



Zadání bakalářské práce

Název:	Anketa ČVUT - redesign vyplňovacího modulu
Student:	Nam Nguyen Hai
Vedoucí:	Ing. Michal Valenta, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Webové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2021/2022

Pokyny pro vypracování

Cílem práce je návrh, implementace a otestování nové verze uživatelského rozhraní vyplňovacího modulu Ankety ČVUT verze 3.

Postupujte v těchto krocích:

- 1) Analyzujte současné řešení vyplňovacího modulu Ankety ČVUT verze 3 zejména z hlediska uživatelského rozhraní.
- 2) Specifikujte požadavky na implementaci nové verze vyplňovacího modulu. Spolupracujte s vývojovým týmem Ankety ČVUT.
- 3) Implementujte nové uživatelské rozhraní dle požadavků z předchozího kroku.
- 4) Implementované uživatelské rozhraní vhodně otestujte.
- 5) Implementaci zdokumentujte.



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalárska práca

Anketa ČVUT - redesign vyplňovacího modulu

Nam Nguyen Hai

Katedra softwarového inženýrství

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

27. júna 2021

Pod'akovanie

Ďakujem Ing. Michalovi Valentovi, Ph.D. za vedenie mojej bakalárskej práce, podporu a možnosť pracovať na reálnom projekte. Ďalej chcem poďakovať vývojovému tímu Anketa ČVUT, najmä Bc. Jakubovi Junovi za jeho rady a pomoc. Tiež chcem poďakovať svojej rodine, skupine FITnamci a Bc. Ivanke Hacajovej, ktorí mi poskytli mentálnu podporu počas celého štúdia.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 27. júna 2021

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2021 Nam Nguyen Hai. Všetky práva vyhrazené.

Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.

Odkaz na túto prácu

Nguyen Hai, Nam. *Anketa ČVUT - redesign vyplňovacího modulu*. Bakalárska práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Táto práca sa venuje redesignu vyplňovacieho modulu Anketa ČVUT, ktorá slúži na zber spätnej väzby od študentov na univerzite. Pretože pôvodný modul obsahuje chyby a nedostatky, tím Anketa ČVUT sa ho rozhodol prepísať do nového frameworku Vue.js.

Pri návrhu a implementácii sa riešia základné problémy ako smerovanie, prepínanie jazyka, rôzne UX/UI návrhy užívateľského rozhrania a redesign nového vyplňovacieho modulu vo frameworku Vue.js.

Výsledok tejto práce je funkčná, otestovaná aplikácia, ktorú budú používať v budúcnosti študenti z univerzity. Ďalším prínosom práce je zjednodušenie práce pre tím Anketa ČVUT pri ďalšom vývoji a udržateľnosť systému vďaka kompletnej dokumentácii nového modulu. Spolu s vývojovým tímom Anketa ČVUT sa predpokladá integrácia a nasadenie nového modulu do produkčného používania do konca roku 2021.

Kľúčová slova Anketa ČVUT, Vue.js, NuxtJS, frontend, užívateľské rozhranie, JavaScript

Abstract

This thesis deals with the redesign of the filling module of the CTU Survey, which is used to collect feedback from students at the university. Because the original module contains bugs and flaws, the CTU Survey team decided to rewrite it in the new Vue.js framework.

During the design and implementation, basic issues such as routing, language switching, various UX/UI designs, and redesign of the new filling module in the Vue.js framework were addressed.

The result of this work is a functional and tested application that will be used in the future by students from the university. Another benefit of the work is the simplification of the work of the CTU Survey team in further development and sustainability of the system due to the complete documentation of the new module. The CTU Survey development team expects the new module to be integrated and deployed for production use by the end of 2021.

Keywords CTU Survey, Vue.js, NuxtJS, frontend, user interface, JavaScript

Obsah

Úvod	1
1 Cieľ práce	3
2 Analýza pôvodného vyplňovacieho modulu	5
2.1 Užívateľské rozhranie	5
2.1.1 Popis rozhrania	5
2.1.2 Sťažnosti od užívateľov	7
2.2 Implementácia a údržba zdrojového kódu	7
3 Požiadavky na implementáciu novej verzie vyplňovacieho modulu	9
3.1 Nefunkčné požiadavky	9
3.2 Funkčné požiadavky	9
4 Návrh designu a technológie	11
4.1 Nový design užívateľského rozhrania	11
4.1.1 Vyplňovací modul	11
4.1.2 Prepínanie jazyka a lokalizácia textu	11
4.1.3 Navigačná lišta	12
4.1.4 Nástenka	12
4.2 Programovacie jazyky a frameworky	12
4.2.1 Framework versus knižnica	12
4.2.2 JavaScript	13
4.2.3 HTML	13
4.2.4 CSS	13
4.2.5 Vue.js	13
4.2.6 NuxtJS	15
4.2.7 Json-server	15
4.3 Návrh štruktúry komponentov nového vyplňovacieho modulu	15

4.4	Návrh nového REST API	16
4.5	Návrh novej nástenky	17
5	Implementácia a dokumentácia	19
5.1	Vývojové prostredie	19
5.2	Príprava dát na testovanie - lokálny server REST API	20
5.3	Smerovanie	21
5.4	Lokalizácia textu a prepínanie jazyka	22
5.5	Rozbor komponentov vyplňovacieho modulu	24
5.5.1	Analýza dát z backendu	24
5.5.2	Zaobstaranie dát z backendu	25
5.5.3	Tvorba dielčích komponentov pre <i>CourseRating</i>	26
5.5.4	Štruktúra šablóny	28
5.5.5	Validácia vstupu	29
5.5.6	Odoslanie užívateľských vstupov do backendu	30
5.6	Nástenka	32
5.7	Navigačná lišta	34
5.8	Autentizácia a autorizácia	35
5.8.1	Autentizácia	35
5.8.2	Autorizácia	35
5.9	Dokumentácia	36
6	Testovanie	37
6.1	Užívateľské testovanie	37
6.1.1	Scenár 1	38
6.1.2	Scenár 2	39
6.1.3	Scenár 3	40
6.1.4	Zhrnutie	40
	Záver	43
	Literatúra	45
	A Prílohy	49
	B Zoznam použitých skratiek	53
	C Obsah priloženého CD	55

Zoznam obrázkov

2.1	Anketový lístok na hodnotenie kurzu	6
2.2	Sledovanie prichádzajúcich zdrojov z API.	8
4.1	Návrh navigačnej lišty desktopovej a mobilnej verzie v programe FIGMA.	12
4.2	Návrh novej nástenky s časovými údajmi.	18
5.1	Štruktúra priečinka <i>pages</i> na generovanie dynamického smerovania.	22
5.2	Markdow editor a možnosť zverejňovať osobné údaje (na obrázku vpravo).	28
5.3	Štruktúra šablón nového modulu.	29
5.4	Výsledok po rozdelení zo zdrojového kódu 5.12 (použitie grid system).	29
5.5	Chybová hláška pri nevyplnenom smajlíkovom hodnotení.	30
5.6	Modálne okno na upozornenie užívateľa pri odoslaní.	31
5.7	Hláška pri úspešnom odoslaní dát.	32
5.8	Chybová hláška pri neúspešnom pokuse o odoslanie dát a zobrazenie príčiny chyby.	32
5.9	Nová nástenka (Moje lístky).	33
5.10	Navigačná lišta na desktopovom zariadení.	34
5.11	Navigačná lišta na mobilnom zariadení.	34
5.12	Popis navigácie v Anketa ČVUT wiki.	36
A.1	Anketový lístok predmetu „Operační systémy.“	50
A.2	Nová nástenka.	51
A.3	Prihlasovacia stránka.	52

Úvod

Anketa ČVUT je informačný systém, ktorý každý semester zbiera cenné hodnotenie študentov k výuke a prevádzke všetkých fakúlt patriacich pod České vysoké učení technické. Anketa pomáha k zlepšeniu výuky tak, že študenti adresujú svoje hodnotenie (či už kladné, alebo záporné) k jednotlivým predmetom. Vyučujúci tým pádom dostávajú spätnú väzbu, podľa ktorej prispôbia svoje materiály a spôsob hodnotenia alebo výučby.

Systém postupne vznikol ako záverečné práce absolventov na tejto univerzite, a preto je vyvíjaný po častiach, v rôznych jazykoch a s rôznymi konvenciami, čiže neexistuje spôsob, ako systém rozumne spravovať.

Táto práca nadväzuje na diplomovú prácu Ing. Vojtěcha Štecha, ktorý vytvoril modul na vyplňovanie anketových lístkov v technológii React. V roku 2020 implementoval ako svoju bakalársku prácu Bc. Jakub Jun modul užívateľské rozhranie pre správu ankety vo Vue.js, a zároveň prepísal väčšinu častí ankety do tohto frameworku. Aby systém bol jednotný a udržateľný, tak je treba navrhnuť nový design pre vyplňovací modul a prepísať ho do Vue.js.

Autor si vybral túto tému, pretože by veľmi rád svojou časťou prispel do reálneho bežiaceho projektu, ktorý používajú tisíce študentov a vyučujúcich na tejto univerzite. Zároveň je táto práca pre autora veľkou príležitosťou naučiť sa novým veciam, konkrétne frontendu, keďže počas štúdia a práce pracoval len s backendovou časťou systému.

Cielom teoretickej časti je analyzovať pôvodný stav vyplňovacieho modulu. V praktickej časti je detailný popis implementácie jednotlivých komponentov, aby splnili funkčné a nefunkčné požiadavky.

Cieľ práce

Hlavným cieľom tejto bakalárske práce je navrhnuť, implementovať a vhodne otestovať nový vyplňovací modul do ankety ČVUT. Preto je potrebné zanalyzovať stávajúce riešenie a jeho nedostatky z hľadiska funkčnosti a udržateľnosti. Ďalším cieľom je oboznámiť sa s frameworkom Vue.js, jeho metaframeworkami¹ a vhodnými frontendovými praktikami a postupmi k tvorbe nového modulu. Posledným cieľom práce je jeho samostatná implementácia a pokrytie testami, aby nedošlo k bezpečnostnému incidentu.

¹Metaframework je framework, ktorý je vybudovaný nad iným frameworkom.

Analýza pôvodného vyplňovacieho modulu

Prvá kapitola sa zaoberá popisom štruktúry a fungovania pôvodného modulu. Tento modul implementoval Ing. Vojtěch Štech v rámci svojej diplomovej práce „Anketa ČVUT - verze 3.0 - vyplňování anketních lístků“ [1] v knižnici React [3] s použitím frameworku pre kaskádové štýly Bootstrap [5].

React je populárna javascriptová knižnica na vývoj mobilnej aplikácie alebo frontendu, vyvíjaná vo Facebooku. Táto knižnica získa stav aplikácie a vykresľuje tento stav do DOM [4] (Document Object Model). Inak povedané, iba pomáha vykresliť stránku pri zmene stavu aplikácie. Aby aplikácia bola plne funkčná, je potrebné doplniť ďalšie knižnice, ktoré zaoštarajú napríklad smerovanie² alebo zaoštaranie dát.

Bootstrap je ďalší veľmi populárny framework, ktorý poskytuje programátorom preddefinované sady nástrojov, designov, štýlov a komponentov. Vhodná je tiež pre mobilný vývoj, kde je ťažké zabezpečiť responzivitu pri zmene veľkosti obrazovky.

2.1 Uživatelské rozhranie

2.1.1 Popis rozhrania

Rozhranie obsahuje 3 veľké komponenty:

- Navigačná lišta, ktorá obsahuje logo ČVUT, meno prihlasovaného užívateľa a čas, kedy vyprší autorizačný token.
- Bočná navigačná lišta, ktorá má v sebe odkazy do rôznych častí aplikácie. Pokiaľ má užívateľ právo vyplniť anketu, ponúkne sa mu tiež odkaz na ňu.

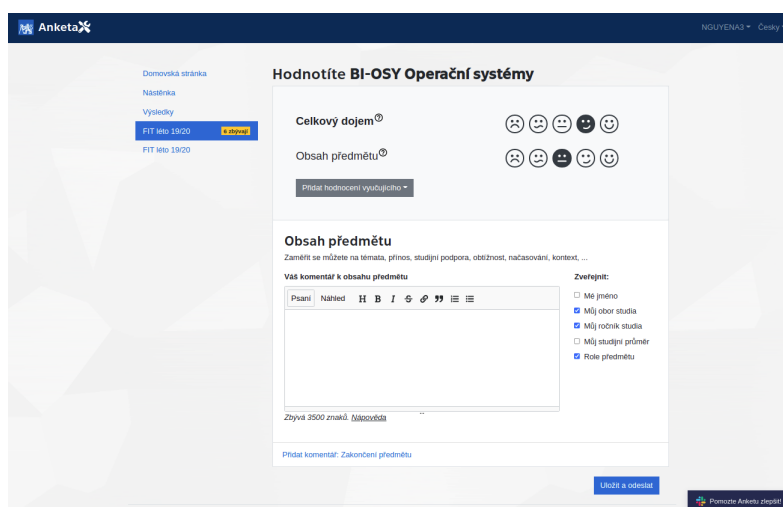
²Smerovanie označuje činnosť, kde aplikácia rozhodne, ktorý kus kódu je spustený pri zmene URL.

