



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Anketa ČVUT - verze 3.0 - vyplňování anketních lístků
Student:	Bc. Vojtěch Štecha
Vedoucí:	Ing. Michal Valenta, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2019/20

Pokyny pro vypracování

1. Seznamte se s návrhem nové koncepce Ankety ČVUT v diplomové práci Davida Knapa.
2. Diskutujte a zvolte vhodnou architekturu aplikace, implementační platformu a model vývoje aplikace.
3. Dále se zaměřte na uživatelské rozhraní. Diskutujte a dokončete návrh části vyplňování anketních lístků.
4. Implementujte část vyplňování anketních lístků.
5. Proveďte uživatelské testování a opravte případné nedostatky.
6. Část vyplňování anketních lístků připravte k beta testování, které proběhne za zimní semestr 2018/2019.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 26. června 2018



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Diplomová práce

Anketa ČVUT - verze 3.0 - vyplňování anketních lístků

Bc. Vojtěch Štecha

Katedra softwarového inženýrství

Vedoucí práce: Ing. Michal Valenta, Ph.D.

10. ledna 2019

Poděkování

Děkuji Ing. Michalovi Valentovi, Ph.D. za vedení této práce s trpělivostí a za podporu při vytváření této práce. Dále bych chtěl poděkovat své rodině za veškerou podporu, kterou mi poskytovala po celou dobu mého studia

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 10. ledna 2019

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2019 Vojtěch Štecha. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Štecha, Vojtěch. *Anketa ČVUT - verze 3.0 - vyplňování anketních lístků*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Tato diplomová práce popisuje vývoj nové aplikace Anketa ČVUT. Analyzuje dostupné požadavky z předchozí práce kolegy Knapa, uvádí a porovnává možné návrhy realizace. Autor popisuje dostupné klient-server technologie a vývojové nástroje. Vybrané technologie a nástroje jsou použity při implementaci. Klient využívá technologie React a pro server byla použita Java s frameworkem Spring Boot. Vývoj probíhal agilním způsobem, kdy výsledky testů jednotlivých iterací byly ihned použity pro další verzi. Implementace byla též podrobena testování vybranými koncovými uživateli. Zdrojový kód je v příloze.

Klíčová slova JavaScript, Java, React, Vývoj webových stránek, Anketa ČVUT, Spring Boot

Abstract

This diploma thesis describes the development of a new application Anketa ČVUT. Requirements from the thesis of my colleague Knap are analyzed and possible designs are listed and compared. Author describes available client-server technologies and development tools. Selected technologies and tools were used for the implementation. The client part uses React and the server

uses Java with Spring Boot framework. Agile development was applied. The test results from individual iterations were used for the next version. The implementation was tested by selected end users. Source code can be found in the attachment.

Keywords JavaScript, Java, React, Web development, Inquiry CTU, Spring Boot

Obsah

Úvod	1
1 Analýza	3
1.1 Rozbor zadání	3
1.2 Anketa	4
1.3 Historie studentské ankety ČVUT	4
1.4 Úvodní návrhy uživatelského rozhraní	9
1.5 Požadavky pro Implementaci	11
1.6 Technologie webových stránek	12
1.7 Databáze	14
2 Návrh	17
2.1 Jednostránková aplikace	17
2.2 Technologie pro vývoj SPA	18
2.3 Výběr technologie	19
2.4 Autorizace a Autentifikace	19
3 Implementace	21
3.1 Vývojové nástroje	21
3.2 Klient	21
3.3 Server	31
4 Testování	33
4.1 Scénáře	33
4.2 Výsledky	36
Závěr	41
Literatura	43

A Seznam použitých zkratek	47
B Obsah přiloženého CD	49

Seznam obrázků

1.1	Návrh vzhledu pro vyplňování předmětů	10
1.2	Návrh psaní textového komentáře	10
1.3	Návrh přehledu o učitelé	11
3.1	Domovská stránka	27
3.2	Domovská stránka na mobilním zařízení	28
3.3	Zobrazení předmětů	28
3.4	Hodnocení předmětu	29
3.5	Hodnocení předmětů na mobilním zařízení	30
3.6	Vyplňování textového hodnocení	30
3.7	Vyplňování textového hodnocení na mobilním zařízení	31
3.8	Databázové schéma	32

Úvod

Technologie se neustále mění, a to obzvláště v oblasti internetu. Existuje mnoho různých způsobů jak se obsah uživateli podává. Jedním z nich jsou webové stránky. Implementace webových stránek může být realizována jednou z mnoha technologií na to určených. Záleží vždy na konkrétních požadavcích a dostupných zdrojích. Stránky musí být přístupné jak koncovým uživatelům, tak správcům pro potřeby budoucích oprav a aktualizací.

V této práci se zabývám implementací nové verze webových stránek pro vyplňování ankety na ČVUT. Analýza požadavků a prototyp vzhledu nové verze jsou popsány v diplomové práci Davida Knapa[1]. S touto prací jsem se detailně seznámil. Navrhl jsem nový způsob implementace, který jsem na základě této práce následně také realizoval.

V kapitole Analýza provádím rozbor zadání, zmiňuji nedostatky současné verze programu Anketa a popisuji požadavky na implementaci. Dále popisuji jednotlivé dostupné technologie webových stránek a typy databází.

Kapitola Návrh uvádí jednotlivé možné způsoby implementace a související technologie. V závěru této části popisuji zvolenou technologii a nástroje.

V následující části provedu čtenáře implementací jak klientské tak serverové části a uvádím některé použité části kódu.

V kapitole Testování jsou popsány jednotlivé testovací scénáře a výsledky testování.

Analýza

1.1 Rozbor zadání

1. Seznamte se s návrhem nové koncepce Ankety ČVUT v diplomové práci Davida Knapa[1].
 - Je zapotřebí se detailně seznámit se zpracovaným návrhem v práci Davida Knapa[1]. Diskutovat proveditelnost a na základě podkladů obsažené v této práci vytvořit návrh na implementaci.
 - Návrh kolegy Davida Knapa a požadavky budoucích uživatelů uvedu v kapitole Analýza.
2. Diskutujte a zvolte vhodnou architekturu aplikace, implementační platformu a model vývoje aplikace.
 - Návrh architektury aplikace a zvolení použitelných a vhodných technologií. Způsob nasazení implementované aplikace na produkční prostředí. Zvolení vhodných vývojových nástrojů.
 - Použitelná řešení ucedu v kapitole Návrh.
3. Dále se zaměřte na uživatelské rozhraní. Diskutujte a dokončete návrh části vyplňování anketních lístků.
 - Implementace uživatelského rozhraní. Analýza stávajícího návrhu obsaženého v práci Davida Knapa[1] a případné rozšíření o další možné prvky.
 - Změny uživatelského rozhraní uvedu v kapitole Implementace a na základě testování může být tato implementace změněna.
4. Implementujte část vyplňování anketních lístků.
 - Realizace funkcionality o vyplňování anketních lístků na základě analýzy a návrhu této práce.

- Implementace aplikace bude přílohou této diplomové práce a v kapitole Implementace uvedu některé důležité části.
5. Proved'te uživatelské testování a opravte případné nedostatky.
- Vyzkoušení implementace testery pro validaci správného zřízení a zpracování případných připomínek.
 - Testovací scénáře a výsledky testování budou uvedeny v kapitole Testování.
6. Část vyplňování anketních lístků připravte k beta testování, které proběhne za zimní semestr 2018/2019.
- Realizovaná aplikace je naplánována na nasazení do produkčního prostředí a možného využití v Zimním Semestru 2018/2019.

1.2 Anketa

Anketa slouží ke sběru zpětné vazby všech studentů ČVUT. Hlavním zaměřením je sběr hodnocení výuky. Jedná se o informace o průběhu předmětů za jednotlivé semestry. Hodnocení zpravidla obsahuje několik částí, které hodnotí různé prvky ve výuce. Jedná se o hodnocení učitelů, samotné látky předmětu, podání informací na přednáškách, laboratořích a proseminářích. Studenti se mohou vyjádřit anonymně a tím mají větší volnost k vyjádření svého názoru. Informace sesbírané pomocí ankety lze využít pro nalezení nedostatků nebo naopak věcí, které prospívají výuce a měly by být více využívány. Anketa se skládá jak s otázek s možností odpovědět jen výběrem z předem nastavených hodnot tak i volných otázek kde se je možné odpovědět komentářem. Otázky s výběrem předpřipravených odpovědí je obvykle více, protože je možné přiřadit hodnoty pro jednotlivé odpovědi a provádět nad výsledky matematické operace.

1.3 Historie studentské ankety ČVUT

Současná verze webové aplikace Anketa ČVUT byla vytvořena v roce 2007. Pokus o jednoduché přidávání nových modulů nevedl k vytyčenému cíli, kterým bylo vytvořit jednotnou aplikaci nejen pro zadávání ale také pro přehledné vyhodnocení. Novela vysokoškolského zákona, která vstoupila v platnost v roce 2016 ukládá sledovat vývoj vnitřního hodnocení v čase pro celou univerzitu a jednotlivé fakulty. Tyto požadavky současná verze nespĺňuje. Tato diplomová práce navazuje na diplomovou práci Davida Knapa „Návrh nového řešení aplikace Anketa ČVUT“ z roku 2018[1]. Kolega Knap se ve své práci zaměřil na sběr požadavků jednotlivých budoucích uživatelů a vytvořil návrh uživatelského rozhraní pro několik scénářů. Dále provedl sběr

požadavků studentů, pedagogů a univerzitního managementu. Vstupy byly dodány z těchto fakult a uživatelů s uvedením hlavních specifik shrnutých do jednotlivých bodů.

