



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

INFORMAČNÍ PORTÁL PRO STUDENTY

INFORMATION PORTAL FOR STUDENTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MIROSLAV ŠVARC

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAROSLAV DYTRYCH, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Švarc Miroslav**
Program: Informační technologie
Název: **Informační portál pro studenty**
Information Portal for Students
Kategorie: Informační systémy

Zadání:

1. Seznamte se se současnými komunikačními kanály studentů a existujícími diskusními fóry využívanými studenty FIT VUT.
2. Prostudujte API pro přístup k diskusím na sociální síti Facebook a k dokumentům a kalendářům od firmy Google.
3. Navrhněte nový informační portál pro studenty FIT VUT, který umožní přehledné strukturované diskuse k jednotlivým předmětům, ukládání dokumentů, práci se studentskými skripty apod. Řešení umožní integraci se sociální sítí Facebook, kalendářem od firmy Google a s dalšími vybranými službami, které studenti využívají. Portál poskytne také podporu pro vybrané činnosti Studentské unie.
4. Implementujte navržené řešení.
5. Vyhodnoťte výsledné řešení a vytvořte stručný plakát prezentující vytvořené řešení

Literatura:

- Dle doporučení vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Dytrych Jaroslav, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 30. října 2020

Abstrakt

Tato práce se zabývá analýzou aktuálně využívaných komunikačních systémů studentů, návrhem nového systému a jeho implementací. Součástí systému je propojení se sociální sítí Facebook, komunikační aplikací Discord a službami společnosti Google, jako jsou kalendáře a sdílený disk. Výsledkem je funkční systém, naprogramovaný v jazyce PHP, který může sloužit pro komunikaci studentů.

Abstract

This work deals with the analysis of current communication systems used by students, the design of a new system and its implementation. The system includes a connection to the social network Facebook, the communication application Discord and Google services such as calendars and shared drive. The result is a functional system, programmed in PHP, which can be used for student communication.

Klíčová slova

informační systém, komunikace studentů, laravel, php, Facebook, služba Google, Discord, L^AT_EX, fórum

Keywords

information system, student communication, laravel, php, Facebook, Google service, Discord, L^AT_EX, forum

Citace

ŠVARC, Miroslav. *Informační portál pro studenty*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Dytrych, Ph.D.

Informační portál pro studenty

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Dytrycha, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Miroslav Švarc
15. května 2021

Poděkování

Tímto bych rád poděkoval mému vedoucímu Ing. Jaroslavu Dytrychovi, Ph.D. za čas strávený na konzultacích a odbornou pomoc na této práci.

Obsah

1	Úvod	2
2	Analýza a zhodnocení současného stavu	3
2.1	Discord	3
2.2	Facebook	3
2.3	Kancelářské aplikace Google a Microsoft	4
2.4	Současný systém Fituška	4
3	Návrh řešení	10
3.1	Požadavky na nový systém	10
3.2	Systém rolí	14
3.3	Použité technologie	15
3.4	Databáze a ukládání dat	19
3.5	Uživatelské rozhraní	20
4	Implementace	24
4.1	Základ fóra	24
4.2	Služba Facebook	28
4.3	Discord robot	29
4.4	Služby Google	30
4.5	Služba L ^A T _E X	32
4.6	Údržba systému	33
5	Testování a nasazení	35
5.1	Testování	35
5.2	Nasazení	37
6	Závěr	38
	Literatura	39

Kapitola 1

Úvod

V současné době je pro studenty samozřejmostí komunikovat mezi sebou prostřednictvím Internetu. Využívají k tomu různé komunikační systémy, které ovšem nejsou primárně určeny pro komunikaci studentů. Důležité informace je obtížné vyhledat a často se nacházejí na různých místech.

Pro řešení problému centralizace informací již v minulosti vznikl systém Fituška¹, který byl studenty hojně využíván, ale momentálně je již zastaralý, pomalý a je obtížné sehnat studenty, kteří by prováděli jeho údržbu. Momentálně ho studenti již téměř nepoužívají a přesunuli se především na sociální síť.

Mezi aktuálně využívané platformy se dostaly především sociální síť Facebook a komunikační aplikace Discord, které studentům vyhovují díky možnosti rychlé komunikace. Pro vytváření studentských skript a dalších materiálů se nejčastěji využívají sdílené dokumenty od společností Google a Microsoft. V obou těchto případech nastává problém s uchováváním informací, jelikož se nacházejí na serverech třetích stran a mohou se kdykoliv ztratit, pokud je nikdo ze studentů nezálouje.

Tato práce se zabývá návrhem a implementací komunikačního systému pro studenty, který umožní přehlednou strukturovanou diskuzi, sdílení souborů, integraci s již používanými komunikačními aplikacemi Facebook a Discord a také se službami společnosti Google, jako jsou kalendáře a sdílené soubory. Práce je rozdělena do 5 kapitol. Kapitola 2 se zabývá analýzou a zhodnocením současného stavu. Shrnuje všechny hlavní druhy komunikace, které studenti aktuálně využívají, a popisuje jejich výhody a nevýhody. Podrobnější analýza je věnována systému Fituška, který by nově vzniklý systém měl nahradit. Kapitola 3 popisuje návrh řešení a požadavky na nově vytvořený systém. Také jsou v ní popsány konkrétní technologie použité pro vývoj systému, architektura, způsob ukládání dat, systém rolí, případy užití a uživatelské rozhraní. Implementací se zabývá kapitola 4. Ve té je detailně popsán vývoj celého systému a jeho součástí. V poslední kapitole 6 je provedeno zhodnocení práce.

¹Fituška je studentské diskuzní fórum určené pro studenty Fakulty informačních technologií Vysokého učení technického v Brně. Fituška je dostupná na webové adrese <https://fituska.eu/>

Kapitola 2

Analýza a zhodnocení současného stavu

Před začátkem návrhu a vývoje nového studentského portálu je nejprve nutné provést analýzu aktuálně používaných komunikačních systémů, které jsou mezi studenty populární. Mezi ně patří především aplikace Discord, sociální síť Facebook, online kancelářské aplikace od společností Google a Microsoft a v poslední řadě současný systém Fituška.

2.1 Discord

Aktuálně nejrozšířenější aplikací, kterou studenti využívají pro komunikaci, je Discord. Studenti zde mají vytvořený svůj server, který slouží pro komunikaci všech ročníků v rámci celé fakulty. Server je rozdělen na kanály, které slouží k základnímu strukturování komunikace. Každý předmět má svůj vlastní kanál a studenti vidí pouze ty, které mají v jejich aktuálním ročníku povinné, nebo které si zvolí ve speciálním kanálu, který slouží právě k výběru sledovaných předmětů. Dále server obsahuje množství kanálů, jejichž většina již nesouvisí se studiem.

Výhodou je, že komunikace probíhá velmi rychle a pokud student položí otázku, odpověď na ni obvykle dostane téměř okamžitě. Tomu napomáhá i přítomnost studijních poradců, kteří odpovídají především na obecné otázky ohledně studia.

Nevýhodou je nemožnost strukturovat diskuzi uvnitř jednotlivých kanálů. Zprávy přibývají rychle a v takovém množství, že ty důležité se brzy ztratí v historii a je poměrně obtížné je dohledat bez znalosti autora nebo slovního spojení, které zpráva obsahuje. Zároveň s přibývajícím počtem kanálů se jejich seznam stává nepřehledným. Celý server vyžaduje poměrně velkou údržbu, která je v rukou studentů a vždy musí existovat jeden nebo více administrátorů, kteří server udrží v chodu.

2.2 Facebook

Před nástupem Discordu byla pro komunikaci studentů nejvíce využívána sociální síť Facebook. Zde každý rok členové Studentské unie zakládají 2 skupiny. První skupina je vždy určena pro diskuze, které se přímo týkají školy nebo výuky. Druhá skupina slouží pro veškeré ostatní diskuze, jako je například seznamování se se spolužáky a často jí také využívají různé společnosti pro své inzeráty s nabídkami práce a brigád pro studenty.

Skupiny jsou vytvářeny ručně a jsou nastaveny tak, že k obsahu mají přístup pouze jejich členové. Student tak musí do každé skupiny zvlášť požádat o přijetí a počkat, až správce jeho žádost potvrdí. To někdy může trvat i několik hodin až dní, pokud se student rozhodne přidat například o víkendu a žádný ze správců momentálně není dostupný.

Nevýhodou je nemožnost strukturovat diskuze. Student má pouze možnost vytvořit příspěvek a musí do něj sám uvést, k jakému tématu nebo předmětu patří. Studenti příspěvky nejčastěji označují tak, že do hranatých závorek vloží zkratku předmětu, nebo klíčové slovo, kterého se příspěvek týká a podle kterého se dá později dohledat. Zde opět nastává stejný problém, jako v případě aplikace Discord. Příspěvků je velké množství a všechny se nacházejí na jedné stránce, seřazené od nejnovějšího. Dohledat některý konkrétní starší příspěvek je tedy poměrně zdlouhavá záležitost.

Facebook má samozřejmě i své výhody. Aktuálně se jedná o jednu z nejrozšířenějších sociálních sítí a většina studentů ho již využívá pro osobní potřebu, takže se nemusejí učit pracovat s novým systémem.

2.3 Kancelářské aplikace Google a Microsoft

Další způsob komunikace, který je studenty často využíván, jsou sdílené dokumenty od společností Google a Microsoft. Obě tyto společnosti mají vlastní kancelářský balík, který je pro uživatele dostupný online. Studenti tyto dokumenty často využívají ke tvorbě studentských skript, nebo při učení na zkoušku. Výhodou je, že veškeré úpravy dokumentu jsou dostupné v reálném čase, takže na jednom dokumentu může zároveň pracovat velké množství studentů. Také si mohou do dokumentu navzájem psát poznámky a návrhy na opravy chyb v textu. Tyto dokumenty jsou často tvořeny během učení na zkoušku a pokud odkaz na ně nikdo nezalohuje, jsou brzo ztraceny a další rok je studenti musí tvořit znovu. Kvůli tomu vznikají desítky různých souborů a je obtížné vyznat se v tom, který z nich je ten správný.

2.4 Současný systém Fituška

Současný systém Fituška vznikl kolem roku 2007 a momentálně je pod správou Studentské unie. Jedná se o diskuzní fórum, postavené na systému phpBB¹. Právě tento systém má být nahrazen řešením, které vznikne v rámci této práce, proto je mu věnována více obsáhlá analýza.

Studenty je Fituška využívána minimálně a většina z nich se přesunula na výše zmíněný Facebook a později Discord. Fituška kvůli tomu postupně upadá, studenti na ni přestávají přispívat a chybí lidé, kteří by byli ochotni ji spravovat a zabezpečovat její chod. To zahrnuje především vytvoření nových kategorií předmětů pro aktuální školní rok, přesunutí starých diskuzí do archivu a přidání e-mailů nových studentů do databáze, aby se tito studenti mohli registrovat. Tyto operace se musí provádět ručně spuštěním skriptů vždy na začátku nového školního roku, a má je na starosti hlavní administrátor Fitušky. Nový účet si na Fitušce nemůže vytvořit kdokoli, ale při registraci je ověřen přes školní e-mail, který musí být v databázi. Tím se ověří, že zadaný e-mail je platný.

Na obrázku 2.1 je zobrazena hlavní strana systému Fituška, na kterou se student dostane po přihlášení. Tato úvodní strana není příliš přehledná, v horní části jsou vidět kategorie

¹phpBB je volně dostupný systém pro vytvoření interaktivního fóra, dostupný na webové adrese <https://www.phpbb.cz/>

s předměty pouze pro aktuální školní rok a většina předmětů je v kategorii Volitelné, kde je obtížné rychle najít požadovaný předmět.

Fórum:
 Hledat
 Členové
 FAQ
 Uživatelský panel
 0 nových zpráv
 Odhlásit [mira135]

Odkazy:
 WIS FIT
 Rozvrh pro 3BIT
 Osobní rozvrh
 Fórum SU FIT
 Pastebin pro fituška
 FIT Wiki
 FIT server

Terminy: (přidat, editovat, log, exportovat)
 Momentálně nejsou žádné termíny.

Poslední návštěva: 15 dub 2021 11:15 am
 Právě je 16 dub 2021 01:02 pm
 Světek má třeba.
 Nyní je ticho.

Vyhledat témata bez odpovědí | Zobrazit aktivní témata

Zobrazit nepřečtené příspěvky | Zobrazit nové příspěvky | Zobrazit vaše příspěvky

fórum		Témata	Příspěvky	Poslední příspěvek
Předměty (2020/2021)				
	1BIT Povinné předměty v 1. ročníku bakalářského studia Subfóra: IDM, ILG, IEL, IUS, IZP, IMA1, INC, IOS, ISU	18	24	10 dub 2021 01:33 am xspria01 ➔
	2BIT Povinné předměty v 2. ročníku bakalářského studia Subfóra: IAL, IFJ, IMA2, INP, IPT, ISS, IDS, IPK, IPP, IZG, IZU	5	5	18 bře 2021 03:50 pm xkoren10 ➔
	3BIT Povinné předměty v 3. ročníku bakalářského studia Subfóra: IIS, IMP, IMS, ISA, ISP, ITU, IBT, ISZ	7	12	13 úno 2021 12:22 pm kalkak ➔
	1MITAI Povinné předměty v 1. ročníku magisterského studia (společně pro všechny specializace) Subfóra: MSP, TIN	10	13	25 led 2021 12:21 pm Silvia ➔
	2MITAI Povinné předměty v 2. ročníku magisterského studia (společně pro všechny specializace) Subfóra: SEP, MSZ, DIP	0	0	Žádné příspěvky
	Povinné volitelné - angličtina (PVA, V) a další jazyky (V) Subfóra: Angličtina (BAN a jiné), BAN1, BAN2, BAN3, BAN4, AEU, FCE, Němčina, Španělština, Ruština	1	1	15 pro 2020 03:19 pm Supnik36 ➔
	Povinné volitelné technické (PVT) Subfóra: ICP, JA, IVH, ICS	0	0	Žádné příspěvky
	Povinné volitelné humanitní (PVH) Subfóra: FIK, HKO, HVR, PRM, RET	0	0	Žádné příspěvky
	Volitelné (V) Subfóra: IW1, IW2, IW5, IAN, IBS, IFS, IIZ, IJC, ILI, IMU, INI, IPA, IPPe, IPS, IPZ, ISC, ISI, ISM, ITP, ITS, ITW, ITY, IVG, IVS, IZA, KRĐ, C3P, AGS, AIS, AVS, BIF, BIN, BIO, BIS, BMS, BZA, DJA, EVO, FAD, FLP, FYO, GAL, GJA, GMU, GUX, GZN, HSC, KKO, KKN, KRY, MBA, MPR, MOG, MUL, NAV, NSB, PBI, PCG, PCS, PDB, PDI, PDS, PFP, FGP, FGR, FIS, FKS, FMA, POS, FOLA, PRL, ROBa, RTSa, SEM, SFC, SIN, SLOa, SNT, SPR, SKI, SUL, SUR, TAD, TANA, THE, TOI, UPA, VGE, VNU, VYPa, WAP, ZPa, ZPO, ZTI, IST1, IST2, IATD, IDZL, IDZZ, IDH1, IDH2, BKPO, IODI, IRO3, IROZ, IIPD, IIPY, IMK, IPGO, IMAE, IMIE, IPMA / POM, IZFI, IZMA, LOG, OPM, IAH5, IHS, IKAH	46	57	01 dub 2021 10:05 am Vladce ➔

Obrázek 2.1: Hlavní strana systému Fituška

Pokud chce student zobrazit příspěvky a materiály z předešlých let, musí si nejdříve vybrat daný rok, jak je vidět na obrázku 2.2. Až po jeho rozkliknutí se zobrazí kategorie předmětů, jako na hlavní straně, ale pro vybraný školní rok.

