

SmallCat Software

Třídni kniha

Semestrální projekt NSS



Dávid Michal Bulko, Václav Smítka, David Hájek, Lukáš
Hofmann, Vladimír Vasko

24.5.2019

Obsah

1. Strategický záměr (stav "TO BE")	3
2. Obchodní přínos	4
3. Stav "As IS"	5
4. Analýza SWOT.....	6
5. Analýza 5 F.....	7
5.1. Konkurenční rivalita	7
5.2. Vyjednávací síla dodavatele.....	7
5.3. Výkonná síla zákazníků	7
5.4. Hrozba nových účastníků.....	7
5.5. Hrozba náhradních výrobků.....	7
6. Analýza PEST(E)	8
6.1. Politické a legislativní faktory	8
6.2. Ekonomické faktory	8
6.3. Sociální a demografické faktory.....	8
6.4. Technologické faktory.....	8
7. Funkční požadavky	9
8. Nefunkční požadavky	10
9. Požadavky na zavedení řešení.....	10
10. Seznam uživatelů.....	11
10.1. Administrátor.....	11
10.2. Učitel.....	11
10.3. Žák.....	11
11. Případy užití.....	12
12. Diagram nasazení	13
13. Diagram komponent.....	14
14. Rozbor a výběr alternativ návrhu řešení	15
14.1. Frontend	15
14.2. Backend.....	15
Databáze	15
14.3. Server	16
15. WBS – rozdělení projektu na dílčí procesy	17
16. Zdroje.....	18
16.1. Lidské	18
16.2. Finanční.....	18
16.3. Časové.....	18
16.4. Technické	18

17. Normy a standardy	19
17.1. Standart kódu na frontendu	19
17.2. Standart kódu na backendu	19
17.3. REST standart	19
18. Matice zodpovědnosti	20
19. Harmonogram GANTT	21
20. Analýza rizik FMEA	22
21. Znovupoužitelnost	25
22. Přílohy	26
22.1. Detailní popis vícevrstvé architektury	26
22.1.1. Prezentační vrstva	26
22.1.2. Webová vrstva	26
22.1.3. Aplikační vrstva	26
22.1.4. Perzistentní vrstva	26
22.1.5. Databázová vrstva	26
23. Návrh ER diagramu	27
24. Návrh UML diagramu	28
25. Seznam obrázků	29

1. Strategický záměr (stav "TO BE")

Nás systém bude poskytovat místo pro studenty a učitele, aby měli přehled o studiu na jednom místě. Naší předností před konkurencí jsou přídavné moduly, které se dají přizpůsobit každé škole.

Student bude moci vidět své předměty a známky bez potřeby hmotné formy. Učitel bude moci vidět předměty, které učí, vypisovat známky a mít přehled o prospěchu jednotlivých studentů.

Výstup tohoto projektu bude nasazen na střední technické škole Františka Křížíka na Praze 3, jež je zadavatelem této zakázky.

2. Obchodní přínos

Naše firma chce se svým produktem vyjít svým zákazníkům maximálně vstříc. Na základě setkání a konzultací společně do specifikovat očekávané kvality produktu. Mezi přidané hodnoty bychom zařadili mimo jiné vlastní nedávnou zkušenost s těmito systémy z pohledu uživatelů a znalost prostředí středních a základních škol. Tímto bychom rádi škole poskytl odbornou pomoc při nasazování našeho řešení.

Náš produkt ušetří čas jak vedení školy, učitelům tak žákům. Čas ušetří hlavně díky minimalistickému řešení celé problematiky. Náš produkt bude modularizován, což umožní školám ušetřit a zvýší variabilitu. Jednotlivé moduly bude následně po jejich výběru možné námi editovat a upravit na míru přání.

V dnešní době na trhu je sice již spousta produktů tohoto druhu, například Škola Online, iŠkola, Etřídnice, ascTimetables nebo Bakaláři. Tyto velké informační systémy pokrývají všechny potřeby škol a mají kladné hodnocení. Oproti těm to gigantům chceme nabídnout větší osobitost a možnost školám vnést svého ducha i do těchto virtuálních prostor školy.

3. Stav "As IS"

Jsme tým lidí, který se dal dohromady nedávno. Každý má povinnosti i mimo tento tým, takže náš projekt, který je zároveň 1. projektem tohoto týmu, není práce na plný úvazek žádného z členů.

Ačkoliv všichni členové týmu mají silnou motivaci pracovat, jeden člen to „zabalil“. To se ale samozřejmě netýká žádného dalšího člena, jelikož jejich smysl pro povinnost je vysoká.

4. Analýza SWOT

Strenghts	Weakness
Silná motivace	Rozpad týmu
Chuť zkoumat nové technologie	Nedostatek času programátorů
Opportunities	Threats
Jakožto menší firma, můžeme nabídnout osobnější přístup	Silná konkurence v oblasti
Jádro systému se dá přenést i na další školy => menší časová náročnost a více zkušeností na další projekty a jejich moduly.	Školy budou chtít spíše „klasiku která funguje“ než naše řešení
Náš systém je jednoduše rozšiřitelný díky modulům	Jakožto malá firma nemáme jistotu, že zákazníci budou chtít náš produkt (nebo o něm vědět)

5. Analýza 5 F

V následující analýze se zaměříme na to, jak by si náš produkt stál v českých podmínkách pro střední a základní školy.

5.1. Konkurenční rivalita

Na českém trhu v dnešní době působí již několik firem se svými produkty, které se snaží obsáhnout veškerou poptávku. Mezi největší hráče patří celé informační systémy, které mimo jiné obsahují evidenci známek a správu rozvrhů. Sem řadíme například Školu Online, iŠkolu, Etřídnicí, ascTimetables nebo Bakaláře. Na trhu tedy panuje intenzivní rivalita. Prostor, kam se ale stále dá cílit, jsou menší školy, kde je dosah těchto gigantů minimální.

5.2. Vyjednávací síla dodavatele

Náš produkt je stavěn na zelené louce, což umožňuje větší variabilitu ve volbě technologií a jejich použití. Z tohoto důvodu budou mít dodavatelé minimální vyjednávací sílu. Největší vliv mezi dodavateli bude mít poskytovatel serveru.

5.3. Výkonná síla zákazníků

Z důvodů poměrně krátkého působení naší firmy na trhu je snaha prodat maximální. Proto budeme chtít vyjít zákazníkům vstříc v jejich námětech. Vyjednávací síla zákazníků bude vysoká.

5.4. Hrozba nových účastníků

Z pohledu naší firmy je hrozba nových účastníků na trhu poměrně vysoká. Nemáme zatím vybudované jméno a příchod produktů od větších firem cílených pro malé školy na trh by mohl způsobit komplikace. Na základě rozvíjejícího se trhu se softwarem je možné očekávat i příchod konkurence ze zahraničí, kde se budeme snažit minimalizovat tuto hrozbu osobním kontaktem a znalostí prostředí českých škol.

5.5. Hrozba náhradních výrobků

Na základě znalostí z předmětu NSS si myslíme, že je vysoce nepravděpodobné, aby v případě zakoupení našeho produktu vznikla později iniciativa ze strany školy přestoupit ke konkurenci. Změna informačního systému totiž nelze být provedena pouze částečně, ale je nutné ji vykonat kompletní, což je relativně náročný a zdoluhavý proces. Vzhledem k tomu, že se snažíme naše produkty co nejvíce personalizovat klientům na míru, by k takovým situacím mělo docházet jen minimálně. Rovněž z finanční situace ve školství je předpokladatelné, že než škola bude moci investovat do nového systému, my budeme již plně schopni na jejich nespokojenost zareagovat.