Uživatel 1 – aktivní hodnotitel

- Důraz na zjednodušení současného složitého a redundantního dotazníku
- Motivace ve formě losování tabletu je vhodná a dostatečná
- Zvážit povinnost hodnocení pro uzavření předmětu
- Možnost adresného i anonymního hodnocení
- Možnost pro nové studenty získat informace hodnocení pro zapisované předměty

Uživatel 2 – současný systém nepoužívá z důvodu složitosti

- Současný systém vyžaduje patnáct minut na vyplnění duplicitních informací
- Pomalá odezva současného systému
- Problémy s rychlostí a stabilitou aplikace
- Zvážit použití emotikonů (symboly používané na moderních sociálních sítích)
- Zamezit tomu, že přednášky jsou evidované v různých systémech

Uživatel 3 – doktorand, který se aktivně podílí na zlepšování

- Zkrácení dotazníku
- Vyplňování by nemělo zabrat více než pět minut
- Důraz na textové komentáře
- Zpřesnění formulace otázek
- Možnost vracet se k rozpracovaným hodnocením
- Největší motivací k vyplnění by mělo být – vidět reálné změny v předmětech
- Zrušit možnost odpovídat anonymně

Uživatel 4 – se zkušenostmi ze zahraničí

- U nás hodnotí pouze pokud došlo k problémům nebo byl nespokojený
- Pro kladné hodnocení se stejně musí proklikat pětistránkovým formulářem
- Zjednodušení dotazníku pro kladné hodnocení
- Rozšíření současného systému odměn jako motivace pro studenty
- Problém se zastaráváním současného systému
- Doporučuje celkové předělání vzhledu aplikace i logiky otázek
- Formát otázek na konci semestru doporučuje zachovat
- Sjednotit výstupy hodnocení na jedno místo

Fakulta informačních technologií

- Chybějící kvantitativní hodnocení jednotlivých osob
- V současné verzi hodnocení cvičení k předmětu bez možnosti hodnotit osobu vyučujícího
- Podporovat zobrazení výsledků za posledních pět let „živě“
- Starší data by mělo být možné načíst na žádost z datového skladu univerzity
- Zdůraznit i pokračující existenci celofakultních otázek
- Nejednoznačnost hvězdičkového systému hodnocení – nutno doplnit popisem nebo smajlíky
- Dynamické otevření komentáře při špatném kvalitativním hodnocení
- Tzv. happy path – možnost udělit pětihvězdičkové hodnocení a anketu uzavřít
- Pohled do systému, který otevře všechny dosud nevyplněné anketní lístky
- Proaktivní nabídka dalších předmětů k hodnocení při ukončení předchozího
- Nultá úroveň – kvantitativní hodnocení semestru či studia samotného
- Zvážit přidání hodnocení aktuálního semestru ve vztahu k minulému
- Rada pro vnitřní hodnocení ČVUT – schválená akademickým senátem – nutno zahrnout mezi schvalovatele nové aplikace

Fakulta elektrotechnická

- Aplikace by neměla suplovat diskuzní fórum
- Současné rozhraní pro vyučující je vnímáno jako velmi nedostatečné
- Proděkan pro studium (a děkan jeho prostřednictvím) by měl mít možnost zasahovat do příspěvků studentů za hranou etiky, ale nemělo by dojít k zkreslení výsledků; cenzurovaný příspěvek by měl být viditelně označen
- Současný názor FEL tak je, že vyučující by měl nést zodpovědnost za obdržení hodnocení a ta by měla být vždy veřejná
- Vyučující by ocenili větší možnosti reakcí na hodnocení
- Jako reakce na studentův komentář by mohla být zavedena určitá forma označení, kterým dá vyučující najevo, že komentář viděl a bere jej na vědomí, ale nemá, co by k němu dodal či jej komentoval
- U jednotlivých hodnocení by mělo být k dispozici více informací o pisateli (zvolil si nepovinný předmět...) – ale pouze na dobrovolné bázi
- Celkové přepracování pohledu vedoucího katedry – pohled jak přes předměty, tak přes vyučující
- Zvýrazňovat skupiny s malým absolutním počtem hodnocení a označovat je jako málo relevantní
- Chybějící pohled na trendy ve vývoji hodnocení učitelů, předmětů, ale také oborů a programů v jednotlivých letech
- Export dat do XLS a XML
- Přehledné hodnocení s oborem či programem
- Možnost hodnotit také spokojenost se studijním oddělením, vedením fakulty, vybavením university pro výuku i sportovní využití, hodnocení čistoty či bufetů
- Usnadnit přepínání mezi přehledy jednotlivých učitelů či předmětů
- Možnost upravovat reakci vyučujícího
- V současné verzi splývají komentáře k jednotlivým vyučujícím na úrovni předmětu a není jasné koho student hodnotil
- Možnost zobrazení fotek vyučujících ze systému Usermap
- Předměty s nedostatečně relevantním hodnocením by měly být vyřazeny z žebříčků hodnocení předmětů

Fakulta dopravní

- Nutnost většího zapojení sběru zpětné vazby do průběhu výuky na fakultě
- V současnosti studenti nemají vůli vyplňovat hodnocení a ústavy nemají na co reagovat
- Podpora aplikace pro sběr prvního dojmu nově nastupujících studentů a porovnávat naopak s výstupy studentů, kteří se chýlí ke konci studia
- Možnost použití anketních lístků také k dalším oblastem (například Studentská unie – studovny, ...)
- Přání fakulty důrazněji sbírat zpětnou vazbu od studentů, kteří v předmětech měli potíže, zejména pak u studentů absolvujících předměty v druhém zápisu apod.

Fakulta stavební

- Fakulta se už od roku 2012 potýká s každoročním výrazným úbytkem vyplněných hodnocení
- Roli jednoduchého portálu s hodnocením vyučujících postupně převzal studentský portál Archa ČVUT
- Popularitě ankety na fakultě také ubírá nestabilita procesů - anketa v tuto chvíli nemá pevně stanovené termíny otevření a uzavření

Fakulta biomedicínského inženýrství

- Ke sběru zpětné vazby se nyní používá jak celouniverzitní anketa, tak alternativní papírová forma za využití anonymních boxů rozmístěných v budovách fakulty
- reakce vyučujících na vlastní hodnocení nejsou příliš obvyklá

Fakulta jaderná a fyzikálně inženýrská

- Nová možnost pro správce ankety nahrát seznam externistů a přiřadit jej k danému předmětu
- Možnost přidat hodnocení závěrečných prací
- Problémy s anonymitou u menších předmětů

Fakulta strojní

- Kvůli nízké míře vyplnění vyučující nemají motivaci se s nasbíranými daty zabývat
- Problémy s anonymitou při povinně uváděném studijním průměru, který je unikátní

Prorektorka pro studium

- Současná anglická verze není plnohodnotná
- Doplnění hodnocení vybavení učeben a laboratoří pro výuku odborných předmětů, dopravní obslužnost, koleje, ubytování, menzy a dalších
- Doplnění hodnocení pro kantýny, kavárny, papírnictví a copycenter umístěných v budovách jednotlivých fakult
- Anketní lístky pro hodnocení mezinárodních programů
- Hodnocení, jakým způsobem studium zasahuje do osobního života

Rada pro vnitřní hodnocení

- Chybí reporty historických dat

1.4 Úvodní návrhy uživatelského rozhraní

Ve své práci jsem vycházel z níže uvedených high-fidelity wireframes, které jsem poté upravoval na základě zvolných technologií a výsledků testování.

1. ANALÝZA

Anketa ČVUT / Hodnocení předmětů / Výpočetní technika 1

Nástěnka

Hodnotit předměty **3 zbývá**

Výsledky předmětů

Výsledky vyučujících **1 reakce**

Další přehledy

Hodnotíte Výpočetní technika 1

Celkový dojem 😞 😟 😐 😊 😄

Obsah předmětu

Ing. Martin Novotný, Ph.D. 😞 😟 😐 😊 😄

Ing. Jana Nováková, Ph.D. 😞 😟 😐 😊 😄

[Přidat dalšího vyučujícího](#)

[Přidat textové hodnocení k předmětu](#)

[Přidat textové hodnocení k Ing. Martin Novotný, Ph.D.](#)

[Přidat textové hodnocení k Ing. Jana Nováková, Ph.D.](#)

[Přidat textové hodnocení ke zkoušce](#)

Otázka k předmětu: Jak jste byli spokojeni s návazností na předmět Praktická technologie 1?

[Zahodit vyplněná data](#) [Uložit data do mezipaměti](#) [Uložit a odeslat](#)

Obrázek 1.1: Vzhled navrženého uživatelského rozhraní pro vyřňování předmětu[1]

Hodnotili jste vyučujícího **Ing. Josefína Smyšlená, Ph.D.** negativně. Abychom mohli předmět zlepšit, potřebujeme další informace o tom, co bylo v předmětu špatně, a jak to můžeme do příštího roku napravit. Zkuste být co nejkonkrétnější, pomůže nám to.

Jaké problémy se ve výuce objevily?

Hodnotíte vyučujícího Ing. Josefína Smyšlená, Ph.D.

Zveřejnit:

- Mé jméno
- Můj obor studia
- Můj ročník studia **1**
- Můj studijní průměr
- Roli předmětu v mém stud. plánu

Nalezen pouze 1 student v předmětu s ročníkem 4.

Obrázek 1.2: Vzhled navrženého uživatelského rozhraní pro psaní textového komentáře[1]



Obrázek 1.3: Vzhled navrženého uživatelského rozhraní pro přehled dat o učitelích[1]

1.5 Požadavky pro Implementaci

Nová verze Ankety musí být dostupná na internetu s využitím nejnovějších technologií. Dále musí umožnit jednoduchý přístup bez potřeby doplňujících instalací a s použitím běžně rozšířených a dostupných technologií. Stávající řešení je webová aplikace a nové řešení bude pro uživatele dostupné stejným způsobem. Aby nové řešení splnilo tyto požadavky, musí být také webovou aplikací, kterou bude možné rozšiřovat a jednoduše spravovat. Musí být přístupná jak z webového prohlížeče, tak z mobilního zařízení, aby se aplikace jednoduše používala uživateli. Ovšem webové aplikace se dají navrhnout a realizovat různými způsoby. Navržená webová aplikace bude přesněji webovou stránkou kompatibilní s moderními webovými prohlížeči.