Archiv			
	Předměty (2019/2020)	189	261
	Předměty (2018/2019)	215	342
	Předměty (2017/2018)	228	403
	Předměty (2016/2017)	260	548
	Předměty (2015/2016)	286	770
	Předměty (2014/2015)	436	2093
	Předměty (2013/2014)	710	4835
	Předměty (2012/2013)	1145	11838
	Předměty (2011/2012)	1550	21374
	Předměty (2010/2011)	2525	49406
	Předměty (2009/2010)	2915	62275
	Předměty (2008/2009)	2690	56398
	Předměty (2007/2008)	2150	44651
	Old Fituška (2006/2007)		

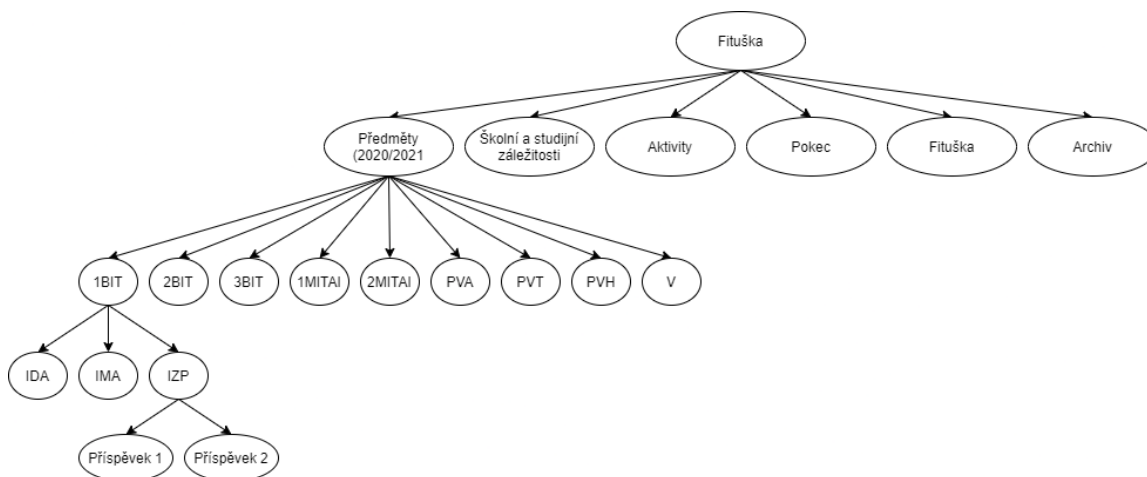
Obrázek 2.2: Archiv předchozích ročníků

Struktura dat

Data jsou v systému Fituška uložena ve stromové struktuře, která je naznačena na obrázku 2.3. Na vrcholu je několik hlavních kategorií, které se dělí na fóra. Ty se dále mohou dělit na subfóra. Tato fóra a subfóra poté obsahují jednotlivá diskuzní vlákna, která mohou studenti sami vytvářet.

Mezi hlavní kategorie patří *Předměty* pro aktuální ročník, *Školní a studijní záležitosti*, *Aktivity*, *Pokec*, *Fituška* a *Archiv*.

Veškeré diskuze z kategorie *Předměty* se vždy na začátku nového školního roku přesunou do nově vytvořeného fóra v kategorii *Archiv* a obsah této kategorie je vytvořen znovu, jen bez diskuzních vláken. Díky tomuto systému jsou oddělena stará data od nových. Toto řešení ovšem není příliš uživatelsky přívětivé, jelikož obsah předmětů se v průběhu let mění jen minimálně a studenti stále navštěvují i staré diskuze. Pokud chce student například získat všechny materiály k nějakému předmětu, musí postupně projít všechna fóra v kategorii *Archiv* a v každém z nich otevřít příslušné subforum. Toto hledání zabírá velké množství času a může to být jedním z důvodů, proč studenti přestávají Fitušku používat.

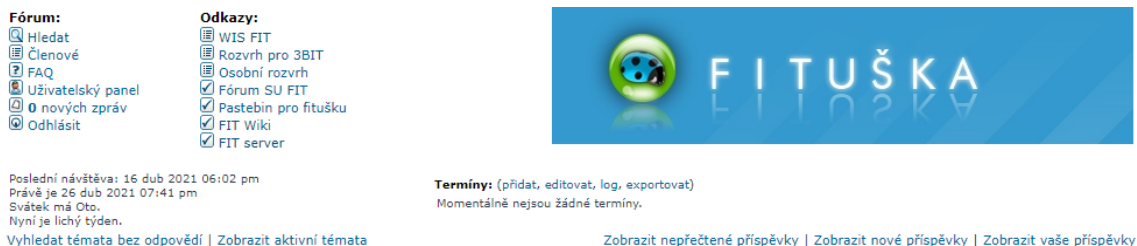


Obrázek 2.3: Struktura uložení dat

Uživatelské rozhraní

Důležitou součástí analýzy je také uživatelské rozhraní, jelikož právě to uvidí uživatel nejčastěji. Systém Fituška má uživatelské rozhraní poměrně jednoduché, ve stylu diskuzního fóra.



Hlavní stránka, kterou můžeme vidět na obrázku 2.4, obsahuje navigační část, která se nachází v horní části stránky. Ta obsahuje odkazy na důležité funkcionality a stránky, jako je vyhledávání, seznam uživatelů nebo tlačítko pro odhlášení, informace o aktuálním času a datu, jaký je týden a přehled termínů. V pravé části dále obsahuje odkazy na nepřečtené a nové příspěvky a také vlastní příspěvky uživatele. Tato část je pro uživatele poměrně nepřehledná, například přehledu termínů jsem si všiml až nyní, při podrobné analýze.



Obrázek 2.4: Navigační část systému Fituška

Druhou částí hlavní stránky je Obsah fóra, který můžeme vidět na obrázku 2.1 z předchozí kapitoly. Ten obsahuje přehled všech fór, rozdělených do kategorií. U každého fóra je vidět seznam subfór, počet témat a příspěvků, čas, datum a autora posledního příspěvku. Pokud se v některém subfóru nachází nepřečtený příspěvek, je jeho název zobrazen oranžovou barvou.

Na obrázku 2.5 je zobrazena patička, která se nachází ve spodní části stránky a obsahuje přehled základních statistik o návštěvnosti, jako je počet aktuálně přítomných uživatelů a jejich seznam a také celkový počet příspěvků, témat a členů.

Kdo je online	
	Celkem je online 1 uživatel :: 1 registrovaný a 0 skrytých (Tato data jsou založena na uživateli, kteří byli aktivní za posledních 5 minut) Nejvíce zde současně bylo přítomno 277 uživatelů dne 30 zář 2009 08:14 pm
	Registrovaní uživatelé: mira135
	Legenda :: Administrátoři , Globální moderátoři
Statistiky	
	Celkem příspěvků 330001 Celkem témat 24449 Celkem členů 8791 Nejnovějším uživatelem je cierto

Obrázek 2.5: Patička systému Fituška

Po rozkliknutí některého subfóra se uživatel zobrazí přehled všech vláken, která obsahuje. Pokud je příspěvků malé množství, je tento seznam přehledný a uživatelsky přívětivý. U každého názvu je vidět, zda příspěvek obsahuje přiložené soubory, počet komentářů a zobrazení. Tlačítko pro vytvoření nového tématu se nachází na začátku i na konci seznamu. Na obrázku 2.6 je zobrazeno subfórum předmětu IDM pro letošní rok. Každé subfórum také obsahuje kategorii *Oznámení*, ve které jsou zobrazena důležitá vlákna od administrátorů.

 Stránka 1 z 1 [3 témat]

Sledovat fórum Ignorovat fórum		Označit všechna témata jako přečtená			
Témata	Ignorovat	Autor	Odpovědi	Zobrazení	Poslední příspěvek
Oznámení					
 FITUŠKA NEEDS YOU	Ignorovat	redteam4ever	0	2775	15 bře 2019 02:24 pm redteam4ever 
Témata					
  2. Opravný termin	Ignorovat	Preparator	0	17	04 úno 2021 02:28 pm Preparator 
  1. opravný termin	Ignorovat	sunsetka	1	119	04 úno 2021 02:23 pm Preparator 
  1. Termin (všechny varianty)	Ignorovat	Preparator	0	22	04 úno 2021 02:14 pm Preparator 

Zobrazit témata za předchozí: Seřadit podle: Sestupně:

 Stránka 1 z 1 [3 témat]



Obrázek 2.6: Přehled příspěvků subfóra IDM pro rok 2021

Další důležitou částí uživatelského rozhraní je textový editor. S tím uživatelé pracují při přidání každého příspěvku nebo komentáře. Textový editor systému Fituška je ukázán na obrázku 2.7. Ovládání tohoto editoru není příliš přehledné, jednotlivé funkcionality se nacházejí v různých částech editoru, včetně levého a pravého okraje. To může být pro uživatele matoucí. Obsahuje základní formátování textu, které pro diskuzní fórum bez problémů stačí.

Odeslat odpověď

Předmět: Re: 2. Opravný termin

Tělo zprávy: **B** / **I** / **u** Quote Code List List= [*] **Img** **URL** Velikost písma:

Smajlíci:  **Barva písma:** 

Tip: Styly mohou být rychle změněny pokud označíte text

Možnosti:
 BBCode je zapnutý Zakázat BBCode
 [img] je zapnutý Zakázat smajlíky
 [flash] je vypnutý Zakázat URL adresy v tomto příspěvku
 Přiložit podpis (podpisy mohou být nastaveny v uživatelském panelu)
 Upozornit mě, přijde-li odpověď

Příloha

Pokud si přejete přidat jednu nebo více příloh, vyplňte následující údaje.

Název souboru Soubor nevybrán

Poznámka

Obrázek 2.7: Textový editor systému Fituška

Celé uživatelské rozhraní na první pohled nepůsobí příliš přívětivě a student, který se do systému poprvé přihlásí, z toho může být zmaten. Ve všech částech se nachází velké množství textů a ikon, které přehlednosti spíše škodí. Vzhled je také poměrně zastaralý, jelikož nebyl dlouhou dobu aktualizován. Určitě by pomohlo výrazné zjednodušení rozložení a také udržování stále stejného podfóra pro předmět, místo každoročního přesunu do archivu a vytváření nového podfóra.

Kapitola 3

Návrh řešení

Cílem této práce je navrhnout a implementovat systém, který bude sloužit pro komunikaci studentů na Fakultě informačních technologií Vysokého učení technického v Brně. Výsledný informační systém nahradí stávající systém Fituška.

3.1 Požadavky na nový systém

Systém bude umožňovat strukturovanou diskuzi k jednotlivým předmětům, a to nezávisle na školním roce. Dále bude obsahovat integraci se sociální sítí Facebook, která umožní přenos zpráv mezi Facebookovými skupinami a nově vzniklým systémem, integraci s komunikační aplikací Discord, které kromě přenosu zpráv, stejně jako v případě Facebooku, umožní navíc studentům získávat data ze systému přímo v aplikaci Discord, a to pomocí robota, který bude nainstalován na fakultním Discord serveru. Systém také bude obsahovat integraci s kalendáři a sdílenými dokumenty společnosti Google. Kalendáře budou sloužit k vytváření událostí, které uvidí všichni studenti u jednotlivých předmětů. Sdílené soubory budou určeny k vytváření studentských skript a jiných materiálů a budou sjednoceny u jednotlivých předmětů, aby bylo pro studenty snadné je nalézt.

Další důležitou součástí systému je podpora pro činnosti Studentské unie. Ta zahrnuje možnost vytváření diskuzí a nahrávání souborů, které budou přístupné pouze pro členy, možnost zasílání hromadných zpráv do Facebookových skupin a do kanálů v aplikaci Discord.

Důležitou součástí systému je také podpora vkládání matematických vzorců do příspěvků. Pro jejich zápis bude využit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

Na základě tohoto popisu byl sestaven detailní seznam požadavků na nový systém. Tento seznam se dá rozdělit na požadavky funkční a nefunkční [6].

Funkční požadavky

Funkční požadavky obsahují funkce, které budou využívat uživatelé a samotný systém.

