



Zadání bakalářské práce

Název:	Anketa ČVUT - refaktoring a rozvoj administrátorského rozhraní
Student:	Monika Vaňová
Vedoucí:	Ing. Michal Valenta, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Softwarové inženýrství 2021
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

Cílem práce je aktualizace vývojové platformy administrátorského modulu aplikace Anketa ČVUT, dále pak návrh a implementace nových funkcí a redesign některých stávajících. Postupujte v těchto krocích:

1. Seznamte se s aktuálním stavem administrátorského modulu aplikace Anketa ČVUT.
2. Proveďte upgrade frontend na podporovanou verzi (18 nebo 20 LTS).
3. Analyzujte, navrhňte a implementujte či opravte níže uvedené funkce (jedná se o backend i frontend části):
 - a) vyhodnocení ankety,
 - b) generování CSV exportů (podklady pro slosování, souhrn textových odpovědí v anketách),
 - c) skrývání nevhodných komentářů (cenzura),
 - d) úprava anketních otázek.
4. Implementaci řádně zdokumentujte a vhodným způsobem otestujte.

Bakalářská práce

**ANKETA ČVUT –
REFAKTORING
A ROZVOJ
ADMINISTRÁTORSKÉHO
ROZHRANÍ**

Monika Vaňová

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: Ing. Michal Valenta, Ph.D.
16. května 2024

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2024 Monika Vaňová. 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, je nezbytný souhlas autora.

Odkaz na tuto práci: Vaňová Monika. *Anketa ČVUT – refaktoring a rozvoj administrátorského rozhraní*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

Chtěla bych poděkovat svému vedoucímu práce Ing. Michalu Valentovi za možnost podílet se na vývoji reálně používaného projektu a za veškerý čas a podporu po celý čas vývoje. Dále bych chtěla poděkovat své rodině za všechnu pomoc a podporu poskytnutou během práce nad bakalářskou prací.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při 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, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona. Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při 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, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 16. května 2024

Abstrakt

Tato bakalářská práce se zabývá aktualizací, implementací nových funkcionalit a dokumentací administrátorského modulu aplikace Anketa ČVUT, který je určen fakultním administrátorům a superadministrátorům pro usnadnění správy anket. Nejdříve byl proveden upgrade použitých technologií frontendové části vytvořené v jazyce Javascript za použití frameworku Nuxt a UI knihovny Vuetify. Následoval návrh a implementace nových funkcionalit na backendu i frontendu. Jedná se o přidání tlačítka pro vyhodnocení ankety přímo v modulu, které je aktivní pouze pokud je uzavřeno vyplňování ankety a možnost reakce na komentáře a pokud anketa doposud nebyla vyhodnocena. Dále implementace tlačítka pro skrývání a odkrývání nevhodných komentářů (cenzura), generování CSV exportů studentů, kteří vyplnili nějaký anketní lístek ankety a CSV export komentářů k předmětům, vyučujícím i fakultám. Byla také navrhuta implementace změny způsobu přihlašování pomocí jména a hesla na SSO (single sign on) přihlašování používané v rámci ČVUT i ve vyplňovací části aplikace Anketa ČVUT. Vše bylo zdokumentováno v návodech pro fakultní administrátory a v komentářích v kódu frontendu i backendu, ze kterých byla k oběma částem vygenerována dokumentace. Pomocí sestavených scénářů užití proběhlo testování administrátorského modulu jako celek a na backendu byly vytvořeny unit testy pro ověření funkčnosti nejdůležitějších servisních tříd.

Klíčová slova Anketa ČVUT, administrátorský modul, správa webové aplikace, technologický upgrade, nové funkcionality, SSO přihlašování, Nuxt 3, Javascript, Vuetify, Java

Abstract

This bachelor thesis deals with actualization, implementation of new functionalities and documentation of the administration module of the application CTU Survey, which is intended for the faculty administrators and superadministrators for easier management of surveys. The technological upgrade of used technologies on the frontend part created in the Javascript language using the framework Nuxt and UI library Vuetify was done first. The design and implementation of new functionalities on the backend and frontend was next. It included adding a button for survey evaluation directly in the module, which is active only if the survey was closed for new answers and reactions to comments and if it wasn't evaluated yet. Next was the implementation of a button for hiding or revealing inappropriate comments (censorship), generating CSV exports of students who filled out some survey card of a survey and CSV exports of comments on courses, teachers and faculties. The implementation of the change of signing in by username and password to the SSO (single sign on) signing in which is used within CTU even in the filler part of the CTU Survey application was designed as well. Everything was documented in the manuals for the faculty administrators and in the comments in the code of both frontend and backend from which was generated the documentation of both parts. By using created use case scenarios a user testing of the administration module as a whole was done and on the backend were created unit tests for verifying the functionality of the most important service classes.

Keywords CTU Survey, administration module, administration of a web application, technological upgrade, new functionalities, SSO sing in, Nuxt 3, Javascript, Vuetify, Java

Obsah

Úvod	1
1 Aplikace Anketa ČVUT	2
1.1 Základní pojmy	2
1.2 Průběh ankety	2
1.3 Vyplňovací část	3
1.4 Administrátorský modul	3
1.4.1 Aktuality	4
1.4.2 Ankety	4
1.4.3 Otázky	5
1.4.4 Seznamy otázek	5
1.4.5 Hlášení chyb	6
1.4.6 Menu pro superadministrátory	6
2 Metodika práce	9
2.1 Iterativní vývoj	9
2.2 Testování	9
2.3 Nasazení	10
3 Architektura aplikace a použité technologie	11
3.1 Architektura jako celek	11
3.2 Uživatelské rozhraní (Frontend)	11
3.2.1 Architektura	11
3.2.2 Použité technologie	12
3.3 Serverová část (Backend)	12
3.3.1 Architektura	12
3.3.2 Použité technologie	13
3.4 Databáze	13
3.5 SSO	13
3.5.1 Obecné nastavení Shibboleth	14
4 Cíle práce, požadavky	15
4.1 Technologický upgrade	15
4.2 Vyhodnocení ankety	15
4.3 Generování CSV exportů	15
4.4 Skrývání nevhodných komentářů	15
4.5 Úprava anketních otázek	16
4.6 SSO přihlašování	16
4.7 Dokumentace	16
4.8 Návody	16

5	Návrh a implementace	17
5.1	Technologický upgrade	17
5.1.1	Základní nastavení	17
5.1.2	Nuxt.js	17
5.1.3	Vuetify	22
5.2	Vyhodnocení ankety	22
5.3	Generování CSV exportů	22
5.4	Skrývání nevhodných komentářů	23
5.5	Úprava anketních otázek	24
5.6	SSO přihlašování	25
5.7	Úprava uživatelů	25
5.8	Úprava přidávání a odebírání učitelů a předmětů	26
5.9	Úprava vzorů	26
5.10	Úprava základních informací anket	27
5.11	Dokumentace	27
5.11.1	Frontend	27
5.11.2	Backend	28
5.12	Návody	29
6	Nasazení a testování	30
6.1	Backend unit testy	30
6.2	Uživatelské testování	31
6.2.1	Úloha 1 – Úprava základních informací ankety	31
6.2.2	Úloha 2 – Přidávání a odebírání předmětů a učitelů v anketě	32
6.2.3	Úloha 3 – Práce s otázkami a sadami otázek	32
6.2.4	Úloha 4 – Skrývání komentářů, generování CSV exportů	33
6.2.5	Celkové připomínky	33
6.3	Nasazení	33
	Závěr	34
	A Instalace a konfigurace	35
	Bibliografie	36
	Obsah příloh	39

Seznam obrázků

1.1	Průběh ankety, převzato z metodického pokynu [1], strana 5	3
1.2	Záložka Aktuality – úvodní obrazovka	4
1.3	Záložka Ankety – vyhledávání anket s obsahujícími názvy fit	4
1.4	Záložka Otázky – filtrované primární otázky a rozbalené menu s výběrem typu	5
1.5	Záložka Seznamy otázek	5
1.6	Dialogové okno pro nahlášení chyby	6
1.7	Seznam nahlášených chyb na záložce Chyby	6
1.8	Seznam všech uživatelů na záložce Uživatelé v menu pro superadministrátory	7
1.9	Seznam aktualit na záložce Aktuality v menu pro superadministrátory	7
1.10	Záložka Nástěnka v menu pro superadministrátory	8
3.1	Schéma SSO přihlašování, převzato ze stránek poskytovatele služeb Shibboleth [20]	14
5.1	Disabled tlačítko pro vyhodnocení ankety, která již byla vyhodnocena	22
5.2	Tlačítko s ikonkou pro CSV export komentářů k předmětům	23
5.3	Ukázka tabulky s komentáři včetně ikonky pro export CSV souboru	24
5.4	Okno pro úpravu vzoru Vyjezd	26
5.5	Okno pro úpravu základních informací ankety	27
5.6	Ukázka dokumentace komponenty AddNewsModal	28
5.7	Ukázka dokumentace třídy QuestionsController	29

Seznam tabulek

Seznam výpisů kódu

5.1	Plugin vue-markdown	18
5.2	Plugin vue3-markdown	18
5.3	Get request pomocí Axios	18
5.4	Get request pomocí fetch API	18
5.5	Zobrazení textu itemToReturn.text ve formátu markdown	19
5.6	Ukázka použití funkce format knihovny date-fns	20
5.7	Ukázka použití knihovny vue-toasted pro zobrazení informační hlášky	20

5.8	Snackbar komponenta	20
5.9	Snackbar úložiště	21
5.10	Funkce komponenty CsvExportButton	23
5.11	RequestMapper v souboru shibboleth2.xml	25
5.12	JSDoc komentář s JSDoc a jsdoc-vuejs anotacemi	27
5.13	Javadoc komentář s Javadoc anotacemi	28
6.1	Metoda objectToBooleanConversion testovací třídy BaseServiceTest	30
6.2	Metoda getFacultyComments testovací třídy CommentServiceTest	30

Seznam zkratek

- API** Application Programming Interface. 11, 13, 18, 19, 28, 29
- CI/CD** Continuous Delivery/Continuous Deployment. 10, 33
- CSV** Comma-separated values. iv, v, 1, 15, 22, 23, 33, 34
- CTU** Czech Technical University. v
- DTO** Data Access Object. 30
- EOL** End of Life. 17
- HTML** Hypertext Markup Language. 20, 28
- HTTP** Hypertext Transfer Protocol. 14, 18
- IDE** Integrated development environment. 10
- IdP** Identity Provider. 13, 14
- JDBC** Java Database Connectivity. 13
- JSON** Javascript Object Notation. 22
- JWT** Json Web Token. 13, 25
- KOS** Komponenta studium – studijní informační systém ČVUT. 2
- LTS** Long-term support. 15
- MVVM** Model-View-ViewModel. 11, 12
- npm** Node Package Manager. 17
- REST** Representational State Transfer. 13, 18, 19, 29
- SAML** Security Assertion Markup Language. 14
- SP** Service Provider. 13, 14
- SPA** Single Page Applications. 12
- SSH** Secure Shell. 33
- SSO** Single Sign on. iv, v, 13, 14, 16, 25, 34

SSR Server-side rendering. 12

UI User Interface. iv, v, 12, 20, 22, 29

url Uniform Resource Locator. 14, 17, 25, 33

WAR Java Web Archive. 33

ČVUT České Vysoké Učení Technické. iv, 1, 2, 14, 16, 25, 26, 29, 34

Úvod

Anketa ČVUT je webová aplikace, pomocí které mohou studenti hodnotit zapsané předměty, vyučující daných předmětů a také svoji fakultu jako celek. Zpětná vazba od studentů umožňuje vyučujícím a vedení fakult zlepšovat výuku a prostředí fakult podle potřeb studentů. Administrátorský modul aplikace Anketa ČVUT slouží pro správu anket a dat s nimi spojených. Je určena fakulním administrátorům a superadministrátorům Ankety. Tato práce se zabývá aktualizací administrátorského modulu a jeho rozvojem za účelem usnadnění správy anket. Cílem teoretické části je analýza tohoto modulu z pohledu uživatelského rozhraní i serverové části. Dále je cílem analyzovat stávající vývojové platformy uživatelského rozhraní modulu za účelem jejich aktualizace.

Cílem praktické části práce je zmíněná aktualizace použité vývojové platformy uživatelského rozhraní na novější dlouhodobě podporovanou verzi. Dále analýza, návrh a implementace nových funkcí, které ve stávajícím modulu nejsou implementovány, a redesign či oprava vybraných existujících funkcí. Nové funkcionality jsou možnost vyhodnocení jednotlivých anket přímo v modulu, generování CSV exportů studentů, kteří vyplnili nějaký anketní lístek, což slouží jako podklad pro slosování, možnost skrývání nevhodných komentářů a možnost úpravy anketních otázek fakulním administrátorům. Implementace bude zdokumentována a vhodným způsobem otestována.

Práce navazuje na bakalářskou práci Jakuba Juna, v rámci které bylo vytvořeno uživatelské rozhraní administrátorského modulu, a na práci Duc Thang Nguyena, která se zabývá implementací serverové části.

Kapitola 1

Aplikace Anketa ČVUT

Anketa ČVUT (dále jen Anketa) je webová aplikace, pomocí které mohou studenti ČVUT hodnotit jak absolvované předměty a jejich vyučující, tak svoji fakultu jako celek. Tyto funkce zajišťuje vyplňovací část aplikace (dále jen Vyplňovací část). V roce 2020 byl vyvinut administrátorský modul, který zajišťuje snadnější správu anket fakultním administrátorům a superadministrátorům.

1.1 Základní pojmy

V aplikaci Anketa hrají důležitou roli zejména anketní lístky, které tvoří jednotlivé ankety. Každý anketní lístek obsahuje seznam otázek, který může obsahovat jednu nebo dvě sady otázek podle toho, jestli se jedná o předmětovou nebo nepředmětovou anketu. Předmětová anketa se týká předem určené množiny předmětů a vyučujících, kteří dané předměty vyučují. Nepředmětová anketa se týká fakulty jako celku. Student má možnost vyplnit tolik anketních lístků, kolik měl daný semestr zapsaných předmětů a navíc jeden anketní lístek vztahující se k jeho fakultě jako celku. Zmíněné sady otázek se skládají z otázek, které mohou být hodnotové, textové, nebo kombinované. Na hodnotové otázky lze odpovědět hodnotou (známkou) od 1 do 5, textové otázky očekávají textovou odpověď a kombinované otázky se skládají z obou předchozích typů. [1]

1.2 Průběh ankety

Anketa prochází během svého životního cyklu čtyřmi základními fázemi, které budou rozebrány níže. V rámci fází existuje několik stavů, ve kterých se anketa může nacházet. [1]

První fáze procesu ankety je Příprava, ve které administrátoři systému provádějí přípravu dat pro nový běh ankety. Tato příprava zahrnuje několik klíčových úkolů zahrnujících natažení dat ze studijního informačního systému KOS a další úpravy podle potřeb jednotlivých fakult. Administrátoři mají možnost kontrolovat zahrnuté nebo vyloučené předměty a také vyučující v jednotlivých předmětech a provést případné změny v těchto datech.

Další fází je Vyplňování, ve kterém je anketa otevřena studentům pro vyplnění. Předmětové ankety mají dva módy vyplňování: Mód K, ve kterém mají studenti možnost vyplnit ankety pouze pro ty předměty, které řádně ukončili, a mód Z, ve kterém je studentům umožněno vyplnit ankety pro všechny předměty, které měli zapsané v daném semestru.

Po uzavření ankety probíhá fáze Vyhodnocení. Vyučujícím a vedení fakult jsou výsledky zveřejněny jako první, aby se mohli vyjádřit ke komentářům, které se na ně vztahují. Také je umožněno vedení fakult skrývat nevhodné komentáře. Poté jsou všechny komentáře dostupné všem vyučují-

cím a vedení bez možnosti reakce. Poslední fází je zveřejnění výsledků ankety, kdy jsou výsledky zveřejněny jak pro vyučující a vedení, tak pro studenty.