Výsledná práce musí splňovat následující nefunkční požadavky.

- Přístupná z moderních webových prohlížečů.
- Možné nasadit a provozovat na serveru dodaným vedoucím práce.
- Responzivní uživatelské rozhraní.

A následující funkční požadavky.

- Uživatel má přístup ke svým anketám zadaných v systému.
- Uživatel má přístup k otázkám v anketě.

- Uživatel se může vrátit k rozpracovaným anketním lístkům na stejném zařízení, kde provedl předchozí úpravy.
- Uživatel může vyplnit otázky v anketě a potvrdit je pro uložení na serveru.
- Uživatel se přihlašuje do aplikace přes ČVUT školní autentifikaci.
- Aplikace se napojuje na databázi dodanou vedoucím práce.

1.6 Technologie webových stránek

Pro vytvoření webových stránek je možné použít řadu technologií. Některé standardy jsou implementovány všemi a ostatní si volíme na základě porovnání několika faktorů. Nejrozšířenější technologie zahrnují použití HTML[7], CSS[9] a JavaScriptu[2]. Moderní webová stránka se bez těchto základních technologií neobejde. Existují však různé nástavby, které lze použít. Tyto technologie poskytují pokročilější funkce pro vývojáře, ale jejich výstup jsou přeloženy do standardních technologií, kterým rozumí webový prohlížeč. Příklady pro rozšíření jsou JSP[5] pro HTML, SASS[4] pro CSS a TypeScript[6] pro JavaScript.

1.6.1 HTML

Jazyk HTML, celým anglickým jménem Hypertext Markup Language, je základním stavebním prvkem každé webové stránky. Pomocí tohoto jazyka se definuje struktura a význam informací na webové stránce. Text se doplňuje o specifické značky, takzvané tags, a tím se mění jejich význam. Informace přiřazené těmito značkami je možné strojově číst a zpracovávat.

1.6.2 CSS

CSS (Cascading style sheet) je technologie určená pro úpravu vizuálního vzhledu dokumentu psaného v HTML. Díky této technologii je možné v internetovém prohlížeči jednoduše měnit barvy, velikosti písma, fonty, animace a další.

1.6.3 JavaScript

Všeobecně používané pojmenování pro ECMAScript[2]. Moderní prohlížeče umožňují automatické spuštění kódu JavaScript. Nejčastější použití toho jazyka je pro přidání dodatečné funkcionality pro zobrazenou HTML stránku v prohlížeči. Možné je i spuštění JavaScriptu mimo webový prohlížeč a to například pomocí Node.js[8].

1.6.4 Server

Předchozí technologie umožňují prezentovat data pro uživatele a zobrazovat a reagovat na ovládací prvky. Tyto základní prvky umožňují uživateli intuitivní ovládání. Další technologie slouží k poskytnutí pokročilé logiky, perzistenci dat a možnost využití dat na webové stránce. Pro uživatele je tato část neviditelná, nezáleží na vnitřní implementaci, ale jen na výsledných datech. Vývojáři mají možnost využít jednu z mnoha dostupných technologií. Následuje seznam technologií, se kterými jsem se seznámil a uvádím jejich jednotlivá specifika.

1.6.5 Java

Java je programovací jazyk, který v čase vydání této práce je na 1. místě popularity podle žebříčku TIOBE[10]. Jeden z důvodů proč Java dosáhla tohoto úspěchu je možnost využití programů napsaných v Javě na většině používaných operačních systémů. Java podporuje psaní aplikací, které mohou být přístupné z internetu. Existují podpůrné frameworky, ve kterých je možné prezentovat stránky vytvořené v HTML. Je možné využít řadu šablonovacích systémů, jejichž syntaxe se podobá HTML, ale výsledná stránka je dynamicky upravena na základě aplikační logiky. Výsledný kód obsahuje pouze technologii, které webový prohlížeč zobrazuje. Pro webové aplikace se nejčastěji používá jeden z dostupných frameworků podporující Javu. Jeden z nejpopulárnějších frameworků je Spring[11]. Spring lze použít ve variantě Spring Boot[11], který přednastavuje určitou konfiguraci pro jednodušší a rychlejší práci. Aplikace napsané pro web lze spustit na počítači, který je připojený s veřejnou adresou, takzvaný server, přímo. Ale bývá zvykem spustit tento kód v jiné aplikaci určené pro správu. Jeden z těchto programů je Apache Tomcat[12] pro spouštění Java Servletů. Je ho možné využít s kombinací s Apache HTTP serverem[13] pro využití mnoha přídatných funkcí, které poskytuje.

1.6.6 C#

C#[14] je programovací jazyk používaný ve frameworku .NET. Standard tohoto jazyka byl vytvořen společností Microsoft, ale existuje více implementací překladače. Ve frameworku .NET (přesněji v APIS.NET[15]) lze vytvářet webové aplikace. ASP.NET framework se obecně používá na počítačích se systémem Windows, ale existuje implementace .NET Core. Tato implementace podporuje různé operační systémy. Webovou aplikaci .NET Core je možné spustit přímo na počítači nebo je možné využít web serveru pro windows IIS[16].

1.6.7 JavaScript

JavaScript je v dnešní době možné použít i pro podporu logiky a práci dat na počítači který poskytuje webové stránky. Node.js[8] je aplikace umožňující

spuštění kódu napsaného v JavaScriptu přímo na operačním systému. Hlavní výhoda spočívá v tom, že vývojář webových stránek, může vytvořit celou potřebnou logiku jak na stránkách v prohlížeči tak i logiku na serveru jen v JavaScriptu. Ovšem JavaScript nebyl s tímto úmyslem vytvořen, a proto bych nedoporučoval použití pro větší a složitější implementaci.

1.7 Databáze

Aplikace musí umět uchovávat data a pro tento účel se používají databáze. Databáze je nedílnou součástí pokročilých aplikací, které umožňují správu dat. Existuje více metod pro perzistenci dat a i různé implementace stejné metody. Každý typ databáze má určité výhody a nevýhody a použití tedy záleží na situaci. Jedno z hlavních dělení databází je rozdělení na SQL[17] a noSQL databáze. SQL, Structured Query Language, je jazyk všeobecně používaný pro provádění akcí nad uloženými daty. Jeho použití je především určeno pro relační databáze. Relační databáze se skládají z tabulek, kde data se ukládají do řádků tabulek. NoSQL databáze mohou také využívat SQL, ale jen s omezenou funkcionalitou.

1.7.1 SQL

Databázový systém se skládá z dat v databázi a systému RDBMS, zkratka pro relational database management system, v překladu relační systém řízení bází dat. Se spojením s jazykem SQL se bavíme o relačních databázových systémech. Mezi používané implementace patří databáze Oracle, MySQL, PostgreSQL a Microsoft SQL Server. Volna databáze pro tvorbu aplikace není vázána výběrem implementační technologie pro server.

1.7.2 NoSQL

NoSQL databáze se buď zaměřují na reprezentaci dat ve složitých útvarech, například ve formátu grafovém, nebo se zaměřují na zpracování rychlost zpracování, práci s neuniformními daty, paralelním zpracováním a objemu dat. V těchto systémech jsou dotazy nad daty složitější pro tvorbu než v SQL databázích.

1.7.3 Výběr technologií

U technologií pro zobrazení webových stránek je zapotřebí použití všech zmíněných technologií a to HTML, CSS a JavaScript. Pro logickou část aplikace po analýze technologií, zvážení osobních dovedností a diskuzích s vedoucím práce jsem rozhodl využít Java Spring Boot. Server který bude dodán vedoucím práce bude z největší pravděpodobností využívat operační systém na základě Linux. Aplikace pro Javu obecně takového systému

podporují. Tímto se splňuje jeden z nefunkčních požadavků pro možnost nasazení aplikace na server dodaným vedoucím práce.

Návrh

Webové stránky je možné navrhnout více způsoby. Základní model probíhá následujícím způsobem. Uživatel přes webový prohlížeč odešle požadavek pro stažení stránky, která je jednoznačně identifikována pomocí URL. Server vytvoří stránku pokud se jedná o dynamický web a nebo odesílá stránku z paměti. Pomocí odkazů na jedné stránce je možné přecházet na jiné stránky. Každá stránka má unikátní URL pro její identifikaci a webový prohlížeč přes URL se stránky nahrává. Dalším způsobem je vytvoření stránky, která obsahuje všechny ostatní stránky v sobě a přes interní logiku mění obsah který zobrazuje. Tento styl stránek se nazývá jednostránková aplikace.

2.1 Jednostránková aplikace

Z anglického názvu single page application, dále jen SPA, jsou takové aplikace které jsou spuštěny ve webovém prohlížeči. Na první pohled působí stejně jako ostatní webové stránky. Tradiční stránky mohou být buď statické nebo dynamicky vykreslené. Statické stránky jsou tvořeny pomocí HTML a mohou být dále obohaceny pomocí CSS a Javascriptu. Důležité je, že se stránky na serveru nemění a posílají se pro každého klienta stejně. Staticky také jsou poskytovány různé soubory jako obrázky, oddělené soubory s CSS a Javascriptem nebo jakýkoliv jiný soubor. Dynamické stránky se liší tím, že se stránka před odesláním klientům na serveru vygeneruje. Neboli server vytrváří stránku na základě dané logiky takovou aby ji prohlížeč rozuměl. Jednostránkové aplikace se chovají jako dynamické stránky, ale načítají se jako statické. Chovají se tak, protože místo toho aby se logika vykonávala na serveru, který by vrátil celou vykreslenou stránku, tak místo toho se logika vykonává přímo ve webovém prohlížeči. Od vytvoření ECMAScriptu[2], standardizované jméno pro JavaScript, tak valná většina dnešních prohlížečů tento jazyk podporuje. Proto jednostránkové aplikace jsou napsány v JavaScriptu. Klient si stáhne aplikaci a ta na základě vnitřní logiky ukazuje dynamický obsah. Ovšem i tyto

aplikace potřebují komunikovat po síti, například pro stažení dat, odeslání nových dat na server nebo aktualizace dat.