1. Systém

- Vyhledávání příspěvků
- Zobrazení seznamu členů
- Zobrazení seznamu předmětů a vyhledávání v něm
- Import nových uživatelů
- Import předmětů
- Vytvoření kalendáře a složky na disku Google pro nově importovaný předmět

2. Uživatel

- Vytvořit účet
- Přihlásit/odhlásit se
- Editovat/smazat účet
- Přidat/odebrat roli
- Sledovat předmět
- Zobrazení kontaktů, seznamu předmětů, diskuzí

3. Diskuze

- Vytvoření/smazání tématu
- Vytvoření/zobrazení/editace/smazání příspěvku
- Ohodnocení příspěvku
- Okomentování příspěvku/komentáře
- Nahrání souboru do příspěvku
- Vložení matematické rovnice do příspěvku/komentáře

4. Soubory/zadání/skripta

- Nahrání/zobrazení souboru
- Vytvoření/otevření sdíleného dokumentu
- Nahrání/otevření zadání
- Zobrazení studentských skript

5. Studentská unie

- Kontaktování unie pomocí formuláře
- Zobrazení přijatých formulářů
- Zobrazení členů unie a kontaktů na ně
- Sklad souborů pro členy

6. Kalendář

- Zobrazení nadcházejících událostí u jednotlivých předmětů
- Vytvoření nové události u předmětu

7. Facebook

- Prohledání skupin na Facebooku a získání příspěvků
- Zařazení příspěvků ke správným předmětům a uložení do databáze
- Přenesení příspěvku ze systému do Facebookových skupin
- Poslání hromadné zprávy do Facebookových skupin

8. Discord

- Přenos příspěvků z kanálů v aplikaci Discord do systému pomocí bota
- Zařazení příspěvků ke správným předmětům a uložení do databáze
- Získávání příspěvků ze systému pomocí příkazů v aplikaci Discord
- Přenos příspěvků ze systému do správného kanálu v aplikaci Discord
- Poslání hromadné zprávy do kanálů v aplikaci Discord

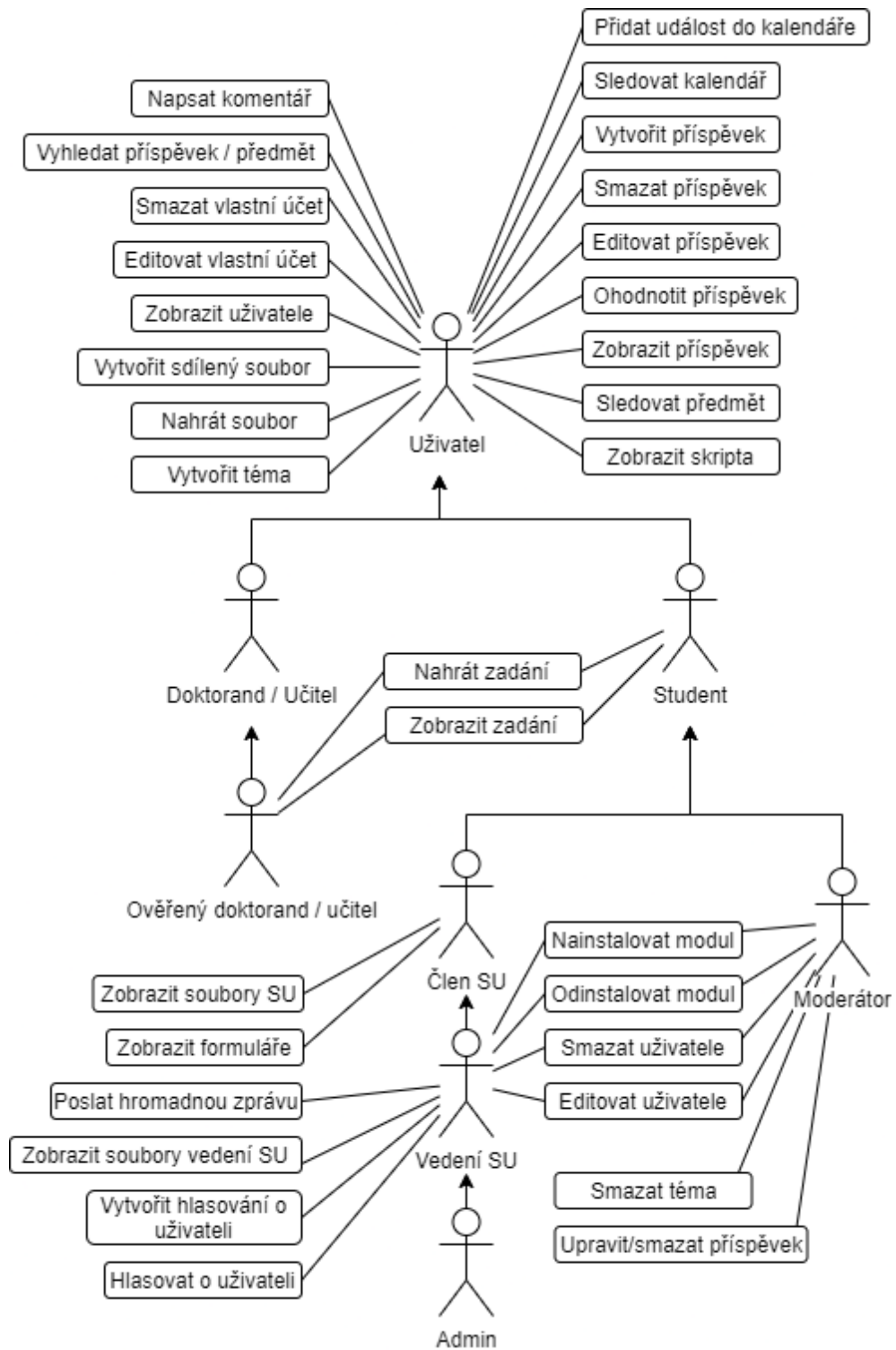
9. Administrace

- Instalace/Odinstalace modulů
- Vytvoření hlasování o ověření uživatele
- Hlasování Ano/Ne o ověření uživatele

Nefunkční požadavky

- Zamezení přístupu nepřihlášeným uživatelům
- Zamezení přístupu k částem systému na základě přidělených uživatelských rolí
- Zamezení vzniku situace, kdy by neexistoval žádný uživatel s rolí Administrátor – nový bude automaticky vybrán ze členů vedení studentské unie, popřípadě ostatních členů unie. Pokud by ani tak nebyl vybrán nový administrátor, vybere se ze členů Akademického senátu FIT, jejichž seznam je k dispozici na webu fakulty.
- Zamezení přístupu k funkcím modulu, pokud je odinstalován

Na základě funkčních požadavků byl vytvořen UML diagram případů užití, který je vidět na obrázku 3.1. Tento diagram popisuje typické interakce uživatelů se systémem a představuje, jak bude systém používán. Jednotliví uživatelé se označují jako *Actors* (česky aktér) a v našem případě je představují jednotlivé role [3]. Role své případy užití dědí, takže například *Ověřený doktorand* může provádět stejné operace jako *Doktorand* a navíc ještě operace příslušící jeho roli. Nejvýše v hierarchii je role *Administrátor*.



Obrázek 3.1: Diagram případů užití

3.2 Systém rolí

V systému existují různé role, podle kterých se řídí přístup k jeho různým částem. Tyto role se dají rozdělit do několika úrovní. První úrovní jsou role, které uživatel získá po registraci do systému. Tyto role odpovídají statusu, jaký má daný uživatel na fakultě.

Jedná se o tyto role:

1. Student – přístup k diskuzím, souborům, písničkám, kalendářům,
2. Doktorand – přístup k diskuzím, kalendářům,
3. Učitel – přístup k diskuzím, kalendářům,

Na druhé úrovni jsou role, které lze získat až po ověření daného uživatele s rolí Učitel nebo Doktorand. To probíhá pomocí hlasování, kterého se účastní vedení Studentské unie. Tyto role umožňují uživateli přístup k souborům a písničkám.

Jedná se o tyto role:

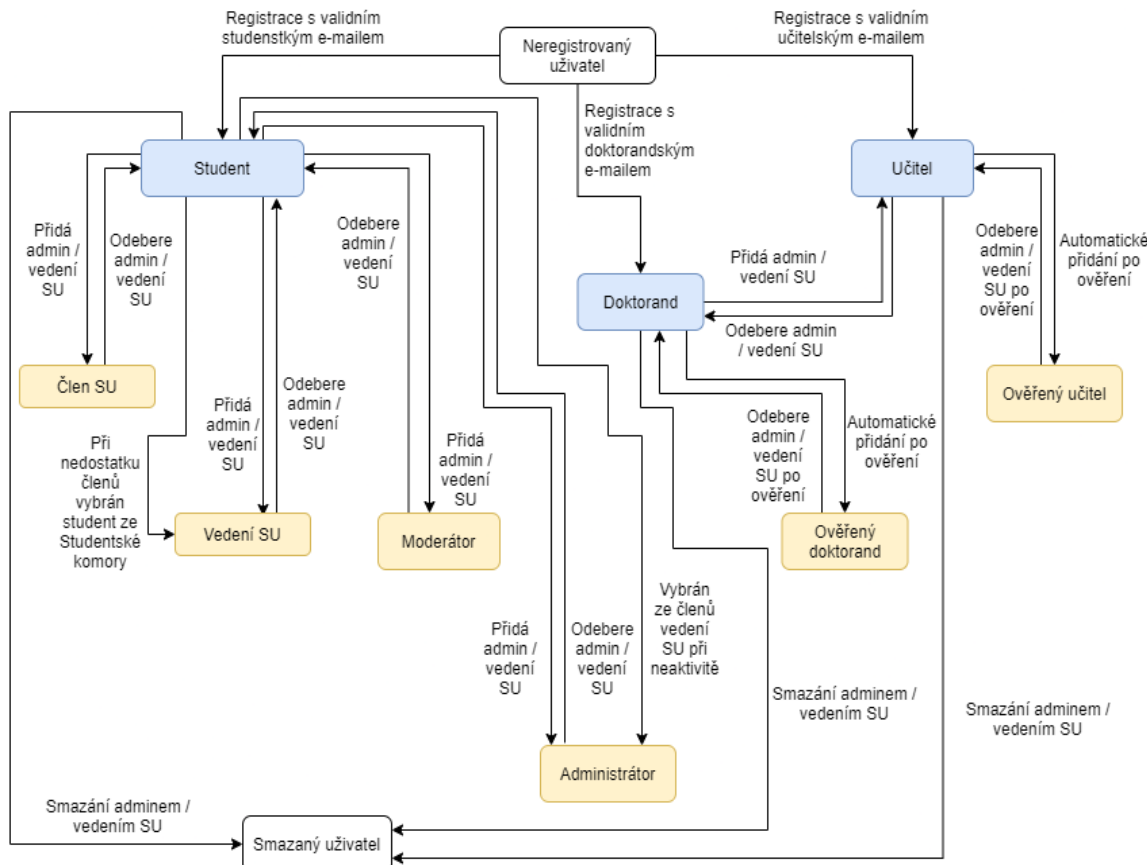
1. Ověřený doktorand
2. Ověřený učitel

Poslední úrovní jsou speciální role. Ty lze získat pouze přidělením administrátorem, moderátorem nebo členem vedení Studentské unie.

Jedná se o tyto role:

1. Člen SU – umožňuje přístup do sekce Studentské unie
2. Vedení SU – umožňuje přístup do sekce Studentské unie, posílání hromadných zpráv, vstup do administrace
3. Moderátor – umožňuje přístup do všech částí systému, kromě částí pro vedení studentské unie
4. Administrátor – umožňuje přístup do všech částí systému

Na obrázku 3.2 je zobrazen diagram systému rolí a přechody mezi nimi.



Obrázek 3.2: Diagram systému rolí

3.3 Použité technologie

Výběr správných technologií je důležitým krokem vývoje informačního systému. Pro vývoj tohoto systému byl zvolen programovací jazyk PHP verze 7.4, konkrétně jeho framework¹ Laravel² verze 7. Laravel je volně dostupný framework, který využívá softwarové architektury MVC³. Byl zvolen především díky své jednoduchosti a použitelnosti pro rozsáhlé aplikace. Jeho použití je jednoduché, velké množství komponent je již předpřipraveno a lze je vygenerovat pomocí jednoduchých příkazů. To značně urychluje celý vývoj systému.

Pro vytvoření robota pro aplikaci Discord byl zvolen jazyk Python a knihovna discord.py⁴. Důvodem tohoto výběru byla možnost jednoduchého a rychlého vývoje robota, obsáhlá dokumentace a komunita kolem projektu discord.py. Pro zpracovávání požadavků, zaslaných ze systému na robota byl využit framework Quart⁵. Ten přidává do Pythonu možnost směřování cest podle adresy URL.

¹Framework je sada nástrojů, knihoven, konvencí a osvědčených postupů, které vytváří abstrakci nad rutinními úkoly do obecných modulů, které mohou být lehce znovu využity [1]

²<https://laravel.com/>

³Model-View-Controller je softwarová architektura, která aplikaci rozděluje do tří na sobě nezávislých komponent, kterými jsou datový model aplikace, uživatelské rozhraní a řídicí logika

⁴discord.py je jednoduchá nadstavba nad rozhraním Discordu, dostupná na adrese <https://discordpy.readthedocs.io/en/stable/>

⁵<https://pgjones.gitlab.io/quart/>

Laravel

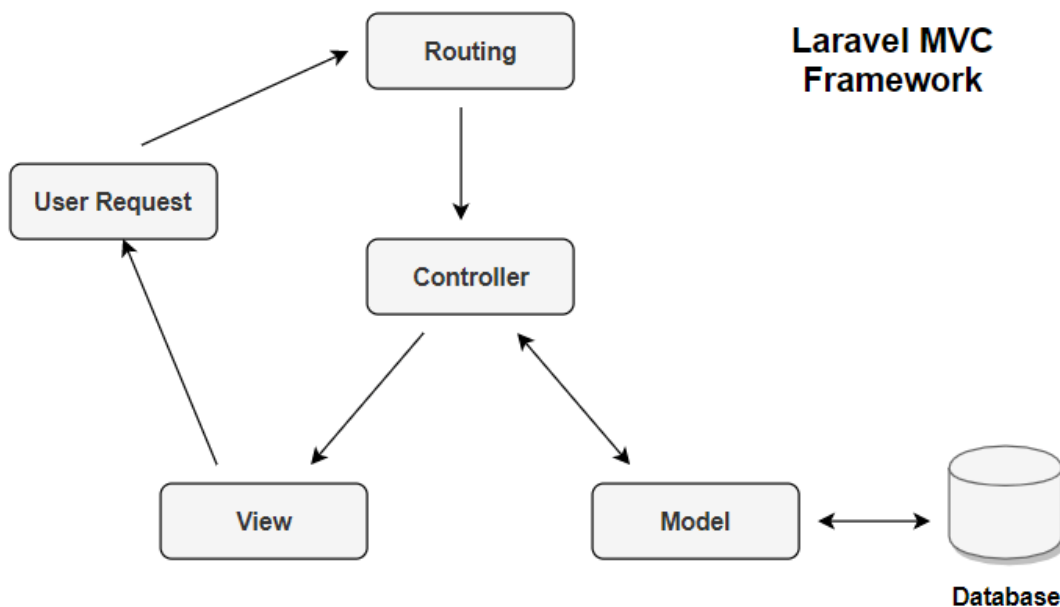
Laravel je jeden z nejpopulárnějších frameworků pro jazyk PHP. Podle webové stránky Builtwith je momentálně využíván na provoz 661 355 webů, z toho 2 314 z nich je provozováno v České republice [7].