Tento cyklus se opakuje každý semestr a zahrnuje různé úrovně přístupu a možnosti interakce pro jednotlivé role v systému [1]. Správné provedení každé fáze je klíčové pro získání relevantní zpětné vazby a úspěšné hodnocení výuky. Detailnější popis průběhu ankety je uveden v Metodickém pokynu [1]. V tabulce na obrázku 1.1 převzatého z Metodického pokynu [1] je vizuálně znázorněn průběh ankety.

Přístupující role	Fáze	Stav	Událost / činnost	
Správci (ANSS, ANFS)	příprava	P	D0 (kontrola před spuštěním)	
			D1 - otevření ankety	
Studenti	vyplňování	K (ank. lístky ukon. př.)	D1_2 přepnutí stavu K -> Z	
		Z (ank. lístky zaps.př.)	D1_3 podklady pro slosování	
			D2 - uzavření ankety	
Vyučující, vedení, komentátoři (ANVED, ANKOM)	vyhodnocení	U	O (Owner)	
				vyjádření ke komentářům
				skrytí nevhodných komentářů
		A (All)	D2_2 přepnutí stavu O -> A	
			D3 – publikace výsledků	
Studenti, vyučující, vedení (ANNAH)	zveřejnění		RO (Read Only)	nahlížení výsledků

■ Obrázek 1.1 Průběh ankety, převzato z metodického pokynu [1], strana 5

1.3 Vyplňovací část

Vyplňovací část ankety je využívána zejména ve fázi Vyplňování a je určena pro studenty, kteří přes ni mohou vyplnit ankety vztahující se na předměty, které měli daný semestr zapsané, a ankety vztahující se k jejich fakultě. Dále je určena pro vyučující, kteří si přes ni mohou prohlížet výsledky anket, které se na ně vztahují, a reagovat na hodnocení. Kromě vyplňování a reagování na komentáře je v této části aplikace umožněno všem uživatelům prohlédnout si výsledky všech uplynulých vyhodnocených anket a zobrazit si obecné informace o anketách jak podle fakulty, tak podle kategorie ankety. Zmíněné funkce jsou umístěny na třech základních záložkách, mezi kterými je možné libovolně přecházet. Detailnější popis vyplňovací části je uveden v bakalářské práci Jakuba Juna [2] v kapitole 2.2 Vyplňovací část.

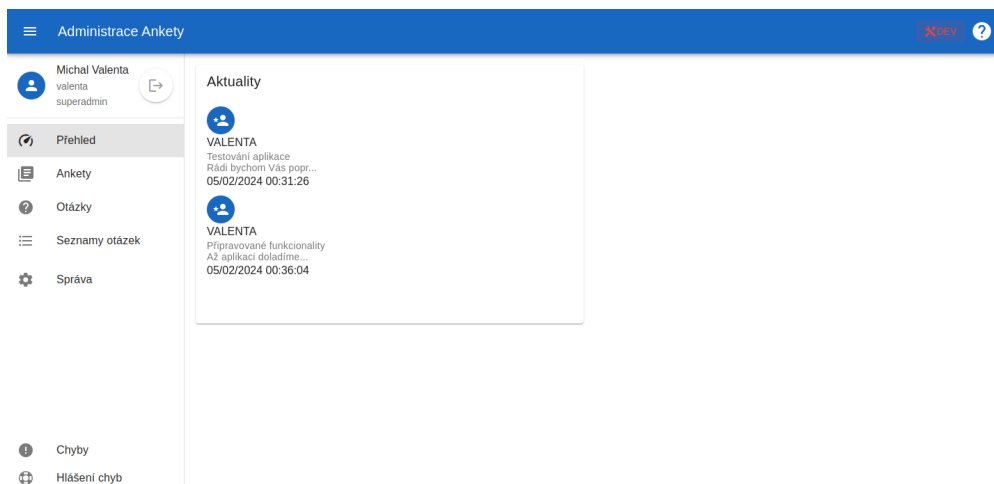
1.4 Administrátorský modul

Tato práce se zabývá především administrátorským modulem Ankety. Uživatelé této části mohou být buď fakultní administrátoři, kteří mají základní práva na prohlížení a upravování dat, nebo superadministrátoři, kteří mohou upravovat také údaje jiných administrátorů a zveřejňovat aktuální informace k průběhu Ankety.

Administrátorský modul obecně slouží ke snadnější organizaci dat Ankety, jako je přepínání stavů anket či úprava anketních otázek, a k pohodlnějšímu zobrazování dat jednotlivých anket včetně textových komentářů. Má jednoduše ovladatelné uživatelské rozhraní s navigačním panelem po levé straně. Pomocí tohoto panelu je možné se pohybovat ve čtyřech základních částech aplikace a pro superadministrátory je v panelu zobrazeno také rozbalovací menu se záložkami určenými pro správu uživatelů, aktualit a nástěnků Vyplňovací části.

1.4.1 Aktuality

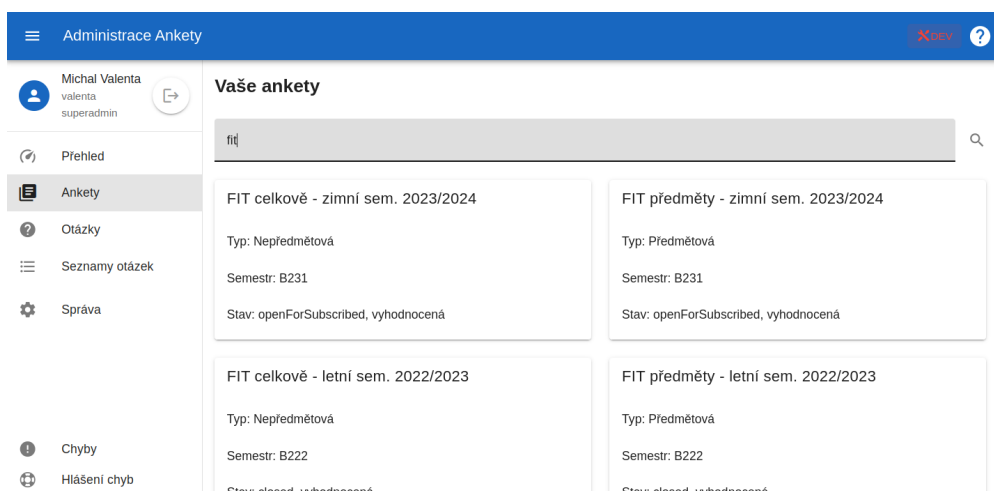
První záložkou jsou Aktuality 1.2. Zde se zobrazují aktuální informace zveřejněné superadministrátory všem administrátorům. Jedná se také o úvodní stránku aplikace po úspěšném přihlášení.



■ Obrázek 1.2 Záložka Aktuality – úvodní obrazovka

1.4.2 Ankety

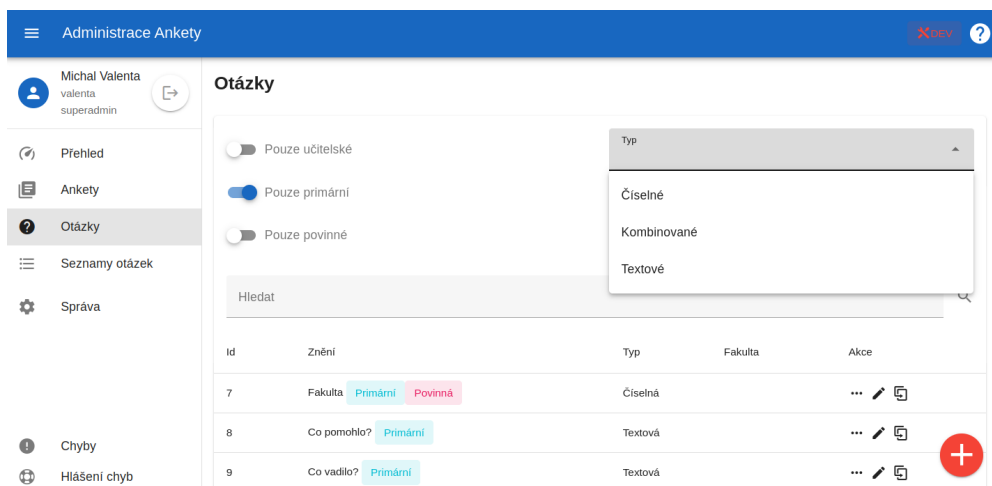
Další záložkou jsou Ankety. V existujících anketách je možné vyhledávat podle jejich jména 1.3 a zobrazit si jejich detail, ve kterém jsou uvedeny obecné informace anket, jako jsou aktuální stavy anket, její typ a použitá sada otázek. Pro předmětové ankety se zobrazují také zahrnuté předměty, zahrnutí učitelé a vzory pro vyloučení předmětů. Tyto údaje je možné upravovat. Dále jsou v detailu anket zobrazeny textové komentáře studentů vztahující se na otázky v dané anketě.



■ Obrázek 1.3 Záložka Ankety – vyhledávání anket s obsahujícími názvy fit

1.4.3 Otázky

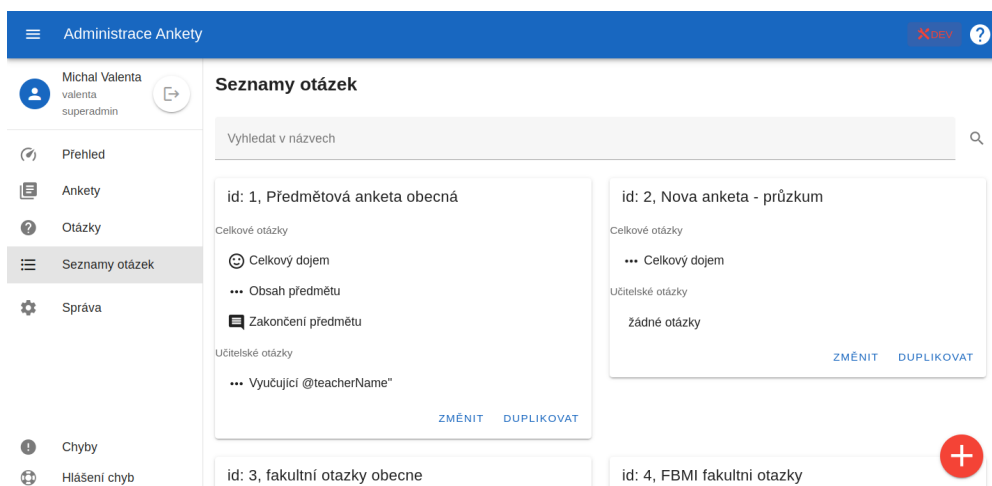
Následuje záložka Otázky 1.4, ve které se zobrazují všechny existující otázky a je možné je filtrovat podle jejich typu. Přidání nové otázky se provádí pomocí červenobílého tlačítka se znakem "+" v pravém dolním rohu. Vytvořené otázky lze zobrazit, upravit nebo duplikovat pomocí ikonek na konci každého řádku.



Obrázek 1.4 Záložka Otázky – filtrované primární otázky a rozbalené menu s výběrem typu

1.4.4 Seznamy otázek

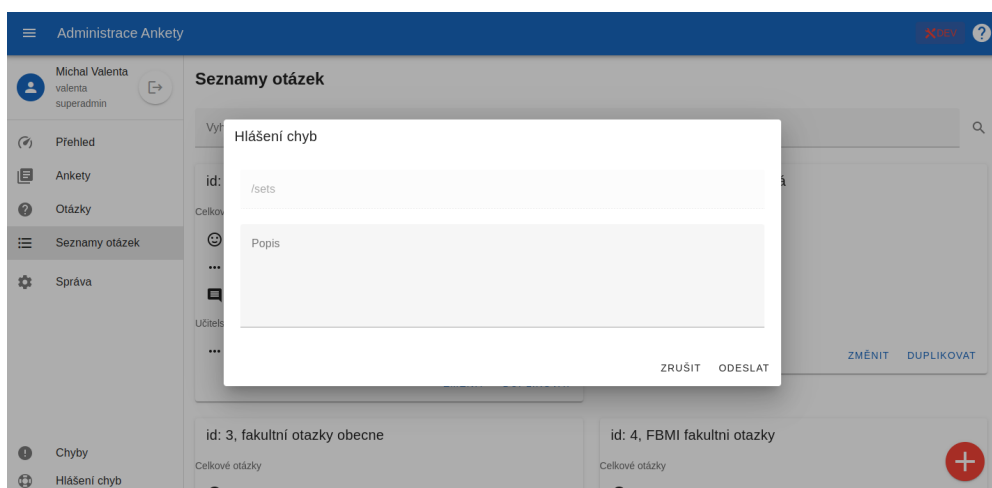
Poslední základní záložkou jsou Seznamy otázek 1.5. Zde je možné seskupit několik otázek do skupin, které lze následně použít v jednotlivých anketách. V seznamech je možné vyhledávat podle názvu, opět se přidávají pomocí tlačítka v pravém dolním rohu a je možné je upravovat a duplikovat pomocí tlačítek ve spodní části dané karty se seznamem.



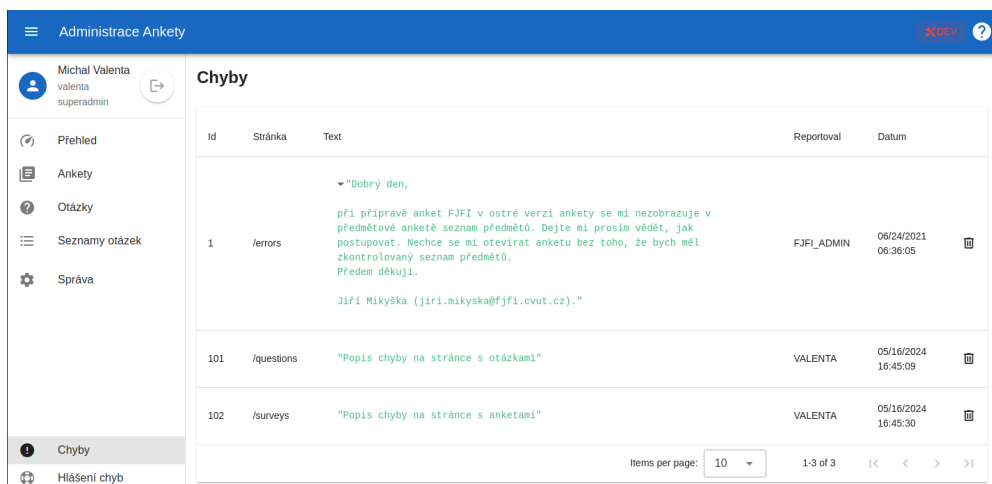
Obrázek 1.5 Záložka Seznamy otázek

1.4.5 Hlášení chyby

Kromě základních čtyř záložek obsahuje administrátorské rozhraní také tlačítko Hlášení chyb, které je umístěno v levém dolním rohu v navigačním panelu. Po jeho stisknutí se otevře dialogové okno 1.6 s textovým polem, ve kterém je již vyplněn konec url stránky, na které se administrátor nachází a ve které nastala chyba, kterou chce administrátor nahlásit. Toto pole není možné měnit. Následuje pole Popis určené pro popsání chyby. Každý administrátor si může nahlášené chyby zobrazit na záložce Chyby umístěné v navigačním panelu nad tlačítkem Hlášení chyb 1.7.