Kvůli rozdělení aplikace na dvě části se otevírají nové možnosti. Jedna s těchto možností je, že může existovat více různých klientů kteří využívají sdílený API server. Klientská část je schopná fungování bez internetového připojení pokud proběhlo načtení dat. Na API server se může připojit i klient ve formě například mobilní aplikace nebo i jiný server.

2.2 Technologie pro vývoj SPA

Pro vývoj SPA je možné začít s pomocí hotových JavaScriptových knihoven nebo celých frameworků. Nejpoužívanější jsou AngularJs[18], Angular[19], React[20], Vue[21], Ember[22] a Meteor[23]. Dále uvažuji jen o relevantních technologiích.

2.2.1 AngularJs

AngularJs je plnohodnotný MVC framework. Proto obsahuje komponenty pro vytvoření datové, vzhledové a aplikační části. AngularJs poskytuje vlastní rozhraní pro AJAX požadavky, ale je možné použít i jiné knihovny. Popularita AngularJs je na druhém místě spolu s React, kde Angular je více vyhledáván na StackOverflow[33] ale React má lepší skóre na GitHub[32].

2.2.2 Angular

Angular je budoucnost AngularJs, kde framework je postavený místo JavaScriptu na TypeScriptu[6]. TypeScript je jazyk vytvořený společností Microsoft. TypeScript je superset ECMAScriptu verze 2015 a kompiluje se do této verze ECMAScriptu. Účel TypeScriptu je poskytnout staticky typované proměnné a další funkce pro zjednodušení přehlednosti a lepší organizaci kódu. Výhoda Angularu je rozšíření o možnost nasazení pro PC a telefony.

2.2.3 VueJs

VueJs je MVC framework který byl vytvořen Evanem You. Evan You pracoval pro google s AngularJs a tyto zkušenosti ho inspirovali pro vytvoření VueJs. VueJs převzal vlastnosti z AngularJS a i z dalších knihoven jako například Redux, používaných pro doplnění logiky k Reactu.

2.2.4 React

React je pouze JavaScriptová knihovna. Z architektury MVC plní čistě funkci view, neboli zajišťuje jen vykreslování stránek. Práce s daty a jejich ukládání a načítání je potřeba řešit samostatně. Výhodou Reactu je rychlé vykreslování stránek. Pro vytvoření SPA s Reactem je tedy zapotřebí použít další

JavaScriptové knihovny pro složení všech nutných funkcionalit pro vytvoření plné SPA.

2.3 Výběr technologie

Pro implementaci aplikace Ankety jsem se rozhodl rozdělit uživatelské rozhraní, klienta, do SPA a API serveru ,který bude poskytovat funkce pro manipulaci s daty. Pro správu dat bude využita Oracle Databáze. Klient bude využívat knihovnu React a následující JavaScriptové knihovny:

- Redux[25] je použit pro kontrolu a uchování stavu aplikace, neboli všech dat, které aplikace potřebuje.
- React-Redux[26] je knihovna použitá pro spojení dat uložených pomocí Reduxu s komponenty v Reactu.
- React-Router[27] je použit pro zpřístupnění navigace pomocí URL.
- Redux-Thunk[28] je použit pro rozšíření schopností Reduxu.
- Bootstrap[29] poskytuje připravené CSS.
- Reactstrap[30] poskytuje React komponenty stylizované v Bootstrapu.
- Axios[31] je HTTP klient, využitý pro komunikaci se serverem.

2.4 Autorizace a Autentifikace

2.4.1 Shibboleth[34]

ČVUT používá systém single sign on, dále SSO, s implementací pomocí technologie Shibboleth. SSO služba zajišťuje přístup do všech systémů přes centrální přihlášení v jedné aplikaci. Uživatel po přihlášení může po určitou dobu navštěvovat aplikace podporující SSO. Aplikace Shibboleth implementuje standard SAML[37] pro autentifikaci. SAML standard definuje jak mají mezi sebou komunikovat části systému pro autentifikaci a autorizaci. Přesněji se jedná o 2 části a to poskytovatele identity, identity provider, dále jako IDP a poskytovatele služby, service provider, dále jako SP.

IDP reprezentuje centrální místo, kde je možné přihlásit se pomocí uživatelských přihlašovacích údajů. Pokud se uživatel úspěšně autentifikuje, tak je možné využívat systémy chráněné pomocí SP, které jsou nakonfigurovány s použitým IDP.

SP umožňuje aplikacím využívat jednotnou autentifikaci. Při přístupu na chráněnou službu, SP zkontroluje zda už byl uživatel autentifikován u IDP a pokud ne, tak uživatel bude přesměrován na IDP pro přihlášení. Po přihlášení u IDP bude uživatel přesměrován zpět na SP, který předá informace službě, kterou chrání a ta může tyto údaje využívat.

2.4.2 JWT zabezpečení

JWT je zkratka pro JSON Web Token a je definován standardem RFC7519[3]. JSON (JavaScript Object Notation) je formát dat ve kterém se informace v tokenu ukládají. Struktura tokenu je složená ze 3 částí odělených tečkou. Části jsou zakódovány pomocí kódování Base64[38]. První část po dekodování tvoří JSON, který obsahuje metadata o tokenu. Data o jaký typ tokenu se jedná, v tomto případě je využito typu “jwt” a algoritmus pro podepsání tokenu, například typu “HS256” (HMAC SHA256). Druhá část po dekodování z base64 tvoří JSON, který obsahuje údaje které chceme využít při autorizaci a autentifikaci. Může to být například uživatelské jméno uživatele, jeho přístupové oprávnění, nebo uživatelské Id v databázi. Datum doby platnosti je také v této části. JWT token autentifikace je bezstavová, neboli na využívaném serveru nejsou uloženy žádné informace o tom kolik a jakých tokenů bylo vydáno. Jediné co server využívá je tajný klíč pro podpis tokenů. Tokeny obsahují v sobě údaj o datu ukončení platnosti a tedy kromě změny tajného klíče není způsob jak by server zrušil vydaný token. Platnost Tokenu vyprší po uplynutí doby platnosti.

Zabezpečení tokenu před úpravou obsahujících dat zajišťuje podpis, který je součástí tokenu. Pro podpis tokenu je použit algoritmus HMAC[39] společně s hashovacím algoritmem SHA256. Server vytváří podpis pomocí svého tajného klíče a bez znalosti tohoto klíče nemůže útočným zfalšovat obsah tokenu, protože nedokáže vytvořit podpis, který by se rovnal podpisu se změněnými údaji v tokenu. Při používání tokenu musí být komunikace prováděná přes zabezpečené připojení TLS, protože pokud útočným získá token, tak je možné aby se s ním autorizoval a autentifikoval stejně jako uživatel pro kterého byl token vydán.

2.4.3 Souhrn

Aplikace vytvořená v rámci této práce využívá autentifikaci Shibboleth spolu s autentifikací pomocí tokenů JWT. Aplikace je rozložená na 2 části. Jedna část tvoří uživatelské rozhraní v podobě SPA pro vizualizaci dat. Druhá část tvoří API server, který poskytuje data a funkce pro jejich změnu. Pro získání identity v aplikaci je potřeba přihlášení pomocí Shibbolethu. API server vydá na základě údajů získaných ze Shibbolethu JWT token. Tento token se používá při komunikaci mezi API serverem a klientem. Toto schéma umožňuje využít API server s aplikacemi, které nemusí být zapojeny do schématu shibboleth a zjednodušuje využití API serveru jinými klienty.

Implementace

3.1 Vývojové nástroje

Při vývoji aplikace byly použity nástroje pro zrychlení a zlepšení práce. GIT je VCS, souborový verzovací systém, určený pro spravování zdrojových souborů pro přehlednou správu jednotlivých verzí souborů. S použitím této aplikace je možné obnovit předchozí data při nechtěném smazání či změně souborů. Podporuje jednoduché nahrávání na GIT server. Pro ukládání tohoto projektu byl využit ČVUT git server na adrese gitlab.fit.cvut.cz jak pro část klienta tak i API serveru. Visual Studio Code je IDE, Integrované vývojové prostředí, které slouží k podpoře vývojáře při psaní kódu. Toto prostředí bylo využito pro psaní klientské části. IntelliJ IDEA je IDE specializované pro vývoj v programovacím jazyce Java. Prostedí bylo využito pro psaní kódu pro část API serveru.