Laravel využívá architekturu MVC. Aplikace je rozdělena na tři hlavní části:

- Modely – obsahují data a jim příslušící logiku. Představují data, která jsou přenášena mezi kontroléry. Každý model představuje jednu tabulku z databáze.
- Kontroléry – části aplikace, které zpracovávají uživatelské požadavky přijaté z pohledu. Získávají data z modelů a posílají je zpět pohledům, kde jsou prezentována uživateli.
- Pohledy – představují části aplikace, které prezentují data. Tyto části vidí uživatel a interaguje s nimi [8].

Směrování

Další důležitou součástí je směrování cest podle adresy URL, kterou uživatel zadá. Úkolem směrování je podle této zadané adresy provést příslušnou akci, kterou může být zavolání funkce z některého kontroléru nebo pouze provedení jednoduché operace, určené přímo v souboru s definovanými cestami. Všechny cesty jsou definovány v souboru `app/routes.php` [11]. Pro jednotlivé cesty je také možné nastavit filtrování požadavků pomocí komponenty anglicky zvané Middleware, která je popsána níže.



Obrázek 3.3: Schéma architektury frameworku Laravel. [4]

Komunikace s databází

Pro práci s databází se využívá objektivě relační mapování Eloquent⁶. Každá tabulka v databázi má svůj vlastní model, který s ní komunikuje. Kromě získávání dat tento model také umožňuje vkládání, úpravy a mazání záznamů. Model pro konkrétní tabulku se dá jednoduše vygenerovat pomocí konzole Artisan⁷ příkazem `php artisan make:model nazev_modelu`. Zároveň je možné spolu s modelem vygenerovat i Migraci tabulky⁸, a to přidáním přepínače `-m` nebo `-migration` do předešlého příkazu.

Migrace slouží k definici schématu databáze, které je možné jednoduše upravovat, například při spolupráci více programátorů. Po vytvoření všech souborů migrací se pomocí příkazu `php artisan migrate` vygenerují tabulky, které jsou v nich definované. Migrace se dá vygenerovat i samostatně, bez modelu, a to pomocí příkazu `php artisan make:migration nazev_migrace`.

Ke komunikaci s databází lze kromě objektivě relačního mapování Eloquent využít i Laravel Query Builder⁹. Ten obsahuje funkce, které umožňují sestavit dotazy v jazyce SQL¹⁰, a to i bez pokročilé znalosti tohoto jazyka. Porovnání přímého dotazu v jazyce SQL, využití Laravel Query Builderu a využití ORM Eloquent pro získání všech záznamů z tabulky `users` je naznačeno v ukázce 3.4. V této práci byl využit Eloquent, díky jednoduše použití a přímé návaznosti na modely jednotlivých entit. Oproti Query builderu není nutné pamatovat si názvy tabulek v databázi a stačí používat pouze název konkrétního modelu. Používání přímých dotazů v jazyce SQL není vhodné, jelikož nechrání proti napadení databázové vrstvy, které je popsáno níže.

```
SQL: SELECT * FROM 'users'  
Query Builder: DB::table('users')->get();  
Eloquent: User::All();
```

Obrázek 3.4: Porovnání SQL dotazu s Laravel Query Builder a Eloquent ORM

Middleware¹¹

Middleware zajišťuje kontrolu a filtrování požadavků protokolu HTTP, které přicházejí na aplikaci. Může být globální, který kontroluje všechny příchozí požadavky, nebo může být přiřazený jen k určité cestě, která je definována pomocí směrování. Jako příklad lze uvést Middleware, který kontroluje, zda je uživatel přihlášený. Pokud ano, požadavek je obslužen a uživatel je přesměrován na požadovanou stránku. Pokud ovšem není, požadavek je odmítnut a uživatel je přesměrován na přihlašovací obrazovku. Další z vlastností Middlewaru je například možnost zaznamenávání všech příchozích požadavků. Laravel obsahuje několik přednastavených pravidel Middleware a uživatel si může vytvořit libovolné množství nových podle toho, co zrovna potřebuje.

⁶<https://laravel.com/docs/8.x/eloquent>

⁷<https://laravel.com/docs/8.x/artisan>

⁸<https://laravel.com/docs/8.x/migrations>

⁹<https://laravel.com/docs/8.x/queries>

¹⁰SQL je zkratka pro standardizovaný strukturovaný dotazovací jazyk, který je používán pro práci s daty v relačních databázích.

¹¹<https://laravel.com/docs/8.x/middleware>

Frontend

Laravel používá šablonovací systém Blade. Ten využívá syntaxi značkovacího jazyka HTML, ale rozšiřuje ji o funkce a příkazy jazyka PHP. Díky tomu je možné v šabloně zobrazovat data přijatá z kontroleru a definovat podmínky a cykly.

Pro použití funkcí a příkazů jazyka PHP se používá klíčový znak `@` a funkce je ukončena příkazem začínajícím klíčovým slovem `@end`. Obsah proměnných se vypisuje zapsáním mezi dvojité složené závorky `{{navez_promenne}}`.

Funkce a příkazy, které nemají přímo podporu v syntaxi šablony Blade, lze volat dvěma způsoby. Pokud se jedná pouze o jednu funkci, stačí jí stejně jako proměnnou uzavřít do dvojitých složených závorek. Pokud chce programátor do šablony vložit delší úsek kódu v jazyce PHP, je nutné jej zapsat mezi klíčová slova `@php` a `@endphp`.

Pro psaní kaskádových stylů¹² byl využit preprocesorový skriptovací jazyk SASS¹³, který je do CSS později zkompileován. Tento jazyk rozšiřuje syntaxi jazyka CSS. Mezi jeho hlavní výhody patří například možnost využívání proměnných, zanořování stylů nebo dědičnost. Zapisuje se do souborů s příponou `.scss`. Pro kompilaci je využíván balík Laravel Mix¹⁴, který slouží jako nadstavba populárního modulu Webpack¹⁵. Ten kromě souborů jazyka SASS umožňuje zkompileovat i skripty napsané v jazyce Javascript, včetně knihoven VueJS¹⁶ a React¹⁷.

Zabezpečení

Framework Laravel implementuje základní zabezpečení systému, jako je ochrana proti Cross-site request forgery¹⁸ nebo SQL Injection¹⁹.

Proti útokům typu CSRF se Laravel brání pomocí CSRF tokenů. Tento token je vytvořen pro každého uživatele a je uložen v jeho relaci. Při každém odeslaném požadavku je tento token ověřen a tím je dokázáno, že požadavek opravdu odeslal daný uživatel. K tomuto ověření slouží middleware `VerifyCsrfToken`. Při každém novém vytvoření relace je token přegenerován a tím je zabráněno tomu, aby ho škodlivá aplikace mohla získat. Při vytváření nového formuláře v systému by měl programátor zahrnout neviditelné pole s hodnotou `{{ csrf_token }}`, pomocí kterého middleware ověří, že se shoduje s tím, který je uložený v relaci [9].

Proti SQL Injection Laravel chrání pouze při použití ORM Eloquent nebo Laravel Query. Využívá k tomu PDO parameter binding²⁰. PDO²¹ vytvoří šablonu pro dotaz v jazyce SQL, která ovšem neobsahuje žádná data, ale pouze proměnné, které je reprezentují. Tuto šablonu uloží a připraví pro pozdější odeslání. Data, která mají být uložena do databáze, jsou poté zasílána do PDO samostatně a až poté jsou vložena do příkazu v jazyce SQL. Tato data mají danou strukturu a díky tomu není možné místo některé z hodnot poslat škodlivý dotaz v jazyce SQL [10].

¹²<https://www.w3.org/Style/CSS/Overview.en.html>. Používá se anglická zkratka CSS

¹³<https://sass-lang.com/>

¹⁴<https://github.com/JeffreyWay/laravel-mix>

¹⁵<https://webpack.js.org/>

¹⁶<https://vuejs.org/>

¹⁷<https://reactjs.org/>

¹⁸Používá se anglická zkratka CSRF. <https://portswigger.net/web-security/csrf>

¹⁹<https://portswigger.net/web-security/sql-injection>

²⁰<https://webapplog.com/how-to-use-pdo-and-parameter-binding/>

²¹<https://www.php.net/manual/en/intro.pdo.php>

3.4 Databáze a ukládání dat

Pro správu databáze byl zvolen volně dostupný databázový systém MySQL²², jehož autorem je Michael Widenius [2].

V relační databázi je každý řádek označen jednoznačným identifikátorem, který se nazývá *klíč*. Tabulka obsahuje množinu atributů, které tvoří sloupce tabulky. Ty mohou nabývat různých hodnot. Každý atribut má určen svůj datový typ. Ten určuje, jaká data může sloupec obsahovat [14].

Na obrázku 3.5 je zobrazeno schéma databáze.

Hlavní částí je tabulka *users*. Ta uchovává informace o jednotlivých uživateli a jejich osobních informacích. Obsahuje dva atributy pro e-mail, jelikož uživatel se registruje s fakultním e-mailem a zároveň si v profilu může vyplnit svůj osobní e-mail.

Další důležitou částí je tabulka *posts*. Ta uchovává informace o jednotlivých příspěvcích. Atributy *upvotes* a *downvotes* udržují informaci o hodnocení příspěvku uživateli a jejich výchozí hodnota je nastavena na 0. Pokud se jedná o příspěvek importovaný ze sociální sítě Facebook, je jeho id uloženo v atributu *facebook_post_id*.

S tabulkou *posts* přímo souvisí tabulka *comments*. Ta udržuje informaci o komentářích. Obsahuje speciální atribut *parent_id*, který je vyplněn v případě, že se jedná o odpověď na komentář, a obsahuje id komentáře, ke kterému odpověď patří.

Pro udržování informace o předmětech slouží tabulka *courses*. Ta obsahuje zkratku předmětu, jeho název a také id jeho kalendáře Google. Pokud kalendář dosud nebyl vytvořen, obsahuje hodnotu *null*.

Pro uchovávání informací o fórech, které nesouvisí s předměty, slouží tabulka *forums*.

Informace o tématech, které dále dělí diskuze v předmětech i fórech, slouží tabulka *topics*. Ta má vyplněnou hodnotu id buď u atributu *course_id* nebo *forum_id*, a to podle toho, jestli patří k nějakému předmětu, nebo k fóru.

Pro uchovávání nahraných souborů slouží tabulka *file*. Ta kromě informací o názvu, typu a cestě k souboru obsahuje také několik značek. Značka *is_exam* slouží k označení souboru, který patří do kategorie zadání. Značka *is_su* slouží k označení souborů, které jsou přístupné členům studentské unie. Poslední kategorií jsou soubory určené pouze pro vedení studentské unie a pro ty slouží značka *is_su_private*.

Tabulka *contact_form* slouží k uchovávání přijatých formulářů z kontaktní stránky studentské unie. Tato tabulka není vázána na žádnou jinou tabulku, jelikož tento formulář není vázaný na konkrétního uživatele, ale k identifikaci slouží pouze vyplněné kontaktní údaje, aby mohl být odeslán anonymně.

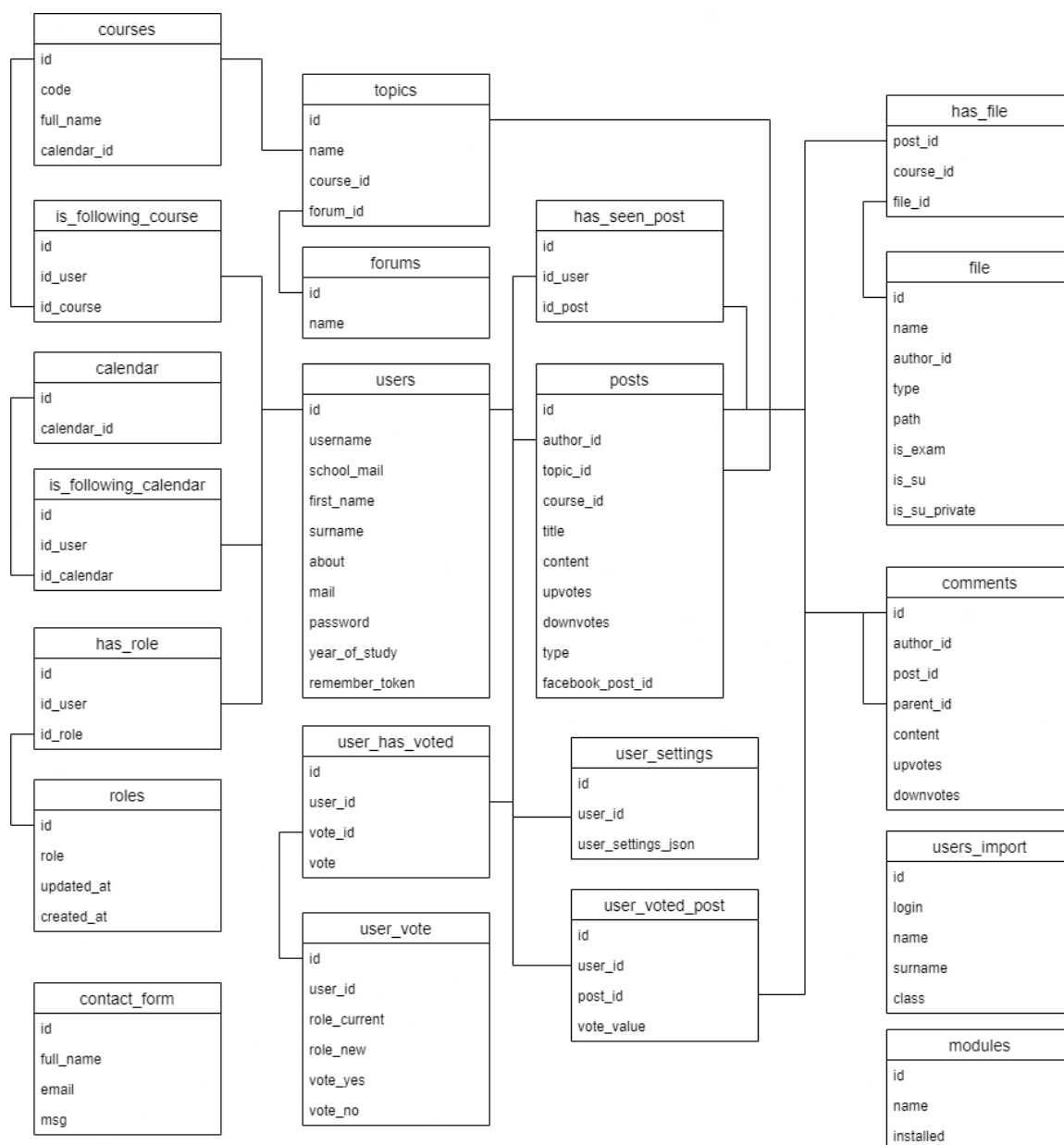
Tabulka *user_voted_post* slouží k uchovávání záznamů o tom, které příspěvky daný uživatel hodnotil a jak. To slouží především k zajištění toho, aby uživatel mohl hlasovat pouze jednou.

