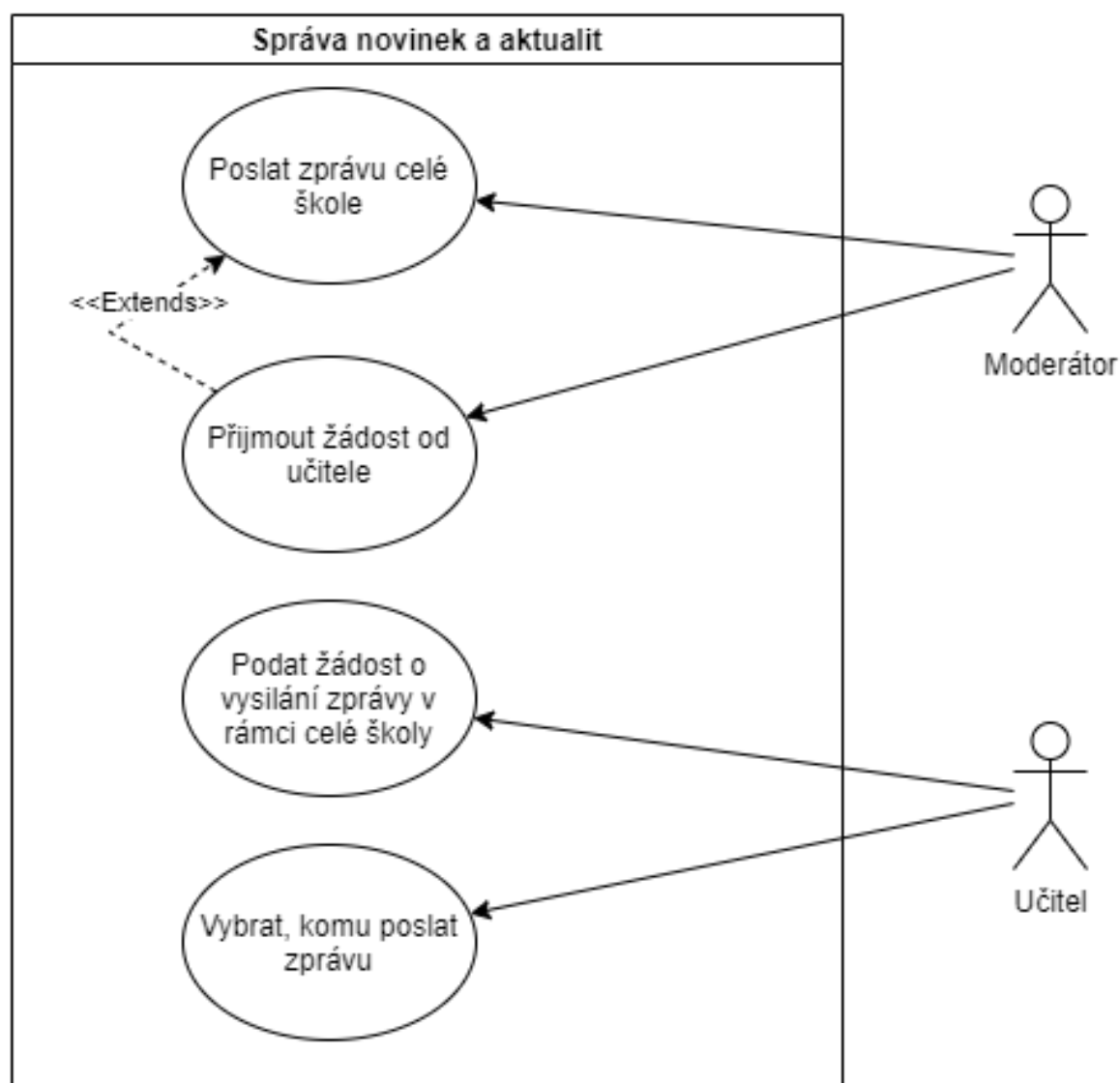
11.7. Správa předmětových stránek



11.8. Správa projektu

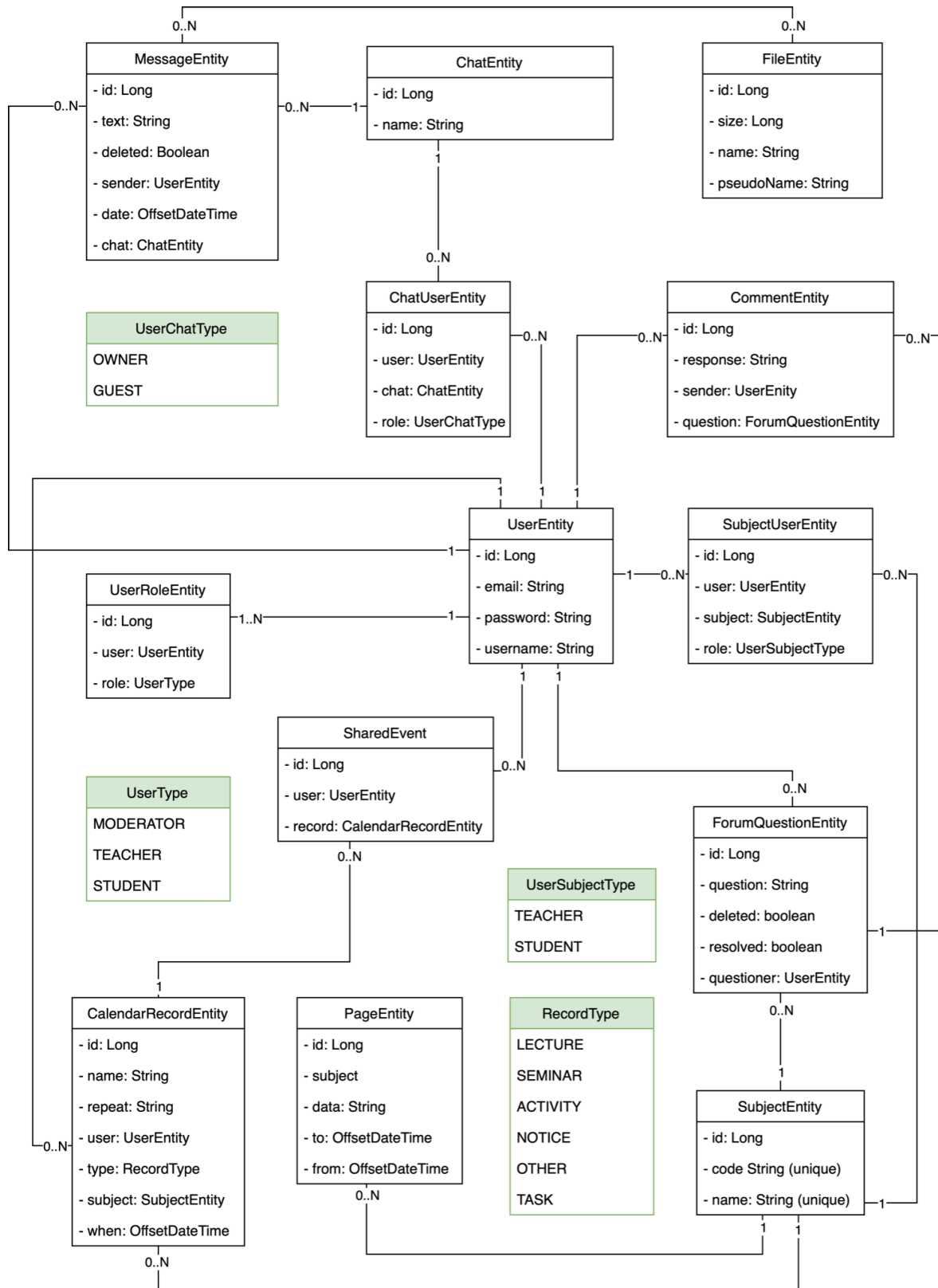


11.9. Správa novinek a aktualit

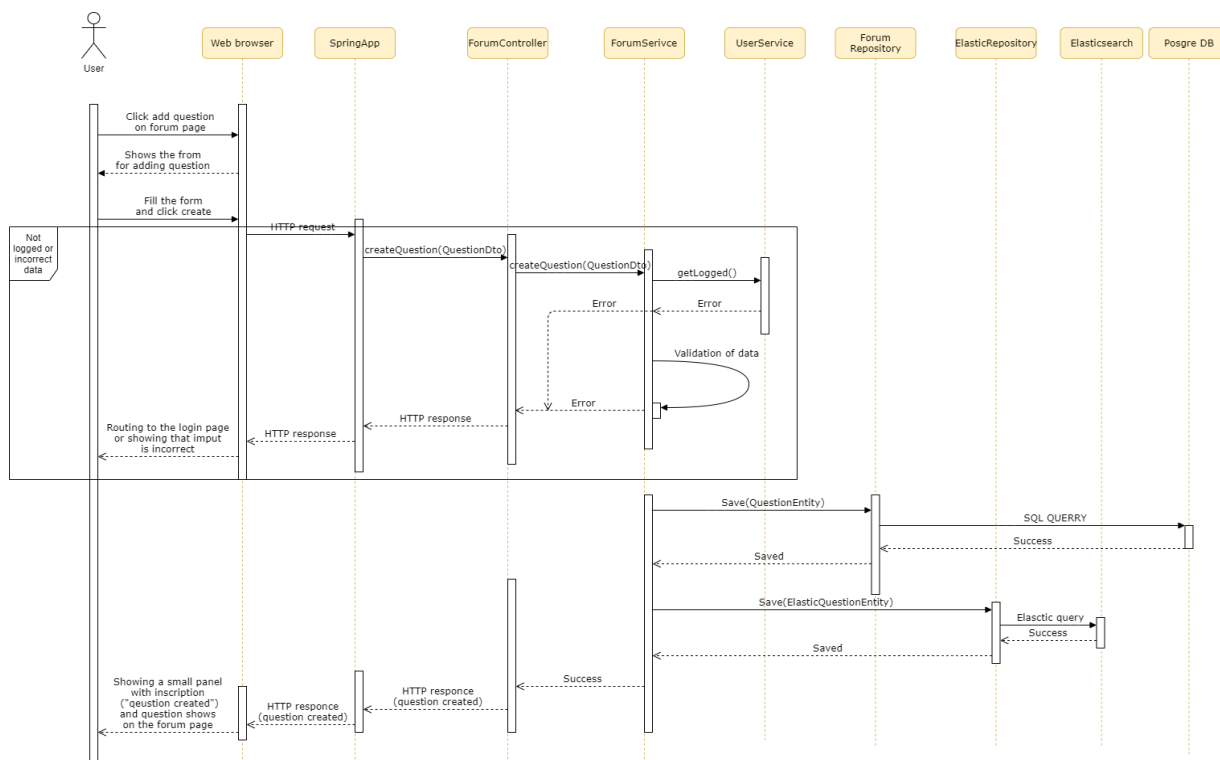


12. UML diagramy

12.1. Class diagram



12.2. Sequence diagram



Sekvenční diagram "Ukládání otázky na fóru".

13. Architektura aplikace