6. Analýza PEST(E)

Zde zanalyzujeme vnější vlivy naší nedávno vzniklé firmy skládající se z vysokoškoláků. Jsme nadšenci a chceme své nově nabyté vědomosti nabídnout světu.

6.1. Politické a legislativní faktory

Legislativa České republiky nijak více neomezuje prodej softwaru. Týkají se ho stejná pravidla jako jiných produktů ošetřených smluvně. Značný vliv na legislativu má politika Evropské Unie. Mezi největší hrozby se řadí GDPR a článek 13, jenž je součástí Evropské směrnice. Dalším faktorem je možný přeprdej a diskutabilní volba podpory.

6.2. Ekonomické faktory

Vývoj HDP České republiky za poslední rok vzrostl o 3 %. Ekonomika je na vzestupu. Investice do softwaru jsou čím dál vyšší a zákazníci jsou za ně ochotni více platit. Naše firma bude cílit na střední a základní školy, kam dle Českého statistického úřadu rovněž teče rok od roku více peněz. Celkově za všechny školy to pro rok 2017 činí 194 823 miliard korun. V roce 2017 bylo v České republice 4155 základních a 1308 středních škol.

6.3. Sociální a demografické faktory

V dnešní době je velmi populární vše elektronizovat a automatizovat procesy notifikace, analýzy a dalších, které se často opakují. Těchto procesů je ve školách celá řada. Mezi školami vznikl trend zavádění elektronických žákovských knížek pro efektivnější komunikaci nejen se žáky, ale i s rodiči. Je třeba zmínit i vliv poptávky ze strany rodičů, kteří usilují o větší přehled o výsledcích svých dětí a dění ve škole.

6.4. Technologické faktory

Vzhledem k odvětví podnikání je na první pohled zřejmé, že technologický faktor je zde naprosto klíčový. V dnešní době je velmi rychlý vývoj nových a dokonalejších technologií. Pro nové hráče na trhu je naprosto klíčové jejich použití z důvodu rychlého stárnutí. Naše firma nebude výjimkou a bude se snažit aplikovat nejnovější technologie.

7. Funkční požadavky

FRO 1 - Systém bude umožňovat uživatelům se přihlašovat do uživatelské sekce.

FRO 2 - Systém bude umožňovat přidávat do systému nové uživatele.

FRO 3 - Systém bude umožňovat změnu údajů uživatelů: hesla, mailu, adresy.

FRO 4 - Systém bude umožňovat tvorbu nových učitelů, žáků a předmětů.

FRA 1 - Systém bude umožňovat vedení školy vypnout celý systém.

FRA 2 - Systém bude umožňovat vedení školy zapnout celý systém.

FRA 3 - Systém bude umožňovat vedení školy vypsat veškeré informace o daném uživateli systému.

FRA 4 - Systém bude umožňovat vedení školy smazat libovolného uživatele.

FRA 5 - Systém bude umožňovat vedení školy vypsat veškeré informace o daném předmětu.

FRA 6 - Systém bude umožňovat vedení školy přidat na předmět konkrétního žáka.

FRA 7 - Systém bude umožňovat vedení školy odebrat na předmět konkrétního žáka.

FRA 8 - Systém bude umožňovat vedení školy přiřadit na předmět konkrétního učitele.

FRA 9 - Systém bude umožňovat vedení školy odebrat na předmět konkrétního učitele.

FRA 10 - Systém bude umožňovat vedení školy vytvořit předmět.

FRA 11 - Systém bude umožňovat vedení školy smazat předmět.

FRT 1 - Systém bude umožňovat učiteli napsat známku konkrétnímu žákovi.

FRT 2 - Systém bude umožňovat učiteli přepsat známku konkrétnímu žákovi.

FRT 3 - Systém bude umožňovat učiteli zapsat docházku konkrétnímu žákovi.

FRT 4 - Systém bude umožňovat učiteli vypsat veškeré informace o žákovi, který je zapsán na jeho předmět.

FRT 5 - Systém bude umožňovat učiteli vypsat svůj rozvrh všech svých vyučovaných hodin v daném týdnu.

FRT 6 - Systém bude umožňovat učiteli zobrazit profil svých kolegů.

FRT 7 - Systém bude umožňovat učiteli napsat komentář k vytvořené známce.

FRS 1 - Systém bude umožňovat žákovi vypsat své známky z konkrétního předmětu.

FRS 2 - Systém bude umožňovat žákovi vypsat profil konkrétního vyučujícího.

FRS 3 - Systém bude umožňovat žákovi vypsat své absence.

FRS 4 - Systém bude umožňovat žákovi vypsat profily všech žáků, které jsou ve stejné třídě.

FRS 5 - Systém bude umožňovat žákovi vypsat průměr z daného předmětu.

8. Nefunkční požadavky

NFR 1 - Bude zhotovena k systému dokumentace obsahující: instalační příručku, popis celého systému, použité technologie, popis uživatelského prostředí.

NFR 2 - Systém bude plně funkční v prohlížečích Firefox, Chrome, Opera, Safari.

NFR 3 - Systém bude splňovat zásady GDPR.

NFR 4 - Systém poběží na serverech školy.

NFR 5 - Systém bude v případě vzniklé chyby do 24 hodin opraven dodavatelskou společností.

NFR 6 - Systém bude rozšiřitelný o další funkcionalitu.

NFR 7 - Systém bude schopen spravovat informace alespoň o 1000 žáků, 100 vyučujících, 2000 předmětů.

NFR 8 - Systém bude schopen paralelně komunikovat s 500 uživateli.

NFR 9 - Veškerá práva bude vlastnit škola provozující systém.

NFR 10 - Systém bude intuitivní pro uživatele.

9. Požadavky na zavedení řešení

TR 1 - Systém bude potřebovat testovací server pro testování před nasazením.

TR 2 - Systém umožní migraci dat o studentech ze školního systému.

TR 3 - Tvůrci bude zorganizována zaškolovací přednáška pro vyučující.

TR 4 - Bude dodána k systému smlouva na support.

10. Seznam uživatelů

Náš produkt bude cíleně vytvořen pro střední a základní školy. Na základě toho bude též kopírovat hlavní role těchto institucí.

10.1. Administrátor

Tento typ uživatele bude mít všechna práva k systému a bude zodpovědný za jeho fungování. Bude spravovat ostatní uživatele, třídy, ale i rozvrhy a tvorbu předmětů a z toho jednotlivé hodiny. Jeho hlavním úkolem bude vytvářet strukturu nad kterou budou ostatní uživatelé pracovat.