Pro uložení importovaných uživatelů slouží tabulka *users_import*. Ta obsahuje informace o všech osobách, které mají něco společného s fakultou, a slouží pro ověření při registraci.

Další hlavní tabulky jsou *roles* pro uchovávání rolí, *user_settings* pro uživatelská nastavení, *calendar* pro kalendáře a *modules* pro informace o rozšiřujících modulech.

Ostatní tabulky slouží jako vazební mezi hlavními tabulkami.

²²<https://www.mysql.com/>



Obrázek 3.5: Schéma databáze

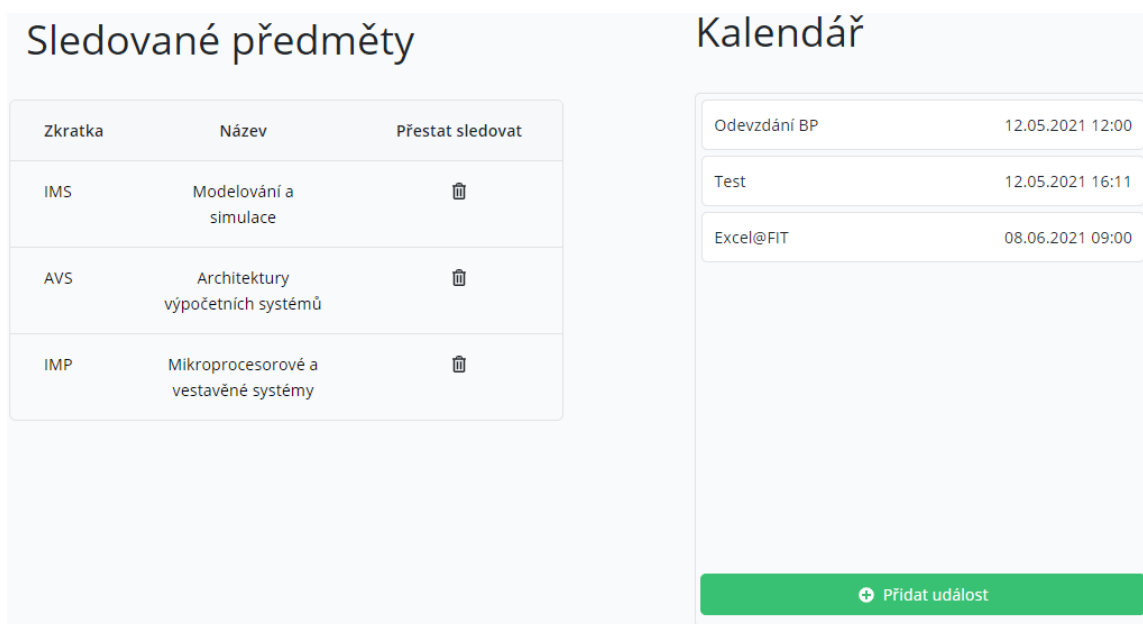
3.5 Uživatelské rozhraní

U návrhu uživatelského rozhraní je důležité myslet především na přehlednost a jednoduchost použití. Nevíme, jaké schopnosti ovládání webových aplikací bude mít uživatel, takže je dobré počítat spíše s horší variantou, že uživatel bude méně zdatný. Pokud se vyvíjí systém, který má být určen pro širokou veřejnost, tak je určitě lepší variantou vytvořit uživatelské rozhraní jednodušší a vyhovět tak méně i více zdatným uživatelům, než složitější, zaměřené pouze na pokročilé uživatele [5].

Tento systém však bude určen pro studenty informačních technologií na vysoké škole, od kterých se očekává spíše nadprůměrná schopnost ovládání webových aplikací. Díky tomu

je možné v některých ohledech dělat ústupky a zaměřit se spíše na funkčnost, než na použitelnost.

Hlavní částí návrhu bylo celkové rozložení stránky a umístění důležitých komponent tak, aby byly co nejvíce přehledné a uživatel nebyl zahlcen příliš velkým množstvím informací. První, co uživatel po přihlášení do systému uvidí, je domovská stránka. Ta by měla obsahovat stručný přehled důležitých informací. Na levé straně se tedy nachází seznam předmětů, které uživatel sleduje. Díky tomu se uživatel okamžitě dostane na předmět, který ho zajímá, aniž by ho musel hledat v seznamu. Na pravé straně se nachází hlavní kalendář s jednoduchým seznamem nadcházejících událostí. Uživatel tedy hned uvidí, jaké důležité události ho čekají. U obou přehledů bylo dbáno na co největší jednoduchost zobrazení, jak je vidět na obrázku 3.6.



Obrázek 3.6: Seznam sledovaných předmětů a kalendář na hlavní stránce

Další důležitou částí návrhu je stránka jednotlivých předmětů. K předmětu patří seznam témat a příspěvků, nahrané soubory, zadání, studentská skripta a kalendář. Na obrázku 3.7 je zobrazeno horní menu, které obsahuje několik ovládacích tlačítek. Hned vedle názvu předmětu se nachází tlačítko pro jeho sledování. Pod názvem se nachází čtyři tlačítka pro přepínání záložek. Toto přepínání nevyžaduje znovu načtení stránky a umožňuje uživateli téměř okamžitou změnu záložky. Pod nimi se nachází tlačítko pro vytvoření nového tématu a tabulka se samotným seznamem témat. U každého tématu je dále uvedeno, kolik příspěvků obsahuje.

V pravé části se nachází kalendář, který je zobrazen stejně jako na úvodní straně.

[IMS] Modelování a simulace ✔ Sledováno

Diskuze
Soubory
Zadání
Skripta

+ Nové téma

Název	Příspěvky
Projekt	3
Půlsemestrálka	0

Obrázek 3.7: Detail stránky předmětu

Po rozkliknutí některého z témat se zobrazí seznam příspěvků, které obsahuje. Seznam je zobrazen stejně jako u témat, jen navíc u každého příspěvku obsahuje typ příspěvku, jméno autora, datum vytvoření a počet komentářů. Pokud si uživatel bude chtít v rychlosti projít všechny příspěvky bez jejich otevírání, má možnost otevřít je v režimu rychlého náhledu. Pokud na příspěvek v seznamu klikne, objeví se pod ním jeho obsah a tlačítko *Otevřít příspěvek*, které ho odkáže na detail příspěvku. Pro ještě rychlejší prohlížení je možnost takto otevřít všechny příspěvky najednou a to pomocí tlačítka *Rozbalit vše*, které se nachází nad seznamem. Příklad rychlého zobrazení příspěvku přímo ze seznamu je ukázán na obrázku 3.8.

Název	Typ	Datum	Autor	Komentáře
Kdy bude opravený projekt?	Diskuze	2021-05-04 18:20:01	ms	🗨️ 0
Nevíte někdo kdy bude opravený projekt?				
				Otevřít příspěvek

Obrázek 3.8: Příklad rychlého zobrazení příspěvku přímo ze seznamu

Další důležitou částí uživatelského rozhraní jsou formuláře pro vytváření témat, událostí a sdílených souborů. Aby nemusel být uživatel pro vytvoření některého z výše uvedených objektů přesměrováván na samostatnou stránku, jsou formuláře vloženy do modálních dialogových oken. Takové okno se zobrazí okamžitě po kliknutí na tlačítko a překryje aktuální načtenou stránku. To umožňuje rychlé odeslání formuláře bez několikanásobného přesměrování. Příklad takového modálního dialogového okna je na obrázku 3.9.

Obrázek 3.9: Příklad modálního okna s formulářem pro vytvoření události do kalendáře

Stránkou, která bude také velmi často využívána, je seznam všech předmětů. Tento seznam by měl být co nejpřehlednější a měl by obsahovat možnost vyhledávání. Uživatel by měl mít možnost vyhledat předmět nejen podle jeho zkratky, ale také podle celého názvu, nebo jeho části. Nalezené předměty by se měly po vyhledání co nejrychleji zobrazit, aby uživatel zbytečně neztrácel čas, pokud například napíše název s chybou. Na obrázku 3.10 je ukázán příklad vyhledání všech předmětů, v jejichž názvu se nachází slovo *Angličtina*.

Seznam předmětů		Angličtina
Zkratka	Název	
BAN2	Angličtina 2: mírně pokročilí 2	
BAN3	Angličtina 3: středně pokročilí 1	
BAN4	Angličtina 4: středně pokročilí 2	
FCE	Angličtina: praktický kurz obchodní konverzace a prezentace	
JA6D	Angličtina pro doktorandy	
AEU	Angličtina pro Evropu	
AIT	Angličtina pro IT	

Obrázek 3.10: Příklad vyhledání předmětu podle části názvu

Kapitola 4

Implementace

4.1 Základ fóra

Základním stavebním kamenem celého systému je možnost vytváření diskuzí. Diskuze se dělí na několik úrovní. První úroveň má dva typy podle toho, jestli se jedná o diskuzi u konkrétního předmětu, nebo o obecnou diskuzi. Toto rozdělení je určeno v databázi v záznamu daného tématu. Pokud se jedná o diskuzi k předmětu, je ve sloupci *course_id* hodnota id předmětu, ke kterému diskuze patří, a sloupec *forum_id* obsahuje hodnotu null. Pokud jde o obecnou diskuzi, je ve sloupci *forum_id* hodnota id fóra, ke kterému téma patří, a sloupec *course_id* obsahuje hodnotu null.

Další úroveň je již zmíněné téma. To umožňuje oddělení skupin příspěvků a pomáhá přehlednosti celé diskuze.

Pod úrovní Tématu se již nacházejí jednotlivé příspěvky. V databázi je u každého příspěvku vyplněn údaj *topic_id*, který určuje, ke kterému tématu patří.

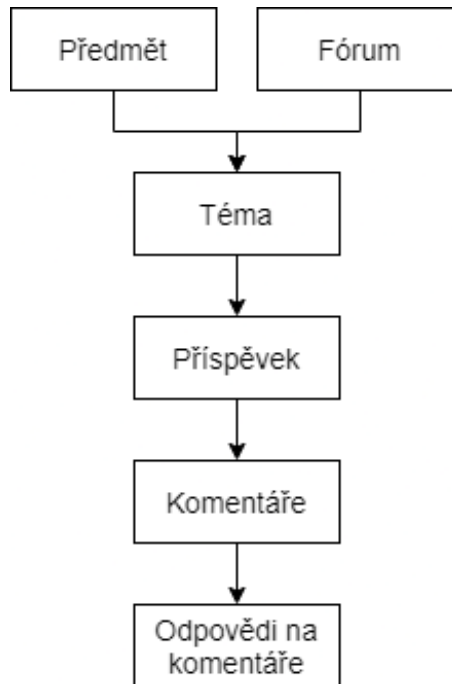
Poslední dvě úrovně jsou komentáře a odpovědi na komentáře. Komentář patří k nějakému konkrétnímu příspěvku a odpovědi poté k danému komentáři. V databázi mají společnou tabulku, jediným rozdílem je, že odpověď na komentář má vyplněn sloupec *parent_id*, který obsahuje id komentáře, ke kterému odpověď patří.

Nákres schématu diskuze je zobrazen na obrázku 4.1.

Pro přidání nového příspěvku slouží formulář, který se otevře na nové stránce. Ke každému příspěvku lze také nahrát libovolné množství souborů. Ty jsou poté zobrazeny přímo v detailu příspěvku a také v záložce *Soubory* v samostatné kategorii.

Také je možnost u příspěvku zvolit jeho typ. Na výběr jsou možnosti *Zadání*, *Materiály*, *Diskuze* a *Ostatní*. To slouží především ke zlepšení přehlednosti, jelikož je ze seznamu příspěvků hned jasné, jakého tématu se příspěvek týká.

Důležitou částí přidávání příspěvků je také pocit bezpečnosti studentů. Data, která jsou v systému sdílena, mohou být občas poměrně senzitivní, proto je nutné přidat možnost ochránit autora příspěvku, který taková data obsahuje. Jedná se například o zadání ke zkouškám, s jejichž zveřejňováním mají někteří vyučující na fakultě problém. To je vyřešeno možností zakliknout pole *Vložit příspěvek anonymně*. Příspěvek, který je vytvořen s touto zvolenou možností, má jako autora uveden účet *FITuška* a jeho skutečný autor je díky tomu skrytý.



Obrázek 4.1: Schéma úrovní diskuze

Seznam předmětů

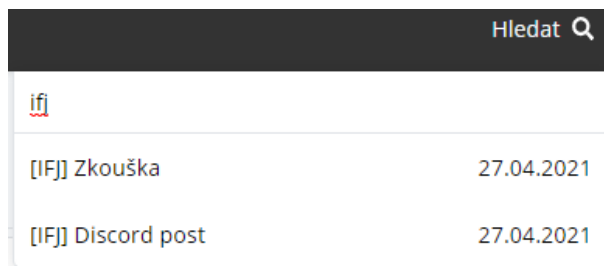
Pro jednoduché nalezení požadovaného předmětu, je implementován jejich seznam. Ten je vypisován pomocí technologie AJAX¹, která umožňuje dynamicky získávat data z databáze bez nutnosti znovu načítání stránky. Ve výchozím stavu jsou zobrazeny všechny předměty, které se v databázi nacházejí.