2. ANALÝZA PÔVODNÉHO VYPLŇOVACIEHO MODULU

- Obsahová časť, ktorá vykresľuje obsah daného odkazu.

Aby sa užívateľ dostal do anketového lístku, je potrebné kliknúť na odkaz príslušnej ankety, a potom si zvoliť príslušný kurz (predmet), ktorý by chcel zhodnotiť. Na obrázku 2.1 je vidieť anketový lístok vybraného predmetu „BI-OSY Operační systémy“.

Tento design navrhol Ing. David Knap v rámci svojej diplomovej práce „Návrh nového řešení aplikace Anketa ČVUT“ [6]. Podľa spätnej väzby sa užívateľom veľmi páčilo pôvodné užívateľské rozhranie.



Obr. 2.1: Anketový lístok na hodnotenie kurzu

Anketový lístok obsahuje:

Smajlíkové hodnotenie Na výber existuje 5 možných smajlíkov, ktoré predstavujú známky od 1 až po 5 a vyjadrujú spokojnosť užívateľov ku predmetom.

Textové hodnotenie V textovom poli je možné používať Markdown prvky, ktoré ponúkajú viac možností formátovania textu. Tento formát je potom upravený na strane serveru, aby nedošlo k XSS útokom [7]. Podľa toho aké hodnotenie užívateľ vybral, sa zmení nápoveda pri textovom hodnotení.

Možnosť zverejňovať užívateľské údaje Užívateľ sa rozhodne, ktoré údaje (napr. meno, fakulta, apod.) chce o sebe zverejniť pri odoslaní anketového lístku. V niektorých anketách táto možnosť nie je, pretože je táto informácia dôležitá pri hodnotení relevantnosti podnetu na strane fakulty.

2.1.2 Sťažnosti od užívateľov

Od ostrého nasadenia v priebehu jedného roku dostal vývojový tím cennú spätnú väzbu od užívateľov. Vývojový tím zanalyzuje každý podnet a zhodnotí, či sa jedná o chybu alebo požiadavku k implementácii. Veľmi často prichádzajú aj chyby alebo podnety, ktoré nie sú relevantné pre túto aplikáciu. Jedna z najkritickejších chýb bola strata dát v aplikácii [8], ktorá bola ale hneď opravená.

Medzi najčastejšie sťažnosti patrí:

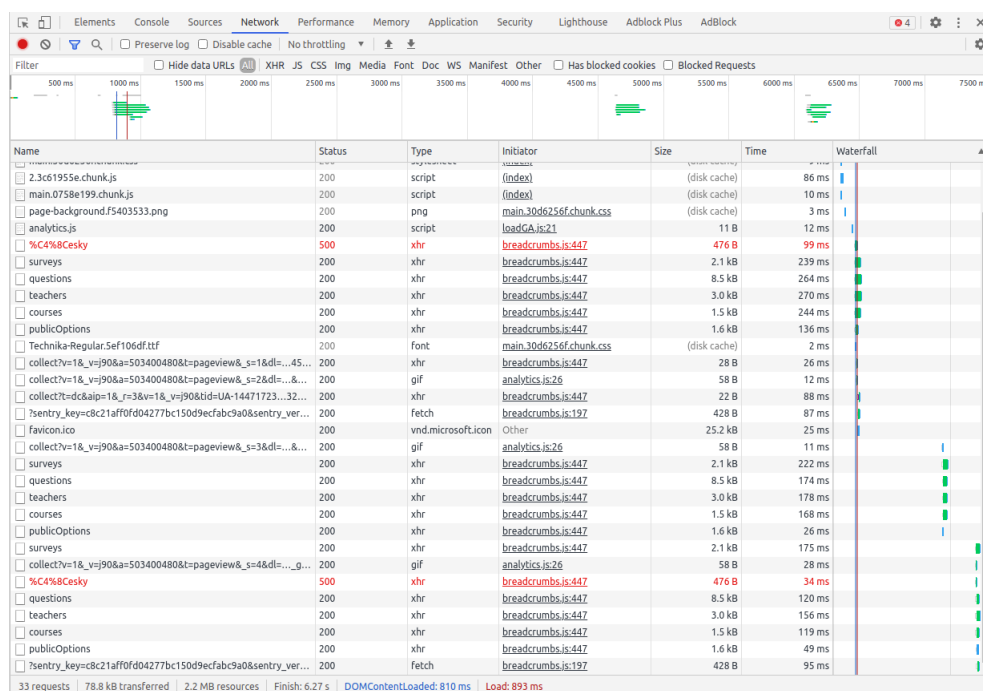
- Nie je možné prísť priamo na URL [9], ktoré smeruje k anketovým lístkom daného predmetu. Namiesto toho presmeruje aplikáciu späť na úvodnú stránku. Na želaný prístup aplikácia pôvodne nebola vôbec navrhnutá.
- Nie je možné zmeniť jazyk cez URL. Tento problém tiež ovplyvňuje funkcionality, ktorá je spomenutá vyššie, kedy vyučujúci nemôžu poslať odkaz na anketu pre študentov anglického štúdia.
- Po zmene jazyka presmeruje aplikácia užívateľa späť na úvodnú stránku, bez ohľadu na to, na akom odkaze sa užívateľ nachádza.
- Nie je možné nastaviť rôzne zverejňovacie nastavenia ku každému textovému hodnoteniu samostatne.
- Pri každej chybe sa zobrazuje „Problém s pripojením k sieti“, čo výrazne komplikuje prácu vývojovému tímu, schopnosť dohľadať a opraviť chybu [10].

2.2 Implementácia a údržba zdrojového kódu

Pri prvom pohľade do zdrojového kódu a pri použití aplikácie je hneď vidieť, že to mala byť jednostránková aplikácia (singlepage application), ďalej len SPA. Tento typ aplikácie prepíše stránku, namiesto načítania zo serveru a jej znovu vykreslenia. Tým pádom jednoznačne znižuje záťaž serveru, ale aj čas užívateľom, ktorí nemusia čakať, kým prehliadač stránku znova vygeneruje. Je dôležité povedať, že SPA **si pýta dáta, len keď ich potrebuje** [11]. Po dlhšom pozorovaní cez vývojársku časť prehliadača je vidieť, že aplikácia momentálne vyžaduje všetky možné dáta z API, čiže aj tie, ktoré v danej chvíli nepotrebuje. Obrázok 2.2 ukazuje prichádzajúce zdroje API pri načítaní úvodnej stránky, pri čom žiadne dáta nepotrebuje.

V zdrojovom kóde chýba veľká časť dokumentácie. Chýba aj začlenenie komponentov do jednotlivých zložiek, ktoré komunikujú so sebou. Občas je možno vidieť aj nekonzistentné pomenovanie funkcií. Z histórie Git repozitáru [12] sa dá ľahko zistiť, že kód nebol priebežne udržiavaný a vôbec nie je pripravený na ďalší vývoj.

2. ANALÝZA PÔVODNÉHO VYPLŇOVACIEHO MODULU



Obr. 2.2: Sledovanie prichádzajúcich zdrojov z API.

Požiadavky na implementáciu novej verzie vyplňovacieho modulu

Počas spolupráce s vývojovým tímom Anketa ČVUT boli upresnené požiadavky na nový vyplňovací modul. Požiadavky vychádzajú zo zozbieranej spätnej väzby a skúseností s anketou. Funkčnosť modulu musí tiež zodpovedať požiadavkám z pôvodnej práce „Anketa ČVUT - verze 3.0 - vyplňování anketních lístků“ [1].

3.1 Nefunkčné požiadavky

NFP-1 Vue.js Nový modul musí byť implementovaný vo frameworku Vue.js.

NFP-2 Responzivita Uživatelské rozhranie musí byť responzívne nie len na desktopovom, ale aj na mobilnom zariadení [13].

NFP-3 Bezpečnosť Nový modul nesmie obsahovať kritické bezpečnostné chyby, ktoré sú známe. Najmä sa jedná o stratu dát alebo XSS útoky [14].

3.2 Funkčné požiadavky

FP-1 Navigácia Prvý požiadavok prichádza priamo od vývojového tímu, a to, aby sa zjednotili navigačné lišty. V rozhraní sú teraz 2 komponenty, ktoré zdieľajú svoju funkcionálnosť [15].

FP-2 Pristúpiť k anketovému lístku cez URL Aplikácia musí umožniť prihlásenému užívateľovi pristupovať cez URL k anketovému lístku. Toto

3. POŽIADAVKY NA IMPLEMENTÁCIU NOVEJ VERZIE VYPLŇOVACIEHO MODULU

uľahčí prácu najmä vyučujúcim na univerzite, ktorí posielajú študentom URL na anketu k svojmu kurzu.

FP-3 Zmena jazyka cez URL Požiadavok je úzko spojený s požiadavkom FP-2, pretože na škole máme anglicky hovoriacich študentov, ktorí by určite radi zdieľali svoje názory na výuku.

FP-4 Po zmene jazyka neprichádza k presmerovaniu Pri zmene jazyka nedochádza k presmerovaniu na úvodnú stránku.

FP-5 Redesign REST API V predošlej kapitole bolo spomínané, že modul nepracuje rozumne s dátami z backendu. Z tohto dôvodu práca zahŕňa aj redesign REST API na backendovej časti systému.

FP-6 Nástenka Pôvodná nástenka obsahuje odkazy na ankety. Tieto informácie sú však aj v navigačnej lište. Nová nástenka bude ukazovať štatistická dáta o anketách a prístupy k nej.

FP-7 Prihlasovanie Aplikácia umožňuje jednoduché prihlásenie užívateľov, zachytávanie tokenov a limitovanie prístupu pre neprihlásených užívateľov.

Návrh designu a technológie

Tretia kapitola sa zaoberá návrhom nového designu vyplňovacieho modulu. Nechýba však detailný popis novej technológie, jej spôsob fungovania a výhody oproti pôvodnému riešeniu, ktoré uľahčujú prácu programátorom pri tvorbe aplikácií.

4.1 Nový design užívateľského rozhrania

Ako bolo spomenuté v kapitole 2, pôvodný modul nie je možné jednoducho rozšíriť a vyriešiť problémy, ktoré boli uvedené v zozname spätných väzieb. Po analýze problému sa rozhodlo, že je najlepšie prepísať celý modul do frameworku Vue.js s použitím metaframeworku NuxtJS [22]. Jeden z dôvodov je tiež ten, že je to ľahko udržateľný a veľmi jednoduchý framework na učenie. Okrem toho je súčasťou tímu Anketa ČVUT osoba, ktorá je expertom v týchto frameworkoch. Nasledujúce návrhy riešia problémy z hľadiska funkčnosti. Grafický návrh zostáva v pôvodnej implementácii od Ing. Vojtěcha Štecha [1] s návrhom Ing. Davida Knapa [6].

4.1.1 Vyplňovací modul

Pred návrhom modulu je treba preskúmať pôvodný zdrojový kód a komunikáciu medzi komponentami. Nový vyplňovací modul bude mať v podstate obdobné komponenty ako pôvodné riešenie. Očakáva sa, že bude schopný prispôbovať sa podľa rôznych dát načítavaných z backendu.

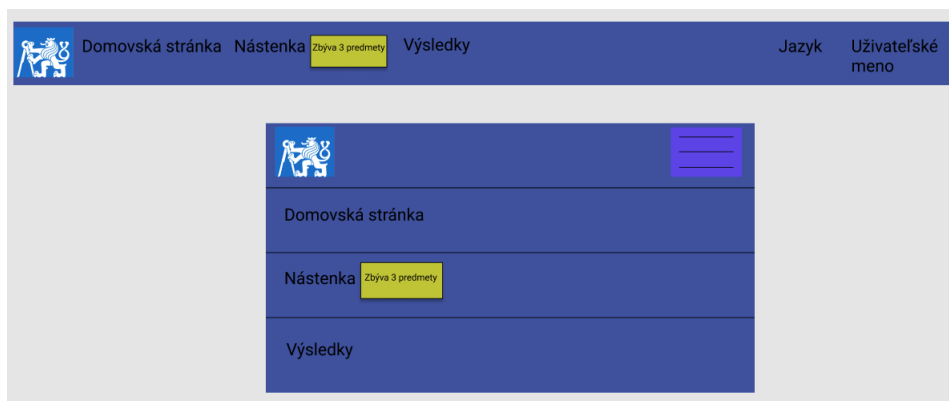
4.1.2 Prepínanie jazyka a lokalizácia textu

Framework NuxtJS umožňuje pre vývojára jednoducho a dynamicky definovať smerovacie linky podľa štruktúry priečinka v *pages*. Existuje mnoho podporovaných frameworkov na lokalizáciu textu. Najpopulárnejšie riešenie je *vue-i18n* [28], ktoré má derivát na podporu NuxtJS. Táto knižnica tiež rieši problém

prepínania jazyka za behu aplikácie, a to prostredníctvom URL alebo cez možnosť v navigačnej lište. Pri návrhu sa počíta s tým, že nový modul bude kompletne lokalizovaný v českom a anglickom jazyku.