Naše architektura byla inspirována podle už existujících projektů, čili klient-server. Náš server bude běžet na Javě EE, bude použit spring, RESTful api a bude integrováno na frontend. Java je vybraná pro svou rychlost pro první nasazení a jednodušší škálovatelnost. Samotný frontend běží na javascriptovém enginu node.js, na kterém probíhá develop. Je použit angular a k němu open source knihovna komponent PrimeNG, které dohromady tvoří silný a intuitivní základ pro stavbu one-page aplikací. Api, respektive jeho popis a generování klienta na frontend probíhá automaticky přes OpenApi generátor a Swagger pro kompletní dokumentaci.

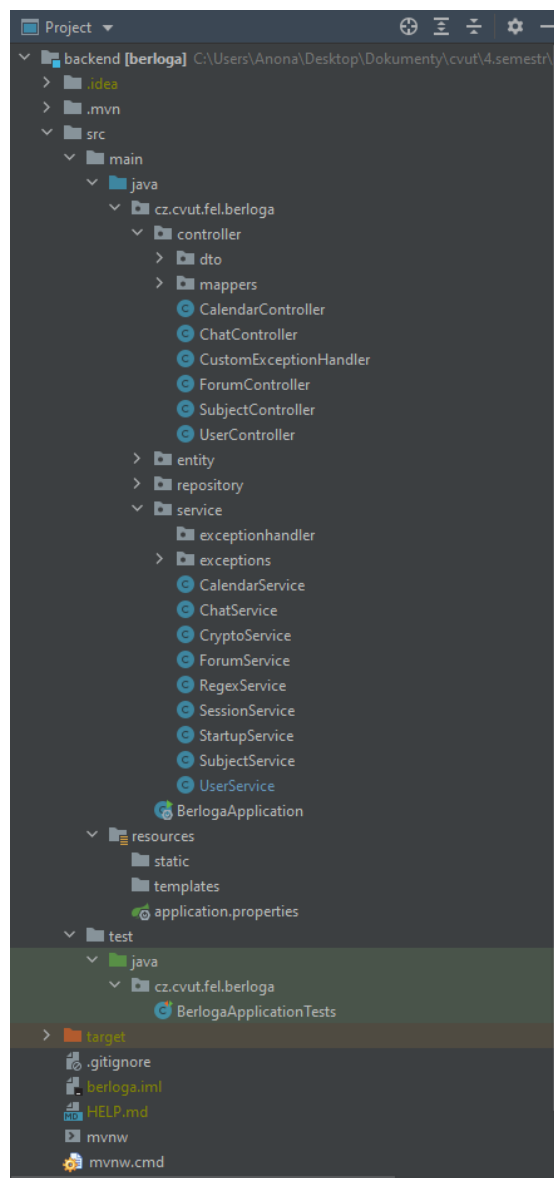
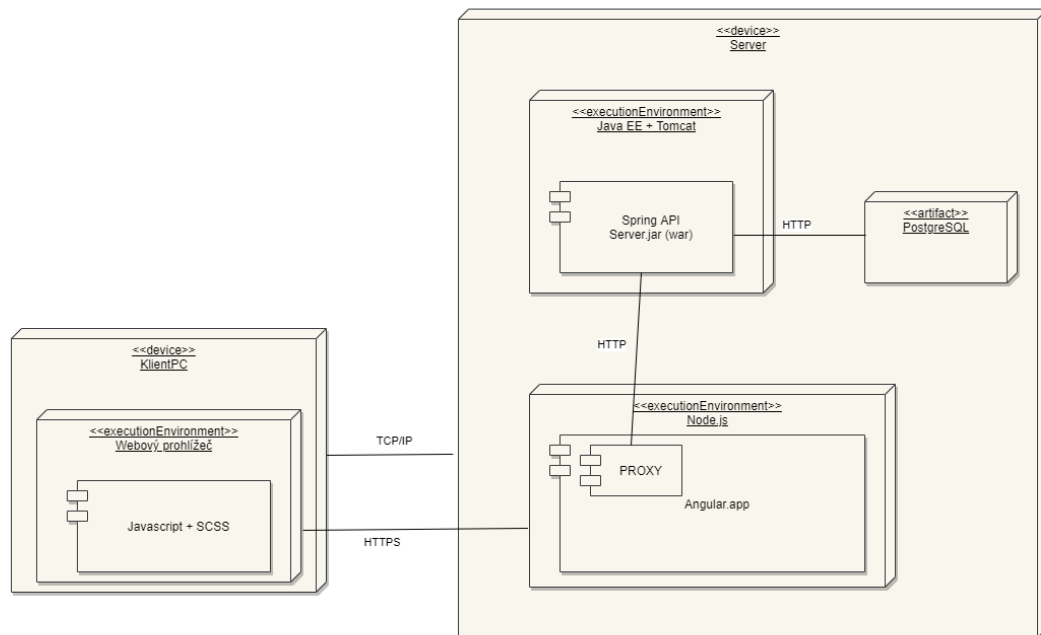