V tabulce je možné vyhledávat, a to pomocí vstupu nad tabulkou. Vyhledávání probíhá v reálném čase, to znamená, že systém reaguje na text napsaný do vstupu a okamžitě vrací výsledky. Například po zadání pouze jednoho písmena najde všechny předměty, které dané písmeno obsahují. Vyhledávat lze jak pomocí zkratky předmětu, tak podle jeho celého názvu.

Vyhledávání

V hlavním menu se nachází vyhledávání, které slouží k rychlému vyhledání příspěvků podle zadaného klíčového slova. Vyhledávání probíhá stejně jako v tabulce předmětů, tedy v reálném čase. Systém podle zadaného klíčového slova vrací příspěvky, které toto slovo obsahují v názvu, nebo přímo v textu. Další možností je vyhledání podle zkratky předmětu. Výhodou tohoto přístupu je jeho rychlost. Uživatel není přesměrováván na jiné stránky, dokud neklikne na požadovaný nalezený příspěvek. Na obrázku 4.2 je zobrazen příklad vyhledání příspěvků, které patří k předmětu IFJ.

¹<https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>



Obrázek 4.2: Příklad vyhledávání příspěvků

Sekce administrace

Do sekce administrace mají přístup pouze uživatelé s rolí Administrátor, nebo Vedení SU. Nachází se v ní několik funkcionalit, a to konkrétně správa modulů a hlasování o uživateli.

Správa modulů spočívá v možnosti nainstalovat a odinstalovat části systému, které závisí na nějaké externí službě. Jedná se o moduly Google API, Discord API a Facebook API. Pokud by některá z těchto služeb přestala fungovat, stačí jí jednoduše odinstalovat. Poté nebude v systému k dispozici a nedojde k neočekávaným chybám. Po obnovení její funkčnosti je možné ji znovu nainstalovat a dále využívat.

Hlasování o uživateli slouží k ověřování uživatelů s rolí Učitel a Doktorand. Tyto role mají zamezený přístup k některým částem systému. Administrátor nebo člen vedení studentské unie může vytvořit hlasování o ověření takového uživatele. Po vytvoření hlasování mohou členové vedení hlasovat. To lze pouze jednou a hlas již zpětně nelze změnit. Pokud všichni členové hlasují kladně, nebo po uplynutí 14 dnů od vytvoření hlasování alespoň 2 členové hlasovali pro a žádný proti, je uživatel ověřen a získá roli Ověřený učitel nebo Ověřený doktorand.

Souborový sklad

U každého předmětu je možnost nahrávání souborů do souborového skladu. Tento sklad je rozdělen na několik částí. První částí jsou soubory, které jsou vloženy přímo do příspěvků. Ty lze nalézt buď u daného příspěvku, nebo na stránce předmětu v záložce *Soubory* v kategorii *Soubory z příspěvků*. V této záložce se nachází i druhá část skladu, a to soubory nahrané přímo zde, které nepatří ke konkrétnímu příspěvku. Ty se nachází v kategorii *Ostatní soubory*. Dále se v této záložce nachází kategorie *Sdílené soubory*, která slouží k vytváření souborů na sdíleném disku společnosti Google a jejich implementace je popsána zvláště v kapitole 4.4. Poslední část se nachází v záložce *Zadání* a slouží, jak již vyplývá z názvu, k nahrávání souborů se zadáními k písemkám a zkouškám. Pro nahrávání souborů je využit souborový systém Laravelu². Ten umožňuje jednoduše ukládat soubory nahrané pomocí formuláře díky funkci *store*. Ta ve výchozím stavu pro nahraný soubor vygeneruje unikátní identifikátor, který použije jako jméno souboru pro uložení. Takový přístup není pro tento systém vhodný, jelikož by značně znepřehlednil hledání těchto souborů. Každému souboru je tedy nastaveno jméno podle toho, jak se jmenoval originální soubor. Toto jméno lze získat pomocí funkce *getClientOriginalName*. Pokud by již soubor s takovým jménem v databázi existoval, je ke jménu nového souboru přidána časová značka, která tyto soubory odliší.

²<https://laravel.com/docs/8.x/filesystem>

Sekce Studentská unie

Tato sekce poskytuje podporu pro vybrané činnosti Studentské unie. Hlavní součástí je možnost zaslání hromadných zpráv do skupin na sociální síti Facebook a do kanálů v aplikaci Discord. To může sloužit zejména pro sdílení důležitých informací, které jsou zaslány jménem FITušky a ne přímo pomocí účtu některého ze členů.

Další součástí je oddělený souborový sklad. Ten slouží k ukládání souborů, které mají být přístupné pouze členům studentské unie. Tento sklad je ještě dále rozdělen na dvě části. První část je přístupná pouze uživatelům s rolí *Vedení SU*. Druhá část je přístupná i uživatelům s rolí *Člen SU*. Díky tomu lze soubory rozdělit do dvou kategorií podle přístupu.

Další funkcionality slouží především ke kontaktování Studentské unie. Jedná se o seznam členů, který je opět rozdělen na klasické členy a vedení, a také o kontaktní formulář, přes který lze zaslat dotaz, který poté členové unie vidí v sekci *Formuláře*. Tento dotaz je zaslán částečně anonymně, jelikož není přiřazen k žádnému uživateli, ale obsahuje povinná pole pro kontaktní údaje, aby bylo možné na něj odpovědět.

Editor příspěvků

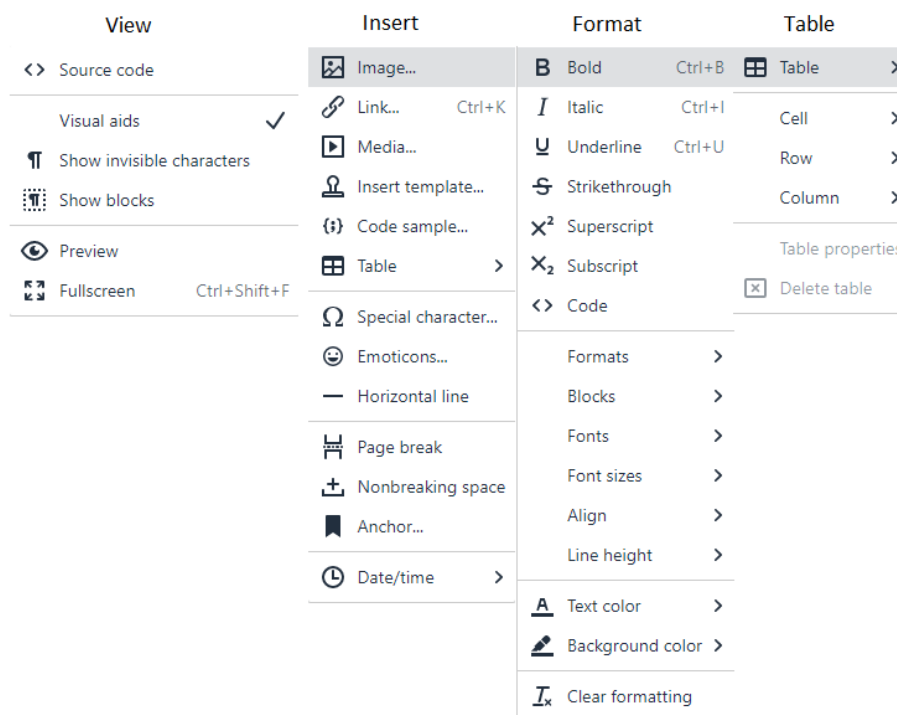
Pro editor příspěvků byl zvolen volně dostupný WYSIWYG [13] editor TinyMCE³. Ten umožňuje široké možnosti formátování psaného textu, vkládání obrázků nebo ukázek kódu. Díky jeho velké uživatelské základně existuje nespočet rozšíření, která do něj lze nainstalovat a rozšiřují ho o další funkce. Hlavní, tedy nejčastěji používané funkce, se nacházejí v okamžitě dostupném menu, jak je vidět na obrázku 4.3.



Obrázek 4.3: Menu hlavních funkcí editoru TinyMCE

Další funkce, které jsou méně používané, jsou skryté pod záložkami v horním menu. Na obrázku 4.4 můžeme vidět příklad funkcí pro záložky *View*, *Insert*, *Format* a *Table*.

³<https://www.tiny.cloud/>



Obrázek 4.4: Příklady dalších funkcí editoru TinyMCE

Dále editor podporuje také vkládání matematických vzorců. K jejich zpracování byl využit program \LaTeX . Podrobně je implementace popsána v kapitole 4.5.

4.2 Služba Facebook

Pro implementaci služby Facebook bylo využito rozhraní GraphAPI⁴, které slouží jako základní služba pro čtení a zapisování dat na tuto sociální síť. Ze služby Facebook byla implementována pouze funkcionalita získávání příspěvků a komentářů z ročníkových skupin. Ostatní funkcionality nemohly být implementovány, jelikož vyžadují oprávnění, která lze získat pouze po ověření aplikace (*App review*) společností Facebook. Toto ověřování je bohužel v době psaní této práce až do odvolání pozastaveno z důvodu probíhající pandemie. Funkcionalita, které závisí na těchto rozšířených oprávněních, bude nutné implementovat po obnovení ověřování aplikací. Z toho důvodu bylo implementováno rozšíření pro komunikaci s aplikací Discord, které je popsáno v kapitole 4.3.

Příspěvky z ročníkových skupin jsou získávány zasláním požadavku na konkrétní skupinu podle jejího jedinečného identifikátoru. K tomu slouží funkce `getPostsFromGroup()`, která jako parametr přijímá id skupiny, které odešle spolu s požadavkem. Facebook poté vrací odpověď ve formátu JSON⁵, která obsahuje všechny příspěvky z dané skupiny a komentáře k nim. V ukázce 4.5 je zobrazen příklad požadavku, zasílaného na Graph API. Proměnná `$group_id` obsahuje identifikátor facebookové skupiny a `$this->ACCESS_TOKEN` obsahuje přístupový řetězec vygenerovaný službou Facebook.

⁴<https://developers.facebook.com/docs/graph-api/>

⁵<https://www.json.org/json-en.html>


```
https://graph.facebook.com/'.$group_id.'/feed?
fields=message,comments&access_token='.$this->ACCESS_TOKEN
```

Obrázek 4.5: Příklad požadavku typu GET na Facebook Graph API

Dalším krokem je analýza jednotlivých příspěvků. K té slouží funkce *parsePost()*, která zkontroluje, zda se v obsahu příspěvku nachází značka v hranatých závorkách, podle které by se příspěvek dal přiřadit k některému z předmětů. Pokud se v příspěvku taková značka nenachází, je zahozen, jelikož nelze přesně určit, k jakému tématu patří. V opačném případě se značka předmětu uloží a později je podle ní příspěvek správně zařazen. U každého příspěvku je také uveden jeho jedinečný identifikátor, pomocí kterého dojde ke kontrole, zda se daný příspěvek už nachází v databázi. Pokud se v ní nenachází, je do ní, včetně komentářů, uložen. K procesu uložení příspěvků a komentářů slouží funkce *getFacebookPosts()* a *storeFacebookPosts()*. Zde přichází další omezení neověřené aplikace, a to takové, že u příspěvků není uveden jejich autor. Při ukládání takových příspěvků je tedy jako autor zvolen uživatel *FITuška*.

4.3 Discord robot

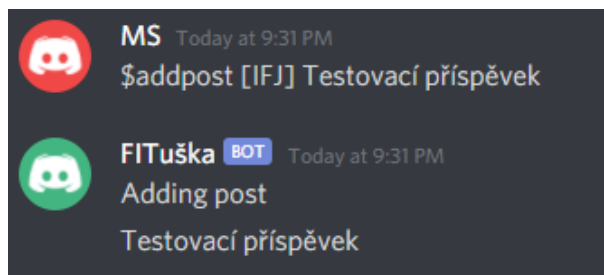
Discord robot je určen ke komunikaci s diskuzní aplikací Discord. Má za úkol preposílat příspěvky z nového systému do diskuzních kanálů v aplikaci a naopak získávat z aplikace příspěvky označené příslušnou značkou. Mezi další funkcionality patří získávání posledních a konkrétních příspěvků ze systému přímo v prostředí aplikace Discord.

Pro implementaci Discord robota byl zvolen programovací jazyk Python a knihovna *discord.py*. Tato knihovna poskytuje funkce pro komunikaci s Discord API⁶.

Robot je rozdělen na dvě části. První část umožňuje přistupovat k systému pomocí zadávání příkazů v aplikaci Discord. K tomu slouží příkazy *\$help* pro zobrazení nápovědy, *\$getposts zkratka_předmětu*, který vypíše názvy posledních 10 příspěvků k danému předmětu, *\$getpost id_příspěvku* pro zobrazení konkrétního příspěvku, *\$addpost zkratka_předmětu text* pro přidání příspěvku k příslušnému předmětu a *\$getuser id_uzivatele* pro zobrazení informací o uživateli. Nápovědu lze vyvolat označením robota pomocí *@FITuska*. Robot kontroluje zprávy zaslané v aplikaci a pokud některá z nich odpovídá přednastaveným příkazům, zašle požadavek na server, který jej zpracuje a zpět odešle požadovaná data. Robot je poté pošle jako zprávu na Discord.

Na obrázku 4.6 je ukázán příklad přidání příspěvku s textem *Testovací příspěvek* označeného zkratkou předmětu *IFJ*. Odesílány jsou pouze takto označené příspěvky, jelikož v místnostech jednotlivých předmětů přibývá obrovské množství příspěvků, u nichž z velké části není možné zjistit, k jakému tématu patří. Discord sice obsahuje funkcionality odpovídání na zprávy, ale uživatelé jí často nepoužívají, takže výsledná struktura přenesené diskuze by nedávala smysl, jelikož by postrádala část zpráv.