3.2 Klient

Klient byl vytvořen v programovacím jazyce JavaScript za použití knihovny React a dalších dále uvedených knihoven. Začátek vývoje začíná od hi-fi wireframu aplikace Anketa z práce Davida Knapa[1]. V prvním kroku implementace bylo zapotřebí převést původní návrh do vytvořených React komponent. Aplikace začíná HTML stránkou ,ve které se pomocí react knihovny manipuluje s obsahem určitého elementu. Je možné mít více jednotlivých aplikací v Reactu na jedné stránce, ale v případě této aplikace to není zapotřebí.

```
<!DOCTYPE html>
<html>
  <body>
    <div id="root">
      <!-- React content -->
    </div>
  </body>
```

3. IMPLEMENTACE

```
</html>
```

Vyvolání startu aplikace se děje následujícím způsobem ve kterém se specifikuje HTML element, který se má použít pro vykreslování obsahu v Reactu. Volaná funkce `root` vrací prvotní komponenty pro vykreslení.

```
ReactDOM.render(root(), document.getElementById("root"));
```

Aplikace v Reactu se skládají z logických celků zvaných komponenta. Komponenta může být buď třída či funkce.

```
class ExampleClass extends Component {
  render() {
    return (
      <p>class<p>
    );
  }
}
```

```
function ExampleFunction(props) {
  return (
    <p>function<p>
  );
}
```

Třídy definují výsledek, který se vykresluje uživateli v metodě `render`. U funkcí se definuje výsledek hodnotou vrácenou z této funkce. Vrácená hodnota je definována ve formátu JSX. JSX je syntaxové rozšíření Reactu, které podporuje HTML a JavaScript. Použit lze i Reactové komponenty, které se liší od HTML tagů tím, že začínají velkým písmenem. Data která se do komponent posílají jsou přístupná v proměnné `this.props`.

```
class Root extends Component {
  return (
    <div>
      <Title value="title"/>
      <p>paragraph</p>
    </div>
  )
}
```

```
class Title extends Component {
  return (
    <h1>{this.props.value}</h1>
  )
}
```

Aplikace je tvořena z hierarchicky složených komponent stejným způsobem jakým jsou tvořeny HTML stránky. Tyto komponenty mohou obsahovat uvnitř funkční logiku pro poskytnutí funkcionality.

3.2.1 Navigace

Při této implementaci by se však všechny obsah zobrazoval najednou. Je tedy zapotřebí nějakým způsobem oddělit logické celky tvořící jednotlivé stránky. Pro tuto úlohu je vhodné využít JavaScriptovou knihovnu React-Router. React-Router poskytuje komponenty pro podmíněné vykreslování ostatních komponent. Jako podmínka se používá obsah URL lišty ve webovém prohlížeči. Tato knihovna také přizpůsobuje chování prohlížeče pro podporu změny URL adresy v rámci stránky a podporu pro historii umožňující funkčnost pohybu zpět a dopředu v historii prohlížení aplikace. Používají se tři komponenty a to Router, Link, Route. Do komponenty Router se musí vložit část aplikace, která by měla podporovat tuto funkcionality. V tomto případě je žádoucí, aby celá aplikace podporovala tuto funkcionality, a proto je zapotřebí tuto komponentu využít před komponentami pro obsah.

```
function root(props) {
  return (
    <Router>
      { /* content components */ }
    </Router>
  )
}
```

Nyní lze využít komponentu pro podmíněné vykreslování Route. Komponenta Route přijímá parametr path pro definici cesty, pro kterou se má vykreslit zadaná komponenta v parametru component. Možné je i využít parametru render pro definování funkce používanou při vykreslení. Další možností je definovat obsah jako potomka, neboli obalit obsah do tagů Route, který se však vykreslí vždy a je zapotřebí zpracovat výsledek podmínky vykreslování v potomkovi.

```
<Route path="/path/to/resource" component={ResourceComponent} />
```

Pro navigaci se nepoužívá HTML element a s atributem href, ale Link. Link má parameter to, ve kterém se specifikuje URL. Je zapotřebí používat stejné označení v parametru path v komponentě Route ,aby bylo možné určit shodu.

```
<Link to="path/to/resource">resource</Link>
```

3.2.2 Správa Dat

V aplikaci je zapotřebí manipulovat s daty, předávat data mezi komponentami a pro správu dat získaných z API serveru. Redux knihovna tuto funkcionální podporu poskytuje. Pro spojení dat z knihovny Redux a React komponent se využívá React-Redux. Redux ukládá data na jednom místě jako objekt v JavaScriptu. S tímto objektem se nemanipuluje přímo, ale pomocí akcí. Akce jsou objekty, které obsahují identifikátor akce a další volitelná data. Akce jsou konzumovány pomocí vytvořených funkcí zvaných Reducer. Reducer je funkce, která přijímá objekt s daty, pojmenován State, a Akci. Na základě akce vrátí nový stav objektu State. Je výhodné vytvořit více funkcí Reducer, které se zaměří na podmnožinu objektu State pro zvýšenou přehlednost a rozdělení funkcionality. Funkce Reducer musejí být čisté funkce. Čistá funkce je taková funkce, která vždy se stejnými vstupními daty vrátí stejný výsledek. Logika na úpravu dat musí být provedena před odesláním akce na zpracování. Pokud je zapotřebí změna stavu objektu State je správné dodržovat neměnitelnost, z anglického slovíčka immutability, objektů. Neměnitelnost objektů odebírá obavy o držení referencí na objekty a nechtěné změny dat. Při ladění programu je možné využít plugin pro webový prohlížeč, který zachovává jednotlivé stavy při změně a díky nezměnitelnosti je možné se vrátit na předcházející stavy s originálními daty. Akce se posílají přes funkci dispatch, která je dostupná ve třídě vrácené z volání funkce createStore, kde se jako parametr uvádí funkce Reducer. Tato třída poskytuje přístupový bod pro funkcionální Redux, dále zmíněnou pod jménem store. Funkce Reducer lze poskládat ze specializovaných Reducerů, které se kombinují přes funkci combineReducers z knihovny Redux.

Funkce pro zpracování akcí pro část user centrálního úložiště

```
function user(state, action) {
  switch(action.type) {
    case "ChangeName": {
      return Object.Assign({}, state, { name: action.newName });
    }
    default: {
      return state;
    }
  }
}
```

Kombinování oddělených funkcí pro správu centrálního úložiště

```
const rootReducer = combineReducers({
  user
```

```
});
```

Spuštění knihovny Redux

```
const store = createStore(rootReducer);
```

Příklad akce posílající se do zpracovávajících funkcí

```
const action = {  
  type: "ChangeName",  
  newName: "newName"  
}
```

Posílání akce na zpracování

```
store.dispatch(action);
```

Pro předání dat z úložiště Redux se používá React komponenta Provider a knihovní funkce connect. Stejným způsobem jak se spojuje knihovna React-Router, tak i React-Redux umožňuje spojení s třídou store reprezentující úložiště Redux použitím komponenty Provider před React komponentami, ve kterých chceme využít funkcí Redux. Pro následné využití dat je zapotřebí napojit React komponentu přes funkci connect. Funkce connect podporuje až 4 parametry s nich využiji jen první dvě. Connect vrací funkci volající se nad React komponentou, která vyžaduje data z úložiště, vracející novou React komponentu s daty z úložiště a i případnými daty poskytované z otcovské komponenty. První parametr funkce connect požaduje funkci přijímající vnitřní stav úložiště state a vracející objekt ve které se vybírají potřebná data z úložiště. Druhý parametr požaduje funkci přijímající funkci dispatch používanou pro posílání Akcí do Reducerů a vrací object s funkcemi uplatňující zvolené akce. Objekty vrácené z obou funkcí se spojují s objektem Props, který je dostupný v React komponentě.

Funkce pro výběr potřebných dat z centrálního úložiště

```
function mapStateToProps(state) {  
  return {  
    name: state.user.name  
  };  
}
```

Funkce pro přípravu funkcí využívajících funkci dispatch

```
function mapDispatchToProps(dispatch) {
  return {
    changeName = newName => dispatch({
      type: "ChangeName",
      newName: newName
    })
  };
}
```

Funkce mapující specifikovaná data a funkce vracející funkci pro přidání hodnot do komponenty

```
connect(mapStateToProps, mapDispatchToProps)(ReactComponent);
```

Další rozšíření pro knihovnu Redux je Redux-Thunk. Tato knihovna poskytuje middleware možný použít v Reduxu pro podporu funkcí jako Akce ve funkci dispatch. Důsledek je takový, že je možné využívat asynchronní funkce. Tato skutečnost se využívá dále při používání asynchronních akcí při komunikaci s API serverem. Middleware se přidává jako druhý parametr v metodě createStore vytvářející Redux úložiště. Middleware se přidává pomocí funkce applyMiddleware z knihovny Redux následujícím způsobem kde thunk je název middlewaru z knihovny Redux-Thunk.

```
const store = createStore(rootReducer, applyMiddleware(thunk));
```

3.2.3 HTTP komunikace

Klient pro komunikaci s API serverem využívá knihovnu Axios. Axios umožňuje provádět HTTP komunikaci ve formě JavaScriptových konstruktů Promise. Jedná se o asynchronní provádění kódu, kde je možné ihned specifikovat to, se má provést až po dokončení volání. Při používání klienta Axios je možné nastavit parametry, které se budou používat ve všech požadavcích, pokud není specifikováno jinak. Toto nastavení jsem využil pro nastavení HTTP hlavičky pro JWT autentifikace. Pokud aplikace už token obdržela, je uložen v relačním úložišti (session storage), které platí do té doby, dokud se neuzavře webový prohlížeč.