Diagram nasazení

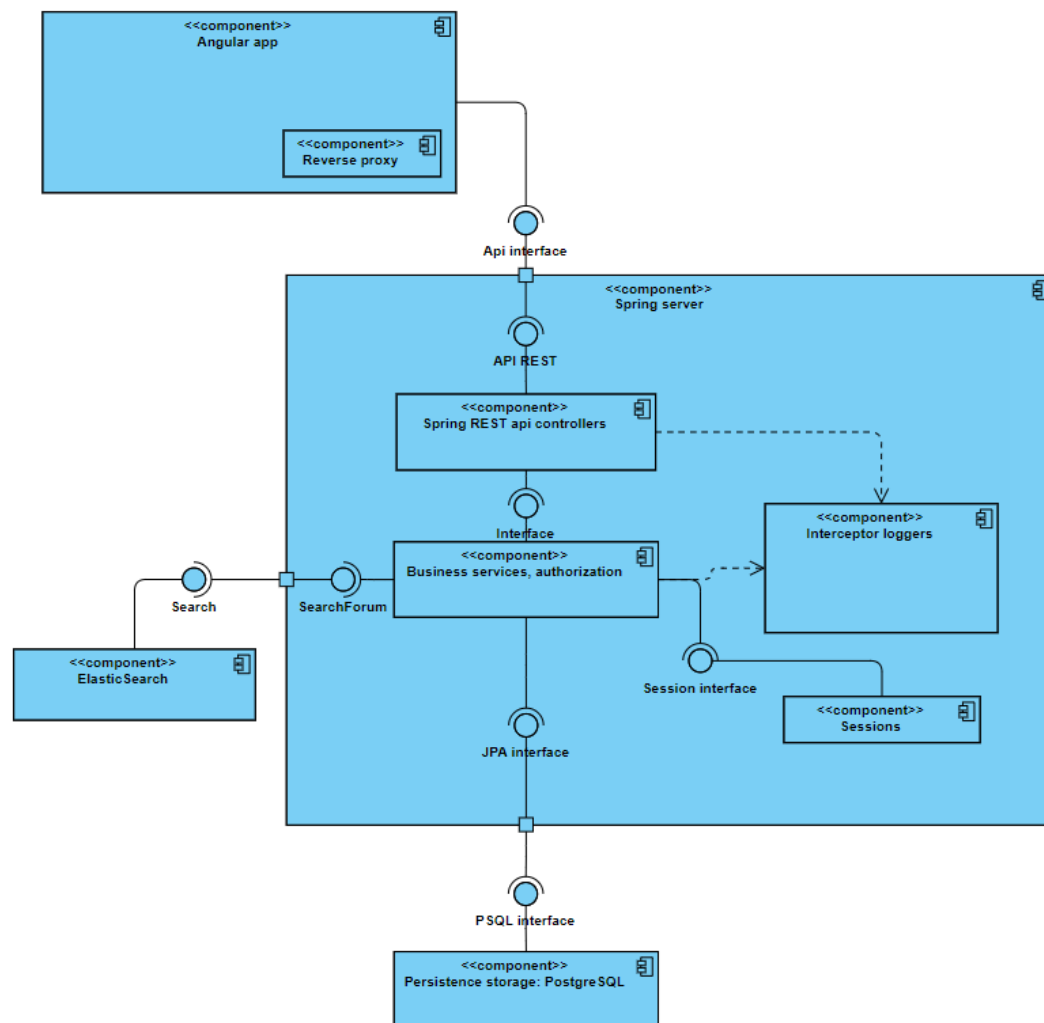
Pro spuštění backendu potřebujeme JRE nebo JDK verze 1.8 a nainstalovaný server PostgreSQL verze 9 a vyš. Pro frontend potřebujeme NODE.js server nejlépe nejnovější verze pro debug a testing vývoj. Pro produkci nám stačí zkompilovaný Angular projekt, který nám může běžet na Apachi nebo jako součást BE, kde bude pod url házet zkompilovaný script.