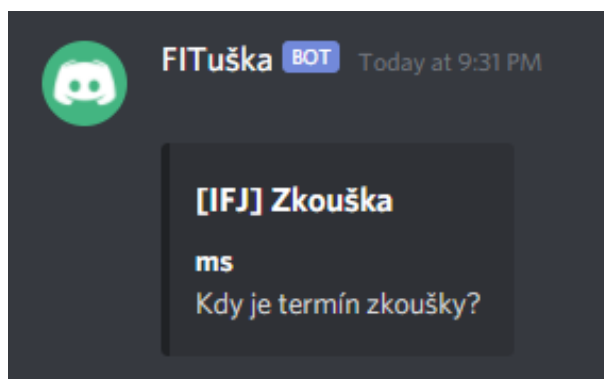
⁶API je ucelený soubor procedur, funkcí, protokolů a knihoven, které může programátor využít pro komunikaci s konkrétním systémem. Popis Discord API je dostupný na adrese <https://discord.com/developers/docs/intro>



Obrázek 4.6: Příklad přidání příspěvku pomocí robota

Druhá část umožňuje zasílat požadavky ze serveru na Discord robota. Obsahuje dvě funkcionality. První z nich je možnost zaslat hromadnou zprávu do vybraného kanálu v aplikaci. Tuto funkcionalitu mají k dispozici členové Vedení Studentské unie a mohou ji využít například k zaslání důležitého oznámení, které se tak dostane ke všem uživatelům Discordu. V systému k tomu slouží jednoduchý formulář s výběrem kanálu z aplikace Discord a polem pro zprávu. Po jeho odeslání tuto zprávu interpretuje robot na discord serveru ve zvoleném kanálu.

Druhou funkcionalitou je preposílání nově vytvořených příspěvků. Pokud někdo vytvoří nový příspěvek u některého z předmětů, bude automaticky zaslán i do příslušného kanálu na Discordu. Na obrázku 4.7 lze vidět příklad zprávy, kterou zaslal robot poté, co uživatel *ms* přidal v systému příspěvek k předmětu *IFJ*.



Obrázek 4.7: Příklad přidání příspěvku ze systému

4.4 Služby Google

Pro implementaci služeb Google byla využita knihovna `google-api-php-client`⁷, která umožňuje pracovat s API společnosti Google v jazyce PHP. V systému jsou implementovány služby Calendar API pro zprávu kalendářů a Drive API pro správu sdílených souborů. Hlavní logiku obstarává model **Google**. Kromě metod pro jednotlivé služby obsahuje také metodu `getGoogleClient()`, která je společná pro všechny dílčí služby a má na starosti inicializaci služby Google a přihlášení servisního účtu.

Servisní účet je speciálním druhem účtu, který nevyužívá žádná osoba, ale sám systém. Aplikace používá servisní účet k autorizovanému volání funkcí z API, během kterého je autorizován jako sám systém, nebo může převzít identitu nějakého reálného uživatele. Servisní

⁷<https://github.com/googleapis/google-api-php-client>

účet je identifikován pomocí e-mailové adresy, která je pro něj jedinečná a je vytvořena při registraci tohoto účtu. Takový účet nemá žádné jméno ani heslo a nedá se s ním přihlásit přes webový prohlížeč [12]. Servisní účet je využit z toho důvodu, aby systém mohl zpracovávat požadavky od uživatelů, aniž by po nich požadoval přihlášení do účtu Google. Díky tomu mají uživatelé možnost vytvářet sdílené soubory, nebo přidávat události do kalendáře.

Calendar API

Služba Google kalendář slouží pro správu kalendářů a událostí, které jsou sdílené pro všechny uživatele systému. Existuje hlavní kalendář, dostupný z domovské stránky systému, a poté jednotlivé kalendáře pro každý předmět. Tyto kalendáře se vytvářejí dvěma způsoby. Prvním je vytvoření při importu předmětu. V tomto případě ovšem nastává problém při importu většího množství předmětů, jelikož společnost Google omezuje počet zaslaných požadavků za určitý čas. Konkrétní hodnoty ovšem nejsou nikde v oficiálních zdrojích uvedené, takže jediným možným řešením je zpomalit zasílání požadavků, což ovšem při velkém množství předmětů není únosné. Kvůli tomuto omezení byla vytvořena druhá varianta. Ta vytvoří kalendář pro předmět až poté, co nějaký uživatel poprvé navštíví stránku daného předmětu. Díky tomu jsou vytvořeny pouze kalendáře pro předměty, které uživatelé v systému navštěvují. Vytvořené kalendáře jsou zároveň uchovávány v databázi systému.

Uživatel si kalendář může přidat do svého účtu Google. K tomu slouží tlačítko *Sledovat*, umístěné nad kalendářem. Kliknutí na toto tlačítko přidá e-mail do ACL⁸ a po potvrzení žádosti zaslané e-mailem je přidání dokončeno.

Přidávání nových událostí do kalendáře je obsluhováno metodou *AddEvent()*. Ta vytvoří novou událost se zadaným názvem, popisem a datem a vloží ji do konkrétního kalendáře.

Drive API

Služba Google drive slouží k vytváření a uchovávání sdílených dokumentů. Pro systém existuje jeden sdílený disk, který je spravován servisním účtem. Na tomto disku má každý předmět svou složku, pojmenovanou po zkratce tohoto předmětu, ve které se nacházejí soubory, které k němu přísluší.

U této služby, stejně jako u kalendářů, existuje omezení na počet zaslaných požadavků. Složka pro předmět je tedy vytvořena také až poté, co nějaký uživatel navštíví stránku daného předmětu. Zároveň se složkou je také vytvořen dokument s názvem *Skripta*, který slouží jako studentská skripta. Tento soubor se nachází na stránce předmětu v samostatné záložce *Skripta*, kde je vložen přímo do stránky jako *iframe*⁹, není tedy nutné ho otevírat v nové záložce přímo ve službě Google drive.

Seznam všech souborů, které se nacházejí ve složce pro daný předmět, se nachází v záložce *Soubory*, odkud je možné soubory otevřít. Vyhledávání souborů na disku Google se provádí pomocí dotazů. Prvním dotazem je *mimeType = 'application/vnd.google-apps.folder' and name contains zkratka_predmetu*, který vrátí všechny soubory typu *folder*, jejichž název obsahuje zkratku daného předmětu. Takový soubor se na disku nachází pouze jeden. Z odpovědi získáme id této složky a můžeme odeslat dotaz *id_slozky in parents*, který vrátí všechny soubory, které se v této složce nacházejí. Stejným způsobem probíhá také vyhledání souboru, který je určený pro studentská skripta.

⁸ACL (access control list) je seznam oprávnění pro připojení k nějakému objektu

⁹https://www.w3schools.com/tags/tag_iframe.ASP

Vytvoření souboru probíhá pomocí metody `createSharedFile()`. Ta nejdříve pomocí metody `getSharedFolderIdByCourse()` získá id složky, ve které má být soubor vytvořen. Poté soubor do dané složky vytvoří. Posledním krokem je nastavení práv souboru, aby k němu získali přístup všichni uživatelé. To probíhá zápisem role `write` pro uživatele `anyone` do ACL. Tím získají k souboru přístup s právem na čtení a zápis všichni uživatelé.

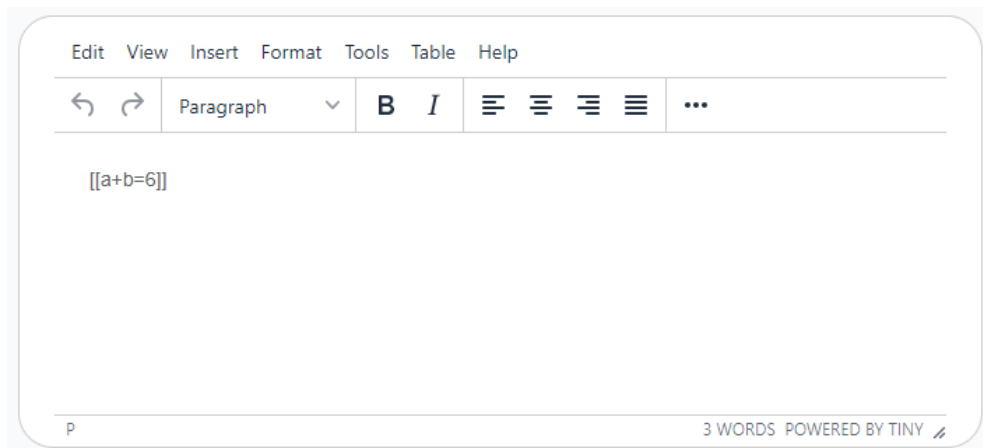
4.5 Služba L^AT_EX

Pro vkládání matematických vzorců do příspěvků a komentářů byl využit program L^AT_EX. Pokud chce uživatel do textu takový vzorec přidat, stačí, když ho označí z obou stran pomocí dvou hranatých závorek. Systém poté vezme obsah těchto závorek, zakóduje ho pomocí procentového kódování¹⁰ a výsledek vloží do parametru v URL serveru, který ho bude zpracovávat.

Požadavek na server je poté odeslán a jako odpověď se vrátí vygenerovaný obrázek se vzorcem ve formátu PNG. Ten je poté vložen do příspěvku, nebo komentáře. Pro zpracování požadavku byla použita služba `iTex2Img`¹¹.

Příklad adresy se zakódovanou rovnicí $a+b=6$ v parametru `eq`
[http://www.sciweavers.org/tex2img.php?](http://www.sciweavers.org/tex2img.php?eq=a%2Bb%3D6&fc=Black&im=png&fs=25&edit=0)
`eq=a%2Bb%3D6&fc=Black&im=png&fs=25&edit=0`

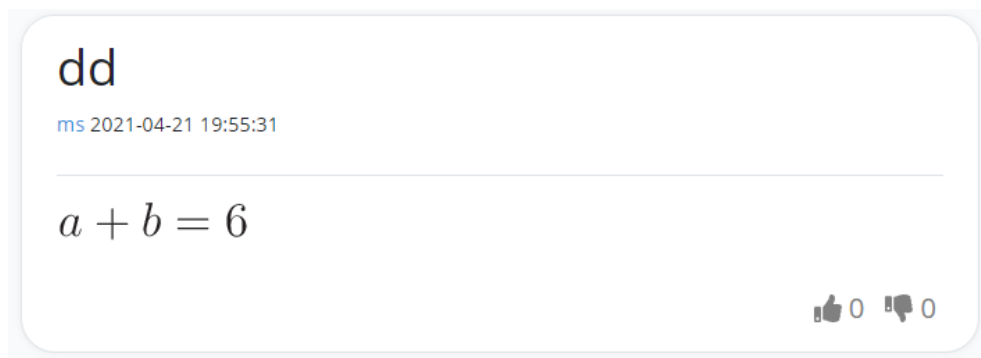
Využití serveru třetí strany bylo zvoleno díky jednoduchosti řešení. U příspěvků se počítá s vkládáním matematických vzorců, pro které je toto řešení dostačující. Pokud by systém měl podporovat rozsáhlejší struktury, bylo by nutné implementovat vlastní řešení.



Obrázek 4.8: Příklad zápisu vzorce v editoru příspěvku

¹⁰Procentové, neboli URL kódování je popsáno v RFC 3986, dostupném na <https://tools.ietf.org/html/rfc3986>

¹¹`iTex2Img` je online převodník z L^AT_EX do obrázku, dostupný na adrese <http://www.sciweavers.org/free-online-latex-equation-editor>



Obrázek 4.9: Výsledný vygenerovaný vzorec

4.6 Údržba systému

Údržba systému zahrnuje několik operací, které se spouštějí vždy po nějakém časovém období. Jedná se o import nových předmětů a studentů, vytvoření kalendářů, složek na disku Google a přidělení přístupů k nim. Tím je zajištěno, že každý nový školní rok systém obsahuje všechna potřebná data a je připraven k použití.

Import předmětů

Seznam předmětů pro aktuální rok je získán z tabulky, která se nachází na webových stránkách fakulty¹². Z této stránky je pomocí funkce `file_get_contents()` získán zdrojový kód stránky, ze kterého je poté pomocí regulárních výrazů získán seznam s názvy předmětů a jejich zkratkami. Pro každý nalezený předmět proběhne kontrola, jestli se již v databázi nenachází. To zajistí, že se importují pouze nově přidávané předměty.

V systému je také implementována služba, která pro všechny nově importované předměty vytvoří kalendáře a složky na sdíleném disku. Tato služba je ve výchozím stavu deaktivována, jelikož po její implementaci bylo zjištěno, že služba Google omezuje počet zaslaných požadavků za určitou dobu. Pokud dojde k překročení tohoto limitu, vrátí služba chybu *Calendar usage limits exceeded*. Poté je nutné počkat, než bude možné zasílat další požadavky. Tyto limity a časové údaje bohužel nejsou nikde uvedené. Z toho důvodu není možné tuto službu využít při importu velkého množství předmětů, a je tedy nahrazena jinou službou, která při navštívení stránky předmětu zkontroluje, zda je pro daný předmět vytvořen kalendář a složka na sdíleném disku. Pokud není, služba je vytvoří. Díky tomu nedochází k zasílání příliš mnoha požadavků na službu Google a nedojde tak k překročení zmíněných limitů.

Import uživatelů

Na začátku každého nového školního roku je také nutné provést import nových uživatelů. Ten se provádí z toho důvodu, aby se u nově registrovaného uživatele dalo ověřit, že je skutečně studentem, nebo zaměstnancem fakulty. Z toho důvodu se při registraci povinně vyplňuje také fakultní e-mail.

¹²<https://www.fit.vut.cz/study/courses/.cs>

Pro import uživatelů je využíván koncový bod ze serveru výzkumné skupiny KNOT¹³. Ten po zaslání dotazu vrací seznam všech osob, které mají nějakou spojitost s fakultou. U každé osoby je uveden login, neboli přihlašovací jméno, reálné jméno a příjmení a také zařazení. Zařazení je identifikátor, který se skládá z několika písmen a určuje, do jakého ročníku studia, výzkumné skupiny, nebo jiné součásti fakulty osoba patří. Z tohoto seznamu je poté vytvořeno pole osob, které je uloženo do databázové tabulky *users_import*.

Samotná kontrola poté probíhá během registrace uživatele. Systém zkontroluje zadaný fakultní e-mail, který se skládá z loginu a domény. Pokud se v databázi nachází osoba s loginem, který byl zadaný v e-mailu, je jí přiřazena role podle jejího zařazení. Pokud se v databázi nenachází, získá uživatel pouze roli *Registrovaný uživatel* a není mu umožněn přístup do systému.