4.1.3 Navigačná lišta

Z FP-1 vyplýva, že je treba zjednotiť 2 navigačné lišty do jednej. Podľa *best practices* by mala byť navigačná lišta jednoznačná, konzistentná a viditeľná na stránkach [27]. Po konzultácii s vývojovým tímom sa teda rozhodlo odstrániť bočnú navigačnú lištu a zjednotiť ju s hlavnou navigačnou lištou, ktorá je umiestnená v hornej časti aplikácie. Predpokladom je, že po zjednotení sa uvoľní veľká časť priestoru pre iné komponenty.



Obr. 4.1: Návrh navigačnej lišty desktopovej a mobilnej verzie v programe FIGMA.

4.1.4 Nástenka

V pôvodnom module je nástenka bez účelu. Obsahuje linky, ktoré smerujú na jednotlivé ankety. Podobnú funkcionality má aj bočná navigačná lišta. Nová nástenka bude zobrazovať čas a informácie k jednotlivým kurzom.

4.2 Programovacie jazyky a frameworky

4.2.1 Framework versus knižnica

[16] Veľmi často sa v tejto práci spomínajú slová „framework“ a „knižnica“. Pri oboch platí, že sa jedná o cudzí kus kódu, ktorý rieši nejaký známy problém. Framework volá kód ktorý napísal vývojár a nastavuje funkcionality aplikácie podľa nejakého vzoru. U knižnic je opačný postup, kedy vývojár volá kód knižnice. Tento programovací princíp sa inak nazýva Inversion of Control [17].

4.2.2 JavaScript

JavaScript [20] je interpretovaný, skriptovací a objektovo orientovaný jazyk. Vznikol ako implementácia štandardu ECMA-262 [19]. Ako základ každého webového prehliadača beží JavaScript počas behu aplikácie, to znamená, že netreba kompilovať zdrojový kód do binárky a ručne spúšťať. Vďaka tejto vlastnosti pridáva tento jazyk dynamickosť webovým stránkam, t.j. umožňuje pohyb prvkov po webovej stránke. Jazyk je možné používať pre vývoj webových ale aj desktopových aplikácií.

4.2.3 HTML

HyperText Markup Language [24] je značkovací jazyk pre webové technológie. Tento jazyk definuje štruktúru a sémantiku webovej stránky (dokumentu). Veľmi častá chyba je zneužívanie značiek na aplikovanie kaskádových štýlov. K tejto téme existuje špecifikácia [25], kde je detailný popis významu jednotlivých značiek. Základné prvky HTML sú značky a elementy. Element obsahuje značky a jeho obsah. Značka definuje miesto, kde element začína a končí. Do značiek je možné pridať atribúty, ktoré definujú vlastnosti elementu.

4.2.4 CSS

CSS [26] (Cascading Style Sheet), alebo kaskádové štýly, je jazyk na definovanie vzhľadu dokumentu, ktorý je napísaný v HTML jazyku. Jeho cieľ je oddeliť sémantickú časť a vzhľadovú časť dokumentu. Tento jazyk umožňuje vývojárom definovať kompletný vzhľad (veľkosť, farba, pozícia) jednotlivých elementov na webovej stránke.

4.2.5 Vue.js

Vue.js [21] je frontendový framework vytvorený Evanom You počas pôsobenia v Google. Jeho filozofia spočívala v tom, že zoberie z každého frontendového frameworku to najlepšie a spojí to do jedného. Vue.js používa tzv. komponenty (podobná myšlienka ako v technológii React), ktoré predstavujú kus kódu, funkcionality, ktorá sa prepoužíva. Navyiac si prevzal myšlienku dvojsmerného posielania dát medzi komponentami od frameworku AngularJS [23]. Keďže Vue.js má širokú ponuku knižníc; výborne spracovanú dokumentáciu, ktorá vedie vývojára krok za krokom a každý komponent má pevnú štruktúru, tak oproti Reactu je Vue.js priateľnejšia pre nového vývojára.

V nasledujúcej podkapitole budú popísané základné prvky Vue.js, ktoré treba pochopiť pri návrhu a vývoji v tejto technológii.

Komponent: Komponent [31] je znovu použiteľná inštancia Vue.js s vlastnými prvkami HTML. Z iného uhlu pohľadu sa dá predstaviť aj ako funkcia, ktorá sa dá opakovane používať alebo zavolať v rámci inej funkcie. Komponent vo Vue.js používa koncept „Single File Components“.

Tento koncept znamená, že celá potrebná logika a štruktúra komponentu je definovaná v jednom súbore. Súbor je zakončený *.vue*. Telo komponentu sa skladá z 3 častí:

Template: V tejto časti sa definuje HTML štruktúra komponentu, ktorá je vlastne aj šablónou. V šablóne je možné používať premenné, javascriptový syntax a validné HTML elementy. Vue.js je známy vďaka „Vue directives“. Táto vlastnosť dodá HTML elementom možnosť používať podmienky, cykly a viazať do elementu hodnotu premennej.

Script: Podobne ako HTML stránky, Vue.js tiež používa jazyk JavaScript, ktorý pridáva dynamickosť komponentu. V tomto bloku sa definuje logika (metódy), spôsob fungovania a stavy komponentu. Dokonca môžeme zdefinovať, čo sa bude diať pred vytvorením komponentu, po jeho vytvorení a pri jeho generovaní do DOM.

Style: Ako väčšina HTML stránok, aj Vue.js komponent má možnosť definovať vlastný CSS štýl. Tento CSS blok sa môže často vynechať, pretože sa zvyčajne používa Bootstrap na zjednodušenie vývoja. Jedna špecifická vlastnosť tohoto bloku je možnosť používať `<style scope>`. Štýly v tomto priestore nebudú viditeľné mimo komponent.

Obojsmerné posielanie dát: Aby bol komponent znovu použiteľný, musí vedieť prijímať parametre podľa potrieb vývojára. Vue.js komponent prijíma parametre pomocou *props* v skriptovacom bloku. Počet *props* je ľubovoľný podľa potrieb komponentu. Druhý smer je poslať stav z komponentu cez *event emitters*. V rámci jedného komponentu môže byť len jeden *event emitter*. Komponent komunikujúci s druhým komponentom zachytáva tento stav cez *v-model*.

```
1 <template>
2   <div>
3     {{ variable }}
4   </div>
5 </template>
6
7 <script>
8 export default {
9   data() {
10    return {
11      variable: "Hello World!"
12    }
13  }
14 }
15 </script>
16
17 <style></style>
```

Zdrojový kód 4.1: Príklad Vue.js komponentu zobrazujúci „Hello World!“

4.2.6 NuxtJS

NuxtJS je metaframework používaný na vývoj aplikácií napísaný vo frameworku Vue.js. Tento metaframework poskytuje abstraktné konfigurácie pre aplikáciu a zjednodušuje proces vývoja. V tejto práci jednoznačne pomôže vyriešiť problém smerovania, nastavenia globálnych pluginov a pomocných knižníc.

4.2.7 Json-server

Bežnou úlohou vývojárov frontendu je simulovať backendovú službu REST, ktorá má frontendovej aplikácii doručiť nejaké dáta vo formáte JSON a uistiť sa, že všetko funguje podľa očakávania. Uvažovalo sa nad plnohodnotným backendovým serverom, ale nastaviť a nakonfigurovať ho by stálo veľa času. Json-server je jednoduchý projekt, pomocou ktorého je možné rýchlo nastaviť funkčný REST API s operáciami CRUD [29].

4.3 Návrh štruktúry komponentov nového vyplňovacieho modulu

Pred návrhom je potrebné naštudovať spôsob fungovania Vue.js komponentu a princíp obojsmerného posielania dát, ktoré je vysvetlené v kapitole 4.2.5. Ponúka sa otázka, či je lepšie implementovať všetko do jedného komponentu alebo rozdeliť na menšie komponenty s dielčou funkcionalitou. Pri programovaní je nevyhnutné otestovať svoj produkt, či naozaj funguje podľa očakávania vývojára. Otestovať veľký komponent je pomerne náročné, pretože obsahuje veľa logiky a náväzností na iné komponenty. Ďalším dôvodom je údržba zdrojového kódu a rýchlosť pri zmene logiky komponentu [32].

Nový vyplňovací modul je možné si predstaviť ako veľký komponent, ktorý bude používať menšie komponenty na získavanie vstupu od užívateľov. Zároveň bude primárnym zdrojom dát a bude zaisťovať načítanie z REST API a poslanie získaného vstupu od užívateľov do backendu. Tento komponent sa bude nazývať *CourseRating*.

V ďalšom kroku je treba vytipovať veci, ktoré sa opakujú a je potrebné ich naprogramovať. Tieto komponenty nebudú obsahovať vo svojom stave primárny zdroj dát. Budú len odkazovať svoje stavy k rodičovskému komponentu *CourseRating*.

Zoznam komponentov, ktoré bude používať *CourseRating*:

1. Smajlíkové hodnotenie (*CRSmileyRating*): Výstup bude hodnotenie od 1 po 5.
2. Textové hodnotenie (*CRTextRating*): Výstup bude hodnota vstupného reťazca.

3. Štítok s nápovedami, ktoré ideálne zobrazujú text po nájdení myšou (on hover) (*CRNameWithHelper*): Vstup bude *reťazec* pre štítok a nápovedu.
4. Možnosť zverejňovať osobné údaje o užívateľovi (*CRPublicSetting*): Vstup bude objekt, ktorý zodpovedá možnostiam výberu. Výstup bude možnosť zvolená počas behu aplikácie, prípadne predvolená hodnota z backendu.

4.4 Návrh nového REST API

V pôvodnom module sa z REST API načítajú všetky dáta, niekde sa uložia, a potom sú rozdelené pre jednotlivé komponenty. Na prvý pohľad nie je jasné, ktorý komponent vyžaduje ktoré dáta. Načítať všetky dáta po prihlásení môže byť náročné, pokiaľ užívateľ navštevuje veľa kurzov počas semestra.

Pri podrobnejšom preskúmaní dát sa zistilo, že dáta môžu byť nezmyselné (napríklad sa na vstupe objavujú číselné hodnoty).

Pri návrhu nového REST API sa používajú *best practices* (osvedčené postupy). Jeho štruktúra bude samopopisná, jasná a dá sa použiť vo viacerých komponentoch. Endpoint [18] predstavuje koncový bod, cez ktorý API komunikuje. Počas komunikácie (dotazovania) získava aplikácia dáta podľa tvaru URL a hlavičky, ktoré API dostane.

Formát výstupu je JSON a nové endpointy budú podporovať GET operáciu.

Zoznam nových endpointov, ktoré sú využívané v novom module:

/surveys Vrátí všetky ankety dostupné po prihlásení. Obsahuje identifikačné údaje a zoznam ID kurzov na vyplňovanie. Používa sa v navigačnom komponente, kde sa počíta počet nevyplnených kurzov.

/surveys/surveyId Endpoint vráti údaje o konkrétnej ankete podľa jej ID. Využíva sa pri overovaní platnosti URL, ktoré zadal užívateľ.

/surveys/surveyId/courses Podľa konkrétneho ID ankety, vráti údaje dostupných kurzov.

/surveys/surveyId/pubSettings Každá anketa má svoje vlastné možnosti zverejňovania údajov. Tento endpoint vráti kompletný zoznam týchto možností.

/surveys/surveyId/questions Endpoint vráti všetky otázky prístupné danej ankete podľa ID.

/courses Endpoint vráti údaje všetkých predmetov dostupných užívateľovi k vyplneniu.

/courses/courseId Vrátí jednu konkrétnu inštanciu kurzu podľa jeho ID. Podobne ako pri anketách, užívateľ môže zadať neplatné ID kurzu a skúsiť prísť na neplatný anketový lístok.

/courses/courseId/teachers Endpoint vráti všetkých učiteľov kurzu.

/overviews Nový endpoint pre zobrazenie nástenky. Obsahuje všetky predmety v semestri z KOSu daného užívateľa.

4.5 Návrh novej nástenky

Nová nástenka by mala prinášať informácie o priebehu ankiet (napr. kedy končí možnosť hodnotenia kurzu) a navigovať užívateľa ku svojmu anketovému lístku. Na výpočet časového úseku medzi 2 dátumami sa bude používať knižnica *vue-moment* [30]. Podľa Ing. Michala Valentu, Ph.D., je na backende pripravený SQL pohľad, ktorý je možný poslať do frontendu cez REST API.