Popis architektury nasazení:

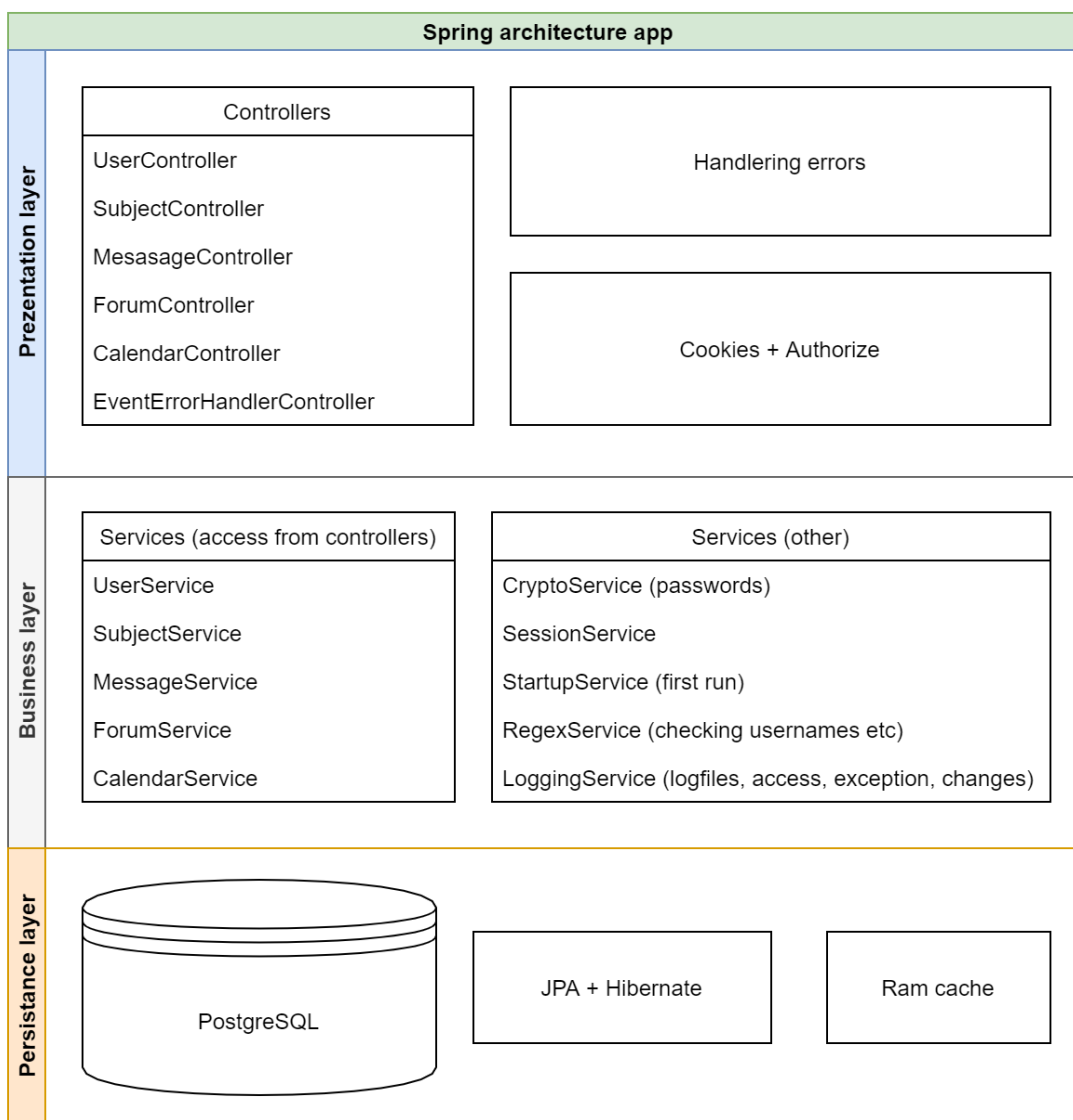
- Backend
 - Aplikace běží na Java EE s Tomcat serverem
 - Aplikace je postavená na Spring frameworku
 - Aplikace běží na portu XXXX
- Frontend
 - Frontend je postavený na Angularu v8 a vyšší, který běží na javascriptovém serveru Node.js
 - Frontend poběží na defaultním portu 80 a 443 (http a https)
 - Kvůli CORS policy, kdy webová aplikace běží na jednom portu 80 a server api na jiném XXXX, je v Node.js udělána routa (reverzní proxy) kdy url na frontend (ip:80/api) je přesměrována na (ip:XXXX/)
- U uživatele je vygenerovaný javascript který zastihuje volání na API i na webovou aplikaci samotnou