Obrázek 1.6 Dialogové okno pro nahlášení chyby



Obrázek 1.7 Seznam nahlášených chyb na záložce Chyby

1.4.6 Menu pro superadministrátory

Jak již bylo zmíněno, pro superadministrátory nabízí administrátorský modul možnost základní správy uživatelů, aktualit a nástěnky Vyplňovací části Ankety. Po přihlášení uživatele s rolí superadministrátora se v navigačním panelu pod základními záložkami zobrazí rozbalovací menu s názvem Správa, které obsahuje záložky Uživatelé, Aktuality a Nástěnka. Na záložce Uživatelé

je tabulka se všemi uživateli a pomocí ikoněk na konci každého řádku tabulky je možné daného uživatele upravit nebo smazat. Nové uživatele je možné přidávat pomocí červenobílého tlačítka se znakem "+", podobně jako otázky a seznamy otázek.

Id ↑	Fakulta	Uživatelské jméno	Jméno a příjmení	Role	Akce
1		VALENTA 👤 To jste Vy!	Michal Valenta	superadmin	
2	Fakulta stavební	DOCKAMAR	Martin Dočkal	admin	
3	Fakulta strojní	HLAVAVLA	Vladimír Hlaváč	admin	
4	Fakulta elektrotechnická	JELINEK	Ivan Jelinek	admin	
5	Fakulta jaderná a fyzikálně inženýrská	MIKYŠJIR	Jiří Mikyška	admin	
6	Fakulta architektury	KASTLVER	Veronika Šindelář Kastlová	admin	
7	Fakulta architektury	IRINKPET	Petr Irínkov	admin	
8	Fakulta dopravní	HORAZPAT	Patrik Horazdovský	admin	
9	Fakulta dopravní	MICHLZDE	Zdeněk Michl	admin	
10	Masarykův ústav	ANDRFPVAV	Pavel Andres	admin	

Obrázek 1.8 Seznam všech uživatelů na záložce Uživatelé v menu pro superadministrátory

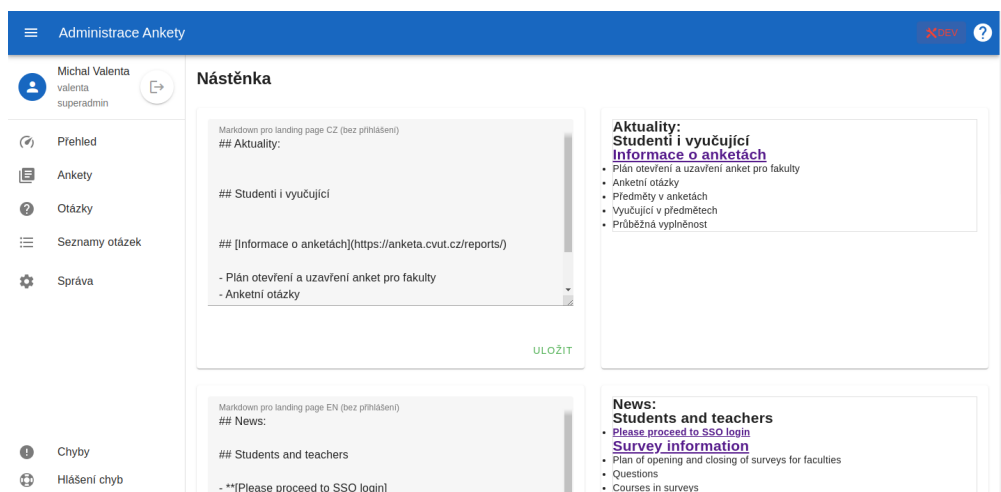
Na záložce Aktuality mohou superadministrátoři vytvářet aktuality, které se zobrazí na první základní záložce administrátorského rozhraní, která je také úvodní obrazovkou po přihlášení. Přidávání probíhá opět pomocí tlačítka v pravém dolním rohu. Pokud je text aktuality delší, je možné ho rozbalit pomocí šipky na začátku daného řádku a dále je možné aktuality upravovat a mazat pomocí ikoněk na konci každého řádku.

Id	Titulek	Autor	Přidáno	Zveřejněno	Text	Akce
23	Testování aplikace	VALENTA	05/02/2024 00:31:26	Ano	Rádí bych...	
24	Připravované funkcionality	VALENTA	05/02/2024 00:36:04	Ano	<p>Až aplikaci doladíme a nasadíme, budete si moci na fakultách sami realizovat některé další věci. Konkrétně:</p> <ul style="list-style-type: none"> • vyhodnotit a publikovat uzavřenou anketu • získávat seznamy studentů, kteří anketu vyplili (např. pro slosování) • získávat exporty textových odpovědí • skrývat nevhodné textové odpovědi studentů (cenzura) 	

Items per page: 10 1-2 of 2 |< < > >|

Obrázek 1.9 Seznam aktualit na záložce Aktuality v menu pro superadministrátory

Záložka Nástěnka 1.10 slouží k úpravě nástěnky Vyplňovací části aplikace Anketa. Obsahuje čtyři textová pole pro úpravu textu s náhledem na každé pole ve formátu markdown. Je možné zde upravit nástěnku Výsledkové části zobrazenou nepřihlášeným a přihlášeným uživatelům v českém i anglickém jazyce.



■ Obrázek 1.10 Zálůžka Nástěnka v menu pro superadministrátory

Metodika práce

2.1 Iterativní vývoj

Při práci na stanovených cílech bylo postupováno formou iterativního vývoje. Iterativní vývoj, někdy nazýván také proces extrémního programování [3], je vývojová metoda, která rozděluje proces vývoje software do opakujících se cyklů nazývaných iterace. Každá iterace zahrnuje plánování, návrh, implementaci a testování části produktu [3].

Iterace začíná plánováním, při kterém si tým rozmyslí a určí, které funkcionality nebo úkoly budou zahrnuty do této iterace. Mohou se rozhodnout prioritizovat určité funkce na základě zpětné vazby, kterou obdrželi od uživatelů nebo zákazníků. Následuje návrh, při kterém se tým začne zabývat návrhem a architekturou části produktu, která má být v této iteraci vyvinuta. Diskutují o tom, jak implementovat požadovanou funkcionalitu a jak bude součástí systému komunikovat s ostatními částmi. Po dokončení návrhu se tým přesune k implementaci. Programátoři začnou psát kód podle specifikací a návrhů definovaných v předchozí fázi. Během implementace mohou vznikat nové nápady nebo požadavky, které mohou být začleněny do této nebo budoucích iterací. Po dokončení implementace části produktu se přechází na testování, aby se ověřilo, že produkt funguje správně a splňuje požadavky. Testování může být automatizované nebo prováděno ručně. Zpětná vazba z testování je následně použita k vylepšení produktu nebo k určení změn potřebných v následujících iteracích. Po dokončení iterace se tým schází, aby zhodnotil, čeho bylo dosaženo, a aby přijal zpětnou vazbu. To může zahrnovat posouzení toho, co fungovalo dobře a co by bylo potřeba zlepšit nebo změnit v příští iteraci. [3]

V každé iteraci se celý proces opakuje, přičemž se postupně přidávají další funkce a vylepšení. Každá iterace by měla být krátká a měla by přinést nějakou přidanou hodnotu produktu. Tímto způsobem iterativní vývoj umožňuje týmu pružně reagovat na změny a požadavky, zatímco postupně buduje a zlepšuje produkt.

Při práci na stanovených cílech byly jednotlivé iterace odděleny konzultacemi s vedoucím práce. Konzultace začínaly zhodnocením aktuálního stavu, které zahrnovalo také identifikaci částí, které bylo třeba zlepšit nebo opravit v následující iteraci. Následovalo plánování úkolů, které by měly být splněny do další konzultace, včetně případné úpravy požadavků na základě výstupu z předchozí iterace. Mezi konzultacemi byl prostor na implementaci a testování. Takto se postupovalo až k úspěšnému splnění vytyčených cílů.

2.2 Testování

Jak již bylo zmíněno v podkapitole 2.1, testování je poslední fází jednoho cyklu iterativního vývoje, podle kterého bylo při vývoji postupováno. Testování software je verifikační proces pro zhod-

nocení kvality a zlepšení software [4]. Obecně lze rozdělit hodnocení kvality software na dvě rozsáhlé kategorie statické a dynamické analýzy [4]. Statická analýza zkoumá kód ve stavu, kdy program není spuštěn, a uvažuje všechny možné případy, které mohou za běhu programu nastat. Statickou analýzu prováděl při vývoji frontendu i backendu kompilátor použitého vývojového prostředí IDE (integrated development environment). Dynamická analýza probíhá po spuštění programu za účelem odhalení chyb, které mohou za běhu nastat. Pro tuto část testování byly na backendu napsány unit testy pro nejdůležitější servisní třídy, které zajišťují práci s databází a přeměnu dat získaných z databáze do formy, která je vracena při volání jednotlivých endpointů. Pro testování byl použit framework JUnit 5 [5], což je jeden z nejrozšířenějších testovacích frameworků pro Java ekosystém [6]. Za účelem toho, aby byla otestována pouze funkčnost jednotlivých metod, byl pro simulaci databázových volání použit framework Mockito [7]. Dále bylo provedeno uživatelské testování neboli user acceptance testing [4]. Byly sestaveny čtyři úlohy pokrývající základní činnosti administrátorů aplikace Anketa a k úlohám byl vytvořen dotazník pro přijetí zpětné vazby. Úlohy a vyplnění dotazníku bylo provedeno fakultními administrátory, pro které je administrátorský modul určen. Skrz vyplněné dotazníky byla obdržela cenná zpětná vazba, na základě které došlo k několika úpravám a přidání nových funkcionalit, aby administrátorský modul lépe vyhovoval požadavkům administrátorů. V kapitole 6 jsou popsány jednotlivé testovací úlohy a úpravy, ke kterým na základě zpětné vazby došlo.

2.3 Nasazení

Nasazení (deployment) je krok v procesu vytváření software, ve kterém je aplikace přenesena do své konečné cílové infrastruktury [8]. Skrz ni je poté přístupná pro koncové uživatele, nebo v případě testovacího prostředí pro vývojáře. Pro nasazení byly při vývoji používány GitLab pipelines [9], které zajišťují CI/CD (Continuous Delivery Continuous Deployment) [10]. Pro vytvoření GitLab pipelines stačí mít v kořenovém adresáři projektu soubor `gitlab-ci.yml`, který obsahuje skripty s jednotlivými fázemi pipelines. Díky tomuto nástroji bylo velmi ulehčeno nasazování aplikace do tří různých využívaných prostředí – vývojové, které sloužilo pro vývoj a testování nových funkcí, staging, ve kterém probíhalo finální testování před nasazením do produkčního prostředí, a produkční, kde je nasazena finální aplikace.

Architektura aplikace a použité technologie

V této kapitole jsou popsány architektura a použité technologie administrátorského modulu Ankety uživatelského rozhraní i serverové části. Jelikož se v práci navazuje na existující aplikaci, použitá architektura a technologie již byly stanovené.

3.1 Architektura jako celek

Jak je zmíněno v bakalářské práci Duc Thang Nguyena [11] v kapitole Třívrstvá architektura aplikace a s ní spojené technologie, architektura administrátorského modulu je založena na třívrstvé architektuře. Třívrstvá architektura aplikací je architektonický vzor, který rozděluje aplikaci do tří hlavních vrstev – prezentační, aplikační a datové. Každá vrstva má svůj specifický účel a zodpovědnost, což usnadňuje správu a údržbu aplikace a zlepšuje její modularitu a škálovatelnost [12].

Prezentační vrstva je zodpovědná za zobrazení uživatelského rozhraní a interakci s uživatelem. Zahrnuje veškeré prvky, které uživatel vidí a pomocí kterých může aplikaci ovládat, jako jsou webové stránky, formuláře, tlačítka a další. Typicky se jedná o část aplikace, která je přístupná přes webový prohlížeč nebo mobilní či desktopovou aplikaci.

Aplikační vrstva obsahuje logiku aplikace a provádí operace nad daty. Zajišťuje zpracování požadavků od prezentační vrstvy, provádí potřebné operace a vrací odpovídající výsledek. Často se zde nacházejí různé servisní třídy, kontroléry a další komponenty, které provádějí činnosti jako zpracování dat, validace vstupů nebo volání externích API (Application Programming Interface).

Datová vrstva obsahuje veškerá data potřebná pro chod aplikace. Zahrnuje databázi a další úložiště dat uchovávající informace, se kterými aplikace pracuje. Datová vrstva je zodpovědná za ukládání, aktualizaci, načítání a odstraňování dat z databáze nebo jiného datového úložiště.

3.2 Uživatelské rozhraní (Frontend)

3.2.1 Architektura

Uživatelské rozhraní neboli frontend reprezentuje prezentační vrstvu aplikace a slouží jako vstupní bod interakce uživatele s aplikací. Architektura samotného rozhraní je založena na principu MVVM (Model-View-ViewModel) [2]. MVVM architektura je architektonický vzor, který se často používá při vývoji aplikací s uživatelským rozhraním. Tento vzor pomáhá oddělit logiku aplikace

od prezentační vrstvy a zlepšuje škálovatelnost, testovatelnost a znovupoužitelnost kódu. Jednotlivé vrstvy MVVM vzoru jsou Model, View a ViewModel [13], [14].

Model reprezentuje datovou strukturu aplikace. Může to být například datová struktura reprezentující data získaná z databáze. Model je nezávislý na uživatelském rozhraní a obsahuje logiku pro práci s daty, jako jsou metody pro načítání, ukládání a aktualizaci dat. Obvykle neobsahuje žádnou logiku prezentační vrstvy a nemá informace o způsobu zobrazení dat.

View představuje uživatelské rozhraní aplikace, které je viditelné pro uživatele. Může to být okno, formulář, stránka nebo jakýkoli jiný prvek, se kterým uživatel interaguje. View obsahuje pouze prezentaci dat a zpravidla neprovádí žádné logické operace. Jeho hlavním účelem je zobrazit data z ViewModel a reagovat na uživatelské interakce, jako jsou kliknutí myši nebo stisknutí kláves.

ViewModel funguje jako spojka mezi Modelem a View. Obsahuje logiku aplikace a datové manipulace. ViewModel obvykle zahrnuje všechny operace, které jsou potřebné pro zpracování uživatelských požadavků a aktualizaci dat v modelu. ViewModel také poskytuje data a chování pro prezentační vrstvu. Může to být například kolekce objektů, která se zobrazuje v seznamu, nebo příznaky, které ovlivňují viditelnost různých prvků ve View.

V architektuře MVVM jsou ViewModel a View úzce propojeny pomocí techniky nazývané data binding, která umožňuje automatickou synchronizaci dat mezi ViewModel a View. Tento přístup umožňuje oddělení logiky aplikace od prezentační vrstvy a usnadňuje testování a údržbu kódu.

Architektura uživatelského rozhraní je detailněji popsána v bakalářské práci Jakuba Juna [2] v kapitole 4 Návrh uživatelského rozhraní administrátorské části.

3.2.2 Použité technologie

Pro tvorbu uživatelského rozhraní byl vybrán [2] jazyk Javascript s použitím frameworku Nuxt a frameworku Vuetify jako UI knihovny. Nuxt.js je open-source framework pro vývoj moderních webových aplikací nebo SPA (Single Page Applications) založený na Vue.js. Jedná se o vysoce produktivní nástroj, který kombinuje mnoho užitečných funkcí a konvencí, aby usnadnil vývoj webových aplikací. Některé z jeho klíčových vlastností jsou směřování založené na souborech, SSR (Server-side rendering), automatické importy, konfigurace od nuly a další. [15]

Jako UI knihovna byl použit framework Vuetify. Jedná se o open-source UI framework sloužící k vytváření moderních a responzivních uživatelských rozhraní pro webové aplikace a webové stránky. Obsahuje rozsáhlou kolekci komponent uživatelského rozhraní, pomocí kterých lze vytvořit jednotné dynamické uživatelské rozhraní zajišťující příjemnou uživatelskou zkušenost. [16]

Diskuze k výběru použitých technologií je uvedena v bakalářské práci Jakuba Juna [2] v kapitole 5 Diskuze a volba implementační platformy.