Návrhom je, aby jednotlivé ankety boli v jednotlivých kartách. V každej karte je zoznam kurzov k daným anketám. Každý kurz môže nadobúdať 4 stavy:

1. Otvorený kurz. Koľko ostáva času na vyjadrenie sa ku kurzu. Pri tomto stave sa zobrazuje tlačidlo, ktoré vedie priamo k anketovým lístkom daného kurzu.
2. Vyplnený kurz. Obsahuje tiež časový údaj, kedy bol vyplnený.
3. Nedokončený kurz. Kurz, ktorý užívateľ ešte nedokončil a preto nie je prístupný k vyjadreniu.
4. Ostatné (menej dôležité kurzy, ako napr. telesná výchova) nezberajú vyjadrenia od študentov.

4. NÁVRH DESIGNU A TECHNOLOGIE

FIT LETO 2021/2022		
BI-OSY	Zbýva 2 hodiny k vyjadrenie	Vyjadriť sa
BI-DBS	Hodnoceno pred 2 dňami	
BI-ZUM	Kurz nedokončen	
BI-TJV	Kurz je nehodnotefný	

Obr. 4.2: Návrh novej nástenky s časovými údajmi.

Implementácia a dokumentácia

V prvej podkapitole sa čitateľ zoznámí s vývojovým prostredím, v ktorom bola táto práca implementovaná.

Pretože produkčné REST API nie je prístupné, bude treba simulovať CRUD operácie na lokálnom stroji, a tiež pripraviť vhodné dáta, ktoré zodpovedajú daným endpointom.

Ďalšie podkapitoly sa budú venovať detailnej implementácii, riešeniu spomenutých problémov týkajúcich sa zberu spätnej väzby a funkčným požiadavkám. Postupy a riešenia vždy vychádzajú z návrhu a technológií v Kapitole 4.

Veľká časť sa venuje rozboru jednotlivých komponentov využitých v rámci nového vyplňovacieho modulu. Nechýba však ani analýza z hľadiska bezpečnosti, aby opäť nedošlo ku strate dát [8].

Posledná podkapitola sa bude venovať dokumentácii, ktorá bude slúžiť budúcim vývojárom pri rozširovaní a úprave nového modulu.

5.1 Vývojové prostredie

Pred implementáciou je dôležité si naštudovať možné vývojové prostredia pre daný framework a programovacie jazyky. Pretože celá práca je vyvíjaná na lokálnom stroji, je potrebné si nainštalovať Vue.js a jeho podporné frameworky. V dokumentácii je odporúčané používať *npm* [33]. Jedná sa o balíčkový systém na manažovanie javascriptových knižníc a frameworkov.

Na vývoj je možné používať všetky operačné systémy. Z dôvodu pohodlnosti a zvyklosti je použitý operačný systém Linux.

Všetok zdrojový kód je napísaný v textovom editore *Visual Studio Code* (*VSC*) s podporným pluginom *Vetur* [34] pre automatické formátovanie a podporu našepkávača pri implementácii. Tento kód je postupne verzovaný prostredníctvom školského *Gitlabu* (odkaz na [35]), kde majú prístup vývojári pracujúci na Ankete ČVUT.

5.2 Príprava dát na testovanie - lokálny server REST API

Po konzultácii s vývojovým tímom Anketa ČVUT sa rozhodlo, že nový modul má počítať s novým návrhom REST API. Nainštalovať a nakonfigurovať plnohodnotný backendový server je pomerne náročné na čas a tím Ankety ČVUT nemá kapacitu na programovanie nových endpointov. Do hry tak vstupuje lokálny server vytvorený pomocou *json-server*.

Testovacie dáta je možné získať cez vývojársku časť prehliadača v starom testovacom module. Tieto dáta je treba filtrovať a navrhnúť nové endpointy spomenuté v Kapitole 4.4.

Na spustenie *json-serveru* stačí iba súbor *db.json*, v ktorom sa definuje štruktúra endpointu vo formáte JSON. Jedna z užitočných vlastností tohto serveru je možnosť zadefinovať väzbu medzi rôznymi objektami.

Každý endpoint predstavuje jeden objekt v JSON formáte. Objekt v relačnej databáze má zvyčajne unikátny a jednoznačný identifikátor. Tento identifikátor je možné pripojiť k inému objektu pomocou syntaxe *[objekt]Id : [číslo ID]*, ktorá je demonštrovaná v kóde 5.1. Po dotaze na endpoint *surveys/1/courses*, server vráti pole dvoch objektov *courses*.

```
1  {
2    "surveys": [
3      {
4        "id": 1,
5        "name": "FIT-anketa"
6      }
7    ],
8    "courses": [
9      {
10       "surveyId": 1,
11       "name": "Predmet1"
12     },
13     {
14       "surveyId": 1,
15       "name": "Predmet2"
16     }
17   ]
18 }
```

Zdrojový kód 5.1: Príklad vzťahu medzi dvoma endpointami.

Server je možné spustiť cez príkaz: *json-server -watch db.json -port 3001*. Po spustení žije server na porte 3001 a je pripravený pre aplikáciu.

Pokiaľ má server zvládnuť zložitejšiu logiku (napr. iné HTTP hlavičky), je treba vytvoriť javascriptový súbor, ktorý obsahuje konfiguráciu tohto serveru. Príklad takejto konfigurácie je v kóde 5.2, kde sa pri dotazovaní na endpoint */surveys/test* vždy vráti stavový kód HTTP 500.

```

1  const jsonServer = require('json-server')
2  const server = jsonServer.create()
3  const router = jsonServer.router('db.json')
4  const middlewares = jsonServer.defaults()
5
6  server.use(middlewares)
7
8  var whitelist = [
9    'http://localhost',
10   'http://localhost:3000',
11  ];
12
13  server.get('/surveys/test', (req, res) => {
14    res.status(500).jsonp({
15      msg: 'testing'
16    })
17  })
18  server.use(router)

```

Zdrojový kód 5.2: Konfigurácia falošného serveru cez súbor *server.js*.

5.3 Smerovanie

V každej webovej aplikácii je potrebné riešiť smerovanie, teda nastaviť prístupné URL, cez ktoré sa dostáva užívateľ k danému obsahu. Vo frontendovom svete existuje objekt volajúci tzv. *router*. Tento objekt zaisťuje konfiguráciu a zodpovedá za to, ktorý komponent sa zobrazuje na akom URL.

Problém s pôvodným modulom je automatické smerovanie pri zmene jazyka a tak isto nie je možné prísť do anketového lístku cez URL. Tieto problémy sa vyriešili použitím NuxtJS.

Pretože v projekte sa používa metaframework NuxtJS, bude použitá jedna z jeho výborných vlastností, a to *dynamické smerovanie*. Pri budovaní projektu si NuxtJS prejde celú štruktúru priečinku *pages*. Načíta všetky súbory a vytvorí vlastnú konfiguráciu routru. Jedinou prácou vývojára ostáva správne zostrojiť štruktúru priečinku *pages*.

Po dohode s vývojovým tímom sa nebudú implementovať nové URL formáty pre aplikáciu, pretože vyučujúci už linky používajú, napr. vo svojich prezentáciách.

Formáty URL jednotlivých ankiet:

Predmetová anketa: *surveys/surveyId/surveySemesterId/course/courseId* s parametrami:

surveyId: Jednoznačný identifikátor ankety.

surveySemesterId: Jednoznačný identifikátor semestru.

courseId: Jednoznačný identifikátor kurzu.

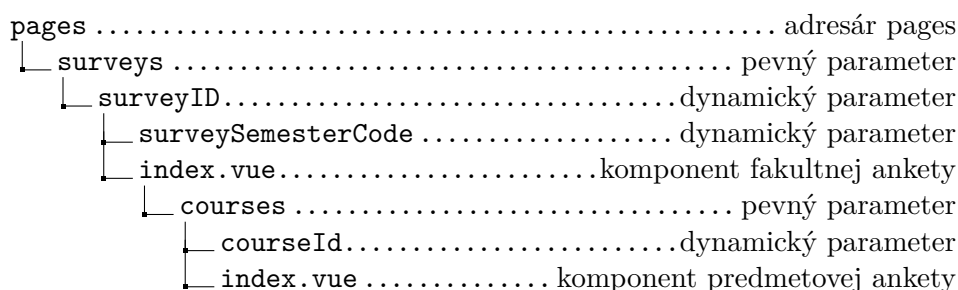
Fakultná anketa: *surveys/surveyId/surveySemesterId* s parametrami:

surveyId: Jednoznačný identifikátor ankety.

surveySemesterId: Jednoznačný identifikátor semestru.

Obsah vyššie uvedených parametrov sa vykresľuje pomocou *dynamických stránok*. Dynamická stránka sa vytvorí pomocou pridania *podčiarkovníku* pred mená súborov zakončených s *.vue* alebo priečinku. NuxtJS si v priebehu budovania aplikácie poznamená tieto súbory a priečinky. Zapisuje si ich potom do routru ako parameter.

Na obrázku 5.1 je štruktúra priečinka *pages*. V komponente *index.vue* sa dajú získať parametre *surveyId* a *surveySemesterCode* pomocou premennej *this.\$route.params.surveyCode*.



Obr. 5.1: Štruktúra priečinka *pages* na generovanie dynamického smerovania.

5.4 Lokalizácia textu a prepínanie jazyka

V dnešnej dobe by každá moderná webová aplikácia mala podporovať minimálne 2 jazyky. Webová aplikácia Anketa ČVUT tiež nie je výnimkou - na univerzite sú anglicky hovoriaci študenti.

Pôvodná aplikácia je kompletne lokalizovaná a podporuje zmenu jazyka cez navigačnú lištu. Pri zmene jazyka presmeruje aplikácia užívateľa na úvodnú stránku bez ohľadu na to, kde sa práve v aplikácii nachádza. Zmena jazyka sa neprejaví v URL odkaze anketového lístku, viď Kapitola 2.

Ako bolo uvedené v návrhu riešenia, bude sa používať framework *i18n-vue* na lokalizáciu textu v aplikácii. Framework má derivát pre rôzne frameworky a knižnice (napr. *i18next* pre React). Pretože metaframework NuxtJS podporuje *i18n-vue*, je možné v konfiguračnom súbore pre NuxtJS nastaviť globálnu konfiguráciu frameworku.

V zdrojovom kóde 5.3 je ukážka nastavenia *i18n-vue* v novom vyplňovacom module. Všetky texty a preklady sú uložené v súbore *locale_EN* a *locale_CS* podľa vybraného jazyka. Pri vstupe do aplikácie je zvolený ako predvolený český jazyk a do URL sa pridá prefix „cs“. Je možné tiež zmeniť jazyk aplikácie cez navigačnú lištu.

```

1 export default {
2   // Modules (https://go.nuxtjs.dev/config-modules)
3   modules: [
4     'nuxt-i18n',
5   ],
6   i18n: {
7     strategy: 'prefix',
8     lazy: true,
9     langDir: './locales/',
10    locales: [
11      {
12        code: 'en',
13        name: 'English',
14        file: 'locale_EN.js'
15      },
16      {
17        code: 'cs',
18        name: 'Cesky',
19        file: 'locale_CZ.js'
20      }
21    ],
22    defaultLocale: 'cs',
23    vueI18n: {
24      fallbackLocale: 'cs',
25    }
26  }
27 }

```

Zdrojový kód 5.3: Konfigurácia i18n-vue

Vo frameworku i18n-vue existuje počas behu aplikácie objekt, ktorý sa postará o preklad textu. Tento objekt je globálny, to znamená, že je prístupný v rámci celého projektu. Podľa dokumentácie i18n-vue je možné lokalizovať text pomocou „\$t()“, meniť jazyk celej aplikácie pomocou metódy „switchLocalePath()“ a smerovacie linky pomocou „localePath()“.

```

1 <b-dropdown-item
2   v-for="locale in availableLocales"
3   :key="locale.code"
4   :to="switchLocalePath(locale.code)"
5 >

```

Zdrojový kód 5.4: Prepínanie jazyka v navigačnej lište.

```

1 <b-dropdown-item :to="localePath('login')">
2   {{ $t("log_in") }}
3 </b-dropdown-item>

```

Zdrojový kód 5.5: Lokalizácia textu a lokalizácia smerovacích odkazov v navigačnej lište.

5.5 Rozbor komponentov vyplňovacieho modulu

Táto kapitola sa zaoberá detailnou implementáciou vyplňovacieho modulu, ktorý je podľa návrhu z Kapitoly 4.3 jedným veľkým komponentom. Čitateľ v tejto kapitole zistí, aká bola myšlienka za návrhom a implementáciou jednotlivých komponentov.