¹³<https://www.fit.vut.cz/research/group/knot/.cs>

Kapitola 5

Testování a nasazení

První část této kapitoly se zabývá popisem testování implementovaného systému. Druhá část popisuje nasazení systému, včetně všech externích služeb.

5.1 Testování

Testování bylo rozděleno do několika fází. První fáze se skládala z testování během implementace, a to jak jednotlivých částí, tak systému jako celku. Druhou fází bylo testování případů užití pomocí vytvořených fiktivních uživatelů. Třetí fází bylo testování reálnými uživateli, které bylo zaměřeno především na použitelnost systému a otestování uživatelského rozhraní.

Testování během implementace

Během vývoje systému probíhalo pravidelné testování. Každá nová komponenta byla ihned po její implementaci otestována samostatně a na základě tohoto testování byly opraveny nalezené chyby. Po této fázi přišla na řadu integrace nové komponenty do systému. Poté proběhlo testování znovu, tentokrát již včetně komunikace se zbytkem systému. Díky tomu bylo ověřeno, že nově přidaná komponenta se systémem správně komunikuje a nedochází k žádným neočekávaným událostem.

Po dokončení implementace všech součástí byl systém znovu testován jako celek. Součástí testování byla také kontrola ošetření všech vstupů, která byla prováděna zadáváním velkého množství různých vstupních kombinací a parametrů a měla odhalit, zda je systém připraven na neplatné údaje.

Testování případů užití

K tomuto testování byli vytvořeni fiktivní uživatelé, kteří měli předem připravené scénáře, podle kterých bylo poté postupováno. Jednalo se jak o chování typického uživatele, které obsahuje především vytváření příspěvků, přidávání komentářů, nahrávání a vytváření souborů, tak i uživatele s vyšším oprávněním, kteří mají možnost měnit nastavení systému.

Příklad vytvořených scénářů:

1. Jan Novák, role Student
 - Přihlásit se

- Vyhledat předmět IFJ
 - Vytvořit nové téma
 - V novém tématu přidat nový příspěvek s příloženým souborem
 - Okomentovat nově vzniklý příspěvek
 - Smazat příspěvek
2. Petr Zelený, role Student, Vedení SU
- Zaslát hromadnou zprávu na Discord
 - Najít předmět IMS a vytvořit k němu nový sdílený soubor
 - Přidat událost do kalendáře
3. Tereza Červená, role Student, Moderátor
- Otevřít administraci
 - Odinstalovat modul Facebook api
 - Vytvořit hlasování o ověření uživatele

Během tohoto testování byly zapisovány veškeré neočekávané události. Po jeho skončení proběhla analýza těchto událostí a byly zapracovány opravy. Jednalo se především o chyby v nastavení oprávnění jednotlivých rolí a přidělování špatných rolí po registraci.

Testování reálnými uživateli

Poslední fáze sestávala z testování reálnými uživateli. Tito uživatelé dostali systém k vyzkoušení a jejich úkolem bylo zaznamenat veškeré chyby a neočekávané události. Také byla sbírána jejich zpětná vazba ohledně použitelnosti a přehlednosti uživatelského rozhraní.

Jejich hlavním úkolem bylo pokusit se používat systém tak, jako by ho používali během studia. To zahrnovalo především vytváření příspěvků, přidávání komentářů, sdílení souborů, vytváření událostí do kalendářů a procházení předmětů.

Všechny neočekávané události při komunikaci s databází byly během testování zaznamenávány samotným frameworkem Laravel v souboru *laravel.log*. Z jeho analýzy bylo zjištěno, že nedošlo k žádným závažným chybám.

Samotnými uživateli bylo nahlášeno několik chyb. Jednalo se zejména o chyby v nastavení přístupů pro různé kombinace rolí, kdy systém zamezil přístup do částí, které měly být přístupné. Většina uživatelů uvedla, že na žádnou neočekávanou událost nenarazila. Na základě analýzy těchto chyb byla poté provedena jejich oprava.

Samostatnou částí bylo testování uživatelského rozhraní. To bylo více specifické, jelikož ho každý uživatel vnímá trochu jinak. Většina uživatelů uvedla, že rozhraní je poměrně přehledné a intuitivní. Několik uživatelů uvedlo, že by uvítali více možností personalizace, jako je například přepnutí vzhledu do tmavé verze, nebo možnost určit si, co bude zobrazeno na hlavní stránce. Několikrát bylo také zmíněno pomalé načtení stránky předmětu při prvním přístupu. To může být zaviněno vytvářením kalendáře a sdílené složky, pokud tento předmět uživatel navštívil jako první, jelikož je nutné zaslat několik požadavků na servery společnosti Google a odezva někdy může trvat až několik vteřin. Toto ovšem postihne u každého předmětu pouze jednoho uživatele, který jeho stránku navštíví jako první. Při dalších návštěvách je již kalendář i sdílená složka k dispozici a načtení proběhne téměř okamžitě.

Na základě tohoto testování bylo provedeno několik drobných úprav a systém je připraven k nasazení.

5.2 Nasazení

Nasazení aplikace do provozu sestává z několika částí. První fází je nasazení samotného systému. U toho je nutné nejdříve připravit prostředí, kde systém poběží. Je tedy nutné nakonfigurovat webový server, například Apache¹ nebo Nginx². Na serveru musí být nainstalováno PHP verze 7.4 nebo vyšší. Dále je nutné nainstalovat databázovou službu MySQL ve verzi 5.6 nebo vyšší. Poslední důležitou aplikací je Composer³, který slouží ke správě závislostí jazyka PHP.

Pokud je toto všechno připraveno, může se přejít k samotnému spuštění systému. Prvním krokem je nakopírování zdrojových souborů na připravený server. Poté je nutné v kořenové složce projektu spustit instalaci závislostí pomocí programu Composer. Díky tomu se nainstalují všechny potřebné balíčky, které jsou definované v souboru *composer.json*.

Dalším krokem je vytvoření a nastavení databáze. Nejdříve si vytvoříme prázdnou databázi v databázovém systému MySQL. Po jejím vytvoření můžeme přistoupit k vytvoření tabulek, které je možné provést dvěma způsoby. Prvním z nich je import zálohy databáze, která je dostupná v kořenové složce projektu pod názvem *fituska.sql*. To zajistí vytvoření celé databáze s předem připravenými daty. Po vytvoření databáze je potřeba nastavit její přihlašovací údaje v systému. Ty se definují v souboru *.env*, který se nachází v kořenové složce projektu. Druhou možností je vytvoření tabulek pomocí migrací, které jsou definované ve složce */database/migrations*. Ty je možné provést spuštěním příkazu *migrate* z konzole artisan⁴ po nastavení přihlašovacích údajů ve výše zmíněném souboru..

Tímto je systém připraven k použití. Nyní je ovšem ještě nutné uvést do provozu všechny externí služby.

První takovou službou jsou kalendáře a sdílené soubory od společnosti Google. Zde je nutné vytvořit projekt v Google cloud platform⁵. Po jeho vytvoření je potřeba vytvořit přístupový klíč. Ten se poté vloží do souboru *Google.php*. Dalším krokem je povolení potřebných služeb. K tomu slouží tlačítko *ENABLE APIS AND SERVICES*. Zde je nutné přidat API pro kalendář a sdílený disk. Aby bylo možné službu využívat bez ověřování uživatelů, je potřeba vytvořit servisní účet a soubor s jeho údaji ve formátu *json* nahrát do kořenové složky projektu pod názvem *fituska_service_acc.json*. Tím jsou služby Google připraveny k použití.

Druhou službou je robot pro aplikaci Discord. K tomu je potřeba na webový server nainstalovat jazyk Python ve verzi 3.6 nebo vyšší. Poté je možné spustit oba skripty, které se nacházejí ve složce *fituska_bot*. Poté je nutné robota nainstalovat na vybraný server přímo v aplikaci Discord. To je možné provést pomocí přistupení na adresu⁶ pro autorizaci, kde se vybere server, na který chceme robota nainstalovat. V dalším kroku potvrdíme oprávnění a po potvrzení je robot nainstalován a připraven k použití.

Poslední službou je integrace se sociální sítí Facebook. Zde stačí nainstalovat již vytvořenou aplikaci *FITuška* na vybranou skupinu a služba je připravena k použití. Tuto instalaci momentálně může provést pouze vlastník aplikace, jelikož kvůli již dříve zmíněnému pozastavení ověřování aplikací, není tato aplikace k dispozici ve vyhledávání aplikací připravených pro instalaci.

¹<https://httpd.apache.org/>

²<https://www.nginx.com/>

³<https://getcomposer.org/>

⁴<https://laravel.com/docs/8.x/artisan>

⁵<https://console.cloud.google.com/apis/dashboard>

⁶https://discord.com/api/oauth2/authorize?client_id=812333296028876811&permissions=2147613760&scope=bot

Kapitola 6

Závěr

Cílem této práce bylo seznámit se s aktuálně využívanými platformami pro studentskou komunikaci, provést jejich analýzu a prostudovat API pro přístup k diskusím na sociální síti Facebook a k dokumentům a kalendářům od firmy Google. Na základě této analýzy měl poté vzniknout návrh a implementace nového systému, který může sloužit pro komunikaci studentů a zároveň spolupracovat i s ostatními platformami.

Byla provedena analýza platforem, které studenti používají nejčastěji. Mezi ně patří diskuzní aplikace Discord, sociální síť Facebook, sdílené dokumenty společnosti Google a také studentské fórum Fituška. Na základě této analýzy byl poté sestaven seznam požadavků na nový systém. Podle těchto požadavků vznikl návrh nového systému, podle kterého byl systém implementován.

Během vývoje systému došlo k neočekávané situaci, jelikož společnost Facebook z důvodu probíhající pandemie pozastavila ověřování aplikací na jejich síti. Kvůli tomu bylo možné implementovat pouze základní funkce, které nevyžadují vyšší oprávnění, podmíněné oficiálním schválením aplikace. Z toho důvodu byla navíc implementována integrace s aplikací Discord, která je popsána v kapitole 4.3.

Pro implementaci systému byl vybrán programovací jazyk PHP s frameworkem Laravel. Tato volba se v průběhu ukázala jako správná, jelikož tento framework obsahuje spoustu funkcionalit, které značně zjednodušují a zrychlují vývoj základního systému. Další výhodou je možnost jednoduchého nasazení, které nevyžaduje žádnou složitou konfiguraci serveru.

Cíle práce se podařilo naplnit a vznikl systém, který je schopný samostatně sloužit pro komunikaci studentů, bez větších nároků na údržbu. Obsahuje propojení se službami Google, jako jsou kalendáře a sdílené soubory. Dále umožňuje získávání příspěvků z ročníkových skupin na sociální síti Facebook a jejich ukládání do systému. Součástí je také robot, kterého je možné nainstalovat na server v aplikaci Discord a který umožňuje přeposílání zpráv z tohoto serveru do hlavního systému a naopak. Také umožňuje zasílání hromadných zpráv do vybraných kanálů, díky kterému je možné zasílat důležité informace. Systém i robota je možné ihned nasadit a používat v praxi.

Jako další možné rozšíření je možné implementovat přidání dalších funkcionalit v rámci propojení se sociální sítí Facebook, které by například umožnily zasílání hromadných zpráv do skupin, nebo přeposílání nově přidávaných příspěvků ze systému. To je ovšem podmíněno obnovením schvalovacího procesu a poté samotným schválením aplikace.

Literatura

- [1] CROFT, J. *Frameworks for Designers* [online]. Červen 2007 [cit. 2021-05-05]. Dostupné z: <https://alistapart.com/article/frameworksfordesigners/>.
- [2] DRAKE, M. *What is MySQL?* [online]. Prosinec 2020 [cit. 2021-04-26]. Dostupné z: <https://www.digitalocean.com/community/tutorials/what-is-mysql>.
- [3] FOWLER, M. *UML Distilled*. 3. vyd. Addison-Wesley Professional, září 2003. ISBN 978-0321193681.
- [4] INTERVIEWGURU. *LARAVEL INTERVIEW QUESTIONS* [online]. 2019 [cit. 2021-04-26]. Dostupné z: <https://interview-guru.club/laravel>.
- [5] KOHOUT, M. *Jak na tvorbu přívětivého uživatelského rozhraní?* [online]. Červen 2018 [cit. 2021-05-05]. Dostupné z: <https://freebit.cz/jak-na-tvorbu-privetiveho-uzivatelskeho-rozhrani-ui-user-interface/>.
- [6] KOČÍ, R. a KŘENA, B. *Úvod do Softwarového inženýrství - Analýza a specifikace požadavků* [online]. Brno: Fakulta informačních technologií VUT v Brně, říjen 2020 [cit. 2020-04-17]. Dostupné z: <https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FUIUS-IT%2Flectures%2FUIUS2.pdf&cid=12208>.
- [7] *Laravel Usage Statistics* [online]. 2021 [cit. 2021-05-09]. Dostupné z: <https://trends.builtwith.com/framework/Laravel>.
- [8] *MVC Tutorial for Beginners: What is, Architecture & Example* [online]. 2021 [cit. 2021-05-09]. Dostupné z: <https://www.guru99.com/mvc-tutorial.html>.
- [9] OTWELL, T. et al. *CSRF Protection* [online]. 2021 [cit. 2021-04-27]. Dostupné z: <https://laravel.com/docs/8.x/csrf>.
- [10] *PHP MySQL Prepared Statements* [online]. 2021 [cit. 2021-04-27]. Dostupné z: https://www.w3schools.com/php/php_mysql_prepared_statements.asp.
- [11] REES, D. *Laravel: Code Bright*. 1. vyd. Leanpub.com, říjen 2013. ISBN 978-1471777493.
- [12] *Service accounts* [online]. 2021 [cit. 2021-05-05]. Dostupné z: <https://cloud.google.com/iam/docs/service-accounts>.
- [13] ŠTRÁFELDA, J. *WYSIWYG editor* [online]. 2021 [cit. 2021-04-22]. Dostupné z: <https://www.strafelda.cz/wysiwyg>. Path: Štráfelda.cz, Know-how, Encyklopedie, WYSIWYG editor.

- [14] *Základní pojmy databáze* [online]. 2021 [cit. 2021-05-05]. Dostupné z:
<https://sites.google.com/site/xvinformatika/databaze/zakladni-pojmy-databaze>.