10.2. Učitel

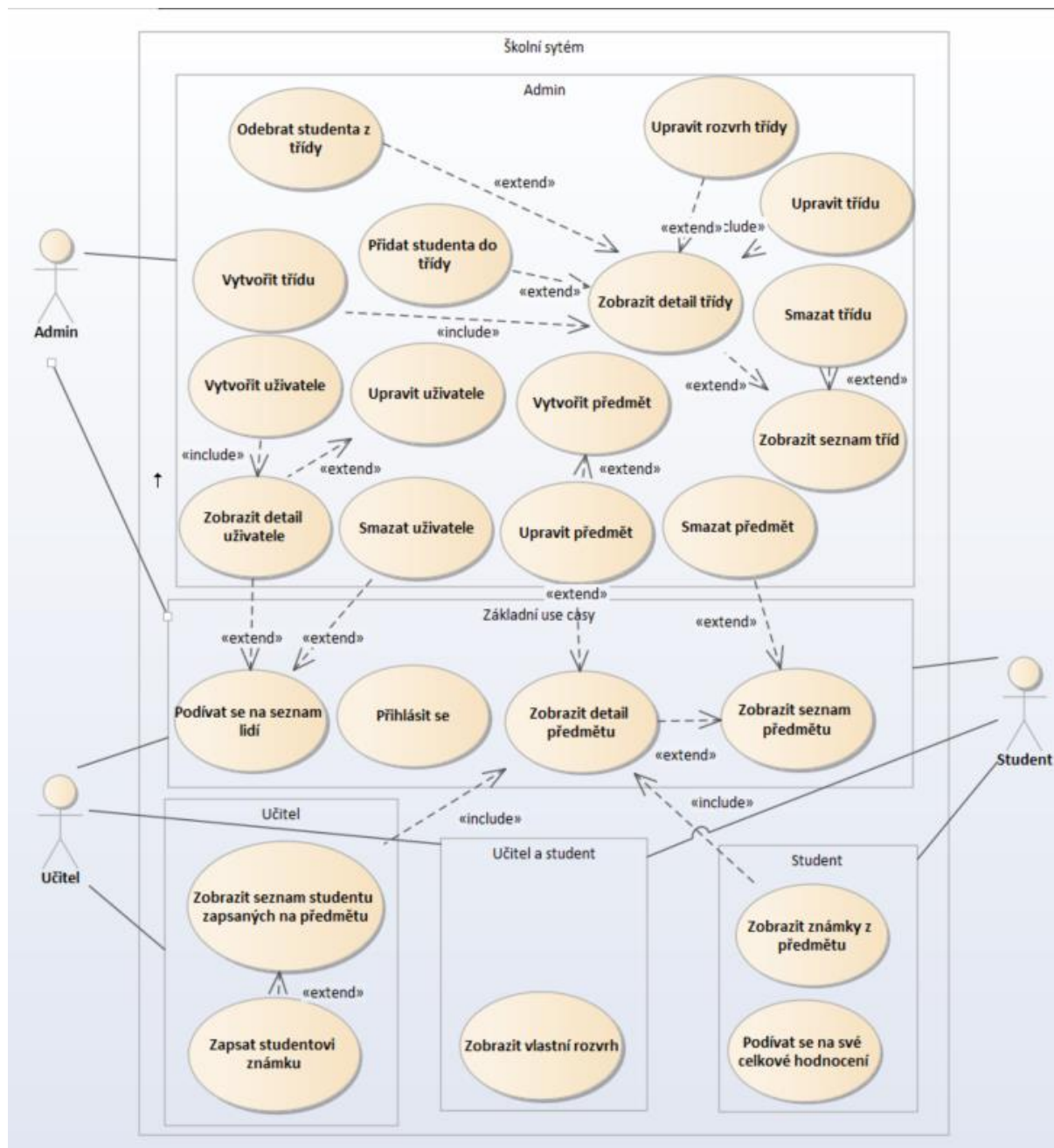
Druhou rolí systému bude učitel, který dostane administrátorem vytvořený rozvrh svých hodin. Bude mít možnost náhledu do svých hodin, kde uvidí žáky, které bude na dané hodině vyučovat a možnost vytvořit test a napsat k němu žákům známky. Dané známky bude moci i zpětně editovat.

10.3. Žák

Poslední rolí systému bude žák, který bude mít pouze možnost číst jednotlivé informace. Bude mít přístup ke svým známkám a vyhledávání profilů ostatních uživatelů na škole.

11. Případy užití

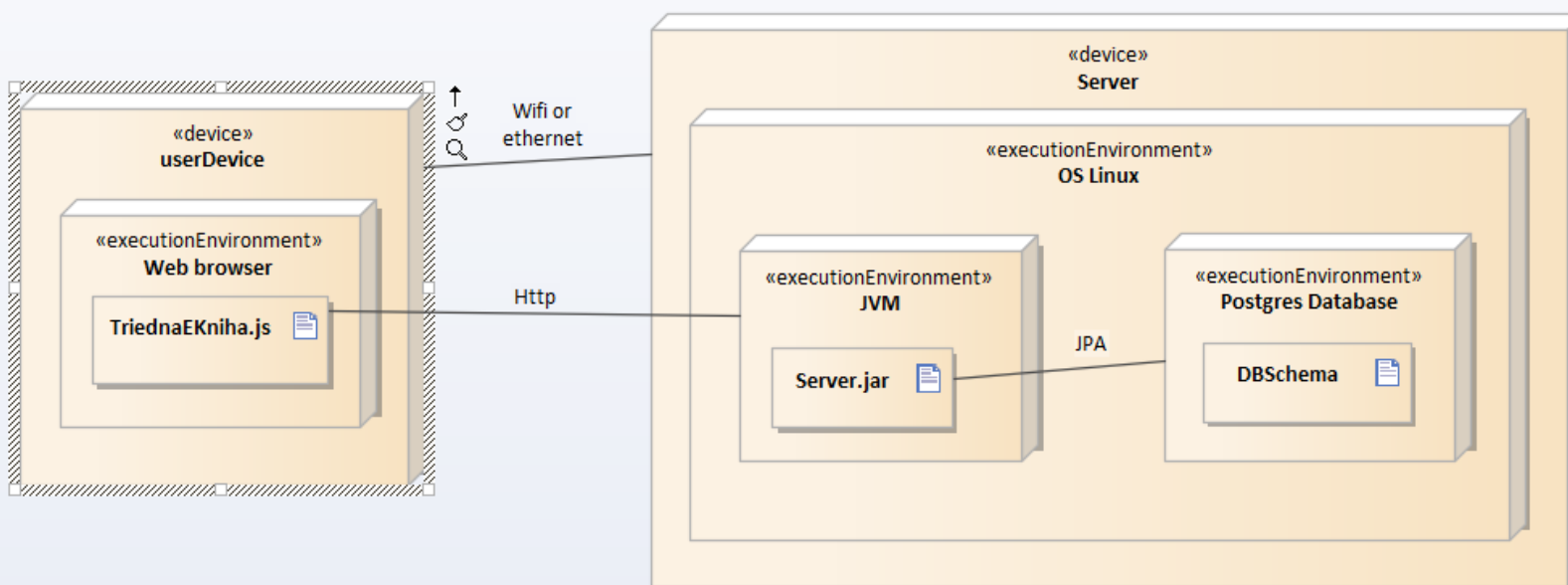
Pro přehlednější pohled na funkcionalitu našeho systému přikládáme Usecase diagram. Lze z něho dobře vyčíst, jaké možnosti mají jednotlivé role a jak se od sebe liší.