Vychádza sa z návrhu jednotlivých komponentov z Kapitoly 4.3. V ďalšej časti budú uvedené informácie o komponente *CourseRating*.

5.5.1 Analýza dát z backendu

Pred samotnou implementáciou treba analyzovať dáta poskytované endpointami. Každá anketa obsahuje otázky. Tieto otázky sa môžu líšiť podľa typu ankety, ale sú rovnaké pre všetky kurzy danej ankety. Pre spravovanie tohto typu dát sa používa endpoint: *surveys/surveyId/questions*.

```

1      "id": 2,
2      "surveyId": integer
3      "hasValueAnswer": boolean
4      "hasTextAnswer": boolean
5      "valueQuestionCz": string
6      "valueQuestionEn": string
7      "textContextCz": string
8      "textExplNormalCz": string
9      "textExplNormalEn": string
10     "textExplBadCz": string
11     "textExplBadEn": string
12     "textQuestionNormalCz": string
13     "textQuestionNormalEn": string
14     "textQuestionBadCz": string
15     "textQuestionBadEn": string
16     "placeholderCz": string
17     "placeholderEn": string
18     "priority": int
19     "textContextEn": string
20     "mandatory": boolean
21     "primary": boolean
22     "teacherQuestion": boolean

```

Zdrojový kód 5.6: Štruktúra objektu „question“ v tvare JSON z backendu.

V zdrojovom kóde 5.6 sú vidno zaujímavé atribúty, ktoré sú kľúčové pre analýzu:

1. Atribút „hasValueAnswer“ a „hasTextAnswer“ hovorí o možnosti hodnotenia otázok.
2. Textové vysvetlenie - názov v českom a anglickom jazyku daného objektu.
3. Atribút „teacherQuestion“, ktorý sa viaže k učiteľovi.

Z analýzy štruktúry objektu „question“, musí *CourseRating* počítať s nasledujúcimi bodmi:

1. Dokáže správne vkladať textové vysvetlenie a názov podľa jazyka aplikácie.
2. Pred odoslaním užívateľských vstupov musí správne validovať všetky povinnosti vyplnenia.
3. Pri predmetovej ankete sa pre pridanie učiteľa k hodnoteniu musia správne pridať oba typy hodnotenia.

Pretože komponent *CourseRating* sa používa v oboch anketách, budú sa musieť pred vytvorením načítať dáta z backendu o *učiteľoch*, *otázkach* a *možnostiach zverejňovania*. Lokálne stavy komponentu *CourseRating* majú nasledujúcu štruktúru, ktorá je znázornená v zdrojovom kóde 5.7.

```
1 data() {  
2   return {  
3     teachers: [],  
4     courseQuestions: [],  
5     pubSettingOptions: [],  
6     teacherQuestion: {},  
7   };  
8 }
```

Zdrojový kód 5.7: Zdrojový kód lokálneho stavu komponentu *CourseRating*.

5.5.2 Zaoberanie dát z backendu

Ďalší krok je získavanie dát z backendu, ktoré následne zaplnia lokálne stavy, aby mohol komponent správne zostrojiť otázky. Podľa *best practices* [36] sa načítavajú dáta z backendu pred vytvorením komponentu. Zdrojový kód 5.8 znázorňuje spôsob načítania dát v komponente *CourseRating*.

```
1 created() {  
2   this.fetchTeachers();  
3   this.fetchQuestions();  
4   this.fetchPubSettings();  
5 }
```

Zdrojový kód 5.8: Načítanie dát z backendu pred samotným vytvorením komponentu.

V zdrojovom kóde 5.9 je príklad spracovania dát - objekt typu *question*. Do každého objektu tohto typu sa pridáva premenná pre smajlíkové a textové hodnotenie. Otázka typu *teacher* je vždy najviac jedna a preto môže existovať v komponente sama. Podobným spôsobom sa načítavajú a spracovávajú *učitelia* a *možnosť zverejňovania*.

```

1 fetchQuestions() {
2   if (!this.surveyId) return; // Anketa neexistuje
3   this.$axios
4     .get(`/surveys/${this.surveyId}/questions`)
5     .then((response) => {
6       for (let i = 0; i < response.data.length; i++) {
7         if (response.data[i].teacherQuestion) {
8           const question = response.data[i];
9           this.teacherQuestion = question; // otázka typu teacher
10        } else {
11          var question = response.data[i];
12          // pridavanie nových premenných do objektu s
13            reaktivitou
14          let newItem = Object.assign({}, question, {
15            valueAnswer: undefined,
16            textAnswer: "",
17            pubSettingSelected: [],
18          });
19          this.courseQuestions.push(newItem);
20        }
21        this.courseQuestions.sort((a, b) => a.priority - b.priority
22          );
23      });
24    });

```

Zdrojový kód 5.9: Načítanie a spracovanie dát typu *question*.

5.5.3 Tvorba dielčích komponentov pre *CourseRating*

Komponent *CourseRating* má dáta a je pripravený na vykresľovanie štruktúry formulára. Počas návrhu z kapitoly 4.3 boli vytipované pomocné komponenty. Táto kapitola obsahuje rozbor týchto dielčích komponentov.

1. *CRSmileyRating* má za účel zobrazovať hodnotenie a odkazovať svoj výsledok (stav) ku *CourseRating*. Smajlíky sa zobrazujú v aplikácii ako obrázkov typu *SVG*, pretože formát *SVG* sa dá programovať ako *XML* súbor [37]. Pri zmene veľkosti (rozmeru obrázku) sa neznižuje kvalita. Pri implementácii zobrazenia smajlíkov sa uvažovalo o používaní BootstrapVue komponenty *icons*. Ich výhodou sú nastaviteľné veľkosti, farby a animácie. Po konzultácii sa nakoniec rozhodlo používať obrázky, pretože BootstrapVue nemá vhodné komponenty. Ako jediný vstupný parameter je hodnota, na základe ktorej sa zvýrazní daný smajlík. Výstupná hodnota je *ID* smajlíka od 1 do 5.
2. *CRNameWithHelper* je jediný komponent, ktorý nikam neodkazuje svoj stav. Tento komponent má v sebe len text a BootstrapVue [38] komponent *question-circle*, ktorý je zmenšený a posunutý. Aby sa pri nájdení myšou zobrazila nápoveda, používa Vue.js direktívu *v-b-tooltip*, ktorá

sa o to sama postará. Vstup komponentu je textový reťazec štítku a nápvoda pri nájdení myšou.

3. *CRPublicSetting* je komponent zobrazujúci možnosti zverejňovania podľa ankiet. Na vykresľovanie jednotlivých možností sa používa komponent *CRNameWithHelper*. Pri návrhu sa počítalo s tým, že v tomto komponente sa načítajú dáta z backendu cez endpoint `/surveys/surveyId/pub-Settings`.

Pri implementácii sa zistilo, že pri každej textovej otázke sa vykresľuje jeden *CRPublicSetting* a komponent by sa tak dotazoval do endpointu toľko krát, koľko je textových odpovedí.

Ako vhodné riešenie bol zvolený návrh, v ktorom sa ako vstup dostane pole možností z rodičovského komponentu *CourseRating*. Výstup je pole zvolených možností.

4. *CRTextRating* ako textový editor podľa požiadavku musí podporovať *markdown* prvky a dokázať zobrazovať priebežné vstupy užívateľa v *markdown* formáte. Pri implementácii tohto komponentu sa používa knižnica tretích strán *v-md-editor* [39], ktorý podporuje požiadavky uvedené vyššie. Zvažovala sa implementácia vlastného textového editoru, ale z časového dôvodu bol tento návrh zamietnutý. Z hľadiska bezpečnosti môže teoreticky vzniknúť *XSS* útok, toto je ale riziko každého textového editora na strane frontendu. Proti takým útokom musí byť ošetrovaný vstup na strane backendu. Výstup komponentu je reťazec.

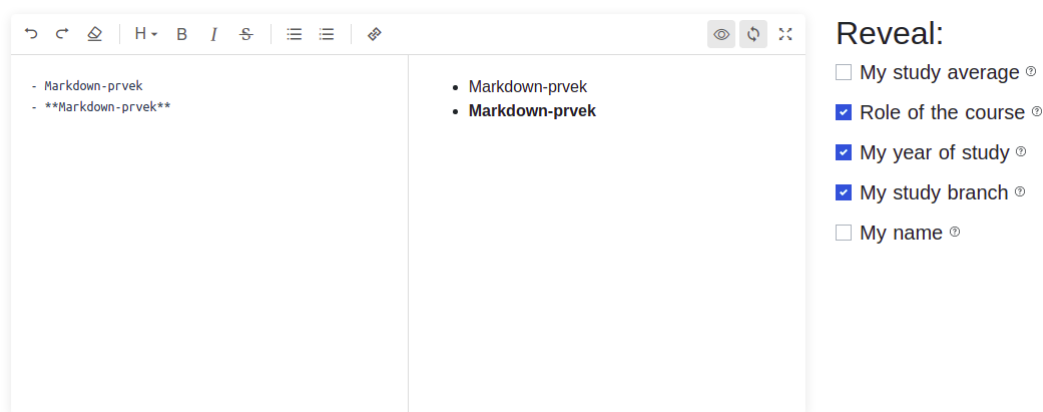
```
1 <p>
2   {{ title }}
3   <b-icon
4     icon="question-circle"
5     font-scale="0.5"
6     shift-v="8"
7     @mouseover="hover = true"
8     @mouseleave="hover = false"
9     v-b-tooltip="helpText"
10  >
11 </b-icon>
12 </p>
```

Zdrojový kód 5.10: Implementácia *CRNameWithHelper* s použitím *BootstrapVue*

5. IMPLEMENTÁCIA A DOKUMENTÁCIA

```
1 <v-md-editor
2   v-model="value"
3   height="400px"
4   @input="$emit('input', value)"
5   left-toolbar="undo redo clear | h bold italic strikethrough |
      ul ol | link"
6   right-toolbar="preview sync-scroll fullscreen"
7 ></v-md-editor>
```

Zdrojový kód 5.11: Použitie *v-md-editor* a odkazovanie stavu cez *event emitter*.

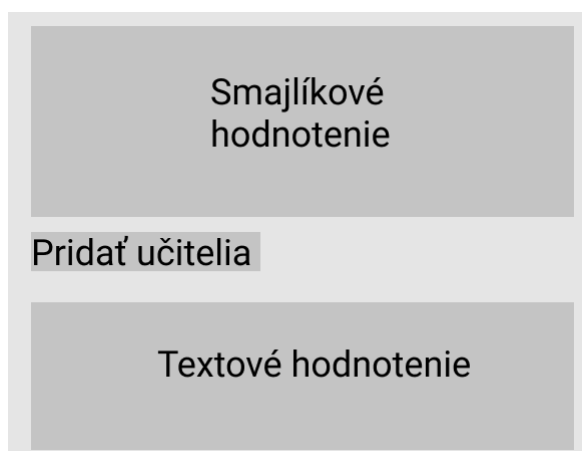


Obr. 5.2: Markdown editor a možnosť zverejňovať osobné údaje (na obrázku vpravo).

5.5.4 Štruktúra šablóny

V ďalšom kroku je treba správne zakomponovať vytvorené komponenty do jedného zmysluplného celku. Šablóna navrhnutého modulu je znázornená na obrázku 5.3. V prvej časti je vždy hodnotenie pomocou smajlíka, potom nasleduje textové hodnotenie. Pri pridaní nového učiteľa k hodnoteniu musí komponent *CourseRating* správne zareagovať podľa nastavenia otázok.

Na implementáciu šablóny komponentu *CourseRating* sa používa *Bootstrap Vue grid system* [40]. Grid system je spôsob, ako rozdeliť obsah do pravidelnej mriežky. V zdrojovom kóde 5.12 je použitie grid system na rozdelenie komponentu a na obrázku 5.4 je možné vidieť výsledok po rozdelení. Zároveň je vidieť spôsob odkazovania výstupných stavov komponentu cez *v-model* zo zdrojového kódu 5.12 pri komponente *CRSmileyRating*.

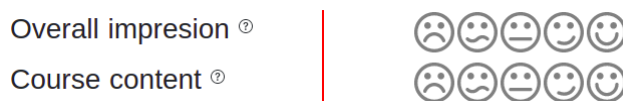


Obr. 5.3: Štruktúra šablón nového modulu.

```

1 <b-container fluid flush="true">
2   <b-row>
3     <b-col> // stlpec 1
4       <NameWithHelper
5         :title="getText(question, 'valueQuestion')"
6         :helpText="getText(question, 'textContext')"
7         class="h3 mb-2"
8       />
9     </b-col>
10    <b-col> // stlpec 2
11      <SmileyRating v-model="question.valueAnswer" />
12    </b-col>
13  </b-row>
14 </b-container>

```

Zdrojový kód 5.12: Použitie *grid system* na zarovnávanie komponentov.Obr. 5.4: Výsledok po rozdelení zo zdrojového kódu 5.12 (použitie *grid system*).