3.3 Serverová část (Backend)

3.3.1 Architektura

Serverová část aplikace neboli backend reprezentuje aplikační a částečně datovou vrstvu třívrstvé architektury. Je rozdělena do několika skupin tříd, které odpovídají různým aspektům aplikace. Architektura je detailněji popsána v bakalářské práci Duc Thang Nguyena [11] v kapitole 5.1 Architektura. Níže jsou popsány jednotlivé skupiny tříd.

Kontrolery jsou určeny k vystavení REST (Representational State Transfer) služeb klientům a jsou označeny anotací `@RestController`. Každá metoda daného kontroleru představuje jeden endpoint vytvořeného REST API a musí mít anotace k určení typu dotazu (GET, POST, PUT, DELETE) a adresy, pomocí které lze endpoint zavolat.

Entitní třídy představují obvyčejné objekty v Javě, které reprezentují jednotlivé tabulky v databázi, a jejich instance reprezentují jeden řádek v databázové tabulce. Pro definici entitní třídy se používá anotace `@Entity`, která umožňuje mapování objektů na tabulky v databázi. Pro namapování atributů jednotlivých entitních tříd se používají anotace `@Column`, případně jsou použity anotace `@Converter` pro převod dat databáze na data ve formátu programovacího jazyka Java či další anotace.

Repository třídy jsou abstraktní třídy rozšiřující `JpaRepository` [17]. Tyto třídy poskytují základní funkce pro komunikaci s databází, jako je stránkování dat nebo serializace.

Servisní třídy obsahují business logiku aplikace a jsou označeny anotací `@Service`. Tyto třídy provádějí manipulace s databázovými daty a vracejí získaná data z databáze do kontrolérů, které je při zavolání jednotlivých endpointů posílají jako tělo odpovědi.

Security třídy zajišťují bezpečnost aplikace a obsahují veškeré důležité části pro implementaci JWT (Json Web Token), včetně filtrace, generování tokenů a konfigurace požadavků.

Utils třídy obsahují autentizační mechanismy a nachází se v balíčku `Utils`.

3.3.2 Použité technologie

Serverová část aplikace je psaná v programovacím jazyku Java za využití frameworku Spring boot. Jako nástroj pro sestavení byl vybrán nástroj Maven. Diskuze k volbě implementační platformy je uvedena v bakalářské práci Duc Thang Nguyena [11] v kapitole 4 Diskuze a volba implementační platformy.

3.4 Databáze

K ukládání dat slouží Oracle databáze vytvořená Ing. Michalem Valentou Ph.D. Na backendu se pro připojení k databázi používá konektor JDBC (Java Database Connectivity) [18]. Nejdříve bylo potřeba nainstalovat konkrétní JDBC konektory `ojdbc8` a `orai18n` a následně se připojit k databázi pomocí přípojovacího řetězce (connection string) [11].

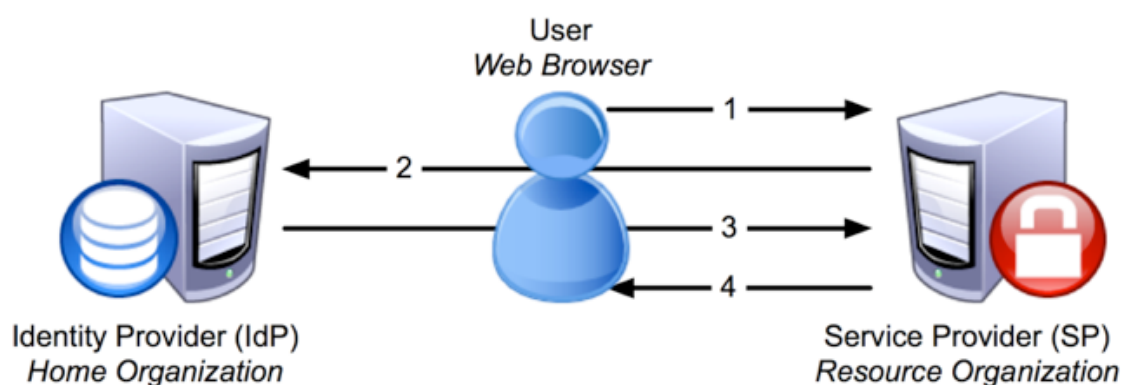
3.5 SSO

SSO je technika autentizace, která umožňuje uživatelům přihlásit se k různým aplikacím nebo systémům pomocí jediné sady přihlašovacích údajů, jako je uživatelské jméno a heslo. První přihlášení uživatele k jednomu ze systémů jim umožňuje přístup ke všem ostatním systémům bez potřeby opakovaného zadávání přihlašovacích údajů [19]. Tento způsob autentizace je výhodný jak pro uživatele aplikací, kterým stačí zapamatovat si pouze jedny přihlašovací údaje, tak pro vývojáře, kterým SSO usnadňuje práci s uchováváním uživatelských dat a údržbou aplikace [19].

Důležitou roli v procesu SSO přihlašování hrají především Poskytovatel identity IdP (Identity Provider), který je zodpovědný za autentifikaci uživatele a zaopatření informací o uživateli Poskytovateli služeb SP (Service Provider), a dále zmíněný Poskytovatel služeb SP, který je zodpovědný za ochranu online zdrojů a získávání informací od IdP. [20]

Na obrázku 3.1 převzatého ze stránek poskytovatele služeb Shibboleth [20] je vidět obecné schéma SSO přihlašování. Vysvětlení obrázku 3.1, přeloženo z [20]:

1. SP detekuje uživatele, který se snaží přistoupit k omezenému obsahu v rámci zdroje.
2. SP vygeneruje autentifikační požadavek, následně pošle žádost a přesměruje uživatele k jeho IdP.
3. IdP autentifikuje uživatele, následně pošle autentifikační odpověď a přesměruje uživatele zpátky k SP.
4. SP ověří odpověď od IdP a odešle požadavek zdroji, který vrátí původně požadovaný obsah.



■ **Obrázek 3.1** Schéma SSO přihlašování, převzato ze stránek poskytovatele služeb Shibboleth [20]

V rámci systémů ČVUT se používá SSO přihlašování pomocí poskytovatele služeb Shibboleth, který je jedním z nejrozšířenějších systémů pro správu identity a je používán zejména akademickými institucemi a komerčními organizacemi [21]. Je založený na protokolu SAML (Security Assertion Markup Language), což je standard, který umožňuje poskytovatelům identity, aby autentifikovali své uživatele a předali autentizační token požadované aplikaci pomocí poskytovatele služeb [22].

3.5.1 Obecné nastavení Shibboleth

Obecný návod na instalaci a konfiguraci Shibboleth jako poskytovatele služeb je dostupný na stránkách Federace identit eduid.cz [23] a konkrétní návod pro poskytovatele identity ČVUT, který je definován na url <https://idp2.civ.cvut.cz/idp>, se nachází na Wiki stránkách s návody ČVUT [24].

Postup je takový, že je potřeba nejdříve stáhnout a instalovat Shibboleth SP včetně modulů pro Apache HTTP server (v operačním systému Linux `apache2` a `libapache2-mod-shib`). Dále je zapotřebí nakonfigurovat soubor `/etc/shibboleth/shibboleth2.xml`, ve kterém se nachází nejdůležitější nastavení SP Shibboleth. V sekci `ApplicationDefaults` se nastavují údaje jako entity ID, metadata provider informace, nastavení inicializace session a další potřebné konfigurační volby. Dále je v souboru `/etc/shibboleth/shibboleth2.xml` sekce `Handler`, ve které lze nastavit použití souboru s metadaty a v sekci `MetadataProvider` lze nastavit metadata federace (IdP). V konkrétním návodu na konfiguraci Shibboleth pro IdP ČVUT [24] je také k dispozici soubor `attribute-map.xml` se všemi atributy poskytovanými ve federaci `cvutID`, který má být umístěn ve složce `/etc/shibboleth`. [23]

Návrh implementace SSO přihlašování administrátorského modulu je popsán v kapitole 5.6.

Cíle práce, požadavky

4.1 Technologický upgrade

Administrátorský modul Ankety byl vyvinut před čtyřmi roky a od té doby se významně posunuly technologie použité pro tvorbu jeho uživatelského rozhraní. Z toho důvodu bylo nezbytné před začátkem implementace nových funkcí aktualizovat technologie použité pro tvorbu uživatelského rozhraní neboli provést technologický upgrade. Úkolem bylo povýšení prostředí pro Javascript Nodejs na verzi 18 LTS (Long-term support).

4.2 Vyhodnocení ankety

V původní verzi administrátorského modulu vždy, když došlo na vyhodnocení anket, bylo nutné manuálně spustit proces v databázi, který vyhodnotil danou anketu, a poté bylo možné zobrazit výsledky. Z toho plynula potřeba vytvořit novou funkcionalitu vyhodnocení ankety přímo v modulu, která umožní spustit proces vyhodnocení kliknutím na tlačítko v detailu ankety a následně zobrazí, zda se podařilo anketu vyhodnotit.

4.3 Generování CSV exportů

Některé fakulty se snaží motivovat své studenty k vyplňování anketních lístků zejména u předmětových anket tím, že během vyplňování vylosují několik studentů, kteří již vyplnili nějaký anketní lístek dané ankety, a odmění je. Administrátorům pro zmíněné slosování chyběla možnost vygenerovat CSV export všech studentů, kteří vyplnili nějaký anketní lístek dané ankety. Další požadavek tedy byl vytvořit tlačítko, pomocí kterého si mohl administrátor stáhnout tento exportovaný CSV soubor. Dále bylo vhodné umožnit CSV export všech typů komentářů, který je dostupný po vyhodnocení ankety.

4.4 Skrývání nevhodných komentářů

Občas se stává, že se v textových komentářích objeví neslušné nebo hrubé výrazy, které by neměly být zveřejněny. V administrátorském modulu bylo možné komentáře prohlížet, ale pokud administrátor narazil na nevhodný komentář, musel o tom informovat správce Ankety, který daný komentář upravil manuálně v databázi. Z toho vznikl požadavek na možnost skrývání nevhodných komentářů neboli jejich cenzuru pomocí tlačítka umístěného vedle komentáře.

4.5 Úprava anketních otázek

Původně bylo v administrátorském modulu superadministrátorům umožněno vytvářet, upravovat a mazat anketní otázky a seznamy otázek. Fakultní administrátoři mohli otázky a seznamy otázek pouze vytvářet, což se ukázalo být poněkud omezující. Na druhou stranu umožněním mazání otázek a jejich seznamů mohly vznikat problémy v rámci anket v případě, že byly smazány otázky nebo seznamy, které byly použity v některé z anket. Z těchto důvodů bylo jedním z požadavků povolit úpravu otázek a seznamů otázek i fakultním administrátorům a zároveň zakázat superadministrátorům mazání otázek i jejich seznamů. Dále bylo kvůli snazšímu vytváření anketních otázek a seznamů otázek vhodné umožnit všem administrátorům otázky i jejich seznamy duplikovat.

4.6 SSO přihlašování

V průběhu technologického upgrade vznikl požadavek na implementaci SSO přihlašování, které velmi usnadní přihlašování uživatelů do aplikace a uchovávání s nimi spojených dat v databázi. Jak již bylo zmíněno v podkapitole 3.5, v rámci ČVUT se používá přihlašování pomocí poskytovatele služeb Shibboleth, který je použit i pro přihlašování do Vyplňovací části Ankety [25]. Z toho důvodu bylo pro přihlašování do administrátorského modulu rozhodnuto také použít Shibboleth a byl navrhnout způsob implementace SSO přihlašování.

4.7 Dokumentace

Velká část kódu frontendu i backendu administrátorského modulu postrádala komentáře pro lepší orientaci v jednotlivých komponentách a modulech. Bylo tedy nezbytné porozumět všem částem programu a zdokumentovat je takovým způsobem, že vznikla ucelená dokumentace obou částí.

4.8 Návody

K původnímu administrátorskému modulu byly vytvořeny návody pro fakultní administrátory, které jim usnadňovaly práci s aplikací. K novým funkcionalitám byl třeba vytvořit návod a u upravených funkcionalit byl třeba návod aktualizovat.

Návrh a implementace

5.1 Technologický upgrade

Technologický upgrade byl prováděn v části uživatelského rozhraní neboli frontendu administrátorského modulu. Jak již bylo zmíněno v podkapitole 3.2.2, pro vývoj této části byl vybrán jazyk Javascript za použití frameworků Nuxt.js a Vuetify. Níže jsou popsány hlavní kroky, které bylo potřeba při technologickém upgrade provést.

5.1.1 Základní nastavení

Jako první bylo nutné nastavit v projektu použití správné verze Nodejs, konkrétně 18.19.0. Dále byl proveden upgrade npm balíčků. Pomocí příkazu `npm outdated` bylo možné zobrazit balíčky, jejichž verzi bylo zapotřebí povýšit, a následně byly verze upraveny v souboru `package.json` na požadované verze, případně byly vyřešeny konflikty způsobené novými verzemi balíčků.

5.1.2 Nuxt.js

Původní implementace administrátorského modulu používala verzi frameworku Nuxt 2. Od doby vzniku byl vyvinut Nuxt 3 kompatibilní s Node 18, na který bylo požadováno provést upgrade. Dále bude 30. června 2024 verzi Nuxt 2 končit doba údržby EOL (End Of Life) a pro rozvíjející aplikaci, jako je Anketa, je vhodné používat aktuální technologie [26]. Bylo tedy nezbytné udělat upgrade na verzi Nuxt 3. Jak je uvedeno i v dokumentaci frameworku Nuxt [27], Nuxt 3 je víceméně kompletní přepsání Nuxt 2. Při provádění upgrade tedy došlo k mnoha podstatným úpravám.

Nejprve byly provedeny změny ve struktuře projektového adresáře, v základní konfiguraci, která se nachází v souboru `nuxt.config.js` a v názvech dynamicky generovaných souborů, aby fungovalo správně vytváření jimi generovaných stránek (např. stránky s detaily jednotlivých anket, jejichž url adresa obsahuje vždy kód semestru a klíč dané ankety).

Dále byla upravena konfigurace použitých Nuxt pluginů a modulů, přičemž bylo nutné některé pluginy či moduly nahradit jinou variantou kompatibilní s Nuxt 3. Nuxt pluginy se nachází ve složce `plugins/`, mají definovanou strukturu a v administrátorském modulu slouží pro začlenění některých knihoven a komponent třetích stran do programu. V kódu 5.1 je ukázka pluginu `vue-markdown` použitého v původním administrátorském modulu a kód 5.2 ukazuje plugin `vue3-markdown`, který nahradil plugin `vue-markdown`. Použité moduly také slouží k začlenění knihoven třetích stran, ale na rozdíl od pluginů jsou definovány v souboru `nuxt.config.js` a při spuštění ve vývojovém módu nebo při sestavení programu jsou začleněny dříve než pluginy [28].

```
import Vue from 'vue'
import VueMarkdown from 'vue-markdown'

Vue.component('vue-markdown', VueMarkdown)
```

■ **Výpis kódu 5.1** Plugin vue-markdown

```
import { VMarkdownView } from 'vue3-markdown'
import { defineNuxtPlugin } from 'nuxt/app'

export default defineNuxtPlugin((nuxtApp) => {
  nuxtApp.vueApp.component('VMarkdownView', VMarkdownView)
})
```

■ **Výpis kódu 5.2** Plugin vue3-markdown

Níže jsou popsány pluginy a moduly, které bylo potřeba nahradit nebo odstranit.