Obrázek 1: Usecase diagram

12. Diagram nasazení

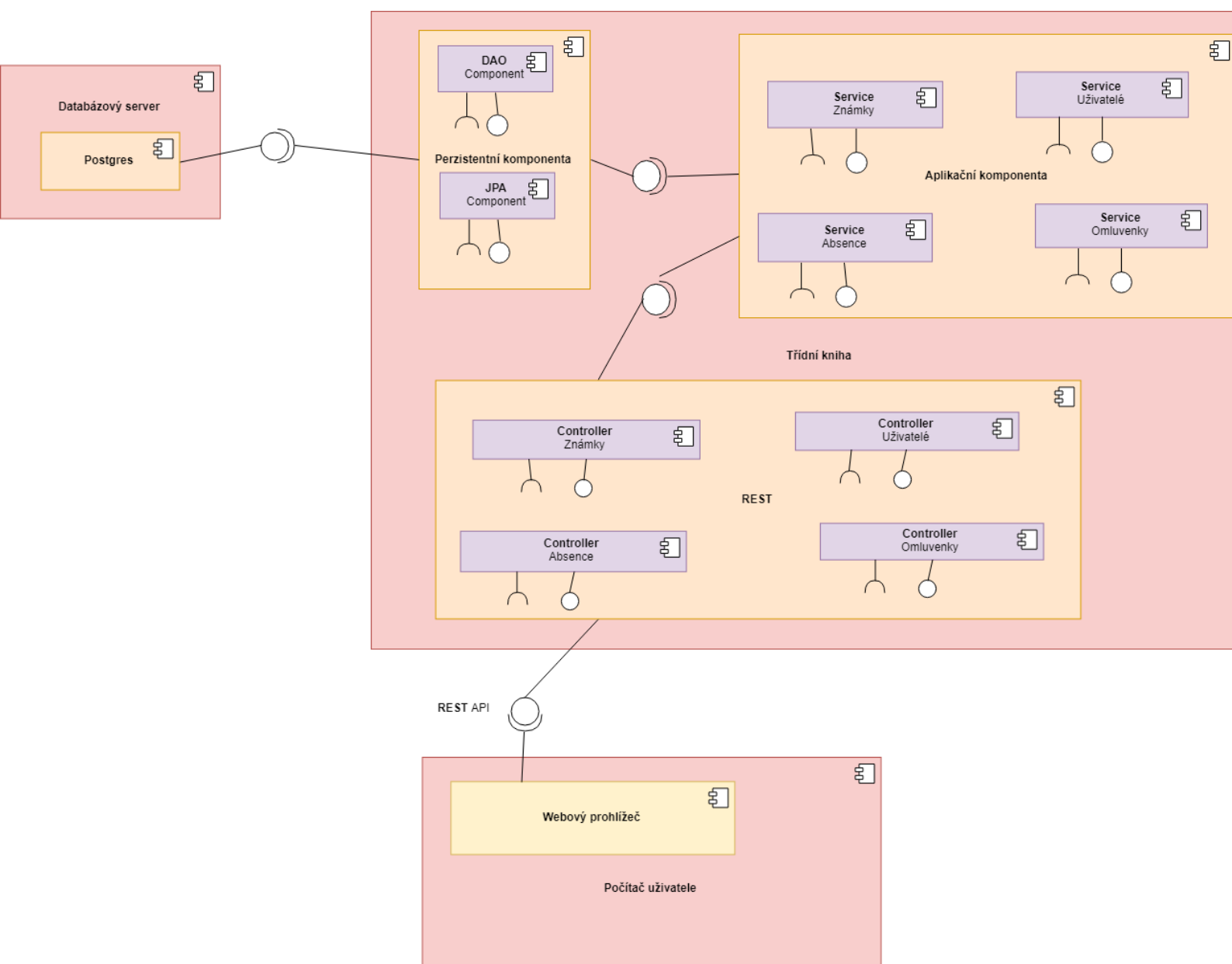
Zde uvádíme diagram nasazení, který zachycuje rozmístění zdrojů našeho systému.



Obrázek 2: Diagram nasazení

13. Diagram komponent

Níže uvádíme detailně rozepsaný diagram komponent. Je z něj zřejmé rozdělení do více vrstev.



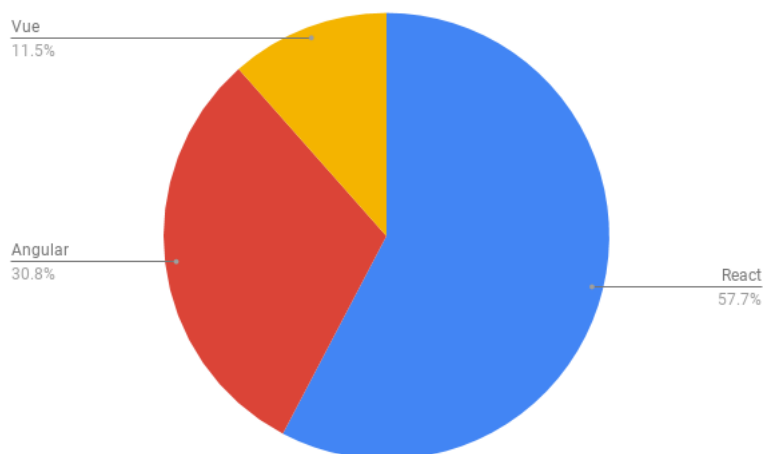
Obrázek 3: Diagram komponent

14. Rozbor a výběr alternativ návrhu řešení

14.1. Frontend

Na základě požadavků, které vznikly v průběhu návrhu softwaru, jsme se rozhodli, že nebudeme ani uvažovat nad něčím jiným než Javascriptovým frameworkem. Možnost psát to v čistém Javascriptu nebo generovat obsah pomocí PHP jsme tedy zavrhnuli v rámci snahy držet krok s moderní dobou.

Mezi uvažované frameworky přišely React, Vue.js a AngularJS. Na základě zájmu o seznámení se s nejpopulárnějším jsme vybrali React na základě počtu stažení balíčků podle Google Statistics.



Obrázek 4: Používáno JS frameworků

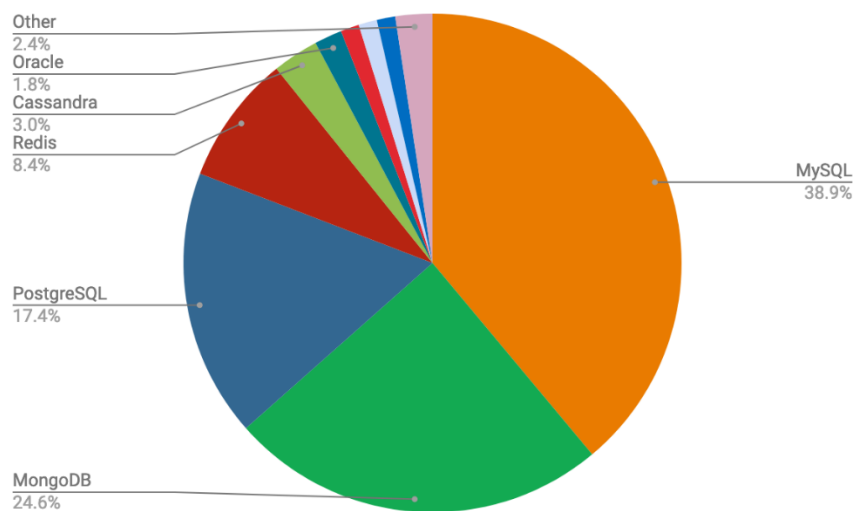
14.2. Backend

Pro backend je možné v dnešní době použít skoro jakýkoliv známý programovací jazyk např.: Java, C#, Python, Javascript, atd. V každém z nich existuje i řádka frameworků, které lze pro dané účely použít. Z povahy našeho studovaného oboru jsme se rozhodli dát přednost Javě, neboť je zde upřednostňována. Ve snaze ušetřit si práci jsme se rozhodli dát přednost nějakému frameworku nad JavaEE.

Java frameworků je celá řada a výběr by byl velmi komplikovaný, kdyby ve výběru opět nefigurovaly zkušenosti ze školy. Zde jsme se setkali s frameworkem Spring, který je programátorsky přívětivý a spoustu práce vykoná sám jen za pomoci notací. Mezi alternativy patří například JSF (JavaServer Faces), který je podstatně komplikovanější, STRUTS, který není moc flexibilní nebo Hibernate, jehož výhody by se na tomto projektu neprojevily.

Databáze

V dnešní době jsou databáze hlavně dvou druhů, a to SQL a noSQL. Vzhledem k povaze našich dat a chuti držet se tradičních metod s kterými jsme byli seznámeni ze školy, jsme se rozhodli pro SQL. Zde existuje velká škála všech možných databází. Rozlišují se svou robustností, enginem a prací s nimi. My jsme se rozhodli vybírat na základě vlastních zkušeností, protože jsme své znalosti chtěli rozšiřovat na jiných místech tohoto projektu. Rozhodli jsme se pro Postgres, jehož alternativou bylo MySQL.