5.5.5 Validácia vstupu

Vstupné dáta treba pred odoslaním validovať podľa definovaných povinností. Pokiaľ komponent zistí pri validácii nezgodu, musí dokázať zobrazit chybovú hlášku a správne odkázať na otázku, ktorú treba hodnotiť. Pre zobrazenie chybových hlášok sa používa modul *vue-toasted* [41]. Pri validácii sa prejde cyklom všetkých otázok, kontroluje sa ich povinnosť a vstupy od užívateľov.

5. IMPLEMENTÁCIA A DOKUMENTÁCIA

V zdrojovom kóde 5.13 je uvedená časť validácií otázok. Výstupná hodnota funkcie vráti „false“, pokiaľ nie je splnená validačná podmienka.

```
1 checkMandatory() {
2   for (let i = 0; i < this.courseQuestions.length; i++) {
3     let question = this.courseQuestions[i];
4     if (question.mandatory) {
5       // Pokud otazka je povinna
6       if (question.hasValueAnswer && question.valueAnswer
7         == undefined) {
8         this.$toast.error(
9           this.$t('smiley_face_question_not_evaluated', {
10            question: this.getText(question, '
11              valueQuestion')})
12         );
13         return false;
14       }
15     }
16   }
17   return true;
18 }
```

Zdrojový kód 5.13: Validácia vyplnenosti otázok.

A red rectangular toast message box with white text. The text reads: "Otázka "Celkový dojem" u smajlíkové hodnotenie není vyhodnocená. CANCEL".

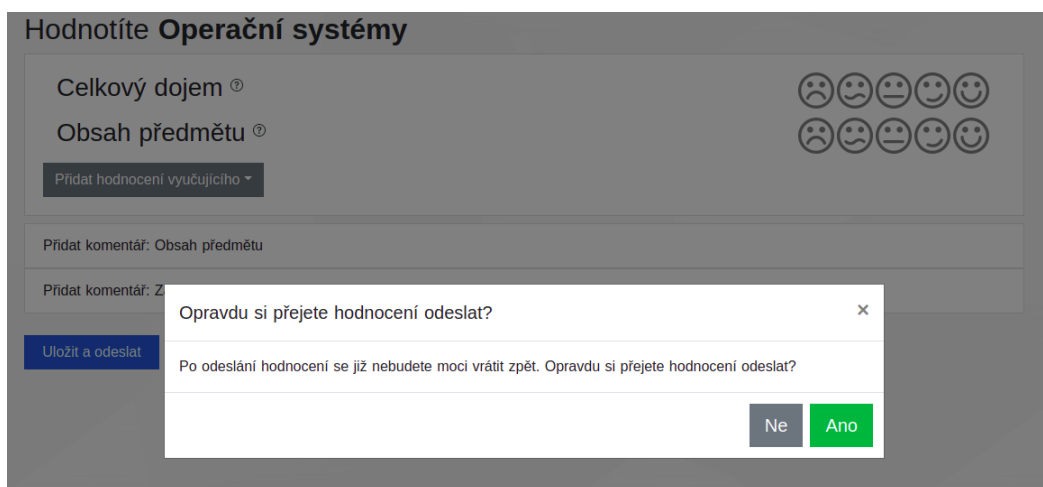
Obr. 5.5: Chybová hláška pri nevyplnenom smajlíkovom hodnotení.

5.5.6 Odoslanie užívateľských vstupov do backendu

Pokiaľ je užívateľ spokojný so svojimi hodnoteniami, môže poslať dáta do backendu cez tlačidlo *Uložiť a odeslať*. Po kliknutí sa zobrazuje modálne okno pre potvrdenie, že užívateľ naozaj chce poslať svoje hodnotenie, viď 5.6. Pri výbere *NE* sa okno zavrie a nič sa neudeje. Naopak pri výbere *ANO* sa vykonáva validácia vstupných dát popísaná v predošlej podkapitole.

Po úspešnej validácii je možné poslať vstupné dáta od užívateľa do backendu prostredníctvom REST API. Pred posielaním treba pripraviť dátový balíček. V zdrojovom kóde 5.14 je ukážka procesu vytvorenia balíčka.

5.5. Rozbor komponentov vyplňovacieho modulu



Obr. 5.6: Modálne okno na upozornenie užívateľa pri odoslaní.

```
1  createDataToPost() {
2    let data = []; // array of objects send to API
3
4    // push non-teacher evaluation
5    for (let i = 0; i < this.courseQuestions.length; i++) {
6      let question = this.courseQuestions[i];
7      let evaluation = Object.assign(
8        {},
9        {},
10       {
11         questionId: question.id,
12         valueAnswer: question.valueAnswer,
13         textAnswer: question.textAnswer,
14         pubSettingSelected: question.pubSettingSelected,
15       }
16     );
17     data.push(evaluation);
18   }
19   return data;
20 }
```

Zdrojový kód 5.14: Proces vytvorenia balíčka dát pre poslanie do backendu.

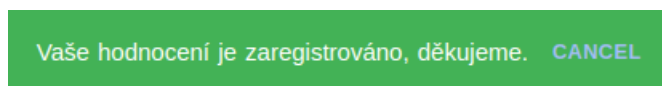
Proces posielania dát je zaistený POST metódou do backendu. Pokiaľ backend úspešne prijme dáta, vráti užívateľovi úspešnú hlášku a presmeruje ho na úvodnú stránku. V opačnom prípade sa zobrazí chybová hláška s príčinou, ktorá je odoslaná ako odpoveď z backendu alebo priamo z aplikácie.

```

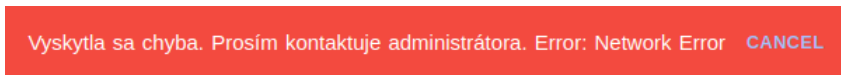
1 async sendData() {
2     if (this.checkMandatory() === false) return;
3     let data = this.createDataToPost();
4     await this.$axios
5         .post("/evaluations", data)
6         .then((response) => {
7             this.$toast.success(this.$t('evaluation_success')
8                 , {
9                     duration: 5000
10                });
11            this.$router.push('/')
12        })
13        .catch((error) => {
14            this.$toast.error(this.$t('send_data_error'), {
15                error: error });
16        });
17 }

```

Zdrojový kód 5.15: Logika pri posielaní dát do backendu.



Obr. 5.7: Hláška pri úspešnom odoslaní dát.



Obr. 5.8: Chybová hláška pri neúspešnom pokuse o odoslanie dát a zobrazenie príčiny chyby.

5.6 Nástenka

Pred implementáciou je treba navrhnuť dáta, ktoré by mohol vrátiť endpoint */overview*. Na rozdiel od ostatných endpointov, ktorých dáta boli navrhnuté podľa existujúcich v pôvodnej aplikácii na produkcii, dáta pre endpoint */overview* sú testovacie a nezaručujú funkčnosť na produkcii. Pri návrhu sa uvažovalo nad štruktúrou podobnou ako u endpointu */surveys*. Jediné rozdiely sú stavy a dátumy posledných zmien pri jednotlivých kurzoch alebo anketách.

Z návrhu Kapitoly 4.5 sa bude každá anketa zobrazovať vo svojej karte. V karte bude zoznam kurzov podľa ankiet. Pokiaľ je to fakultná anketa, nebude obsahovať kurzy, ale len ročník.

Možné stavy pre kurzy a fakultné ankety:

Evaluated: Anketový lístok daného kurzu je vyplnený. Pri názve kurzu sa zobrazuje žltý štítok s časovými údajmi, kedy bol vyplnený.

Unevaluated: Anketový lístok daného kurzu je dostupný k vyplneniu. Zobrazuje sa modrý štítok s časovými údajmi, do kedy je možné vstúpiť do kurzu a ohodnotiť ho. Navyše sa zobrazuje ešte tlačítko, ktoré navedie užívateľa priamo k danému anketovému lístku.

Uncompleted: Užívateľ ešte nedokončil kurz (nedostal známku).

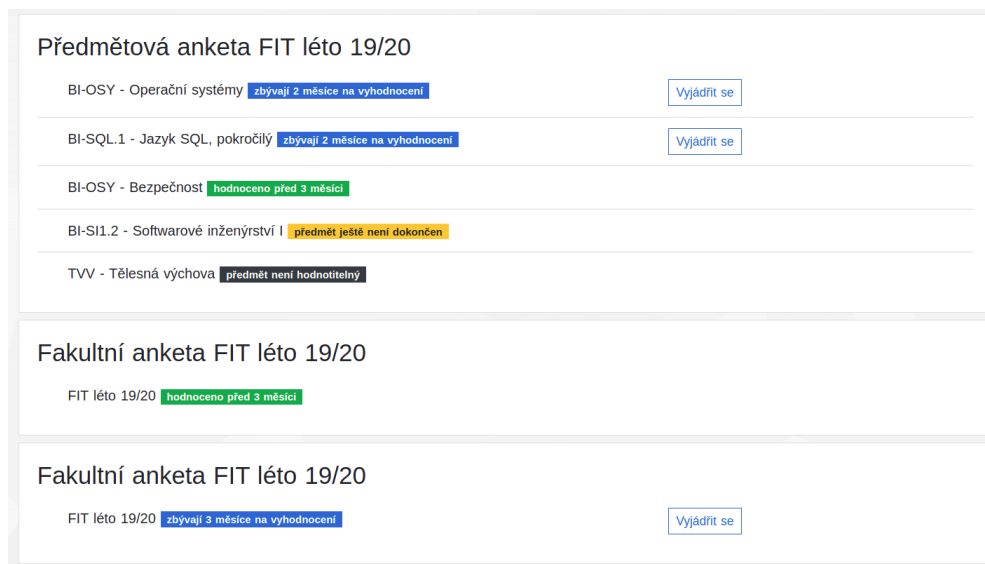
Unevaluable: Kurz sa nehodnotí v ankete.

Na výpočet časového úseku medzi 2 dátumami sa používa knižnica *vue-moment*. Vue-moment ponúka len integrácie medzi knižnicou *moment* [42] a Vue.js. Táto knižnica má širokú ponuku konfigurácie, aby sa prispôbil formát podľa jazyka, času, atď. Pre účel aplikácie stačí používať metódu „*fromNow()*“, ktorá vypočíta rozdiel medzi dvoma dátumami a je znázornená v zdrojovom kóde 5.16. Pomocou bodkovej notácie je možné zreťaziť ďalšie funkcie na získavanie vhodného formátu.

```
1 $moment(survey.date).locale($i18n.locale).fromNow(true)
```

Zdrojový kód 5.16: Príklad použitia *vue-moment* na výpočet rozdielu 2 časov.

Po konzultácii s vývojovým tímom Anketa ČVUT sa rozhodlo, že sa „Nástenka“ premenuje na „Moje lístky“. Nástenka totiž zobrazuje všetky kurzy, prípadne aj prístup do anketových lístkov užívateľa.



Obr. 5.9: Nová nástenka (Moje lístky).

5.7 Navigačná lišta

Z požiadavkov a z návrhu Kapitoly 4.1.3 sa zjednocujú dve navigačné lišty do jednej. Implementácia používa BootstrapVue komponent *Navbar* [44]. Podobne ako ostatné komponenty od BootstrapVue, *Navbar* dokáže prispôbiť rôzne nastavenia podľa potrieb vývojára. Do komponentu treba pridať smerovacie odkazy a lokalizáciu týchto odkazov.

Nová zmena oproti pôvodnému modulu je, že na prepínanie jazykov sa používajú vlajky. Implementácia je znázornená v zdrojovom kóde 5.17. Na zobrazenie vlajky sa používa knižnica tretích strán *country-flag* [45].

```

1 <b-nav-item-dropdown right>
2   <template #button-content>
3     <country-flag
4       :country="$i18n.locale == 'cs' ? 'cz' : 'gb'"
5       size="normal"
6     />
7   </template>
8   <b-dropdown-item
9     v-for="locale in availableLocales"
10    :key="locale.code"
11    :to="switchLocalePath(locale.code)"
12    style=""
13  ><country-flag
14    :country="locale.code == 'cs' ? 'cz' : 'gb'"
15    size="normal"
16  />
17    {{ locale.name }}
18  </b-dropdown-item>
19 </b-nav-item-dropdown>

```

Zdrojový kód 5.17: Implementácia prepínania jazyka s použitím vlajky.



Obr. 5.10: Navigačná lišta na desktopovom zariadení.



Obr. 5.11: Navigačná lišta na mobilnom zariadení.

5.8 Autentizácia a autorizácia

Pre správny beh modulu je treba implementovať jednoduché prihlasovanie (autentizácia) a zakázať prístup (autorizácia) nepovoleným osobám do určitej časti. Táto implementácia a konfigurácia je základ, ktorý nahradí pôvodnú implementáciu pri nasadení na produkciu.