5.1.2.1 Axios

Axios je HTTP klient založený na typu odpovědi zvané promise [29] vytvořený pro Nodejs a webový prohlížeč [30]. Pomocí Axios byla původně v administrátorském modulu prováděna veškerá volání pro získání a ukládání dat z backend REST API. V kódu 5.3 je ukázka volání typu GET pomocí Axios. Axios byl začleněn pomocí modulu @nuxt/axios [31] a pomocí pluginu axios. Uvedený modul je však stavěný pro Nuxt 2 a jak je uvedeno v dokumentaci k přechodu z Nuxt 2 na Nuxt 3 [32], bylo nutné použít globálně dostupné fetch API [33] nebo ofetch API [34]. Jako výměna za Axios byl vybrán klasický fetch, zejména kvůli tomu, že se jedná o API zabudované do Nodejs. Není tedy potřeba používat knihovny třetích stran a proto je také rychlejší[35].

```
this.$axios.$get('/news').then(responseArray => {
  this.news = responseArray
})
```

■ **Výpis kódu 5.3** Get request pomocí Axios

```
await fetch(useRuntimeConfig().public.apiBase + '/news', {
  method: 'GET',
  headers: {
    "Content-Type": "application/json",
    "Authorization": "Bearer " + useUserStore().token
  },
}).then(async res => {
  const resObj = await res.json()
  this.news = resObj
})
```

■ **Výpis kódu 5.4** Get request pomocí fetch API

5.1.2.2 NuxtAuth

Pro autentizaci se původně používal modul @nuxtjs/auth určený pro Nuxt 2 [36]. Pro snadnější údržbu kódu a také kvůli přihlašování pomocí Shibboleth v produkčním prostředí nebyl modul @nuxtjs/auth nahrazen jiným modulem, ale bylo použito fetch API pro volání endpointů z backend REST API spolu s využitím úložiště Pinia (uvedeno v podkapitole 5.1.2.3) pro uchování uživatelských dat na frontendové části.

5.1.2.3 Vuex

Úložiště Vuex je knihovna pro řízení stavu (state management library) určené pro Vue.js aplikace, ale jak je uvedené na stránkách Vuex [37], oficiální knihovnou pro řízení stavu je nyní Pinia, která se označuje také jako Vuex 5. Původně bylo používáno pouze zřídka, jedno úložiště sloužilo modulu @nuxtjs/auth 5.1.2.2 k automatickému uložení uživatelských dat a další bylo použito pro ukládání nahlášených chyb. Při odstranění knihovny @nuxtjs/auth bylo třeba použít kompatibilní knihovnu pro ukládání uživatelských dat získaných z backend REST API. Výhoda Pinia úložiště je také to, že ho lze použít přímo v kódu pomocí importované funkce vytvářející úložiště. Knihovna Pinia byla začleněna do programu jako modul @pinia/nuxt. Na obrázku 5.4 je vidět použití úložiště s uživatelskými daty pomocí funkce useUserStore() a jeho atributu token.

5.1.2.4 Vuedraggable

Vuedraggable je komponenta pro Vue 2 zajišťující schopnost přesouvání položek seznamů (drag-and-drop) [38]. Této funkcionalitě se v administrátorském modulu využívá při vytváření, úpravě a duplikaci seznamů otázek. Po přidání celkových i učitelských otázek je možné otázky přesunout stylem drag-and-drop, aby byly seřazeny v požadovaném pořadí. Jelikož pořadí otázek hraje důležitou roli při vyplňování anket, avšak komponenta vuedraggable není kompatibilní s Vue 3, bylo nutné místo ní použít komponentu VueDraggableNext [39], která splňuje požadavek kompatibility s Vue 3. Narozdíl od předchozí komponenty, která byla do aplikace včleněna jako plugin, je nově použitá komponenta v kódu importována přímo pomocí import statement `import VueDraggableNext from 'vue-draggable-next'`.

5.1.2.5 Vue-markdown

V administrátorském modulu se na několika místech zobrazuje text ve formátu markdown. Jedná se o větší bloky textu, jako jsou například aktuality nebo náhled nástěnek pro Vyplňovací část. Pro zobrazení textu ve formátu markdown byla původně použita knihovna vue-markdown, ale kvůli kompatibilitě s Vue 3 byla tato knihovna nahrazena knihovnou vue3-markdown [40]. Do projektu byla začleněna pomocí pluginu jako komponenta a v kódu byla použita podobně jako jiné komponenty. Rozdíl mezi pluginem vue-markdown a vue3-markdown je uveden v kódu 5.1 a 5.2. Kód 5.5 ukazuje zmíněnou komponentu zobrazující text `itemToReturn.text` ve formátu markdown.

```
<v-card-text>
  <VMarkdownView :content="itemToReturn.text" />
</v-card-text>
```

■ **Výpis kódu 5.5** Zobrazení textu `itemToReturn.text` ve formátu markdown

5.1.2.6 Moment

Pro formátování datumů byly dříve použity filtry balíčku vue-moment, které ke svému fungování potřebují knihovnu MomentJS. Jelikož MomentJS přechází do stavu pouhé údržby [41] zejména z toho důvodu, že objekty knihovny MomentJS jsou měnitelné (mutable), což způsobuje, že jejich použití činí kód náchylnější k chybám, komplikovanější pro porozumění a méně uzpůsobený pro změny [42]. Další velkou nevýhodou je velikost knihovny, která zvětšuje rozsah výsledného projektu [41].

Místo balíčku vue-moment a knihovny MomentJS byla pro formátování datumů v programu vybrána a použita knihovna date-fns. Tato knihovna poskytuje mnoho výhod jako je modularita, díky které je v projektu možné použít pouze ty funkcionality, které jsou opravdu potřeba, použít

neměnných (immutable) objektů, které jsou vráceny jako výsledek funkcí knihovny, a mnoho dalších [43]. Kód 5.6 ukazuje použití funkce `format` z knihovny `date-fns` pro zobrazení datumů a například na datum `created` objektu `item` je aplikována následujícím způsobem:

```
<span>
  {{item.created ? $dateFormat(item.created, 'MM/dd/yyyy HH:mm:ss') : ''}}
</span>
```

■ **Výpis kódu 5.6** Ukázka použití funkce `format` knihovny `date-fns`

5.1.2.7 Vue-toasted

`Vue-toasted` je plugin používaný pro zobrazování informačních hlášek, které informují uživatele aplikace například o úspěšném či neúspěšném provedení akce [44]. Tento plugin byl původně integrován do administrátorského modulu stejně jako ostatní pluginy a 5.7 ukazuje příklad volání funkce, která zobrazí informační hlášku s textem 'Chyba byla uspesne nahlasena!' a ikonkou s názvem `done`, která se po pěti vteřinách skryje.

```
this.$toasted.success('Chyba byla uspesne nahlasena!', {
  icon: "done",
  duration: 5000});
```

■ **Výpis kódu 5.7** Ukázka použití knihovny `vue-toasted` pro zobrazení informační hlášky

Jelikož `vue-toasted` je plugin pro `Vue 2` [45], bylo zapotřebí tento plugin nahradit. Za účelem omezení používání knihoven a pluginů třetích stran, byl plugin `vue-toasted` nahrazen komponentou `v-snackbar` již používané UI knihovny `Vuetify`, která je použita v `Nuxt` komponentě `Snackbar` představující informační hlášku. 5.8 ukazuje HTML část kódu komponenty `Snackbar`.

```
<template>
  <v-snackbar
    v-model="show"
    :color="toast.color"
    :timeout="timeout"
    bottom
    right
    :multi-line="toast.multiLine"
    :vertical="toast.vertical"
    :absolute="toast.absolute"
    :outlined="toast.outlined"
    class="custom-snackbar"
  >
  <div class="snackbar-content">
    {{ toast.message }}
    <v-icon @click="closeSnackbar">mdi-close-box</v-icon>
  </div>
</v-snackbar>
</template>

<script> ... </script>

<style scoped>
.custom-snackbar {
  max-width: auto;
  min-width: 0;
  padding: 0;
}
```

```
.snackbar-content {
  display: flex;
  align-items: center;
  padding: 6px 16px;
  font-size: 1.25rem;
}
</style>
```

■ Výpis kódu 5.8 Snackbar komponenta

Dále bylo vytvořeno Pinia úložiště `snackbar` obsahující data, která ovládají viditelnost, barvu a text hlášky, a také funkci pro zobrazení a skrytí hlášky. Tyto data a funkce jsou dostupné po importování funkce `useSnackbarStore` vytvářející úložiště z kterékoli komponenty aplikace a zavoláním zmíněných funkcí je ovládána viditelnost a obsah hlášky. 5.9 ukazuje kód úložiště `snackbar`.

```
import { defineStore } from 'pinia'

export const useSnackbarStore = defineStore({
  id: 'snackbarStore',
  state: () => {
    /** Snackbar data */
    return {
      message: '',
      color: '',
      show: false
    }
  },
  actions: {
    /**
     * Show a snackbar with a message
     * @param {String} message - message to display
     * @param {String} color - color of the snackbar
     */
    showSnackbar(message, color) {
      this.message = message
      this.color = color
      this.show = true
    },

    /** Close the snackbar */
    closeSnackbar() {
      this.message = ''
      this.color = ''
      this.show = false
    },
  },
})
```

■ Výpis kódu 5.9 Snackbar úložiště

Výhodou tohoto řešení je eliminace závislosti na další knihovně či pluginu mimo ty, které jsou již v projektu používané. Další výhodou je možnost úpravy vzhledu hlášky podle vlastní představy. Oproti tomu současné řešení neumožňuje zobrazit více informačních hlášek najednou, ale za pomoci několika úprav nebude problém v budoucnu komponentu a úložiště upravit tak, aby bylo umožněné zobrazit více informačních hlášek najednou.

5.1.3 Vuetify

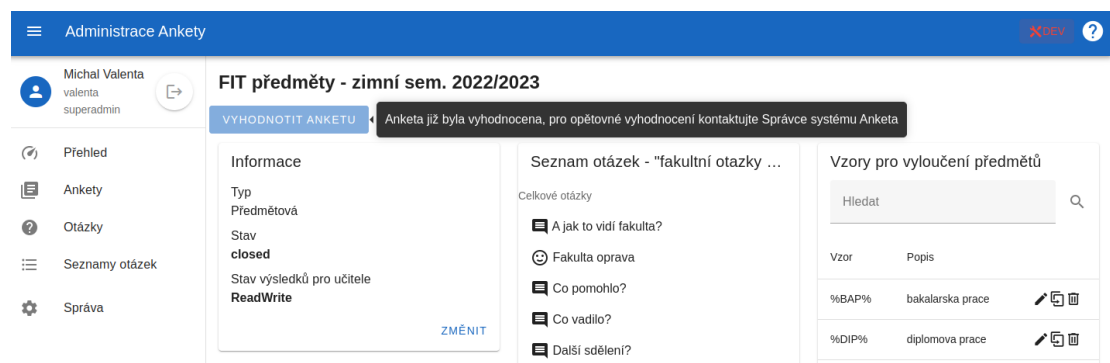
Jak již bylo několikrát zmíněno, Vuetify je v projektu použit jako UI framework. Jelikož Vuetify 3 je kompatibilní s Vue 3 [46] a Nuxt.js je založený na Vue.js [47], bylo nutné provést upgrade také frameworku Vuetify. Upgrade zahrnoval zejména úpravu jeho konfigurace v projektu jako plugin, podobně jako ostatní pluginy zmíněné v podkapitole 5.1.2, a zároveň jeho začlenění do projektu jako modul. Dále upgrade zahrnoval úpravu, odstranění nebo nahrazení některých komponent a jejich argumentů (props), které již nejsou kompatibilní s Vuetify 3 [48].

5.2 Vyhodnocení ankety

Základem pro vyhodnocení ankety jsou dvě funkce v databázovém balíčku `eval_sem` vytvořené Ing. Michalem Valentou, PhD. Jedná se o funkce `course_eval_create` pro předmětové ankety a `faculty_eval_create` pro nepředmětové ankety, které vyhodnotí výsledky a vrátí jejich souhrn spolu se statusem úspěšnosti vyhodnocení. Vracená data jsou typu textový řetězec a pro snadný rozklad řetězce na backendu jsou data ve formátu JSON (Javascript Object Notation).

Na backendu byl pro vyhodnocení ankety vytvořen endpoint ve třídě `SurveyController`. Funkce pod tímto endpointem používá servisní třídu funkci `executeSurveyEvaluation` třídy `SurveyServis`, která volá jednu ze zmíněných funkcí z databáze podle toho, jestli se jedná o předmětovou nebo nepředmětovou anketu.

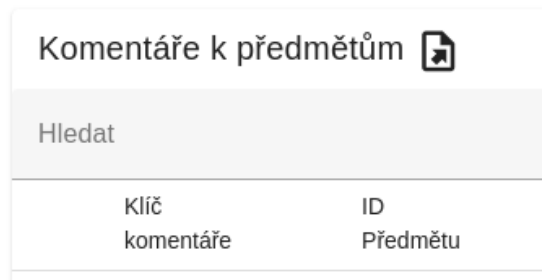
Na frontendu bylo pro vyhodnocení ankety vytvořeno tlačítko umístěné na stránce s detailem ankety, jehož stisknutím se volá endpoint vytvořený na backendu. Po vyhodnocení jsou získaná data ve formátu JSON zobrazena v dialogovém okně. Během vyhodnocování není možné na tlačítko kliknout a tlačítko je nepřístupné (disabled) také v případě, že anketa již byla vyhodnocena nebo pokud doposud nebyla uzavřena. Vyhodnocení je tedy možné pouze pokud je anketa uzavřena a nebyla vyhodnocena. V případě, že vyhodnocení není možné, je po najetí myši na tlačítko administrátorům zobrazena hláška s důvodem a informací, aby administrátor případně kontaktoval správce systému Anketa. Na obrázku 5.1 je ukázka disabled tlačítka pro vyhodnocení ankety, která již byla vyhodnocena.



Obrázek 5.1 Disabled tlačítko pro vyhodnocení ankety, která již byla vyhodnocena

5.3 Generování CSV exportů

Implementace generování CSV exportů probíhala pouze na frontendu administrátorského modulu. Export je vygenerován pomocí tlačítka s ikonkou exportu vedle názvu dané tabulky s komentáři nebo studenty (znázorněno na obrázku 5.2).



■ **Obrázek 5.2** Tlačítko s ikonkou pro CSV export komentářů k předmětům

Tato funkcionality je zajištěna komponentou `csvExportButton`, která je v kódu vložena vedle názvu tabulky, a jako argumenty (props) jsou do ní vložena data k exportování. Obsahuje metody pro převod dat do CSV formátu a následné stažení souboru. Kód 5.10 ukazuje zmíněné metody. Metoda `exportCsv` je volána stisknutím tlačítka a používá metodu `convertToCsv` k převodu dat do CSV formátu a metodu `downloadCsv` ke stažení souboru.

```
exportCsv(data) {
  const csvData = this.convertToCsv(data);
  this.downloadCsv(csvData);
},

convertToCsv(data) {
  if (data.length == 0) {
    return ''
  }
  const header = Object.keys(data[0]).join(',');
  const rows = data.map(item =>
    Object.values(item).map(value => value !== null ?
      String(value).replace(/"/g, '"') : '').join(","));

  return `${header}\n"${rows.join("\n")}`;
},

downloadCsv(csvData) {
  const blob = new Blob(['\uFEFF' + csvData],
    { type: 'text/csv;charset=utf-8;' });
  const url = URL.createObjectURL(blob);
  const link = document.createElement('a');
  link.href = url;
  link.setAttribute('download', 'export.csv');
  document.body.appendChild(link);
  link.click();
  document.body.removeChild(link);
  URL.revokeObjectURL(url);
}
```

■ **Výpis kódu 5.10** Funkce komponenty `CsvExportButton`

5.4 Skrývání nevhodných komentářů

Pro umožnění skrývání nevhodných komentářů (cenzury) bylo potřeba nejdříve implementovat na backendu příslušné endpointy, které zajistí uložení informace o skrytí/odkrytí komentáře

v databázi. Tyto endpointy se nachází ve třídě `CommentsController`, která obsahuje endpointy pro získávání a ukládání dat týkajících se komentářů. Tato třída používá pomocnou servisní třídu `CommentService`, která obsahuje funkce vykonávající zásahy do databáze. V případě skrývání či odkrývání komentářů bylo potřeba změnit v tabulce s komentáři hodnotu sloupce `IS_CENSORED` a v dalších dvou sloupcích nastavit čas a autora změny.