Nastavení URL serveru

```
axios.defaults.baseURL = "...";
```

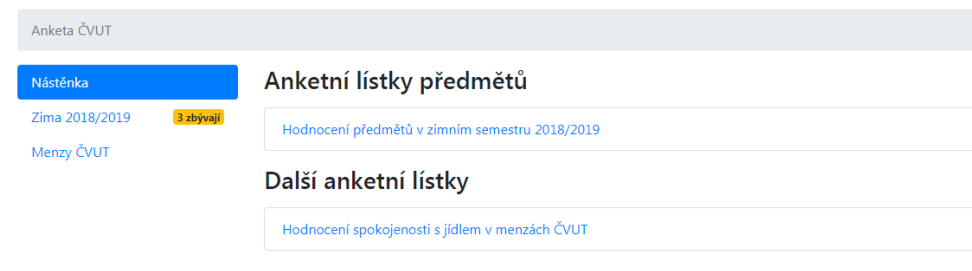

Nastavení HTTP hlavičky „Authorization“

```
axios.defaults.headers.common["Authorization"] = 'Bearer ${sessionStorage.Authorization}'
```

Posílání GET požadavku a přijímání dat

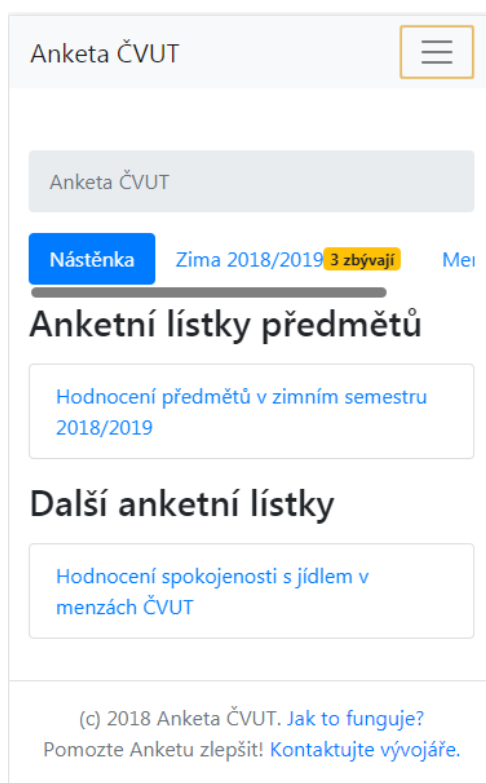
```
axios.get("/surveys")
  .then( result => {
    /* result.data contains data from server */
  });
```

3.2.4 Vzhled implementovaného klienta

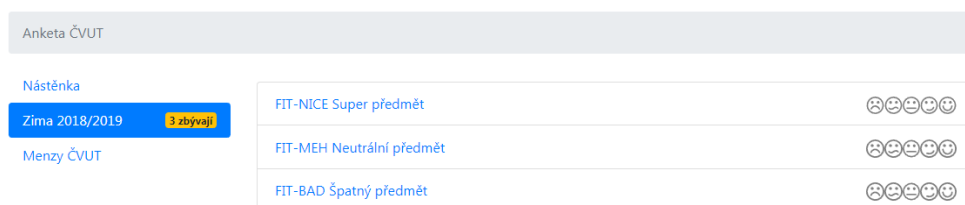


Obrázek 3.1: Vzhled implementovaného uživatelského rozhraní domovské stránky po přihlášení

3. IMPLEMENTACE



Obrázek 3.2: Vzhled implementovaného uživatelského rozhraní domovské stránky po přihlášení na mobilním zařízení



Obrázek 3.3: Vzhled implementovaného uživatelského rozhraní přehledu předmětů v anketě

Hodnotíte FIT-NICE Super předmět

Celkový dojem [?] Ohodnoťte celkový dojem z předmětu (obsah+vyučující). Níže pak můžete přidat hodnocení předmětu a/nebo vyučujících.

Obsah předmětu [?]

Ing. Martin Novotný, Ph.D. [?] ×

[Přidat komentář: Obsah předmětu](#)

[Přidat komentář: Zakončení předmětu](#)

[Přidat komentář: Ing. Martin Novotný, Ph.D.](#)

[Zahodit vyplněná data](#)

Obrázek 3.4: Vzhled implementovaného uživatelského rozhraní pro hodnocení předmětu

3. IMPLEMENTACE

Hodnotíte FIT-NICE Super předmět

Celkový dojem ⓘ

☹️ 😞 😐 😊 😄

Obsah předmětu ⓘ

☹️ 😞 😐 😊 😄

Přidat hodnocení vyučujícího ▾

Obrázek 3.5: Vzhled implementovaného uživatelského rozhraní pro hodnocení předmětu na mobilním zařízení

Vyučující Ing. Martin Novotný, Ph.D. —

Přidejte textový komentář k svému hodnocení vyučujícího Ing. Martin Novotný, Ph.D. (způsob výkladu, jednání se studenty, rovný přístup, apod.).

Komentář k vyučujícímu Ing. Martin Novotný, Ph.D.

Zde můžete vložit svůj komentář k vyučujícímu Ing. Martin Novotný, Ph.D.

Mé jméno ⓘ

Můj obor studia ⓘ

Můj ročník studia ⓘ

Můj studijní průměr ⓘ

Roli předmětu v mém stud. plánu ⓘ

[Přidat komentář: Obsah předmětu](#)

[Přidat komentář: Zakončení předmětu](#)

Obrázek 3.6: Vzhled implementovaného uživatelského rozhraní pro přidání textového komentáře

Zakončení předmětu -

popisek komentující otázku

Váš komentář k zakončení předmětu.

Zde můžete vložit komentář k zakončení předmětu.

Zveřejnit:

- Mé jméno ?
- Můj obor studia ?
- Můj ročník studia ?
- Můj studijní průměr ?
- Roli předmětu v mém stud. plánu ?

Obrázek 3.7: Vzhled implementovaného uživatelského rozhraní pro přidání textového komentáře na mobilním zařízení

3.3 Server

Server pro manipulaci s daty je vytvořen v programovacím jazyce Java za použití frameworku Spring nakonfigurovaného přes Spring Boot. Serverová aplikace je uzpůsobena pro nasazení pomocí Apache Tomcat.

3.3.1 Spring Boot

Spring Boot zajišťuje jednodušší použití frameworku Spring. Umožňuje to přes schovávání základních konfigurací přímo do knihoven, které jsou využívány. Při používání Spring Boot se používané knihovny přidávají přes buď Maven[36] či Gradle[35]. Tyto systémy podporují automatizovaný build a stahování potřebných pluginů definovaných v projektu.

3.3.2 Autentifikace

Pro využívání aplikace je zapotřebí se autentifikovat přes Shibboleth používaný na ČVUT. Autentifikace se odehrává mezi Shibboleth poskytovatelem služby (SP) a poskytovatelem identity (IDP). Poskytovatel identity na ČVUT je k 9.1.2019 definován na URL <https://idp2.civ.evut.cz/idp>. Jako po-

3. IMPLEMENTACE

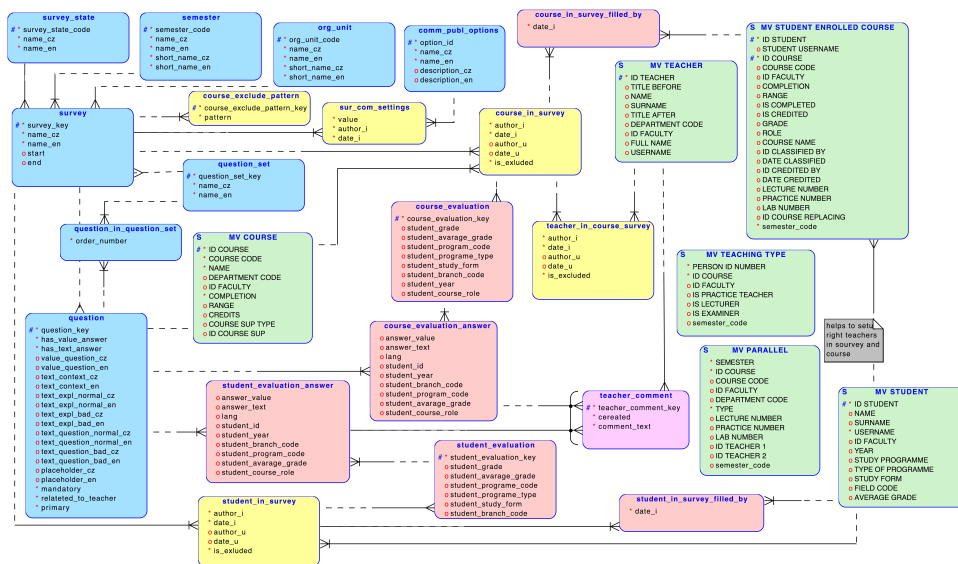
skytovatel služby se používá modul pro HTTP server Apache. V Apache však nelze provozovat aplikace v jazyce Java, a proto je zapotřebí přeměrovávat požadavky na Apache Tomcat. Pro přeměrování se používá modul mod.Jk, který implementuje protokol Apache JServ Protocol zkráceně AJP. Proces tedy začíná na serveru Apache, který použije Shibboleth SP implementaci pro autentifikaci oproti nastavenému IDP. Dále přeošle požadavek na Apache Tomcat, kde aplikace získá údaje zasláné z IDP a přes unikátní identifikátor identifikuje přihlašujícího se uživatele. Poté aplikace vytvoří token typu JWT a přeměruje uživatele na url ,na kterém je spuštěný klient aplikace. Klient obdrží JWT v parametrech URL a může přes token se serverem komunikovat.

3.3.3 Databáze

Použitá databáze je Oracle database. Pro komunikaci s Oracle databází v Java aplikaci je zapotřebí konektor JDBC. JDBC konektory existují pro více databází a proto je zapotřebí použít specifický pro Oracle databázi. Pro možnost stažení konektoru je zapotřebí se registrovat u Oracle. Na databázi se připojuje pomocí přípojovacího textu (connection string). Příklad uveden dále. Návrh a nasazení databáze bylo dodáno Ing. Michalem Valentou, Ph.D.