Diagram komponent



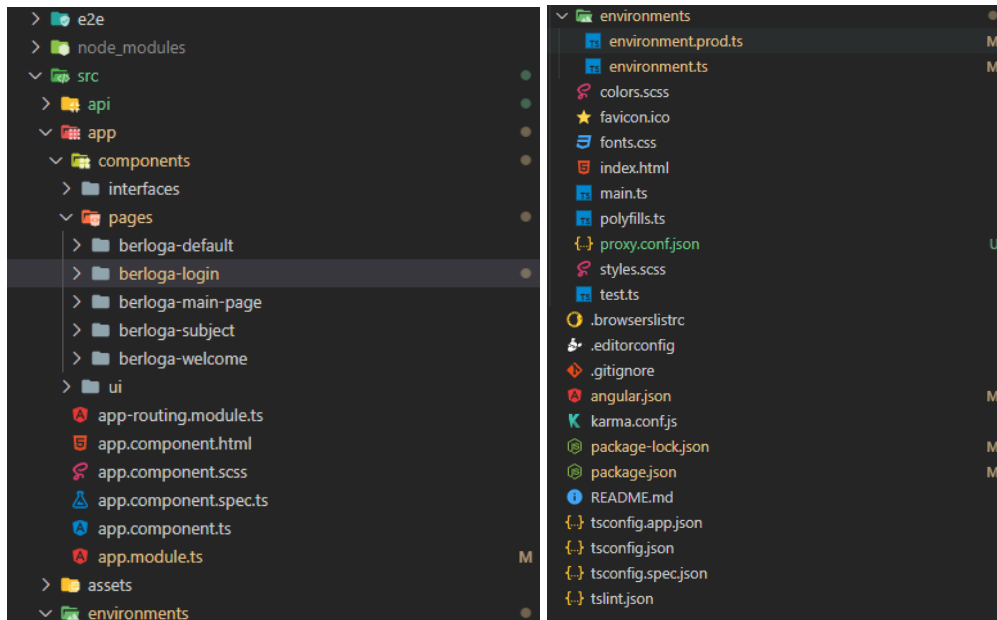
Architektura serverové části

Aplikace Springu je složena v základu z třívrstvé architektury.



Architektura frontendové části

Klasický komponentový vývoj Angularí aplikace, obsahuje rovnou i proxy server na backend.



WBS

Celý projekt je možné rozdělit na tyto dílčí procesy. Procesy jsou seřazené podle časového harmonogramu.

- Příprava zadávací dokumentace
- Příprava prováděcího projektu, analýza a plánování práce
- Vytvoření základu BE a FE
- Inicializace API na BE
- Vytvoření grafických komponent
- Definování servis a entit, implementace API
- Routing a napojování API na frontendu
- Oprava chyb na BE
- Testování (převážně FE)
- Řešení bugu, které se objevily během testování
- Finalizace a odevzdání projektu

Rozbor a výběr alternativ návrhu řešení

Vývoj vlastního produktu

Tato alternativa bude trvat déle při prvním vývoji, ale bude dimenzovaná přímo na naše požadavky, takže při následné podpoře bude daleko jednodušší vyřešení chyb a problémů. Zároveň nebudeme závislí na produktu někoho jiného.

Využití existujícího open-source produktu

Tato alternativa má rychlejší první vývoj, ale poté všechny ostatní případy jsou zde v nevýhodě, jak v případě podpory, tak změn a řešení chyb. Dále je možné, že budeme muset spojit několik projektů dohromady, abychom naplnili naše požadavky. A zde vzniká největší kámen úrazu.

14. Zdroje

Lidské zdroje

Daniil Simon - projektový manažer.

Projektový manažer má na starosti řízení celého projektu, což se jedná o kontrolu deadlinů. Řeší taky management projektu. Dále vede porady celého týmu, na kterých jsou zaměstnanci informováni o stavu projektu.

Miron Grischenko a František Zavázal - hlavní vývojáři a grafici.

Vývojáři řeší jednotlivé části projektu na nejnižší úrovni. Řeší veškeré úkoly spojené s architekturou aplikace, technologickou částí projektu, tvorbou největší části backendu a frontendu a taky grafickými komponentami aplikace.

Nazar Grigorenko - tester a koordinátor projektu.

Tester je zodpovědný za testování všech komponent aplikace. Řeší nejen testování, ale i opravu chyb v backendu. Pracoval taky na úrovni koordinace naplnění, struktury a správného formatování dokumentace projektu.