Na frontendu se komentáře zobrazují v detailu jednotlivých anket. Konkrétně se jedná o komponenty zobrazující tabulky s komentáři a obsahující funkce pro manipulaci s nimi. Pro cenzuru komentářů bylo do každého řádku tabulky přidáno tlačítko s ikonkou zvuku, pomocí kterého se komentář buď skryje, text komentáře zašedne a ikonka se zobrazí přeškrtnutá, nebo se komentář odkryje a text i ikonka tlačítka se vrátí do původní podoby. Na obrázku 5.3 je vidět tabulka s některými skrytými a některými odkrytými komentáři. Delší komentáře je možné rozbalit pomocí šipky na začátku řádku. Příklad rozbaleného komentáře je také na obrázku 5.3, jedná se o komentář s `courseEvaluationKey` 18750.

courseEvaluationKey	Otázka	Text	Cenzura
18730	Co pomohlo?	Nejvíce mi pomohli jiné studenty.	🔊
18736	Co pomohlo?	Kavárny jsou velké p...	🔊
18750	Co pomohlo?	Prváký discord. Na to že to začalo jako parta asi 10 lidí, co se potkali ve vlaku na seznamovák, tak se z toho stala skupina, která právě má přes 300 členů. Nejlepší na tom je, že se člověk nemusí bát zeptat ale úplně na cokoliv, ať už to je sebevětší blbost.	🔊
18754	Co pomohlo?	Prezenční studium. T...	🔊
18760	Co pomohlo?	Seznamovák byl dobrej	🔊
18761	Co pomohlo?	Discord	🔊

■ **Obrázek 5.3** Ukázka tabulky s komentáři včetně ikonky pro export CSV souboru

5.5 Úprava anketních otázek

Úprava anketních otázek a jejich seznamů probíhala tak, že byla v uživatelském rozhraní příslušná tlačítka zobrazena kromě superadministrátorů také fakultním administrátorům. Jak je zmíněno i v podkapitole 6.2.3, byla na základě zpětné vazby z testování upravena také backendová část této funkcionality. Dále byla přidána možnost duplikace otázek i seznamů otázek. Duplikace probíhala stejně jako vytváření nových otázek, akorát místo prázdných dat se použila data duplikované otázky a na konec názvu se přidal text "- kopie". Mazání otázek a jejich seznamů bylo provedeno jejich odstraněním v uživatelském rozhraní. Na backendu byly funkce ponechány pro jejich případné opětovné začlenění do aplikace.

Kromě úpravy práv na manipulaci s otázkami a seznamy otázek byla některá pole dialogového okna pro přidávání, úpravu a duplikaci otázek změněna na nepovinná. Toto bylo provedeno v souladu s požadavky příslušné tabulky v databázi.

5.6 SSO přihlašování

SSO přihlašování bude využito pouze na produkčním prostředí. V testovacím a staging prostředí se kvůli snadnějšímu testování aplikace používá přihlašování pouze pomocí uživatelského jména. Pro implementaci SSO přihlašování na produkčním prostředí bylo v části s uživatelským rozhraním vytvořeno tlačítko, které uživatele přesměruje na přihlašovací stránku ČVUT. Součástí odkazu je také specifikace endpointu, který má po zadání uživatelských údajů Shibboleth využít pro předání uživatelských údajů a získání JWT pro autentifikaci. Jelikož Vyplňovací část ankety již používá přihlašování přes Shibboleth, backend obou částí aplikace je v produkčním prostředí nahráván do stejného adresáře a adresář, kam je v produkčním prostředí nahrán frontend, se liší pouze poslední složkou /admin, do které je nahrán frontend administrátorského modulu, na základě poznatků uvedených v kapitole 3.5 je předpokládáno, že bude stačit upravit soubor shibboleth2.xml v adresáři /etc/shibboleth/, aby bylo SSO přihlašování funkční pro obě aplikace.

Z časových důvodů nebyla v době odevzdání práce implementace SSO přihlašování otestována, budou proto udevedeny dvě hypotézy popisující možný postup úpravy souboru /etc/shibboleth/shibboleth2.xml.

5.6.0.1 Hypotéza 1.

První hypotéza předpokládá, že ve zmíněném souboru /etc/shibboleth/shibboleth2.xml bude stačit přidat novou sekci <ApplicationDefaults>, ve které bude specifikováno entity ID, metadata provider informace, nastavení inicializace session a další potřebné konfigurační možnosti pro administrátorský modul. Všechny atributy budou velmi podobné těm pro Vyplňovací část ankety, protože obě aplikace jsou provozovány pod stejnou doménou a používají stejného poskytovatele identity.

5.6.0.2 Hypotéza 2.

Druhá hypotéza předpokládá, že ve zmíněném souboru /etc/shibboleth/shibboleth2.xml bude stačit přidat nebo upravit sekci RequestMapper [49] tak, aby se správně mapovaly všechny potřebné části url [50]. V kódu 5.11 je příklad sekce RequestMapper, která mapuje url administrátorského modulu.

```
<RequestMapper type="RequestMapper" normalize="true" lowercase="true">
  <RequestMap applicationId="all-applications">
    <Host name="anketa.is.cvut.cz" scheme="https" port="443" />
    <Path name="anketa" /> <!-- velka-anketa -->
    <Path name="html" /> <!-- velka-anketa -->
    <Path name="anketa" /> <!-- velka-anketa -->
    <Path name="admin" /> <!-- admin-modul -->
  </RequestMap>
</RequestMapper>
```

■ **Výpis kódu 5.11** RequestMapper v souboru shibboleth2.xml

5.7 Úprava uživatelů

Jak již bylo zmíněno, v produkčním prostředí je plánováno přihlašování přes Shibboleth, což přineslo také zrušení nutnosti uchovávat hesla uživatelů v databázi produkčního prostředí. Pro usnadnění testování byla však hesla zrušena i ve vývojovém a staging prostředí. Bylo tedy zapotřebí odstranit heslo i z dialogového okna na úpravu uživatelů, které je dostupné pro superadministrátory, a byl zrušen typ uživatele, který dříve mohl být buď interní, nebo SSO, avšak typ SSO

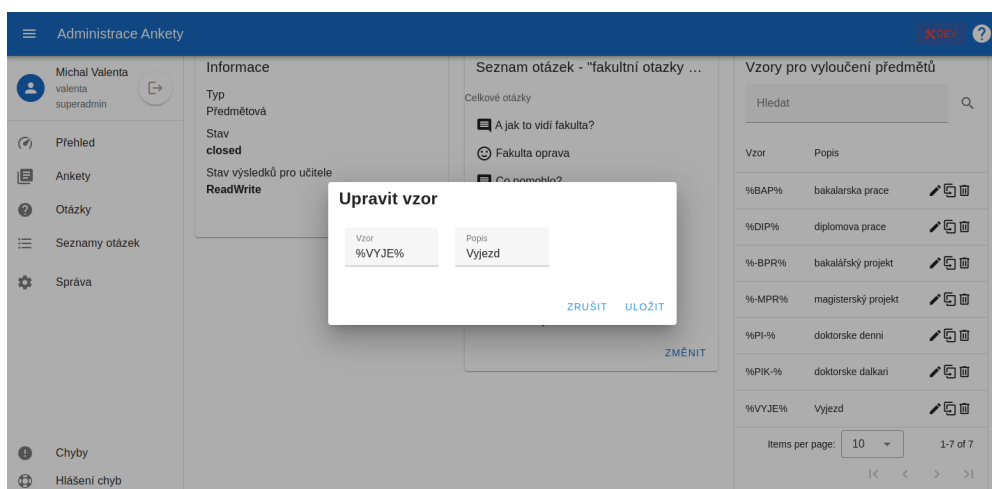
nebyl používán. Nyní je tedy možné uživatele přidávat, upravovat i mazat. Ve vývojovém a staging prostředí je umožněno přihlášení všem administrátorům, což přináší výhodu při testování aplikace, ale v produkčním prostředí bude umožněno přihlašování pouze administrátorům, kteří jsou zaregistrováni jako učitelé či pracovníci ČVUT a mají již přihlašovací údaje pro systém Shibboleth.

5.8 Úprava přidávání a odebírání učitelů a předmětů

Při práci na nových funkcionalitách a provádění úprav na stránce detailu jednotlivých anket bylo zjištěno, že původní implementace přidávání a odebírání předmětů a učitelů v předmětových anketách bylo z uživatelského hlediska vyřešeno komplikovaným způsobem. Detail anket obsahuje mimo jiné tabulku se zahrnutými předměty v dané anketě a tabulku s učiteli zahrnutými v anketě (učitelé, kteří učí nějaký předmět v anketě). Administrátoři potřebovali mít možnost přidat a odebrat předmět v anketě a přidat a odebrat učitele na předmět. Tyto funkce byly původně zajištěny na stránce se dvěma seznamy (zahrnuté a dostupné předměty), mezi kterými šlo předměty spolu s jejich vyučujícími přesouvat. Na tuto stránku se administrátor dostal přes tlačítko ZMĚNIT ve spodní části tabulek se zahrnutými předměty a učiteli. Všechny zmíněné funkce byly přesunuty do tabulky se zahrnutými předměty. Tlačítko ZMĚNIT bylo nahrazeno tlačítkem PŘIDAT, jehož stisknutím se otevře okno s výběrem předmětů, které je možné do anket přidat. Mazání předmětů probíhá pro každý předmět zvlášť kliknutím na ikonku koše na konci řádku tabulky s daným předmětem. Vedle ikonky koše se nachází ikonka sloužící pro přidání učitele na daný předmět. Jejím stisknutím se zobrazí okno s výběrem učitelů dostupných k přidání na předmět. V okně je také tabulka s již přidávanými učiteli, které je možné z předmětu odebrat.

5.9 Úprava vzorů

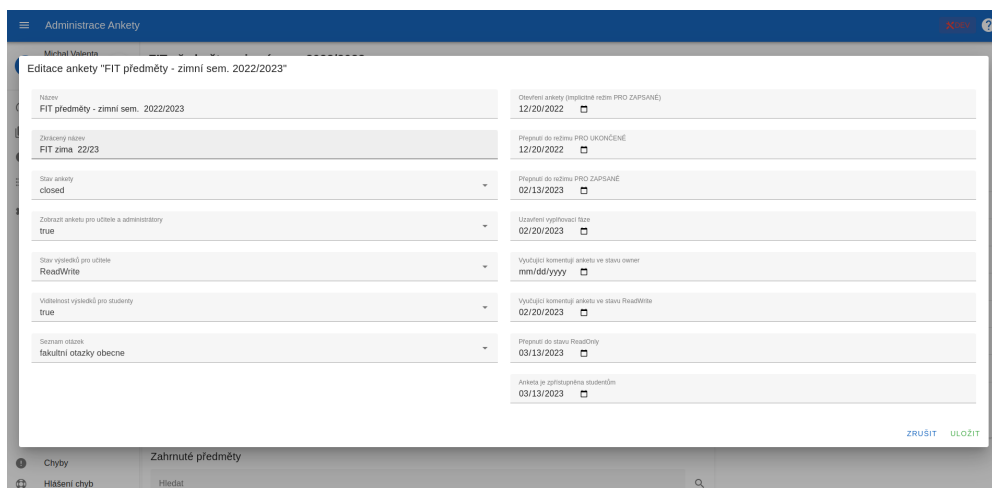
Podobným způsobem, jakým bylo upraveno přidávání a odebírání učitelů na předmět, byla upravena úprava vzorů pro vyloučení předmětů. Ve spodní části tabulky s vyloučenými vzory bylo nahrazeno tlačítko ZMĚNIT, které vedlo na novou stránku určenou k úpravě vzorů, za tlačítko PŘIDAT, pomocí kterého se otevře okno pro přidání nového vzoru. Úprava a mazání vzorů se provádí opět jednotlivě pro každý vzor pomocí ikonky na konci řádku s daným vzorem. Mimo zmíněné funkce byla také přidána možnost duplikace vzoru. Na obrázku 5.4 je vidět okno pro úpravu vzoru Vyjezd.



■ Obrázek 5.4 Okno pro úpravu vzoru Vyjezd

5.10 Úprava základních informací anket

Úprava základních informací anket, jako jsou název, celkový stav a stav výsledků pro učitele, použitý seznam otázek, viditelnost výsledků a data přechodů mezi různými fázemi, probíhala původně na samostatné stránce stejně jako úprava předmětů, učitelů a vzorů. Tyto funkce byly stejným způsobem přesunuty do dialogového okna na obrázku 5.5 aktivovaného tlačítkem ZMĚNIT ve spodní části karty se základními informacemi a karty se seznamem otázek použitým v anketě, které se nachází na stránce s detailem každé anketu.



■ Obrázek 5.5 Okno pro úpravu základních informací anket

5.11 Dokumentace

5.11.1 Frontend

Původně byl pro generování dokumentace frontendu použit nástroj JSDoc spolu s pluginem jsdoc-vuejs. JSDoc je generátor dokumentací určený pro jazyk Javascript. Slouží pro dokumentaci javascriptových aplikací a knihoven a předpokládá, že program se skládá z modulů, jmenných prostorů (namespace), tříd, metod, parametrů a dalších [51]. Komentáře JSDoc mají definovanou strukturu, podle které by měl vývojář kód komentovat. Zpravidla se komentáře píší nad kód, který má být zdokumentovaný a komentáře musí být ohraničeny znaky `/** */`. Pomocí speciálních anotací začínajících znakem `@` je možné rozlišit mezi třídami, metodami, parametry a dalším. Plugin jsdoc-vuejs je plugin umožňující dokumentaci Vue metod, dat, computed dat, závislostí (props) a dalšího z Vue souborů taktéž pomocí speciálních anotací [52]. Komentáře s jsdoc-vuejs anotacemi se narozdíl od klasických JSDoc komentářů píší na začátek části Vue souboru, která definuje zmíněná data, závislosti a metody. V kódu 5.12 je vidět příklad JSDoc komentáře za využití jsdoc-vuejs anotací, jelikož se jedná o .vue soubor.

```
/**
 * @description Modal window for adding/editing news
 * @module components/AddNewsModal
 * @vue-prop {object} itemToDisplay - item to display in form
 * @vue-prop {boolean} propDialog - dialog visibility
 * @vue-data {object} itemToReturn - copy of itemToDisplay for editing
 * @vue-data {boolean} dialog - dialog visibility taken from prop
 * @vue-data {array} titleValidations - validation rules for news title
```



```

* @vue-computed {string} cardTitle - dynamic card title
* @vue-watch {boolean} dialog - watch for dialog for validation reset
* @vue-methods save - emit save event with edited news
* @vue-methods close - emit close event
*/

```

■ Výpis kódu 5.12 JSDoc komentář s JSDoc a jsdoc-vuejs anotacemi

Konfigurace nástroje JSDoc je uvedena v souboru `jsdoc.config.json`. Je zde uvedeno, že má být použit plugin `jsdoc-vuejs`, jsou specifikovány typy souborů, ve kterých se můžou nacházet dokumentační komentáře, název dokumentace a další. Dokumentace je vygenerována ve formátu HTML pomocí příkazu `npm jsdoc run` zpravidla do adresáře `/out`, ale je možné adresář změnit pomocí přepínače `-d` při volání příkazu pro vygenerování [53]. Na obrázku 5.6 je ukázka dokumentace komponenty `AddNewsModal`, která slouží pro přidání nové aktuality.