Příklad connection string pro Oracle JDBC

```
jdbc:oracle:thin:username/password@host:1521:SID
```



Obrázek 3.8: Databázové schéma

Testování

4.1 Scénáře

4.1.1 Scénář 1

Studoval/a jste předmět FIT-NICE, který Vás nadchnul. Líbily se Vám přednášky Ing. Martiny Novákové, Ph.D. a cvičení, která střídavě vedli Ing. John Snow a Ing. Johny Bravo. V anketě chcete pochválit předmět samotný, přednášející a oba cvičící. Zkouška proběhla také skvěle a rád/a byste zdůraznil/a férovost zkoušejícího, který vám přidal 5 minut na vypracování testu, protože musel vysvětlovat nevhodně formulovanou otázku.

1. Začněte hodnotit předmět FIT-NICE, v rámci celkového dojmu mu dejte nejlepší hodnocení
2. Předmět hodnotte také nejlepším možným způsobem.
3. Přidejte vyučujícího Ing. Martinu Novákovou, Ph.D, dejte ji druhé nejlepší hodnocení a do textového pole pro vyučujícího uveďte „Krom pozdních příchodů maximální spokojenost“
4. Přidejte vyučujícího Ing. John Snow, dejte mu druhé nejlepší hodnocení.
5. Vyučujícímu Ing. John Snow přidejte textový komentář: „Je to borec, přežil tolik epizod a ještě umí integrovat z hlavy! Rozhodně není pravda, že nic neví.“
6. Přidejte vyučujícího Ing. Johny Bravo a dejte mu nejlepší hodnocení.
7. Ke zkoušce přidejte textový komentář „Chválím Ing. Johna Brava za to, že nám při testu přidal 5 minut navíc, protože vysvětloval nevhodně formulovanou testovou otázku.“
8. Hodnocení předměty FIT-NICE uložte.

4. TESTOVÁNÍ

Chcete ještě vyplnit anketu, která se týká Menzy ČVUT. Jste veskrze pozitivní člověk s velkým důrazem na výkon, takže Vás těší, že menza je blízko a jiné věci příliš neřešíte. Jinak řečeno: jste plně spokojeni jak s chutí jídla, tak s čistotou.

1. Chuti jídla v menze dejte nejlepší možné hodnocení.
2. Čistotě dejte druhé nejlepší hodnocení.
3. Svoji zkušenost hodnotit nechcete, ponechte ji nevyplněnou.
4. Hodnocení Menzy ČVUT uložte.

4.1.2 Scénář 2

Studoval/a jste předmět FIT-MEH, který byl pro Vás povinný, vůbec Vás nebavil a jste rád/a, že ho máte za sebou. Na druhou stranu musíte uznat, že výkony vyučujících i skladba látky byly v pořádku, dokonce uznáváte, že probíraná témata pro praxi potřebujete. Stejně jako předmět samotný chcete mít i jeho hodnocení rychle za sebou. Celkový dojem z předmětu je neutrální. K předmětu jako takovému máte také neutrální vztah, pouze chcete přidat komentář: „Předmět byl v pořádku, chápu, že je pro praxi důležitý, ale vůbec mě to nebavilo“. Žádnou další zpětnou vazbu dávat nechcete.

1. Začněte hodnotit předmět FIT-MEH, celkový dojem je neutrální.
2. K obsahu předmětu máte také neutrální vztah.
3. K předmětu přidejte komentář: „Předmět byl v pořádku, chápu, že je pro praxi důležitý, ale vůbec mě to nebavilo.“
4. Hodnocení předmětu FIT-MEH uložte.

Dále hodláte vyplnit anketu „Menzy ČVUT“. Jídlo je pro Vás velmi důležité a málokdy Vám chutná něco jiného než, co jste uvařil/a Vy nebo Zdeněk Pohlreich. Zároveň trpíte mírnou obsesí na čistotu. Nejspíš proto je Vaše zkušenost s menzou ČVUT veskrze negativní. Chuti jídla chcete dát nejhorší možné hodnocení, čistotu hodnotíte druhým nejhorším hodnocením. Vaše zkušenost je ovšem neutrální, protože se znáte, očekávali jste katastrofu, ale vlastně to nebylo tak hrozné.

1. Chuti jídla dejte nejhorší možné hodnocení.
2. Čistotě dejte druhé nejhorší hodnocení.
3. Vlastní zkušenost označte jako neutrální.

4. Přidejte textové hodnocení k vaší zkušenosti: „Nikdy, ale opravdu nikdy byste neměli do polévky dávat petržel, když tam správně má být pažitka!“
5. Hodnocení ankety „Menzy ČVUT“ uložte.

4.1.3 Scénář 3

Studoval/a jste předmět FIT-BAD a jste opravdu velmi nespokojen/a. Předmět byl špatně připravený a organizace vážla. Přednášející Ing. Martina Nováková, Ph.D. chodila pozdě, ve výkladu odbíhala od tématu a nedařilo se jí dostatečně jasně odpovídat na dotazy pokládané posluchači. Cvičící Ing. John Snow nerespektoval fakt, že výuka na univerzitě má být apolitická a extra body ze cvičení uděloval výhradně za lajky tweetů @PREZIDENTmluvci. Čas na závěrečný test byl navíc o 20 minut zkrácen, protože dohlížející přednášející byla objednána ke kadeřníkovi.

1. Začněte hodnotit předmět FIT-BAD, v rámci celkového dojmu mu dejte nejhorší možné hodnocení
2. Předmět hodnotte také nejhorším možným způsobem.
3. K předmětu dodejte textovou informaci: „Předmět byl špatně připravený, témata na sebe nenavazovala a nebylo jasné, co je hlavním cílem.“
4. Vyberte vyučující Ing. Martinu Novákovou, Ph.D., dejte jí druhé nejhorší hodnocení.
5. K vyučující Ing. Martině Novákové, Ph.D. přidejte textový komentář: „Chodila pozdě, odbíhala od tématu a nebyla schopna uspokojivě zodpovědět dotazy posluchačů.“
6. Vyberte vyučujícího Ing. John Snow, dejte mu nejhorší hodnocení.
7. K vyučujícímu Ing. John Snow dodejte informaci: „Za získání extra bodů jsme museli na twitteru lajkovat @PREZIDENTmluvci. Béééééééé!“
8. Ke zkoušce přidejte informaci: „Test byl od 20 minut zkrácen, protože přednášející byla objednána ke kadeřníkovi.“
9. Hodnocení předmětu FIT-BAD uložte.

Chcete ještě vyhodnotit anketu „Menzy ČVUT“. Už Vaše maminka vám říkala, že jste trochu zmlsaní. Možná proto vám jídlo v menze příliš nechutná. K čistotě nemáte žádné připomínky, ale protože si těchto věcí příliš nevšímáte, budete ji hodnotit raději neutrálně. Žádnou další zpětnou vazbu pro menzy ČVUT nemáte.

4. TESTOVÁNÍ

1. Chuť jídla označte jako druhou nejhorší.
2. Čistotu označte neutrálně.
3. Vlastní zkušenost hodnotit nechcete, ponechte ji nevyplněnou.
4. Hodnocení „Menzy ČVUT“ uložte

4.2 Výsledky

4.2.1 Tester 1

FIT-NICE

- Jak se liší celkový dojem a předmět? Působí to na mě duplicitně.
- Pozitivně vnímá smajlíky jako způsob vyjádření základní emoce hodnocení předmětu
- BUG John Snow se nezobrazuje jméno

Menza

- OK

FIT-BAD

- Je pěkné, že při špatném hodnocení se rovnou rozbalí možnost vyplnit komentář, plus návodný text

Celkově

- Přehlednější a čistší než aktuální verze
- Smajlíci jsou snadno pochopitelní ve smyslu jestli je to dobré-meh-špatné

4.2.2 Tester 2

FIT-MEH

- Předmět chápe jak na mě působil obsah a využitelnost obsahu
- Celkový dojem se týká lidí, organizace, kvality cvičení
- Zvyklý na opačné pořadí smajlíků
- Co znamená přidat dalšího vyučujícího?
- Proč se mění pořadí textových otázek když něco rozbalím/sbalím?

- Pozitivně vnímá možnost určit si anonymitu hodnocení
 - Co znamená ten vykřičník a jak se mě týká?
- Co je role předmětu?
- Jednoduché
- Přehledné
- Chtěl by výsledky ankety vidět

Menza

- Nic zajímavého nepadlo

4.2.3 Tester 3

FIT-BAD

- Jasný kde jsem
- Dojem = lidský faktor, nějaký konkrétní zážitek
- Předmět = obsah, oprostěno od vyučujících
- Co znamená přidat vyučujícího?
- K čemu je celá sekce zveřejnit?
- Chybí možnost ohodnotit zápočet
- Zveřejňování info by mohlo být předvyplněné podle toho jak jsem to vyplnil jednou
- Nápoředa co se tím kterým dotazem myslí by nebyla na škodu
- Očekává hodnocení vyučujícího jinde (na nové stránce)
- Na HP by uvítal rovnou seznam předmětů

Menza

- Upravit textovou otázku aby bylo jasnější co se tam po uživateli chce

4. TESTOVÁNÍ

4.2.4 Tester 4

FIT-NICE

- Jaký je rozdíl Dojem vs Předmět?
- Chybí uložení rozpracovaného
- Možnost odeslat všechny předměty najednou?
- Jednoduché, rychlé

Menza

- Porovnání menz mezi sebou
- Vybrat kterou navštěvuji

4.2.5 tester 5

FIT-MEH

- Co je to za vyučující?
- Ukládání in progress?
- Řazení rozbalených a sbalených textových otázek!!!
- Zkouška nemá dělení na části, různé zkoušející atp
- Rozlišení hodnocení vyučujícího podle role?
- Hodnocení studijních materiálů?

Menza

- Textová otázka není jasné co se očekává

4.2.6 Tester 6

FIT-BAD

- Smajlíky vypadají na první pohled jako obrázek (ne něco klikatelného)
- Co znamená celkový dojem?
- Předmět = obsah, sylabus
- Zveřejnění oboru u prváků??
- Co je role předmětu a proč mě má zajímat jestli se to zveřejní?