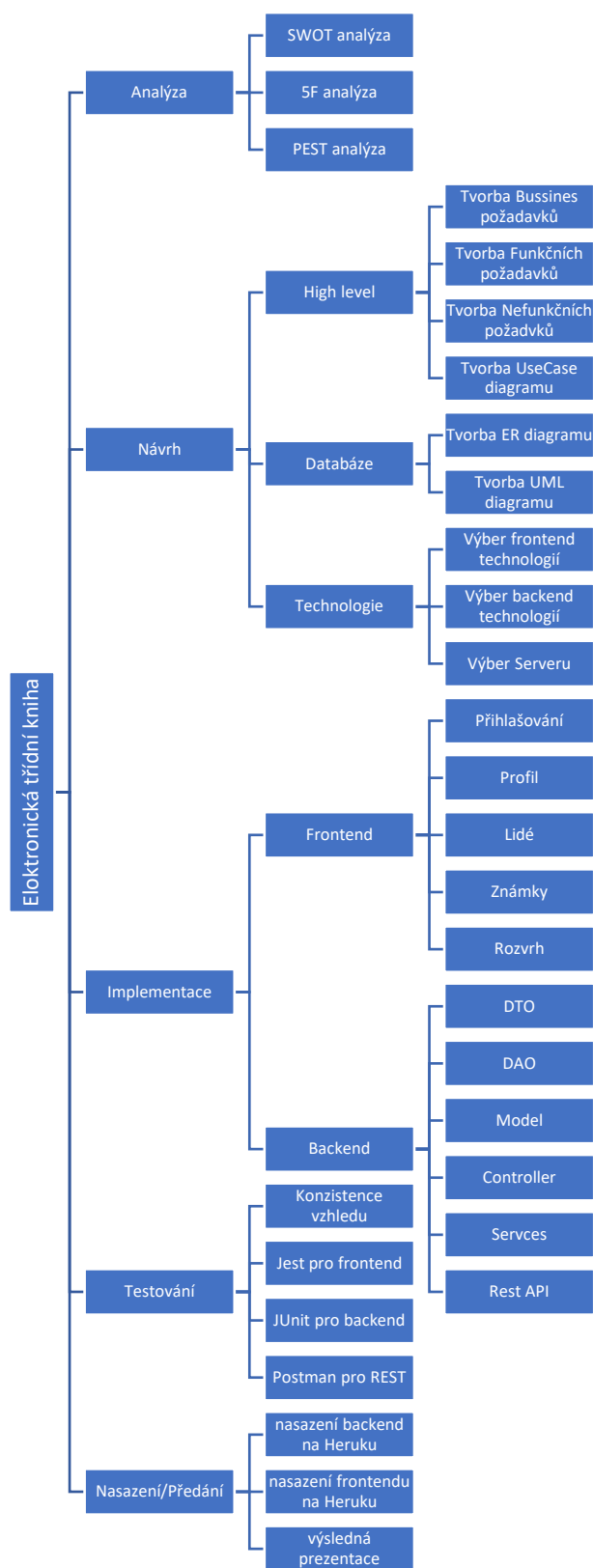
Obrázek 5: [Popularita databází](#)

14.3. Server

Poslední velkou otázkou bylo, kam náš server nasadit. Na základě zvolených technologií viz. výše bylo třeba najít server, který je podporuje. Dalším kritériem byla komplikovanost konfigurace a celkového nasazení. Mezi přívětivější servery patří Heroku, Microsoft Azure, Amazon Web Server a další. Na základě jednoduchosti práce se serverem jsme nakonec zvolili Heroku.

15. WBS – rozdělení projektu na dílčí procesy

Též uvádíme rozpad celého procesu vývoje. Jsou zde zachyceny všechny kroky, které bylo v plánu vykonat. Zároveň se však tímto nezavazujeme, že byly vykonány.



Obrázek 6: WBS

16. Zdroje

Jak již bylo výše zmíněno jsme malá firma, která se skládá ze studentů vysoké školy studujících IT obor. Tento produkt jsme se rozhodli zhotovit v rámci předmětu NSS, podle čehož se odvíjejí i zdroje.

16.1. Lidské

Máme k dispozici 5členný tým studentů. Všichni jsou k dispozici po dobu trvání celého projektu. Tým obsahuje tři pracovníky na backend a dva na frontend. Analytiky a návrháře tvoří 3 lidé z týmu. Testera své části si dělá každý sám. Konzultant je zároveň každý z týmu. O deployment se starají dva lidé týmu -jeden z frontendu a jeden z bacekndu.

16.2. Finanční

Projekt je v tuto chvíli neziskový, členové týmu tedy pracují zadarmo. Jediným výtěžkem tohoto projektu je bodový zisk při jeho prezentaci na hodině NSS.

16.3. Časové

Projekt je časově omezen délkou trvání letního semestru. Výrazný vliv a právo úpravy deadlinů má cvičící předmětu Ing. Jiří Šebek. V tuto chvíli je konec projektu stanoven na 24.5.2019

16.4. Technické

Každý z členů týmu je vybavený vlastním počítačem, což je též jediná potřebná pomůcka pro tento projekt. V rámci projektu je možné využít školních licencí na IntelliJ, Microsoft Office a Enterprise Architect.

17. Normy a standardy

Zde jsme sepsali hrubý seznam toho, čeho se jednotliví členové drželi při implementaci za účelem udržení jednotné struktury dané části projektu.

17.1. Standart kódu na frontendu

- Tab je 4 mezery
- Po každé funkci jeden volný řádek
- Složená závorka vždy na nový řádek
- Mezera za čárkou
- Znaménka oddělena mezerami
- Deklarace proměnných, funkci a metod vždy na nový řádek
- Poznámky vždy na samostatný řádek nad zmiňovanou oblast
- Vždy používat uvozovky na místo apostrofů
- V JSX při více jak tří atributů psát atributy pod sebe
- Používat anonymních funkcí a arrow funkcí na místo bindingu
- Kde je možné využívat postupného vyhodnocování podmínek a jejich zkrácených forem

17.2. Standart kódu na backendu

- Dodržovat zásady OOP (žádné public proměnné apod.)
- Tab je 4 mezery
- Po každé metodě jeden volný řádek
- Mezera za čárkou
- Znaménka oddělena mezerami
- Deklarace proměnných, funkci a metod vždy na nový řádek
- Složené závorky vždy na konec daného řádku s deklarací
- Anotace vždy na samostatný řádek nad danou třídu nebo metodu
- Třídy pojmenovávat dle funkcionality a Upper-CamelCase
- Názvy proměnných Lower-camelCase

17.3. REST standart

- Vytvářet RESTful endpointy
- Slovy za lomítky se konkretizuje výběr zleva doprava
- Volitelné hodnoty náleží vždy slovu nalevo