Frontend of CTU Survey Admin Panel Modules - Global -

Module: components/AddNewsModal

Modal window for adding/editing news

Props

Name	Type	Default value	Required?	Description
itemToDisplay	object	-	Yes	item to display in form
propDialog	boolean	-	Yes	dialog visibility

Data

Name	Type	Default value	Description
itemToReturn	object	-	copy of itemToDisplay for editing
dialog	boolean	-	dialog visibility taken from prop
titleValidations	array	-	validation rules for news title
contentValidations	array	-	validation rules for news content

Computed

Name	Type	Description
cardTitle	string	dynamic card title

Members

<static> watch
Watch for dialog data and validation reset

<inner> contentValidations
Validation rules for news content

■ Obrázek 5.6 Ukázka dokumentace komponenty AddNewsModal

5.11.2 Backend

Pro generování dokumentace backendu byl použit nástroj Javadoc, který generuje dokumentaci vytvořeného API ve formátu HTML na základě komentářů se speciálními anotacemi [54] podobně jako nástroj JSDoc ve frontendové části. Také anotace nástroje Javadoc jsou velmi podobné JSDoc anotacím, nástroje se liší zejména programovacím jazykem, pro který jsou určeny. Kód 5.13 ukazuje komentáře s anotacemi pro generování dokumentace pomocí nástroje Javadoc.

```

/**
 * Gets survey basic information
 * @param semesterCode semester code
 * @param surveyKey survey key
 * @return survey data
 */

```

■ Výpis kódu 5.13 Javadoc komentář s Javadoc anotacemi

Jelikož je backend projekt sestavený pomocí nástroje Maven, výsledná dokumentace je vygenerována příkazem `mvn javadoc:javadoc` do adresáře `/target/site/apidocs`. Na obrázku 5.7 je ukázka části dokumentace třídy `QuestionsController`.

OVERVIEW PACKAGE CLASS USE TREE INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD SEARCH:

Package `cz.cvut.adminBoard.controllers`

Class QuestionsController

`java.lang.Object`
`cz.cvut.adminBoard.controllers.BaseController`
`cz.cvut.adminBoard.controllers.QuestionsController`

`@RestController`
`@RequestMapping("/questions")`
`public class QuestionsController`
`extends BaseController`

Controller for handling requests related to the questions.

Constructor Summary

Constructors	Description
<code>QuestionsController(QuestionService questionService)</code>	

Method Summary

Modifier and Type	Method	Description
<code>org.springframework.http.ResponseEntity<org.springframework</code>	<code>deleteQuestion(Long id)</code>	Deletes a specific question.
<code>org.springframework.http.ResponseEntity<QuestionGetD</code>	<code>getQuestion(Long id)</code>	Returns a specific question, currently unused because it's not implemented on the frontend.

■ **Obrázek 5.7** Ukázka dokumentace třídy `QuestionsController`

Pro dokumentaci jednotlivých endpointů vytvořených backendovým REST API je použit nástroj Swagger UI [55]. Tato dokumentace byla používána i v původním administrátorském modulu, ale chyběly v ní popisy jednotlivých endpointů. Tyto popisy byly do dokumentace doplněny pomocí anotací `@Operation(description = "Popis endpointu.")` v kódu tříd poskytujících jednotlivé endpointy. Anotace byly vždy uvedeny ihned nad anotací popisující typ endpointu. Výsledná dokumentace pro každé prostředí se nachází na adrese serveru s koncem adresy `/swagger-ui`.

5.12 Návody

Na adrese <http://anketa-vyvoj.cvut.cz/wiki/doc-navody-admini> je pro aplikaci Anкета ČVUT zřízena stránka s návody na ovládání administrátorského modulu pro fakultní administrátory. Jelikož návod pro administrátory neobsahoval některé části administrátorského modulu, byl tento návod kromě rozšíření o nové funkcionality a aktualizace upravených rozšíření tak, aby pokryl všechny části administrátorského rozhraní. Původně byly návody převážně ve formě videí ve formátu gif s jejich stručným popisem, ale na základě dotazu ohledně přepínání rychlosti videa, který byl vznesen jedním z administrátorů při uživatelském testování 6.2, byl formát změněn na mp4, díky kterému je možné video pozastavit či upravit rychlost.

Nasazení a testování

6.1 Backend unit testy

Jak již bylo zmíněno v podkapitole 2.2, pro backend administrátorského modulu byly vytvořeny unit testy nejdůležitějších servisních tříd. První třída, pro kterou byly napsány unit testy, je třída `BaseService`, ze které dědí všechny ostatní servisní třídy. Obsahuje metody pro převod dat v datových typech získaných z databáze na data ve formě datových typů jednotlivých DTO (data access object), které jsou posílány na frontend při volání jednotlivých endpointů. Testy této třídy pomáhají zajistit, že data budou poslána ve správném formátu, což je velmi důležité pro správné fungování programu. Testovací třída má název `BaseServiceTest`. Kód 6.1 ukazuje Příklad jedné z jejích testovacích metod, konkrétně metodu `objectToBooleanConversion` ověřující, že se správně převádí reprezentace datového typu Boolean v databázi (reprezentovaný pouze symbolem/řetězcem "Y" a "N") na datový typ Boolean v jazyce Java.

```
@Test
void objectToBooleanConversion() {
    assertTrue(baseService.objectToBooleanConversion("Y"));
    assertFalse(baseService.objectToBooleanConversion("N"));
    assertNull(baseService.objectToBooleanConversion(null));
}
```

■ **Výpis kódu 6.1** Metoda `objectToBooleanConversion` testovací třídy `BaseServiceTest`

Další testy byly napsány pro třídy `CommentService`, `CourseService` a `SurveyService`. Testovány jsou zejména funkce získávající data z databáze a převádějící je na data DTO posílaných jako odpověď volání endpointů na frontend. Kód 6.2 ukazuje metodu `getFacultyComments` testovací třídy `CommentServiceTest`, která ověřuje funkčnost metody získávající z databáze fakultní komentáře k dané anketě identifikované pomocí kódu semestru a klíče ankety. Na začátku testovací metody je vidět příprava dat pro testovací framework Mockito zmíněný v kapitole 2.2. Následuje konfigurace Mockita, aby při určených voláních vracel předem definovaná data. Nakonec je volána metoda `getFacultyComments` třídy `CommentService` a je ověřena správnost výsledků. Některé proměnné jako je instance třídy `EntityManager` `entityManager` nebo instance třídy `Query` `query` jsou v třídě `CommentServiceTest` definované ihned za deklarací třídy s anotací `@Mock` frameworku Mockito, a proto je možné je využít v každé testovací metodě třídy `CommentServiceTest` bez nutnosti jejich další deklarace.

```
@Test
void getFacultyComments() {
    String semesterCode = "B222";
```

```
Long surveyKey = 118L;
PageRequest pageable = PageRequest.of(0, 10);
String text = "test";
List<Object[]> mockData = new ArrayList<>();
for (int i = 0; i < 5; i++) {
    Object[] row = new Object[6];
    row[0] = "ANSWER_TEXT" + i;
    row[1] = 'N'; // vafta.IS_CENSORED
    row[2] = BigDecimal.valueOf(i);
    row[3] = BigDecimal.valueOf(i);
    row[4] = "VALUE_QUESTION_CZ" + i;
    row[5] = BigDecimal.valueOf(i);
    mockData.add(row);
}
when(entityManager.createNativeQuery(anyString())).thenReturn(query);
when(query.getResultList()).thenReturn(mockData);
when(query.getSingleResult()).thenReturn(BigDecimal.valueOf(5));

Page<facultyCommentGetDto> result = commentService.getFacultyComments(
    semesterCode, surveyKey, pageable, text);

assertNotNull(result);
assertEquals(5, result.getTotalElements());
assertEquals(facultyCommentGetDto.class, result.getContent().get(0)
    .getClass());
}
```

■ **Výpis kódu 6.2** Metoda `getFacultyComments` testovací třídy `CommentServiceTest`

6.2 Uživatelské testování

Pro uživatelské testování administrátorského modulu byly vytvořeny čtyři úlohy zahrnující nejběžnější úkony, které mohou fakultní administrátoři i superadministrátoři vykonávat. Úlohy byly sepsány ve formě Google dotazníku. Za každou úlohou následovala otázka, jestli test proběhl úspěšně, hodnocení složitosti úlohy na stupnici od 1 (lehká) do 5 (těžká) a nakonec prostor pro jakoukoli připomínku či postřeh zaznamenané během plnění úlohy. Na konci dotazníku byl také prostor pro připomínky, postřehy a náměty ke zlepšení administrátorského modulu jako celku. Prozatím bylo přes dotazník obdrženo šest odpovědí, což odpovídá více než polovině fakultních administrátorů, kterých je kolem deseti. Připomínky z odpovědí byly zpracovány a dále budou postupně uvedeny jednotlivé úlohy a popsány úpravy a nové funkcionality, které byly na základě zpětné vazby implementovány.

6.2.1 Úloha 1 – Úprava základních informací ankety

První úloha zahrnovala úpravu názvu a zkráceného názvu ankety. Dále obsahovala změnu stavu a viditelnosti výsledků pro vyučující a studenty předmětové i nepředmětové ankety. Tato úloha byla označena jako úspěšně splněna všemi odpověďmi a většina hodnocení složitosti byla stupně 1 až na jedno hodnocení stupně 2. K úloze také nebyly téměř žádné připomínky, byla však jedna připomínka ohledně zpětné vazby týkající se úspěšnosti provedení změn. Na základě toho byla přidána informační hláška s potvrzením úspěšného provedení.

6.2.2 Úloha 2 – Přidávání a odebírání předmětů a učitelů v anketě

Druhá úloha obsahovala přidání předmětu do předmětové ankety, odstranění alespoň jednoho učitele z nově přidaného předmětu a přidání na daný předmět sebe a nakonec smazání přidaného předmětu. Tato úloha byla označena jako úspěšně splněna pouze ve dvou odpovědích a čtyři ji hodnotili jako neúspěšně splněnou. Také složitost byla většinou hodnocena stupněm 3, dále dvě hodnocení byla stupně 2 a jedno hodnocení stupně 1. K této úloze bylo několik připomínek, že se nezobrazovaly žádné předměty k přidání. Bylo to způsobeno tím, že většina anket již měla všechny dostupné předměty zahrnuté. Na základě těchto připomínek bylo upraveno dialogové okno pro přidávání předmětů, aby byl v takovém případě administrátor informován o tom, že všechny předměty jsou v anketě již zahrnuty a nejsou tedy žádné předměty k přidání. Bylo také upraveno zadání této úlohy, aby počítalo s případným chyběním předmětů k přidání.

Jedna připomínka odhalila chybu při mazání předmětu, kdy byli smazáni také všichni učitelé dané ankety až na učitele v daném předmětu. Chyba byla uvedena v jednom z prvních vyplněných dotazníků a byla okamžitě opravena.

Několik výhrad bylo také ke komponentě `v-autocomplete` knihovny Vuetify, která slouží jako rozbalovacího menu s vyhledáváním, ale byla nastavena tak, že nebylo příliš uživatelsky přívětivé přidávat učitele na předmět. Tato komponenta byla upravena takovým způsobem, aby fungovala podobně jako jednodušeji ovladatelné rozbalovací menu s vyhledáváním v dialogovém okně pro přidávání, úpravu a duplikaci seznamu otázek, kde zmíněná komponenta slouží k přidávání otázek do seznamu celkových i učitelových otázek.

6.2.3 Úloha 3 – Práce s otázkami a sadami otázek

Třetí úloha obsahovala přidání nové otázky, duplikaci jednoho seznamu otázek a přidání vytvořené otázky do duplikovaného seznamu. Čtyři odpovědi označily úlohu jako neúspěšně splněnou a dvěma byla označena jako úspěšně splněna. Hodnocení složitosti bylo nejvíce různorodé ze všech testovacích úloh, stupně 5 a 3 obdržely po dvou hodnoceních, stupně 1 a 3 po jednom hodnocení. Na základě několika připomínek bylo nejdříve upřesněno zadání úlohy, které nevysvětlilo dostatečně jasně, která pole dialogového okna pro přidání otázky mají být vyplněna.

K otázkám byla připomínka ohledně smajlíků, které se ukazují u hodnotových a kombinovaných otázek. Tyto otázky mohou nabývat hodnoty 1 až 5, každé hodnotě odpovídá jeden smajlík. Je tedy celkem pět smajlíků, přičemž smajlíky 1 až 3 jsou považovány za kladné a smajlíky 4 a 5 jsou považovány za záporné. Z tohoto důvodu se původně zobrazovalo při tvorbě či úpravě hodnotových a kombinovaných otázek pět smajlíků a jejich přepínáním mezi kladnými a zápornými variantami bylo možné nastavit vysvětlení a výzvu otázky zvlášť pro kladné a zvlášť pro záporné hodnocení, které se ve Vyplňovací části zobrazí podle toho, jaký smajlík byl studentem vybrán. Jelikož z administrátorského pohledu jsou pouze dvě varianty (dobré a špatné hodnocení), bylo přepínání mezi pěti smajlíky shledáno jako zbytečně komplikované. Formulář byl upraven tak, aby se pro nastavení Vysvětlení a Výzvy přepínalo mezi pouze dvěma smajlíky. Na základě další poznámky ohledně nejasnosti významu některých polí dialogového okna pro práci s danou otázkou byla přidána do návodu pro fakulturní administrátory nová stránka, která se věnuje vysvětlení obsahu zmíněného okna. Odkaz na tuto stránku byl také vložen do okna pro práci s hodnotovými a kombinovanými otázkami.

Také byla uvedena připomínka, že by bylo dobré po najetí na ikonky na konci řádku každé otázky zobrazit radu, co otázky dělají. To bylo jednoduše implementováno přidáním atributu `title` do Vuetify komponenty `v-icon` s každou ikonkou.

Dále bylo v rámci zpětné vazby upozorněno na délku názvu seznamu otázek, jelikož nebylo možné přidat seznamy s názvy delšími než 20 znaků. Tento limit byl upraven podle limitu pro sloupec s názvem seznamu v databázi na 100 znaků, což umožnilo ukládání seznamů otázek i s delšími názvy.

6.2.4 Úloha 4 – Skrývání komentářů, generování CSV exportů

Čtvrtá úloha se zabývala skrýváním a odkrýváním komentářů a generováním CSV exportů jak komentářů, tak studentů, kteří vyplnili nějaký anketní lístek dané ankety. Zde byla úspěšnost splnění úlohy u všech odpovědí kladná. Čtyři odpovědi označily složitost úlohy stupněm 1 a stupeň 2 a 3 dostal po jednom hodnocení.

U této úlohy byl v jedné odpovědi uveden požadavek na přidání kódu katedry ke komentářům vztahujícím se na učitele. Vedoucím práce Ing. Michalem Valentou, PhD. byla tyto data přidána do databázového pohledu s komentáři k učitelům, na backendu byla získána z databáze a na frontendu zobrazena administrátorovi. Další požadavek byl na přidání data vyplnění anketního lístku studentem v tabulce a v CSV exportu se studenty, kteří vyplnili nějaký anketní lístek. To bylo provedeno získáním potřebných dat z databáze a zobrazením ve správném formátu v uživatelském rozhraní.

6.2.5 Celkové připomínky

V závěrečné sekci s prostorem na připomínky k administrátorskému modulu bylo poznamenání, že seznamy otázek stále nebyly editovatelné. Původně byly otázky a seznamy otázek editovatelné pouze superadministrátory a v rámci práce měla být editace otázek i seznamů otázek zpřístupněna i fakultním administrátorům, ale kvůli kontrole role na backendu nebylo možné změny v seznamech otázek uložit. Tato chyba byla odstraněna a fakultním administrátorů je nyní umožněno editovat otázky i seznamy otázek. Kromě této připomínky a dalších výše zmíněných byla celková zpětná vazba dosavadních odpovědí na upravené rozhraní administrátorského modulu pozitivní.