- Řazení textových otázek!!!
- Smajlíkové hodnocení zkoušky může svádět k hodnocení podle výsledku zkoušky
- Nelze uložit rozpracované :(
- Historie vyplněných lístků?
- Co jsou ty vykřičníky?

Menza

- Co je vaše zkušenost?

Závěr

Cílem této magisterské práce bylo analyzovat, navrhnout a implementovat novou verzi systému pro provozování Ankety na škole ČVUT. Analyzoval jsem všechny dostupné údaje a na základě dostupných podkladů a seznamu nedostatků předchozí verze jsem vytvořil návrh řešení při použití vhodných technologií pro implementaci. Implementace probíhala pomocí agilního způsobu vývoje, který se vyznačuje jednotlivými iteracemi pro možné úpravy na základě testování. Na základě těchto výsledků jsem řešení upravil a doplnil jej o nové potřebné funkce.

Následuje popis splnění jednotlivých požadavků zadání této diplomové práce. Požadavek č.1 „Seznamte se s návrhem nové koncepce Ankety ČVUT v diplomové práci Davida Knapa“. Návrh a výsledky sběru požadavků budoucích uživatelů nové aplikace jsem detailně popsal kapitole Analýza. Požadavek č.2 „Diskutujte a zvolte vhodnou architekturu aplikace, implementační platformu a model vývoje aplikace“. Zvážil jsem jednotlivé použitelné technologie v Analýze a v následující kapitole Návrh jsem uvedl nejvhodnější architekturu a implementační platformu. Postup vývoje jsem uvedl v kapitole Implementace. Požadavek č.3 „Dále se zaměřte na uživatelské rozhraní. Diskutujte a dokončete návrh části vyplňování anketních lístků.“ Uživatelské rozhraní je uvedeno v popisu klienta v kapitole Implementace. Návrh vyplňování anketních lístků byl diskutován se zadavatelem a na základě výsledků testování jsem uživatelské rozhraní upravil a dopracoval. Požadavek č.4 „Implementujte část vyplňování anketních lístků.“ Tato implementace je součástí výsledného zdrojového kódu, který je přiložený k této diplomové práci. Požadavek č.5 „Proveďte uživatelské testování a opravte případné nedostatky.“ Testování bylo prováděno iterativně a jednotlivé nedostatky odstraňovány průběžně. Jednotlivé testovací scénáře jsou uvedeny v kapitole Testování. Požadavek č. 6 „Realizovaná aplikace je naplánována na nasazení do produkčního prostředí a možného využití v Zimním Semestru 2018/2019.“ Beta testování se uskuteční až po odevzdání této práce.

Práci na tomto díle jsem získal zkušenosti s výběrem vhodných technologií

ZÁVĚR

a jejich použitím na internetu. Ocenil jsem důležitost zabezpečení a potřeby verzovacího souborového systému při vývoji softwarové aplikace. Webové technologie se neustále mění a vyvíjejí kupředu. Pro vývojáře je tudíž nezbytné, aby investovali dostatek času do sebevzdělávání a seberozvoje, který tento obor vyžaduje.

Literatura

- [1] Knap, David. *Návrh nového řešení aplikace Anketa ČVUT*. Diplomová práce. Praha: České vysoké učení v Praze, Fakulta elektrotechnická, 2018.
- [2] Ecma international, *ECMAScript® 2018 Language Specification, 9th edition* [online]. [cit. 2019-01-09]. Dostupné z: <https://www.ecma-international.org/publications/standards/Ecma-262.htm>
- [3] Jones, M., Bradley, J., Sakimura, N., *JSON Web Token* [online]. 2015 [cit. 2019-01-09]. Dostupné z: <https://tools.ietf.org/html/rfc7519>
- [4] *Syntactically Awesome StyleSheets* [online]. ©2006-2018 [cit. 2019-01-09]. Dostupné z: <https://sass-lang.com/>
- [5] Oracle, *JavaServer Pages Technology* [online]. ©2018 [cit. 2019-01-09]. Dostupné z: <https://www.oracle.com/technetwork/java/javaee/jsp/index.html>
- [6] Microsoft, *TypeScript* [online]. ©2012-2018 [cit. 2019-01-09]. Dostupné z: <https://www.typescriptlang.org/>
- [7] WHATWG, *Hypertext Markup Language* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://html.spec.whatwg.org>
- [8] Linux foundation, *Node.js* [online]. [cit. 2019-01-09]. Dostupné z: <https://nodejs.org/en/>
- [9] WWW Consortium, *Cascading Style Sheets* [online]. ©1994-2018 [cit. 2019-01-09]. Dostupné z: <https://www.w3.org/Style/CSS/Overview.en.html>
- [10] TIOBE Index for January 2019, *TIOBE* [online]. ©2018 [cit. 2019-01-09]. Dostupné z: <https://www.tiobe.com/tiobe-index/>

- [11] Pivotal, *Spring* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://spring.io>
- [12] The Apache Software Foundation, *Apache Tomcat* [online]. ©1999-2019 [cit. 2019-01-09]. Dostupné z: <https://tomcat.apache.org/>
- [13] The Apache Software Foundation, *HTTP server project* [online]. ©1997-2019 [cit. 2019-01-09]. Dostupné z: <https://httpd.apache.org/>
- [14] Ecma international, *C# Language Specification 5th edition* [online]. [cit. 2019-01-09]. Dostupné z: <http://www.ecma-international.org/publications/standards/Ecma-334.htm>
- [15] Microsoft, *ASP.NET* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://www.asp.net/>
- [16] Microsoft, *Internet Information Service* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://www.iis.net/>
- [17] ISO/IEC 9075-1:2016, *Information technology – Database languages – SQL – Part 1: Framework (SQL/Framework)* [online]. [cit. 2019-01-09]. International Organization for Standardization, Geneva, Switzerland. Dostupné z: <https://www.typescriptlang.org/>
- [18] Google, *AngularJS* [online]. ©2010-2018 [cit. 2019-01-09]. Dostupné z: <https://angularjs.org/>
- [19] Google, *Angular* [online]. ©2010-2018 [cit. 2019-01-09]. Dostupné z: <https://angular.io/>
- [20] Facebook Inc., *React* [online]. ©2018 [cit. 2019-01-09]. Dostupné z: <https://reactjs.org/>
- [21] Evan You, *Vue.js* [online]. ©2014-2018 [cit. 2019-01-09]. Dostupné z: <https://vuejs.org/>
- [22] Tilde Inc., *Ember* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://www.emberjs.com/>
- [23] Meteor Development Group Inc., *Meteor* [online]. ©2018 [cit. 2019-01-09]. Dostupné z: <https://www.meteor.com/>
- [24] Find your new favorite web framework, *HotFrameworks* [online]. [cit. 2019-01-09]. Dostupné z: <https://hotframeworks.com>
- [25] Abramov Dan and the Redux documentation authors, *Redux* [online]. ©2015 [cit. 2019-01-09]. Dostupné z: <https://redux.js.org>

-
- [26] Abramov Dan and the Redux documentation authors, *React Redux* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://react-redux.js.org>
- [27] React training, *Internet Information Service* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://reacttraining.com/react-router>
- [28] GitHub Inc., *Redux Thunk* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://github.com/reduxjs/redux-thunk>
- [29] Otto m., @fat, maintained by a core team *Bootstrap* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://getbootstrap.com>
- [30] GitHub Inc., *Reactstrap* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://github.com/reactstrap/reactstrap>
- [31] GitHub Inc., *Axios* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://github.com/axios/axios>
- [32] GitHub Inc., *GitHub.com* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://github.com>
- [33] Stack Exchange Inc., *Stackoverflow.com* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://stackoverflow.com>
- [34] Shibboleth consortium Inc., *Shibboleth* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://www.shibboleth.net>
- [35] Gradle Inc., *Gradle* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://gradle.org>
- [36] Apache Software Foundation, *Maven* [online]. ©2002-2019 [cit. 2019-01-09]. Dostupné z: <https://maven.apache.org>
- [37] [SAML v2.0 Errata 05] SAML Version 2.0 Errata 05. 01 May 2012. OASIS Approved Errata. Dostupné z <http://docs.oasis-open.org/security/saml/v2.0/errata05/os/saml-v2.0-errata05-os.html>.
- [38] Josefsson, S., *The Base16, Base32, and Base64 Data Encodings* [online]. ©2006 [cit. 2019-01-09]. Dostupné z: <https://tools.ietf.org/html/rfc4648>
- [39] Kelly, S., Frankel, S., *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec* [online]. ©2007 [cit. 2019-01-09]. Dostupné z: <https://tools.ietf.org/html/rfc4868>
- [40] Roskind, J., Sabim, M., Simon, D., Weinstein, T., Wright, T., Inc., *The Transport Layer Security (TLS) Protocol Version 1.2* [online]. ©2019 [cit. 2019-01-09]. Dostupné z: <https://tools.ietf.org/html/rfc5246>

Seznam použitých zkratk

HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JSON	JavaScript Object Notation
JWT	JSON Web Tokens
SQL	Structured Query Language
VCS	Version control system
SSO	Single sign-on
SAML	Security Assertion Markup Language
IDP	Identity provider
SP	Service provider
IIS	Internet Information Services
JSP	JavaServer Pages
API	Application programming interface
SASS	Syntactically Awesome StyleSheets
SPA	Single page application
AJAX	Asynchronous JavaScript And XML
RDBMS	Relational database management system
JDBC	Java Database Connectivity

A. SEZNAM POUŽITÝCH ZKRATEK

URL Uniform Resource Locator

Obsah přiloženého CD

	readme.txt	stručný popis obsahu CD
	src	
	impl	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu \LaTeX
	text	text práce
	thesis.pdf	text práce ve formátu PDF