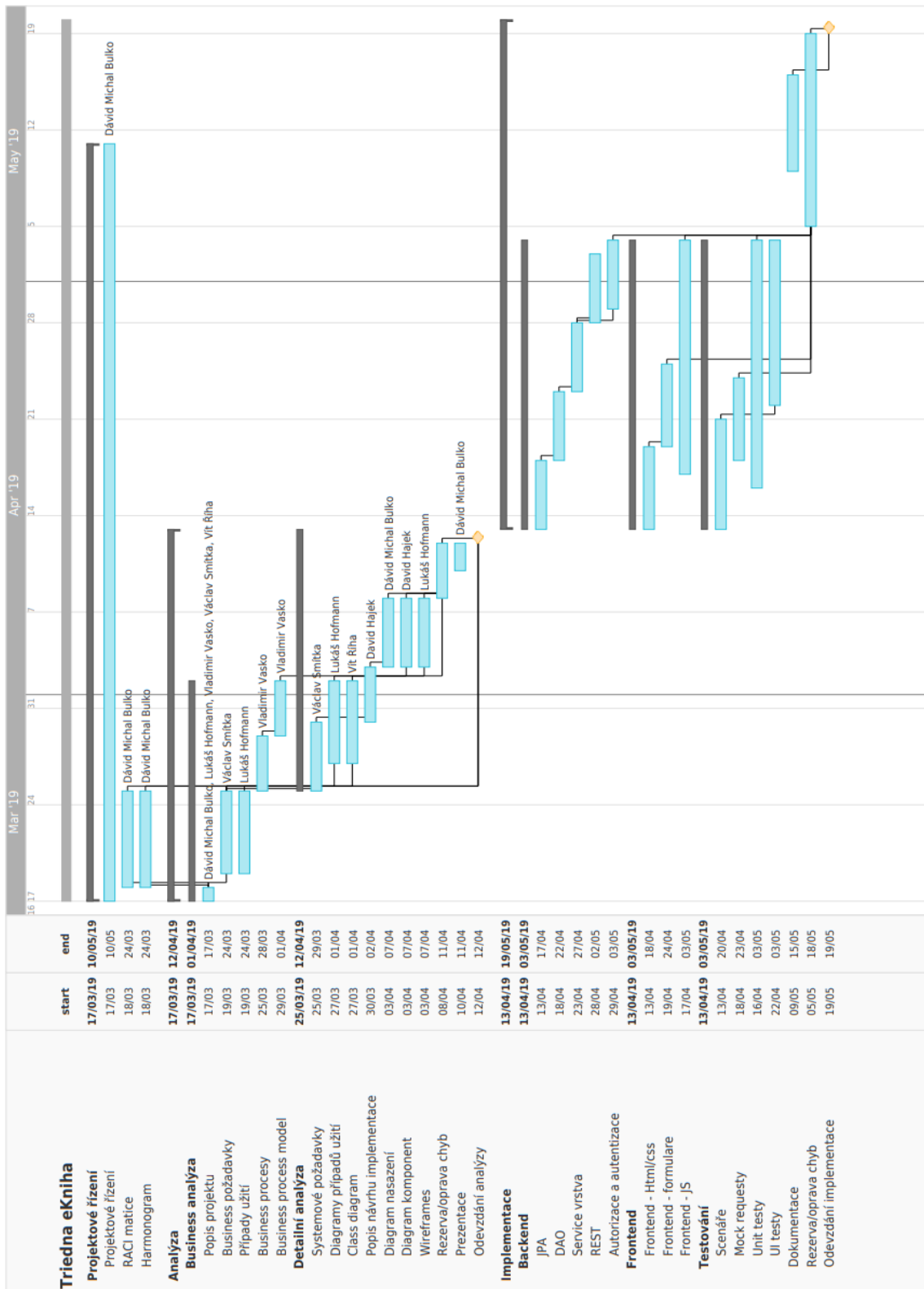
18. Matice zodpovědnosti

V rámci týmu jsme si též rozdělili zodpovědnost za jednotlivé fáze projektu podle možností a schopností jednotlivých členů.

Aktivita	Dávid Michal Bulko	David Hájek	Lukáš Hofmann	Václav Smítka	Vladimír Vasko
Popis projektu	I	R, A	C	I	I
RACI, Harmonogram	R, A	I	I	C	I
Požadavky (Business, Sys.)	I	C	I	R, A	I
Případy užití	I	I	R, A	I	I
Modely tříd (Relační a Class)	I	R, A	C	I	I
Business Proces Model	C	I	I	I	R, A
Popis návrhu implementace	I	R, A	C	I	I
Diagram komponent	C	R, A	I	I	I
Diagram nasazení	R, A	I	I	I	C
Wireframy	C	I	R, A	I	I
HTML, CSS	R, A	I	I	R, A	I
Javascript	R, A	I	I	R, A	I
JPA	I	R, A	I	I	R
DAO	I	R	I	I	R, A
Services	I	R	R, A	I	I
REST	I	R, A	I	I	R
Autentizace a autorizace	I	I	R	R, A	I
Testování UI	R, A	C	R	R	C
Testování REST	I	C	R, A	R	C
Unit Testy	I	R	C	C	R, A

19. Harmonogram GANTT

Zde je vyobrazen původní návrh průběhu celého projektu. Ve skutečnosti se tento harmonogram vůbec nevyplnil.



Obrázek 7: Ganttův diagram

20. Analýza rizik FMEA

Dále představujeme běžná rizika vývoje softwaru. Pro analýzu rizik jsme zvolili kritickou míru $RPN_{krit} = 130$.

Prvek/Funkce	Možná vada	Možné následky vady	Význam	Kritičnost	Možné příčiny (mechanizmy vady)	Výskyt	Stávající opatření proti prevenci	Stávající řízení návrhu / odhalování	Odhalitelnost	Rizikové číslo
Implementace přihlašování	Nedostatečné zabezpečení	Možné nabourání do aplikace	6		Nedostatek času	8	žádné	Průběžné kontory	6	288
	Nefunkční token	Nefunkčnost aplikace	9		Nezkušenost, nedostatek času	9	žádné	Průběžné kontory	10	810
	Špatné navržení komunikace s backendem	Rádoby náhodné chyby	6		Nezkušenost	9	žádné	Průběžné kontory	6	324
Implementace profilu	Nekvalitní řešení avatarů	Pomalé načítání a hustý traffic	3		Nezkušenost	5	žádné	žádné	3	45
	Špatně ošetřená změna hesel	Ukradení hesla/celého účtu	7		Nedostatek času, nezkušenost	6	žádné	Průběžné kontory	3	126
	Špatné ošetření práv zobrazování	Dostání se na cizí účet	8		Zbrkllost	3	žádné	Průběžné kontory	4	96
Implementace lidí	Komplikované filtrování	Zpomalení aplikace, možné chyby	4		Zaladěnost do jednoho problému	4	žádné	Znalost některých best practice	8	128
	Nadměrný traffic	Zpomalení aplikace	5		Nezkušenost	6	žádné	žádné	4	120
	Nedostatečně ošetřená prázdná pole vyhledávaných výsledků	Rozbití vzhledu aplikace	6		Nedostatek času	7	žádné	Průběžné kontory	8	336
Implementace známek	Chybné převádění čísel na řetězce a obrácené	Chyby v zobrazovaných hodnotách nebo v běhu aplikace	8		Komplikovanost celého modulu	3	žádné	Průběžné kontory	8	192
	Špatné ošetření práv zobrazování	Možnost vidět cizí známky	9		Nedostatek času	8	žádné	Průběžné kontory	7	504
Implementace rozvrhu	Nadměrný traffic	Zpomalení aplikace	5		Nezkušenost	5	žádné	žádné	4	100
	Neintuitivní ovládání	Nespokojenost uživatelů	3		Nedostatek času pro věc	6	žádné	Znalost některých best practice	5	90
	Špatné ošetření práv zobrazování	Možnost vidět cizí rozvrh	8		Nedostatek času	8	žádné	Průběžné kontory	4	256