6.3 Nasazení

Jak již bylo zmíněno v podkapitole 2.3, nasazení probíhalo pomocí GitLab pipelines [9] zajišťujících CI/CD [10]. Frontend i backend administrátorského modulu obsahuje vlastní soubor `gitlab-ci.yml` s definovanými fázemi pipelines ve formě skriptů. Oba soubory jsou ale velmi podobné.

První fází obou pipelines je přípravná fáze neboli `before_script`. Během ní jsou instalovány potřebné nástroje pro SSH (Secure Shell) připojení, uložen SSH klíč a vytvořen adresář `.ssh/` pro práci s klíči.

Následuje fáze sestavení nebo také `build`, při které dochází k sestavení aplikace. Na backendu je výsledkem nebo také artefaktem soubor typu WAR (Java Web Archive) a na frontendu je jím soubor obecného typu. Pro každé ze tří používaných prostředí existuje samostatný skript pro sestavení, který zajistí sestavení programu v případě backendu se správnými argumenty a v případě frontendu s použitím správného příkazu definovaného v souboru `package.json`. Jednotlivé skripty se liší také názvem výsledného souboru (artefaktu).

Poslední fází je nasazení nebo také `deploy`, ve které dochází k nahrání artefaktu z fáze `build` do příslušného adresáře na serveru zajišťujícím běh aplikace na definované URL adrese. Nahrání adresáře na server je prováděno pomocí nástroje `lftp` (file transfer protocol), který je na začátku fáze `deploy` nainstalován. Opět je pro každé prostředí definován samostatný skript, pomocí kterého je určen použitý artefakt, koncový adresář a URL adresa daného prostředí.

Závěr

Cílem práce bylo aktualizovat vývojovou platformu uživatelského rozhraní administrátorského modulu aplikace Anketa ČVUT na novější dlouhodobě podporovanou verzi. Dále bylo cílem navrhnout a implementovat či opravit některé funkcionality z hlediska frontendové i backendové části aplikace.

V rámci práce byla úspěšně implementována možnost vyhodnocení jednotlivých anket přímo v modulu, generování CSV exportů studentů, kteří vyplnili nějaký anketní lístek dané ankety, což slouží jako podklady pro slosování, a generování CSV exportů všech typů komentářů. Dále implementace možnosti skrývání a odkrývání komentářů a úprava anketních otázek. Byla upravena implementace přidávání a odebrání předmětů a učitelů v rámci ankety a úprava dat ankety. Byl navrhnut způsob pro implementaci SSO přihlašování přes Shibboleth, které bude využito v produkčním prostředí. Pro otestování nejběžnějších úkonů administrátorů byla sestavena sada testů pro uživatelské testování, ze kterého byla získána cenná zpětná vazba, na základě které došlo k několika dalším úpravám administrátorského modulu. Serverová část aplikace byla doplněna o unit testy ověřující její funkčnost. Vše bylo zdokumentováno v kódu komentáři, pomocí kterých byla vygenerována detailní dokumentace implementace frontendu i backendu aplikace, a byly upraveny stávající a doplněny nové návody pro fakultní administrátory zajišťující snadnější ovládání modulu.

Během konzultací bylo diskutováno také o implementaci funkcionality vytváření nového běhu Ankety přímo v administrátorském modulu, ale jelikož tato akce vyžaduje větší zásahy do databázových pohledů týkajících se všech fakult dohromady, došlo se k závěru, že tato funkcionality by pro administrátorský modul nebyla vhodná. Dále bylo diskutováno o funkcionality úpravy stavů ankety, konkrétně pokud je již jednou uzavřená anketa znovu otevřena, tak se nově přidané odpovědi nepřidávají do celkových výsledků ankety. Prozatím bylo rozhodnuto, že tato funkcionality zůstane povolena kvůli charakteru jejího používání v takovémto krajním případě.

Všechny dílčí body zadání však byly úspěšně splněny a práce přinesla do administrátorského modulu několik užitečných funkcionalit, které fakultním administrátorům i superadministrátorům usnadní spávu anket.

..... Příloha A

Instalace a konfigurace

Pro vývoj administrátorského modulu je používána DevOps platforma GitLab. Frontend i backend je umístěn ve vlastním repozitáři:

- Frontend administrátorského modulu: https://gitlab.fit.cvut.cz/anketa-cvut-v3/admin-v2/survey_admin_frontend
- Backend administrátorského modulu: https://gitlab.fit.cvut.cz/anketa-cvut-v3/admin-v2/survey_admin_backend

Oba repozitáře obsahují soubor `README.md`, ve kterém jsou uvedeny informace k instalaci a konfiguraci obou částí. V obou částech je konfigurace prováděna pomocí konfiguračních souborů.

Bibliografie

1. VALENTA, Michal. *Metodický pokyn č. 3/2022: Anketa ČVUT: dokumentace a metodika provozu systému*. 3/2022. 2022. Dostupné také z: <https://www.cvut.cz/sites/default/files/content/d1dc93cd-5894-4521-b799-c7e715d3c59e/cs/20221007-metodicky-pokyn-c-32022.pdf>.
2. JUN, Jakub. *Anketa ČVUT verze 3 - modul pro správu anket - uživatelské rozhraní*. Praha, 2020. Dostupné také z: <https://dspace.cvut.cz/handle/10467/90395>. Bakalářská práce. České Vysoké Učení Technické.
3. PRESSMAN, Roger S. *Software Engineering: A Practitioner's Approach*. 7th ed. The McGraw-Hill Companies, 2010. ISBN 978-0-07-337597-7.
4. NAIK, Kshirasagar; TRIPATHY, Priyadarshi. *Software testing and quality assurance: Theory and Practice*. John Wiley & Sons, Inc., 2008. ISBN 978-0-471-78911-6.
5. *JUnit 5: The 5th major version of the programmer-friendly testing framework for Java and the JVM* [online]. The JUnit Team. [cit. 2024-05-07]. Dostupné z: <https://junit.org/junit5/>.
6. *A Guide to JUnit 5* [online]. bealdung. [cit. 2024-05-07]. Dostupné z: <https://www.bealdung.com/junit-5>.
7. FABER, Szczepan. *Mockito: Tasty mocking framework for unit tests in Java* [online]. [cit. 2024-05-07]. Dostupné z: <https://site.mockito.org/>.
8. *Deploy and release your application* [online]. GitLab. [cit. 2024-05-10]. Dostupné z: https://docs.gitlab.com/ee/topics/release_your_application.html.
9. *CI/CD pipelines* [online]. GitLab. [cit. 2024-03-30]. Dostupné z: <https://docs.gitlab.com/ee/ci/pipelines/index.html>.
10. FARCIC, Viktor. *The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices*. 1. vyd. Packt Publishing, 2016. ISBN 9781785280313.
11. NGUYEN, Duc Thang. *Anketa ČVUT verze 3 - modul pro správu anket - serverová část*. Praha, 2021. Bakalářská práce. České Vysoké Učení Technické.
12. *What is three-tier architecture?* [online]. IBM. [cit. 2024-04-01]. Dostupné z: <https://www.ibm.com/topics/three-tier-architecture>.
13. OSMANI, Addy. <https://addyosmani.com/blog/understanding-mvvm-a-guide-for-javascript-developers/> [online]. [cit. 2024-04-01]. Dostupné z: [Understanding%20MVVM%20-%20A%20Guide%20For%20JavaScript%20Developers](https://addyosmani.com/blog/understanding-mvvm-a-guide-for-javascript-developers/).
14. *Learning JavaScript Design Patterns by Addy Osmani - MVVM* [online]. O'Reilly. [cit. 2024-04-01]. Dostupné z: <https://www.oreilly.com/library/view/learning-javascript-design/9781449334840/ch10s06.html>.

15. *Nuxt Documentation: Introduction* [online]. Nuxt. [cit. 2024-04-01]. Dostupné z: <https://nuxt.com/docs/getting-started/introduction>.
16. *Vuetify Documentation: Introduction* [online]. Vuetify. [cit. 2024-04-01]. Dostupné z: <https://vuetifyjs.com/en/introduction/why-vuetify/>.
17. *Interface JpaRepository<T,ID>* [online]. Pivotal Software, Inc. [cit. 2024-05-09]. Dostupné z: <https://docs.spring.io/spring-data/jpa/docs/current/api/org/springframework/data/jpa/repository/JpaRepository.html>.
18. *Oracle Java Documentation - JDBC Basics* [online]. Oracle. [cit. 2024-03-30]. Dostupné z: <https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html>.
19. *Single Sign On* [online]. Huntington Ventures Ltd., 2006. [cit. 2024-04-06]. Dostupné z: <https://archive.ph/20140315095827/http://www.authenticationworld.com/Single-Sign-On-Authentication/>.
20. *Home* [online]. Shibboleth. [cit. 2024-04-09]. Dostupné z: <https://shibboleth.atlassian.net/wiki/spaces/CONCEPT/overview>.
21. *What is Shibboleth?* [online]. Shibboleth Consortium. [cit. 2024-04-09]. Dostupné z: <https://www.shibboleth.net/about-us/the-shibboleth-project/>.
22. *What is Security Assertion Markup Language (SAML)?* [online]. Oracle. [cit. 2024-04-09]. Dostupné z: <https://www.oracle.com/cz/security/cloud-security/what-is-saml/>.
23. *Shibboleth SP 3* [online]. CESNET, z.s.p.o. [cit. 2024-04-09]. Dostupné z: <https://www.eduid.cz/cs/tech/sp/shibboleth>.
24. *Shibboleth SP 3.0 - Install&Configure* [online]. České Vysoké Učení Technické. [cit. 2024-04-09]. Dostupné z: <https://wiki.cvut.cz/pages/viewpage.action?pageId=40077405>.
25. ŠTECHA, Vojtěch. *Anketa ČVUT - verze 3.0 - vyplňování anketních lístků*. 2019. Dostupné také z: <https://dspace.cvut.cz/bitstream/handle/10467/82698/F8-DP-2019-Stecha-Vojtech-thesis.pdf?sequence=-1%5C&isAllowed=y>. Diplomová práce. České Vysoké Učení Technické.
26. *Nuxt 2 LTS, EOL & Extended Support* [online]. Nuxt. [cit. 2024-04-03]. Dostupné z: <https://v2.nuxt.com/lts/>.
27. *Docs: Upgrade guide - migrate to Nuxt 3* [online]. Nuxt. [cit. 2024-04-03]. Dostupné z: <https://nuxt.com/docs/migration/overview>.
28. *Module Author Guide* [online]. Nuxt. [cit. 2024-05-15]. Dostupné z: <https://nuxt.com/docs/guide/going-further/modules>.
29. *Lekce 3 - Promises a asynchronní funkce* [online]. ITnetwork s.r.o. [cit. 2024-04-06]. Dostupné z: <https://www.itnetwork.cz/javascript/react/native/react-native-promises-a-asynchronni-funkce>.
30. ZABRISKIE, Matt; SARJEANT, John Jakob "Jake". *Getting Started* [online]. [cit. 2024-04-06]. Dostupné z: <https://axios-http.com/docs/intro>.
31. *The Intuitive Vue Framework* [online]. Nuxt. [cit. 2024-04-06]. Dostupné z: <https://axios.nuxtjs.org/>.
32. *Migrate to Nuxt 3: Component Options* [online]. Nuxt. [cit. 2024-04-06]. Dostupné z: <https://nuxt.com/docs/migration/component-options>.
33. *Using the Fetch API* [online]. MDN Contributors. [cit. 2024-04-06]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch.
34. PARSA, Pooya. *Ofetch* [online]. [cit. 2024-04-06]. Dostupné z: <https://github.com/unjs/ofetch>.

35. BUI, Binh. *Finally, we can use fetch API in node.js* [online]. 2022. [cit. 2024-04-07]. Dostupné z: <https://www.codewithyou.com/blog/finally-we-can-use-fetch-api-in-nodejs>.
36. *Nuxt/auth: Introduction* [online]. Nuxt. [cit. 2024-04-07]. Dostupné z: <https://auth.nuxtjs.org/>.
37. *Introduction: What is Vuex?* [online]. Vuex. [cit. 2024-04-07]. Dostupné z: <https://vuex.vuejs.org/>.
38. DESMAISONS, David. *Vue.Draggable* [online]. SortableJS. [cit. 2024-04-29]. Dostupné z: <https://github.com/SortableJS/Vue.Draggable>.
39. DESMAISONS, David. *Vue.draggable.next* [online]. SortableJS. [cit. 2024-04-29]. Dostupné z: <https://github.com/SortableJS/vue.draggable.next/tree/master>.
40. *Vue 3 markodwn* [online]. npm, 2023. [cit. 2024-05-08]. Dostupné z: <https://www.npmjs.com/package/vue3-markdown>.
41. CHERNEV, Iskren Ivov. *Moment.js Documentation: Project Status* [online]. [cit. 2024-04-30]. Dostupné z: <https://momentjs.com/docs/#/-project-status/>.
42. *Reading 9: Mutability & Immutability* [online]. MIT EECS, 2016. [cit. 2024-04-30]. Dostupné z: <https://ocw.mit.edu/ans7870/6/6.005/s16/classes/09-immutability/>.
43. *Date-fns: Modern JavaScript date utility library* [online]. [cit. 2024-04-30]. Dostupné z: <https://date-fns.org/>.
44. *Vue-toasted* [online]. npm, 2020. [cit. 2024-05-01]. Dostupné z: <https://www.npmjs.com/package/vue-toasted>.
45. SADIKEEN, Shakeeb. *Vue-toasted* [online]. [cit. 2024-05-01]. Dostupné z: <https://github.com/shakee93/vue-toasted>.
46. *Frequently asked questions: What versions of Vue.js are compatible with Vuetify?* [online]. Vuetify. [cit. 2024-04-06]. Dostupné z: <https://vuetifyjs.com/en/getting-started/frequently-asked-questions/>.
47. *Nuxt Blog: Announcing Nuxt 3.0 stable* [online]. Nuxt, 2022. [cit. 2024-04-06]. Dostupné z: <https://nuxt.com/blog/v3>.
48. *Upgrade Guide: Components* [online]. Vuetify. [cit. 2024-04-06]. Dostupné z: <https://vuetifyjs.com/en/getting-started/upgrade-guide/#general-changes>.
49. *RequestMapper* [online]. Shibboleth, 2018. [cit. 2024-04-09]. Dostupné z: <https://shibboleth.atlassian.net/wiki/spaces/SP3/pages/2065334391/RequestMapper>.
50. *HowToRequestMap* [online]. Shibboleth, 2018. [cit. 2024-04-09]. Dostupné z: <https://shibboleth.atlassian.net/wiki/spaces/SP3/pages/2065335006/HowToRequestMap>.
51. *Getting Started with JSDoc 3* [online]. The contributors to the JSDoc documentation. [cit. 2024-05-01]. Dostupné z: <https://jsdoc.app/about-getting-started>.
52. *jsdoc-vuejs: JSDoc for VueJS* [online]. npm, 2021. [cit. 2024-05-01]. Dostupné z: <https://www.npmjs.com/package/jsdoc-vuejs>.
53. *Command-line arguments to JSDoc* [online]. The contributors to the JSDoc documentation. [cit. 2024-05-01]. Dostupné z: <https://jsdoc.app/about-commandline>.
54. *Javadoc* [online]. Oracle. [cit. 2024-05-06]. Dostupné z: <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>.
55. *Swagger UI* [online]. SmartBear. [cit. 2024-05-09]. Dostupné z: <https://swagger.io/tools/swagger-ui/>.

Obsah příloh

Instalace a konfigurace instalace a konfigurace administrátorského modulu