Nástroje potřebné pro vývoj

Editory

Visual Studio Code, IntelliJ IDEA

Jazyky

Java 1.8, Javascript ES6, TypeScript, PSQl

Frameworky

Spring 2.3.0, junit, Swagger API, OpenApi Angular Generator, Angular 11, Primeng

Databáze

PostgreSQL, ElasticSearch

Grafický návrh

Figma

15. Normy a standardy

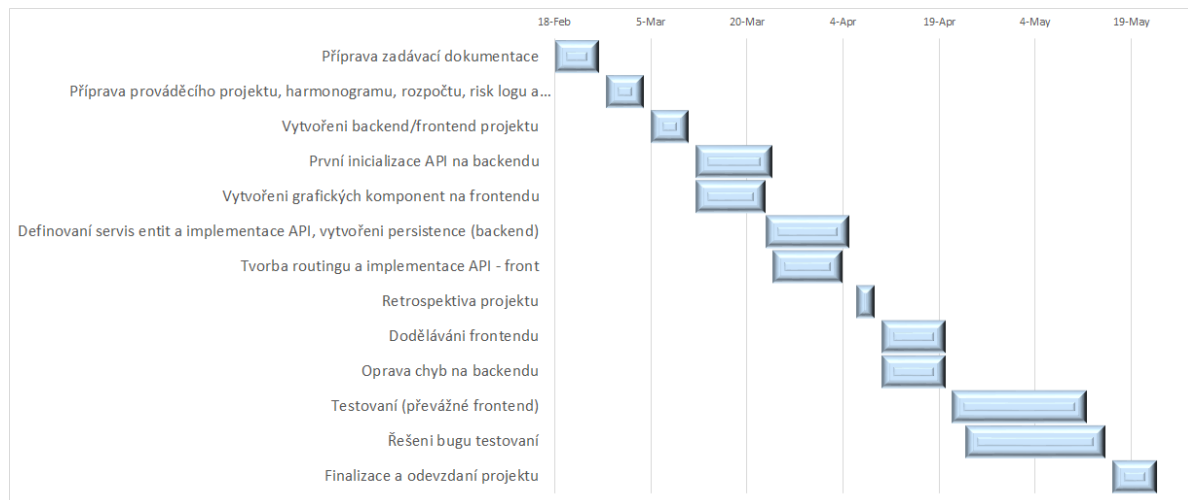
Při vytváření tohoto projektu vývojáři musí dodržovat následující standardy:

- Java (standard W3C)
- HTML 5.0 (standard W3C)
- CSS (standard W3C)
- SQL (standard W3C)
- jQuery (standard W3C)
- Hibernate (standard W3C)
- ISO/IEC 12207 (standard, který se zabývá životním cyklem softwaru)
- JavaScript + TypeScript ES6
- SHA512 (standard FIPS 180-2)

16. Matice odpovědnosti (RACI)

	Daniil Simon	Miron Grishchenko	František Zavázal	Nazar Grigorenko
Návrh týmu	x			
Harmonogram	x			
Vize projektu	x			
Analýza požadavků				x
Vývoj backendu	x	x	x	
Vývoj frontendu	x	x	x	
Testování				x
Oprava chyb		x	x	x
Dokumentace				x
Finalizace projektu	x			

17. Harmonogram GANTT



18. Analýza rizik FMEA

Procesní část výskytu	Závada	Možný dopad na zákazníka	Možná příčina vzniku	S E V	O C C	D E T	R P N	Doporučené opatření	Možný dopad na projekt	Odpovědná osoba
ALL	Odchod člena z týmu	Zpoždění dodání aplikace	Vyhození studenta ze školy	2	9	1	18		Penalizace za nedodržení termínu dodání	Daniil Simon
ALL	Špatná komunikace uvnitř týmu	Zpoždění dodání aplikace	Lidský faktor	3	1	2	6	Častější komunikace v týmu a pravidelné schůzky	Penalizace za nedodržení termínu dodání, nízká kvalita aplikace	Daniil Simon
ANLZ	Podcenění časové náročnosti a složitosti projektu	Zpoždění dodání aplikace	Špatná analýza projektu	1	5	2	10	Kvalitní analýza projektu s ohledem na schopnosti týmu	Penalizace za nedodržení termínu dodání	Nazar Grigorenko
IMPL	Nedostatečná zkušenost developerů	Zpoždění dodání aplikace	Nedostatek zkušenosti	5	6	7	210	Pečlivé prostudování použitých technologií	Penalizace za nedodržení termínu dodání, nízká kvalita aplikace	Miron Grishchenko František Závázal
TEST	Nedostatečné testování projektu	Odevzdání chybného projektu	Nedostatek nutného počtu testů	7	3	3	63	Velký počet unit testů, testování kontinuálně s vývojem	Nízká kvalita aplikace	Nazar Grigorenko
USE	Podcenění školení uživatelů	Uživatelé nebudou chtít aplikaci používat	Chybná realizace projektu	6	7	9	378	User friendly aplikace	Celkový neúspěch projektu	Miron Grishchenko František Závázal

První koeficient je určen u následku poruch a podle závažnosti jim přidělíme koeficienty ve škále od 1 do 10, kdy 10 je pro fatální, havarijní a destruktivní stavy, tedy nejhorší možný konečný stav. (přidělení tzv. „Severity“ koeficient).