Obrázek 8: 1. část rizik

Doporučená opatření	Odpovědnost	Provedená opatření	Význam	Výskyt	Odhaltitelnost	Rizikové číslo
Důkladné otestování	David Michal Bulko	žádné	7	5	8	280
Důkladné otestování, čas na studium	David Hájek	žádné	7	5	8	280
Důkladné naplánování	David Michal Bulko	žádné	6	6	8	288
žádné						
žádné						
žádné						
žádné						
žádné						
Důkladné projití aplikace	David Michal Bulko	žádné	8	5	8	320
Důkladné otestování	Václav Smítka	žádné	7	5	8	280
Důkladné otestování	Václav Smítka	žádné	7	5	8	280
žádné						
žádné						
Důkladné otestování	Lukáš Hofmann	žádné	7	5	8	280

Obrázek 10: Doplněk k 1. části

Doporučená opatření	Odpovědnost	Provedená opatření	Význam	Výskyt	Odhaltitelnost	Rizikové číslo
Studium problematiky	Lukáš Hofmann	žádné	6	4	5	120
žádné						
Studium problematiky	Vladimír Vasko	žádné	5	4	5	100
Studium problematiky	David Hájek	žádné	5	4	5	100
žádné						
Důkladné otestování	David Hájek	žádné	7	5	8	280
žádné						
Důkladné otestování	David Hájek	žádné	7	5	8	280
žádné						
Důkladné otestování	David Hájek	žádné	7	5	8	280
žádné						
žádné						
Vyhrazení času na dokumentaci	Václav Smítka	žádné	8	5	9	360
žádné						

Obrázek 9: Doplněk 2. části

Prvek/Funkce	Možná vada	Možné následky vady	Význam	Kritičnost	Možné příčiny (mechanizmy vady)	Výskyt	Stávající opatření proti prevenci	Stávající řízení návrhu / odhalování	Odhaditelnost	Rizikové číslo
Implementace DAO	Komplikované dotazy	Možné vzniknutí chyb	6		Nezkušenost	6	žádné	Znalost některých best practice	5	180
	Duplicitní dotazy	Složitost v údržbě	4		Rozdělení práce mezi více lidí	4	žádné	Znalost některých best practice	4	64
Implementace modelu	Špatná implementace hierarchie	Chyby v databázi	9		Nezkušenost	4	žádné	Průběžné kontory	7	252
	Špatná implementace m:n vazby	Chyby při dotazování na data přes vazbu	9		Nezkušenost	6	žádné	Průběžné kontory	7	378
	Nedostatečná nebo nadměrná dekompozice	Složitá údržba a škálovatelnost	4		Nezkušenost	2	žádné	žádné	6	48
Implementace controllerů	Nedostatečné otestování	Chyby ve fungování backendu	6		Nedostatek času	7	žádné	Znalost některých best practice	7	294
	Nadměrné velké metody	Problém se škálovatelností a údržbou	4		Zaběhlé zvyky	2	žádné	Průběžné kontory	6	48
Implementace Services	Nesprávné ošetření práv jednotlivých rolí	Možnost odeslání uživateli data na které neměl právo	8		Nedostatek času	4	žádné	Průběžné kontory	5	160
	Nedostatečné otestování	Možné 500 na servru	3		Nedostatek času	6	žádné	žádné	5	90
	Nedostatečně okomentované	Špatná orientace vkodu	3		Nedostatek času	8	Javadoc	Znalost některých best practice	8	192
Implementace Rest API	Nepřehledně a chaoticky naimplementované endpointy	Špatná orientace a údržba	4		Nezkušenost	4	žádné	Znalost některých best practice	7	112
	Velké množství podobných endpointů	Dezorientace, nedorozumění	3		Nezkušenost	5	žádné	žádné	6	90
	Nedostatečná dokumentace	Problém s prezentací a předáníma aplikace	5		Nedostatek času	6	Swagger	Znalost některých best practice	5	150
	Neodstatečné otestování	Možné chyby jak na frontendu tak backendu	5		Nedostatek času	4	žádné	Průběžné kontory	5	100

Obrázek 11: 2. část rizik

21. Znovupoužitelnost

V rámci návrhu i implementace celého projektu se snažíme držet novodobých trendů a pouček nabytých při studiu. Jsme si vědomi potenciálního růstu produktu a též možné potřeby zvýšení výkonu. Tedy snažíme se brát ohled jak na znovupoužitelnost, tak na škálovatelnost. Myslíme si, že vícevrstvá architektura je dobrou volbou pro tyto kritéria.

Dále na fronendu používáme moderní framework React, který je již sám o sobě navržen tak aby se dobře modularizoval. Snažíme se těchto zásad držet a používat best practices daného frameworku. Tímto se snažíme rozdělit vykreslovací logiku od práce s daty a stavu.

Na backendu používáme Spring, který nás nijak více nelimituje v použití typu databáze nebo spojení s frontendem. Též samotné rozdělení na model, services, dao a rest část umožňuje backend jednoduše rozšiřovat a spravovat.

22. Přílohy

22.1. Detailní popis vícevrstvé architektury

Třídní eKniha bude postavena na Vícevrstvé architektuře. Každá vrstva zodpovídá za určitou věc a spolupracují mezi sebou přes definované rozhraní. Pateří aplikace bude framework Spring, který se využije pro backendovou část, a pro frontend použijeme React.

- Prezentační vrstva
- Webová vrstva
- Aplikační (business) vrstva
- Perzistentní a DataAccess vrstva
- Databázová vrstva

22.1.1. Prezentační vrstva

Prezentační vrstva představuje aplikaci napsanou v Reactu. Komunikuje s webovou vrstvou pomocí HTTP protokolu. Webová vrstva zpracovává požadavky a předává je nižší (aplikační) vrstvě. Nakonec prezentační vrstva dostává přes webovou vrstvu odpovědi.

22.1.2. Webová vrstva

Webová vrstva je tvořena servelty a komponentami pro zpracovávání klientských požadavků a vracení odpovědí (contorollery). Při zpracování požadavků komunikuje webová vrstva s business vrstvou. Pro tuto vrstvu je důležitý vypořádání se se všema výjimkami vyhozenýma i ostatními vrstvami. Jelikož je to vstupní bod do naší aplikace musí se postará i o autentizaci.

Využíváme principy REST. A API bude specifikované ve Swaggeru

22.1.3. Aplikační vrstva

V této vrstvě se nachází jádro celé aplikace. Obsahuje Services, ve kterých najdeme veškerou logiku a všechny funkcionality. Aplikační vrstva se také stará o veškerou autorizaci (povoluje, kdo co může dělat).