5.8.1 Autentizácia

Pre kompletnú podporu prihlasovania a prístupy na stránkach sa používa framework *nuxt |auth* [43]. Po inštalácii frameworku sa nastaví v konfiguračnom súbore *nuxt.config.js* spôsob prihlasovania, autentizačný token a endpointy, na ktorých sa framework dotazuje pre autentizačné dáta. Tieto dáta sú pripravené v konfiguračnom súbore pre *json-server*.

```
1 // strategies for nuxtjs auth
2 auth: {
3   strategies: {
4     local: {
5       token: {
6         property: 'token',
7         required: true,
8         type: 'Bearer'
9       },
10      user: {
11        property: false,
12        autoFetch: true
13      },
14      endpoints: {
15        login: { url: '/login/external', method: 'post' },
16        logout: { url: '/logout', method: 'post' },
17        user: { url: '/user/me', method: 'get' }
18      }
19    }
20  },
21  redirect: false
22 }
```

Zdrojový kód 5.18: Konfigurácia autentizácie pre *nuxt |auth*.

5.8.2 Autorizácia

Pre autorizáciu sa používa *middleware*, ktorý je kus logiky, podľa ktorej sa kontrolujú prístupy pred načítaním stránky. Je možné kontrolovať role, užívateľské meno, platnosť tokenu, atď. Na zdrojovom kóde je príklad obmedzenia vstupu na stránku. Pokiaľ užívateľ nesplní autorizáciu, aplikácia presmeruje užívateľa k prihlasovaniu.

5. IMPLEMENTÁCIA A DOKUMENTÁCIA

```
1 export default function ({ store, redirect }) {
2   // If the user is not authenticated
3   if (!store.$auth.$state.loggedIn) {
4     return redirect('/login')
5   }
6 }
```

Zdrojový kód 5.19: Príklad logiky *middlewareu* použitá pri zobrazovaní anketového lístka.

5.9 Dokumentácia

Jeden z dôvodov vzniku tejto práce bol zle zdokumentovaný kód, ktorý zabraňuje tímu Anketa ČVUT k ďalšiemu vývoju. Tak ako pre každý veľký systém, je dokumentácia nového modulu veľmi dôležitá. Pomáha novému vývojárovi rýchlo sa zorientovať pri vstupe do zdrojového kódu aplikácie.

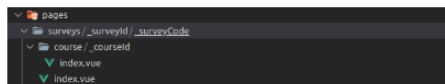
Spôsob dokumentácie bol konzultovaný s vývojovým tímom Anketa ČVUT a s vedúcim projektu Ing. Michalom Valentom, Ph.D. Uvažovalo sa nad dvoma spôsobmi dokumentácie:

1. Priamo v kóde a pomocou frameworkov tretích strán sa vygeneruje dokument.
2. Popis konfigurácie a štruktúru projektu, komponentu do Anketa ČVUT wiki (http://anketa-vyvoj.cvut.cz/wiki/doc-filling_modules).

Po zvážení výhod a nevýhod medzi dvoma spôsobmi sa rozhodlo napísať dokumentáciu na wiki. Dôvodom je ten, že po prečítaní získava nový vývojár prehľad o komponentoch a kde sa používajú. Keďže stránka wiki je len čistý textový editor, dáva voľnosť popisu a pridávanie obrázkov pre lepšie vysvetlenie.

Navigácie je vyriešená pomocou Nuxt frameworku, kde sa definuje štruktúra priečinku pages. Aby fungovalo ten link, musí byť definovaný jeden súbor "index.vue". Ten `_slug` priečinkov slúži ako parameter pri zadávaní `URL`.

Štruktúra je viditeľná na obrázku.



V každom súbore index.vue sa používa komponenta CourseRating s rôznymi props. Jediný rozdiel v tomto čase je, že pre fakultnú anketu je courseId props prázdny.

```
<CourseRating :courseId="null" :surveyId="this.surveyId" />
```

Obr. 5.12: Popis navigácie v Anketa ČVUT wiki.

Testovanie

Táto kapitola sa zaoberá testovaním nového vyplňovacieho modulu. Celá práca je nasadená na testovací server Ankety ČVUT. Testerí sú členovia tímu Anketa ČVUT, ktorí majú prístup do testovacieho portálu. Užívateľské testy sú navrhnuté tak, aby pokrývali funkčné požiadavky z Kapitoly 2. Po testovaní bude zozbieraná spätná väzba, ktorá by mohla byť v budúcnosti spracovaná a implementovaná do modulu.

6.1 Užívateľské testovanie

Každý scenár obsahuje:

1. Účel testu
2. Popis scenára
3. Výsledky
4. Funkčné požiadavky, ktoré sú testom pokryté

6.1.1 Scenár 1

Účel testu	Prihlásenie do aplikácie a odoslanie hodnotenia do systému.
Popis scenára	<ol style="list-style-type: none"> 1. Vstúpte do ankety cez <i>http://anketa-vyvoj.cvut.cz/frontend2/cs</i>. 2. Prihláste sa do aplikácie. 3. Po prihlásení vyberte anketový lístok predmetu, ktorý chcete hodnotiť. 4. Ohodnoťte všetko, okrem smajlíkového hodnotenia „Celkový dojem“. 5. Pri textovom hodnotení môžete zvoliť nezverejňovať vaše osobné údaje (napr. z dôvodu kritického hodnotenia vyučujúceho). 6. Zvoľte „Uložiť a odeslať“.
Výsledky	<ol style="list-style-type: none"> 1. Všetci testeri sa dostali k vybranému anketovému lístku. 2. Testerí upozornili na to, že nešlo odoslať hodnotenie, pretože sa zobrazila chybová hláška <i>Otázka "Celkový dojem" smajlíkové hodnotenie není vyhodnocená..</i> Po použití smajlíkového hodnotenia sa hodnotenie úspešne zaevidovalo do systému. Týmto sa otestovala validácia povinnosti. 3. Kontrolou db.json sa zistilo, že prišli všetky odoslané hodnotenia. U niektorých odpovedí je možnosť zverejnenia osobných údajov prázdna (to znamená, že užívateľ si nepraje svoje osobné údaje zverejniť). <p>Testovací scenár bol otestovaný a dopadol podľa očakávania.</p>
Funkčné požiadavky	FP-1, FP-5, FP-7

6.1.2 Scenár 2

Účel testu	Prepínanie jazyka počas používania aplikácie
Popis scenára	<ol style="list-style-type: none"> 1. Vstúpte do ankety cez <i>http://anketa-vyvoj.cvut.cz/frontend2/cs</i>. 2. Zmeňte jazyk aplikácie podľa vašej preferencie. 3. Skúste sa dostať k anketovému lístku predmetu. 4. Zmeňte jazyk na iný, než momentálne máte v aplikácii editáciou URL. 5. Ohodnotte predmet a pošlite svoje hodnotenie do systému.
Výsledky	<ol style="list-style-type: none"> 1. Všetkým testerom sa podarilo zmeniť jazyk podľa potreby. Niekomu trvalo dlhšiu dobu kým zistil, čo bolo treba zadať do URL. 2. Všetky texty a hlášky boli lokalizované do českého a anglického jazyka. 3. Aplikácia pri zmene jazyka nepresmeruje užívateľa na úvodnú stránku. 4. U jedného testera sa zobrazili hlášky zároveň v češtine a v angličtine. Dôvodom bol vypnutý <i>json-server</i>, druhá časť hlášky je chyba od frameworku v anglickom jazyku. 5. Pri výbere smutného smajlíka sa nezmenila nápoveda textových hodnotení. <p>Testovací scenár bol otestovaný a zistené chyby budú opravené.</p>
Funkčné požiadavky	FP-1, FP-3, FP-5, FP-4, FP-6, FP-7

6.1.3 Scenár 3

Účel testu	Prístup k anketovému lístku cez URL
Popis scenára	<ol style="list-style-type: none"> 1. Zavrite súčasný prehliadač alebo otvorte anonymné okno. 2. Vstúpte do ankety cez <i>http://anketa-vyvoj.cvut.cz/frontend2/en/surveys/1/B192/course/1121406</i> 3. Zmeňte jazyk podľa potreby. 4. Ohodnoťte predmet a pošlite svoje hodnotenie do systému.
Výsledky	<ol style="list-style-type: none"> 1. Pri kliknutí na URL aplikácia správne presmeruje užívateľa k prihláseniu. 2. Všetci testerí sa dostali k anketovému lístku predmetu <i>Operační systémy</i> v anglickom jazyku. 3. Po zmene jazyka nie je užívateľ presmerovaný na úvodnú stránku. <p>Testovací scenár bol otestovaný a dopadol podľa očakávania.</p>
Funkčné požiadavky	FP-2, FP-4, FP-5, FP-7

6.1.4 Zhrnutie

Užívateľské testy dopadli dobre, s drobnými chybami. Tieto chyby budú opravené pred nasadením do produkcie.

Po testovaní uviedli testerí svoj názor a návrhy na funkcionality, ktoré by mal nový modul zvládať:

1. Pri odoslaní navrhujú najprv validovať vstupy, až potom zobrazí modálne okno.
2. Po odoslaní anketového lístku by aplikácia mala užívateľa presmerovať k „Mojim lístkom“, namiesto na úvodnú stránku.
3. Pri textovom hodnotení by mohla byť zobrazená „šípka“, znázorňujúca možnosť rozbalenia po kliknutí.
4. Prostredie príde testerom veľmi intuitívne.

Tím Anketa ČVUT očekává implementáciu týchto návrhov po nasadení súčasného modulu na produkciu.

Záver

Cielom tejto práce bolo navrhnuť, implementovať, otestovať a zdokumentovať redesign vyplňovacieho modulu. Každá časť práce bola popísaná a detailne vysvetlená čitateľom.

Autor otestoval a zanalyzoval pôvodný modul z hľadiska funkčnosti a údržby zdrojového kódu. Kvôli nesprávnej implementácii a použitiu knižnice React, má pôvodný modul veľa nedostatkov. Tak isto udržateľnosť kódu bola problematická. Chýba štruktúra používania jednotlivých komponentov, kvôli čomu má tím Anketa ČVUT veľké problémy pri vývoji. Nesprávny návrh štruktúry REST API môže potenciálne zťažovať aplikáciu pri získavaní dát z backendu.

V spolupráci s vývojovým tímom Anketa ČVUT a s vedúcim práce Ing. Michalom Valentom, Ph.D. autor vyslovil požiadavky na nový modul, ktoré boli následne navrhnuté a implementované. Pri návrhu riešenia jednotlivých problémov boli použité moderné frameworky pre webové aplikácie. Nechýba však detailný popis ich spôsobu fungovania a dôvod pre ich voľbu. Prebehlo sa kompletný redesign navigačnej lišty, anketového lístka a štruktúry REST API.

Keďže autor nemal žiadne poznatky o frontendovom svete, musel si našťudovať nové princípy a osvedčené postupy pri návrhu a implementácii. Jeden z nefunkčných požiadavkov bolo použiť pri implementácii technológiu Vue.js. Pre autora to znamenalo našťudovať a pochopiť nové technológie Vue.js a jazyk JavaScript, s ktorým autor pred tým nikdy neprišiel do styku.

Následne bola prevedená implementácia jednotlivých komponentov podľa návrhu. Autor popísal svoj myšlienkový pochod a spôsob implementácie. Komponenty sú rozdelené do dielčích komponentov, aby sa kód neduplikoval a dal sa ľahko ladiť. Celý nový modul bol kompletne zdokumentovaný na Anketa ČVUT wiki stránke, ktorá slúži ako základ pre nových vývojárov pri vyvíjaní nových funkcionalít do tohto modulu. Výsledná aplikácia bola nasadená a otestovaná užívateľskými testami. Testovanie dopadlo veľmi dobre. Pri testovaní sa našli drobné chyby v module, ktoré budú opravené a znovu otestované.

ZÁVER

Jednotlivé ciele práce boli splnené podľa očakávania vedúceho a vývojového tímu Anketa ČVUT. Po tejto práci bude autor spolupracovať s tímom Anketa ČVUT ohľadom nasadenia nového vyplňovacieho modulu na produkciu. Počas implementácie našiel autor ďalšie možnosti zlepšenie, napríklad automatické testy na detekovanie potenciálnych chýb, ktoré nedokážu pokryť užívateľské testy. Ďalším návrhom je zvýrazniť otázky, ktoré sú povinné a treba na ne odpovedať počas hodnotenia.