Následně popisujeme a hodnotíme jednotlivé příčiny vad a podle očekávané četnosti předpokládaného výskytu opět přidělíme koeficient od 1 do 10, kde 10 je nehorší provozně a technologicky možná kombinace příčin (tzv. „Occurence“ koeficient).

Pak hodnotíme všechny možné indikátory, nebo složitější kontrolní principy a metody, které mohou reálně možné vady a chyby, př.: neshody zachytit nebo jim, pomocí automatizace předejít, resp. zastavit proces. A opět určíme koeficient od 1 do 10, kdy 10 je nejhorší představitelná porucha, tzn.: problém se neprojevuje postupně, tedy bez přirozených technických, a i lidských indikátoru a propuknutí projevu vady je skokové, náraz, s minimální možností odhalení (přidělení tzv. „Detection“ koeficient).

19. Znovupoužitelnost

Možnost opětovného použití naší aplikace není tak velká. Ale na druhou stranu, můžeme použít konfiguraci tohoto projektu pro nové, například generátor API, nastavení na application properties. Možnost opětovného použití Exception Handleru. Kromě toho, vzhledem k velkému množství různých entit, je možné je znovu použít v jiných projektech s minimálními úpravami.

20. Metriky

V projektu jsou sledované tyto metriky:

- Počet unit testů - 92
- Počet process testů - 8
- Celkový čas strávený na projektu - 320 hodin (40 MD)
- Počet lidí potřebných k projektu - 4
- Splněno 25 Use Casů
- Počet end pointu - 48

21. Plán odbavení

- Použití 1 serveru PC (jakýkoliv operační systém)
- Instalace Javy 1.8, PostgreSQL 9 a vyšší, Elasticsearch v projektu (pouze spustit elasticsearch.bat/sh)
- Ve firewallu otevřít port 80 na frontend
- Na frontend instalace HTTP serveru Apache
- Instalace reverzní proxy na CORS policy
 - pc:80 routing -> FE:port
 - pc:80/api routing -> BE:9000
- Defaultní uživatel Administrator přidává ostatní uživatele, předměty a oprávnění

22. Plán podpory

- Oprava chyb, které se objevily během používání aplikace uživateli.
- Rozšíření funkčnosti aplikace, zavedení nových technologií
- Přepracování designu aplikace

První bod je spojen s tím, že aplikace byla vyvinuta studenty v poměrně krátké době a určitě v sobě má mnoho skrytých chyb, které byly nalezeny v průběhu vývoje. Důvodem druhého bodu je také čas a neschopnost v tak krátkém čase vyvinout plnohodnotnou aplikaci s velkou funkčností. Vzhledem k tomu, že mezi námi nebyli designéři, naše aplikace má poměrně slabý design a rádi bychom ho v budoucnu změnili na modernější a adaptivnější (user friendly).

23. Vyhodnocení

Účastník projektu	Vyhodnocení
Daniil Simon	V tomto projektu jsem se naučil spousta nových technologií, ještě jsem nikdy neměl zkušenosti s reálným vývojem v projektu. Byl jsem na pozici projektového manažera a pro mě to byl takový mini start up kde jsme byli primárně zaměřeni na získávání zkušenosti v různých technologiích a byla to taková svoboda v tom, co děláme, já jsem například naučil se dobře angular, taky jsem vyrostl ve springu, a hlavně jsem dostal dobrou zkušenosti s vedením týmu.
Nazar Grigorenko	Zjistil jsem že, vedení skutečného projektu není vůbec jednoduchá činnost. Díky tomuto projektu jsem se skutečně naučil nejen pojmy různých technologií a design patternů, ale jak to v reálné práci používat. Kvůli tomu, že jsem za hlavní úkol dostal testování projektu, pořádně jsem se naučil nové způsoby testování a technologie na vývoj backendu, protože jsem všechny chyby podle testů opravoval. K tomu bych chtěl ještě přidat, že jsem konečně pochopil jak je důležité dávat velký pozor na rozdělení práce mezi členy týmu a hlavně na svůj vlastní time-management
František Zavázal	Tento projekt je jeden z největších, které jsem zatím ve škole měl. Naučil mě spoustu nových technologií, a hlavně jsem si vyzkoušel finální export projektu, a jak bychom to nasazovali u zákazníka. Dále jsem se naučil pomáhat kolegům, a poskytovat jim pomoc v nových technologiích, které já znala oni ne. A dost jsem se naučil při nasazování na heroku, tolik nervů s tím, ale pak ten úspěch byl příjemný
Miron Grishchenko	V tomto projektu jsem výrazně rozšířil své znalosti o poměrně velkém množství aspektů. Například, naučil jsem se poměrně velké množství technologií, jako jsou Angular, Elasticsearch, Kafka, PrimeNG. Dostal více hluboké znalosti v oblasti vedení projektů a naučil lépe plánovat svou práci. Navíc to byl můj největší projekt, na kterém jsem se podílel, a je to pro mě neocenitelná praxe