22.1.4. Perzistentní vrstva

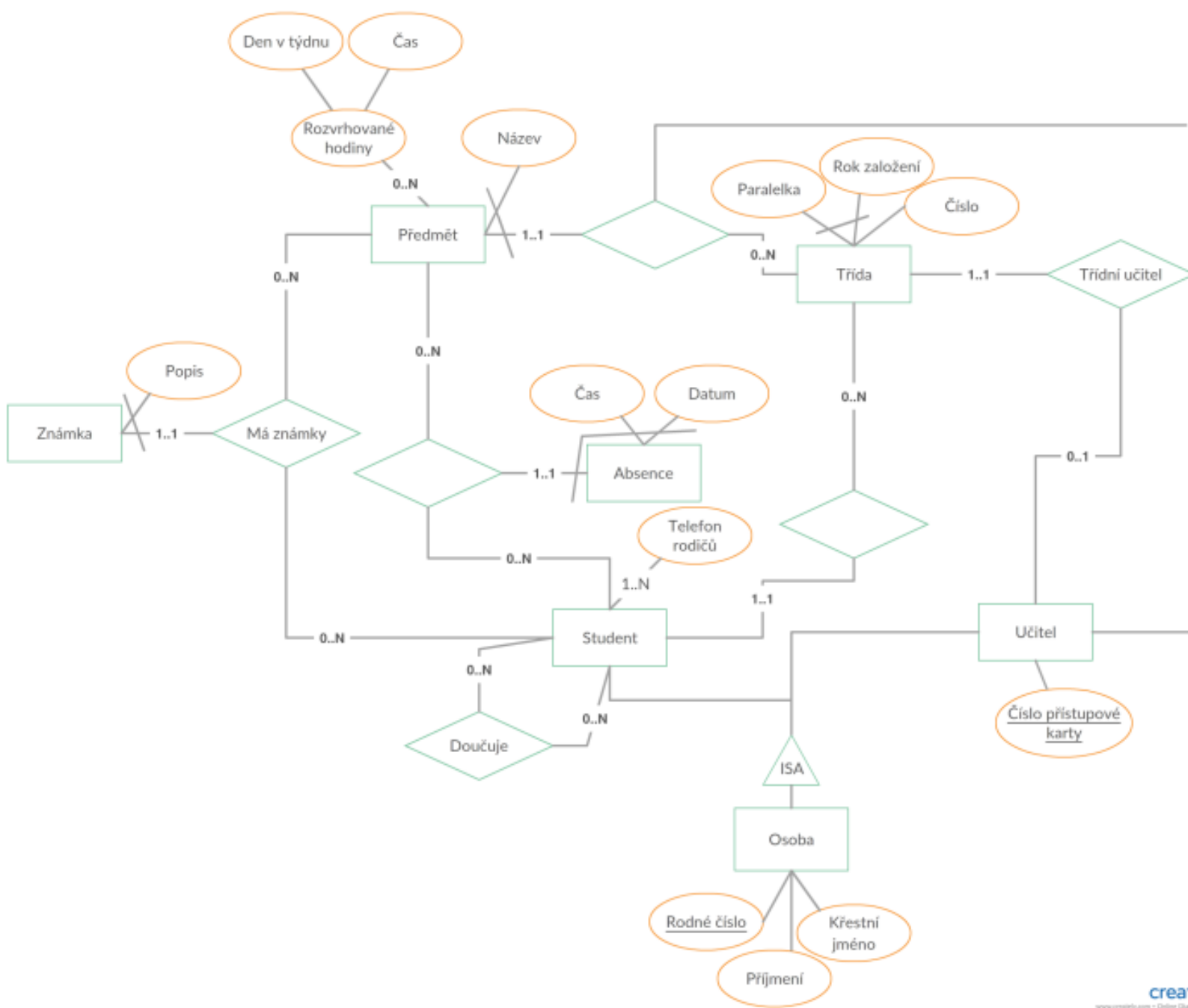
V perzistentní vrstvě je potřeba vytvořit Model, který bude nezávislý na druhu použitých zdrojů. Použijeme Objektově Relační Mapování pomocí technologie JPA. A dále definujeme DAO vrstvu, která nám poskytuje abstraktní rozhraní do nějaké databáze.

22.1.5. Databázová vrstva

Použijeme Postgres.

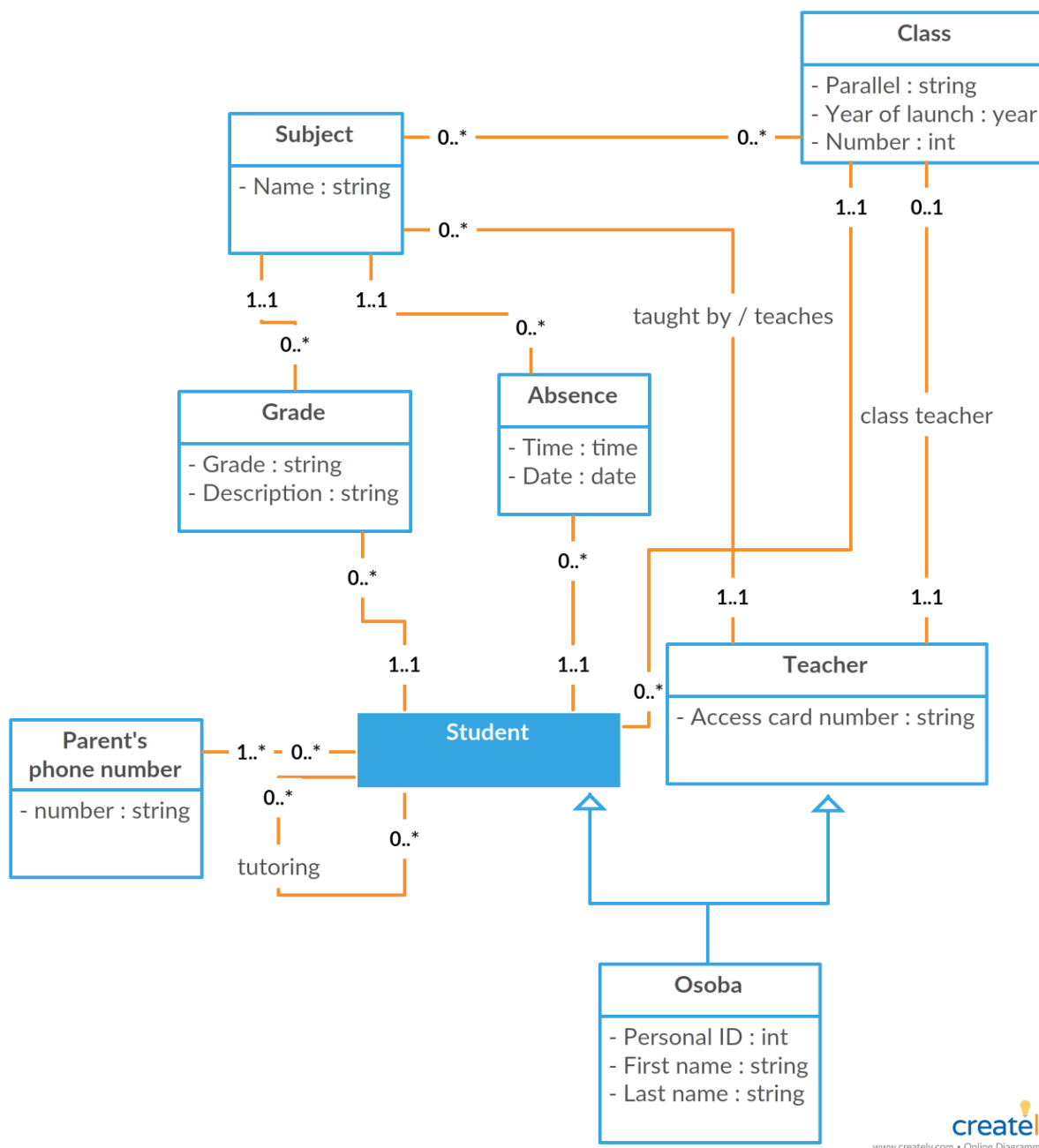
Při navrhování jednotlivých vrstev budeme muset definovat DTO jako nositele dat mezi jednotlivými procesy a vrstvami.

23. Návrh ER diagramu



Obrázek 12: ER diagram

24. Návrh UML diagramu



Obrázek 13: UML diagram

25. Seznam obrázků

Obrázek 1: Usecase diagram	12
Obrázek 2: Diagram nasazení	13
Obrázek 3: Diagram komponent	14
Obrázek 4: Používáno JS frameworků	15
Obrázek 5: Popularita databází	16
Obrázek 6: WBS	17
Obrázek 7: Ganttův diagram	21
Obrázek 8: 1. část rizik.....	22
Obrázek 9: Doplněk 2. části	23
Obrázek 10: Doplněk k 1. části	23
Obrázek 11: 2. část rizik.....	24
Obrázek 12: ER diagram	27
Obrázek 13: UML diagram.....	28