Literatúra

- [1] ŠTECHA, Vojtěch. *Anketa ČVUT - verze 3.0 - vyplňování anketních lístků*. Praha, 2019. Magisterská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, Katedra softwarového inženýrství.
- [2] JUN, Jakub. *Anketa ČVUT verze 3 - modul pro správu anket - uživatelské rozhraní*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020
- [3] Facebook Inc. *React* [online]. [cit. 2021-04-23]. Dostupné z: <https://reactjs.org/>
- [4] Mozilla and individual contributors. *Document Object Model* [online]. 2021 [cit. 2021-04-23]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction
- [5] Otto, M. *Bootstrap* [online]. 2019 [cit. 2021-04-23]. Dostupné z: <https://getbootstrap.com>
- [6] KNAP, David. *Návrh nového řešení aplikace Anketa ČVUT*. Praha, 2018. *Magisterská práce*. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra počítačů.
- [7] Positive Technologies. *What is a cross-site scripting (XSS) attack?* [online]. July 22, 2020 [cit. 2021-04-23]. Dostupné z: <https://www.ptsecurity.com/ww-en/analytics/knowledge-base/what-is-a-cross-site-scripting-xss-attack/>
- [8] VALENTA, Michal. *Částečná ztráta dat - vysvětlení* [online]. Praha: ČVUT, 2019 [cit. 2021-04-29]. Dostupné z: <https://anketa.is.cvut.cz/html/anketa/data-loss>
- [9] Mozilla and individual contributors. *Uniform Resource Locators* [online]. 2021 [cit. 2021-4-23]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_URL

- [10] *Front-End Error Handling* [online]. Amsterdam / Berlín: 5apps, 2014 [cit. 2021-04-30]. Dostupné z: <https://www.staticapps.org/articles/front-end-error-handling>.
- [11] GAJDOS, Andrej. *Single-Page Application vs Multiple-Page Application: Which One To Choose For Your Project* [online]. November 22, 2020 [cit. 2021-05-03]. Dostupné z: <https://andrejgajdos.com/single-page-application-vs-multiple-page-application/>
- [12] Mozilla and individual contributors. *Git repository* [online]. [cit. 2021-05-03]. Dostupné z: <https://git-scm.com/about>
- [13] VITALY, Friedman. *Responsive Web Design - What It Is And How To Use I* [online]. 2011 [cit. 2021-05-06]. Dostupné z: <https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/>
- [14] Shilpi. *Secure by Design: A Web Development Essential* [online]. September 3 2019 [cit. 2021-05-12]. Dostupné z: [Phttps://opensenselabs.com/blog/articles/secure-design](https://opensenselabs.com/blog/articles/secure-design)
- [15] LIN, Stephanie. *The Rules for Modern Navigation* [online]. January 31st, 2017 [cit. 2021-05-12]. Dostupné z: <https://www.uxbooth.com/articles/the-rules-for-modern-navigation/>
- [16] WOZNIEWICZ, Brandon. *The Difference Between a Framework and a Library* [online]. Feb 1, 2019 [cit. 2021-05-12]. Dostupné z: <https://www.freecodecamp.org/news/the-difference-between-a-framework-and-a-library-bd133054023f/>
- [17] FOWLER, Martin. *Inversion of Control Containers and the Dependency Injection pattern* [online]. 23 January 2004n. l. [cit. 2021-05-17]. Dostupné z: <https://martinfowler.com/articles/injection.html>
- [18] SmartBear. *API Endpoints - What Are They? Why Do They Matter?* [online]. 2021 [cit. 2021-05-17]. Dostupné z: <https://smartbear.com/learn/performance-monitoring/api-endpoints/>
- [19] GUO Shu-yu, FICARRA Michael, GIBBONS Kevin. *ECMAScript® 2022 Language Specification* [online]. April 6, 2021 [cit. 2021-05-17]. Dostupné z: <https://tc39.es/ecma262/>
- [20] KANTOR, Ilya. *An Introduction to JavaScript* [online]. 4th January 2021 [cit. 2021-05-17]. Dostupné z: <https://javascript.info/intro>
- [21] AZAM, Shehroz *What is Vue.js, and Why is it Cool?* [online]. 2021 [cit. 2021-05-17]. Dostupné z: https://linuxhint.com/about_vue_js/

-
- [22] NuxtJS Team. *NuxtJs - The Intuitive Vue Framework* [online]. [cit. 2021-05-17]. Dostupné z: <https://nuxtjs.org/>
- [23] Google, *AngularJS* [online]. 2010-2018 [cit. 2021-05-17]. Dostupné z: <https://angularjs.org/>
- [24] Mozilla and individual contributors. *HTML: HyperText Markup Language* [online]. 2021 [cit. 2021-05-17]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [25] W3C. *HTML 5.2 W3C Recommendation* [online]. 14 December 2017n. 1. [cit. 2021-05-17]. Dostupné z: <https://www.w3.org/TR/html52/>
- [26] WWW Consortium. *Cascading Style Sheets* [online]. 1994-2018 [cit. 2021-05-20]. Dostupné z: <https://www.w3.org/Style/CSS/Overview.en.html>
- [27] STEWART, Matt. *Designing the Ideal Navigation Bar* [online]. 2018 [cit. 2021-05-20]. Dostupné z: <https://www.thecreativemomentum.com/blog/designing-the-ideal-navigation-bar>
- [28] NuxtJS Team. *I18n.nuxtjs* [online]. 2021 [cit. 2021-05-20]. Dostupné z: <https://i18n.nuxtjs.org/>
- [29] Typicode. *Json-server* [online]. Nov 15, 2020 [cit. 2021-05-20]. Dostupné z: <https://github.com/typicode/json-server>
- [30] PETRIE, Brock. *Vue-moment* [online]. 2021 [cit. 2021-05-20]. Dostupné z: <https://github.com/brockpetrie/vue-moment?ref=madewithvuejs.com>
- [31] RAKOVSKI, Filip. *Introduction to Vue - key concepts and why it's so awesome* [online]. October 8, 2020 [cit. 2021-05-26]. Dostupné z: <https://buddy.works/tutorials/introduction-to-vue-key-concepts>
- [32] PAVLUTIN, Dmitri. *The Art of Writing Small and Plain Functions* [online]. August 31, 2016 [cit. 2021-05-26]. Dostupné z: <https://dmitripavlutin.com/the-art-of-writing-small-and-plain-functions/>
- [33] YOU, Evan. *Vue.js Installation* [online]. 2021 [cit. 2021-05-26]. Dostupné z: <https://vuejs.org/v2/guide/installation.html>
- [34] PINE, Wu. *Vetur* [online]. 2021 [cit. 2021-05-26]. Dostupné z: <https://vuejs.github.io/vetur/>
- [35] NGUYEN HAI, Nam. *Frontend-v2* [online]. 2021 [cit. 2021-05-28]. Dostupné z: <https://gitlab.fit.cvut.cz/anketa-cvut-v3/frontend-v2>

- [36] GARG, Ashwani. *VueJS created() vs mounted(), Life Cycle Hooks* [online]. Feb 16, 2020 [cit. 2021-05-28]. Dostupné z: <https://medium.com/@akgarg007/vuejs-created-vs-mounted-life-cycle-hooks-74c522b9ceee>
- [37] Xul.fr. *SVG, graphics in XML for the Web* [online]. 2021 [cit. 2021-05-28]. Dostupné z: <https://www.xul.fr/en-xml-svg.php>
- [38] BootstrapVue team. *Bootstrap Icons* [online]. 2021 [cit. 2021-06-20]. Dostupné z: <https://bootstrap-vue.org/docs/icons>
- [39] Code-farmer-i. *Vue-markdown-editor* [online]. 2021 [cit. 2021-06-20]. Dostupné z: <https://github.com/code-farmer-i/vue-markdown-editor>
- [40] BootstrapVue team *Layout and Grid System* [online]. 2021 [cit. 2021-06-22]. Dostupné z: <https://bootstrap-vue.org/docs/components/layout>
- [41] Shakee93. *Vue-toasted* [online]. 2021 [cit. 2021-06-25]. Dostupné z: <https://github.com/shakee93/vue-toasted>
- [42] MIT license *Moment.js* [online]. 2021 [cit. 2021-06-25]. Dostupné z: <https://momentjs.com/>
- [43] NuxtJS Team. *Nuxt \auth* [online]. 2021 [cit. 2021-06-26]. Dostupné z: <https://auth.nuxtjs.org/>
- [44] BootstrapVue team *Navbar* [online]. 2021 [cit. 2021-06-25]. Dostupné z: <https://bootstrap-vue.org/docs/components/navbar>
- [45] PESCATORE, Ubaldo. *Vue-country-flag* [online]. 2021 [cit. 2021-06-25]. Dostupné z: <https://github.com/P3trur0/vue-country-flag/>

DODATOK **A**

Prílohy

The image shows a web-based survey form. At the top left, there is a dark blue header with the text 'Anketa' and 'Domovská stránka. Moje listky 7 zbyvaj'. To the right of the header, the user's name 'Nam Nguyen Hai' is displayed. The main title of the survey is 'Hodnotíte Operační systémy'. Below the title, there are two main sections: 'Celkový dojem' and 'Obsah předmětu'. The 'Celkový dojem' section features a horizontal row of seven smiley face icons representing a rating scale from 1 (sad) to 7 (happy). The 'Obsah předmětu' section has a text input field with a small arrow icon on the right. Below these sections, there are two more text input fields: 'Přidat komentář: Obsah předmětu' and 'Přidat komentář: Zakoření předmětu'. At the bottom right of the form, there is a blue button labeled 'Uložit a odeslat'.

Obr. A.1: Anketový lístok predmetu „Operační systémy.“

Anketa Demovská stránka · Moje listky **3 zbyvaj** · Výsledky **Nam Nguyen Hai**

Předmětová anketa FIT léto 19/20

- BI-OSY - Operační systémy **zbyvaj 2 měsíce na vyhodnocení** [Vydřit se](#)
- BI-SQL.1 - Jazyk SQL, pokročilý **zbyvaj 2 měsíce na vyhodnocení** [Vydřit se](#)
- BI-OSY - Bezpečnost **hodnoceno před 3 měsíci**
- BI-SIL.2 - Softwarové inženýrství I **předmět ještě není dokončen**
- TVW - Tělesná výchova **předmět není hodnocený**

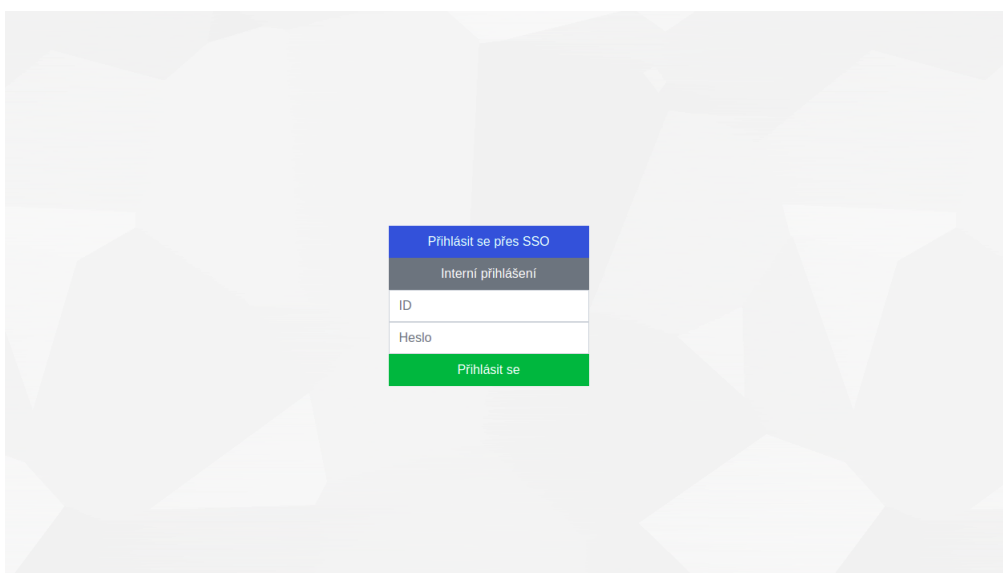
Fakultní anketa FIT léto 19/20

FIT léto 19/20 **hodnoceno před 3 měsíci**

Fakultní anketa FIT léto 19/20

FIT léto 19/20 **zbyvaj 2 měsíce na vyhodnocení** [Vydřit se](#)

Obr. A.2: Nová nástěnka.



Obr. A.3: Prihlasovacia stránka.

Zoznam použitých skratiek

ČVUT České vysoké učení technické v Praze

DOM Document Object Model

XSS Cross Site Scripting

URL Uniform resource locator

SPA Singlepage application

API Application programming interface

REST Representational state transfer

CSS Cascading Style Sheets

CRUD Create, Read, Update, Delete

JSON JavaScript Object Notation

HTTP Hypertext Transfer Protocol

XML Extensible markup language

SVG Scalable Vector Graphics

HTML HyperText Markup Language

Obsah priloženého CD

	readme.txt.....	stručný popis obsahu CD
	src	
	impl	zdrojové kódy implementácie
	thesis	zdrojový kód práce vo formáte \LaTeX
	text	text práce
	thesis.pdf	text práce vo formáte PDF