



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Informa ní systém pro podporu on-line dou ování
Student:	Patrik Neuzer
Vedoucí:	Mgr. Petr Matyáš
Studijní program:	Informatika
Studijní obor:	Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

- * Seznamte se se systémy na podporu individuálního vzd lávání.
- * Na základ p edchozí rešerše navrhnete a implementujete informa ní systém na podporu on-line dou ování, který bude um t:
 - správu student a lektor ,
 - plánování a správu lekcí,
 - po ítání cen a fakturaci,
 - podporu lekcí formou zadávání p íklad z databáze,
 - správu studijních materiál ,
 - propojení lektora a studenta prost ednictvím videop enosu (nap . pomocí programu Skype), vlastní videop enos nebude sou ástí této aplikace.
- * Veškeré softwarové výstupy d kladn otestujte.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 5. prosince 2016

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalárska práca

Informačný systém na podporu online doučovania

Patrik Neuzer

Vedúci práce: Mgr. Petr Matyáš

5. mája 2017

Prehlásenie

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

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov. V súlade s ustanovením § 46 odst. 6 tohoto zákona týmto udeľujem bezvýhradné oprávnenie (licenciu) k užívaniu tejto mojej práce, a to vrátane všetkých počítačových programov, ktoré sú jej súčasťou alebo prílohou a tiež všetkej ich dokumentácie (ďalej len „Dielo“), a to všetkým osobám, ktoré si prajú Dielo užívať. Tieto osoby sú oprávnené Dielo používať akýmkoľvek spôsobom, ktorý neznižuje hodnotu Diela (vrátane komerčného využitia). Toto oprávnenie je časovo, územne a množstevne neobmedzené.

V Prahe 5. mája 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Patrik Neuzer. Všetky práva vyhradené.

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

Odkaz na túto prácu

Neuzer, Patrik. *Informačný systém na podporu online doučovania*. Bakalárska práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Táto bakalárska práca sa zaoberá predovšetkým analýzou, návrhom a realizáciou informačného systému na podporu online doučovania a individuálneho vzdelávania. Na základe požiadaviek zadávateľa bol navrhnutý systém, ktorý umožňuje svojim používateľom správu jednotlivých používateľov (lektorov a študentov), plánovanie a správu lekcí, podporu týchto lekcí formou zadávania domácich úloh a príkladov z databázy, ďalej dovoľuje správu študijných materiálov, tiež počítanie cien a určitú formu fakturácie. To všetko na základe rozličných používateľských práv. Informačný systém navyše spolupracuje s aplikáciou Google kalendár. Samotná aplikácia pritom funguje na PHP s využitím Nette frameworku na pozadí a s využitím Bootstrap frameworku na front-ende. Záver práce je venovaný procesu testovania aplikácie.

Kľúčové slová webový portál, informačný systém, analýza, návrh, online doučovanie, individuálne vzdelávanie, PHP, Nette framework

Abstract

The object of this bachelor thesis is analysis, design and implementation of the information system supporting online tutoring and individual education. According to requirements, the information system was designed to allow its users to manage other users (lecturers and students), to plan and manage lectures and lessons, to support these lessons by assigning homeworks and problems from database, to manage study materials and to count prices with some form of billing process. All of these processes are based on different user privileges. Moreover, information system is cooperating with Google Calendar application. The application itself runs on PHP using Nette framework in the background and using Bootstrap framework on the front end. The final part of the thesis is dedicated to the process of testing the application.

Keywords web portal, information system, analysis, design, online tutoring, individual education, PHP, Nette framework

Obsah

Úvod	1
1 Cieľ práce	3
2 Úvod do problematiky	5
2.1 Základné pojmy	5
2.2 Súčasný stav riešenia problematiky	6
3 Analýza a špecifikácia požiadaviek	9
3.1 Funkčné a nefunkčné požiadavky	10
3.2 Modelovanie prípadov použitia	13
4 Softvérová architektúra a návrh	33
4.1 Voľba implementačného jazyka a frameworkov	33
4.2 Softvérová architektúra	34
4.3 Ukladanie dát	35
4.4 Wireframe	49
5 Realizácia	51
5.1 Použité nástroje	51
5.2 Ukážky kódu	55
5.3 Ukážky obrazoviek aplikácie	61
6 Testovanie	65
6.1 Lokálne testovanie	65
6.2 Testovanie prípadov použitia podľa scenárov	65
6.3 Používateľské testovanie	65
6.4 Testy vo webových prehliadačoch	66
Záver	67

Literatúra	69
A Zoznam použitých skratiek	71
B Obsah priloženého CD	73

Zoznam obrázkov

3.1	Účastníci	14
3.2	UML diagram prípadov použitia pre správu študentov a lektorov .	15
3.3	UML diagram prípadov použitia pre správu lekcií	19
3.4	UML diagram prípadov použitia pre správu konzultačných hodín .	22
3.5	UML diagram prípadov použitia pre správu predmetov a tém . . .	26
3.6	UML diagram prípadov použitia pre správu príkladov	27
3.7	UML diagram prípadov použitia pre správu študijných materiálov	30
4.1	Diagram popisujúci spoluprácu vrstiev MVC architektúry	34
4.2	Doporučená adresárová štruktúra Nette projektu	36
4.3	Databázový model	38
4.4	Wireframe pre vyhľadávanie a zobrazenie príkladov	50
5.1	Datagrid pre aktiváciu používateľských účtov	52
5.2	TreeView komponenta pre zobrazenie stromu predmetov a tém . .	53
5.3	Príklad interaktívnej Select2 komponenty	54
5.4	Zobrazenie chybovej hlášky s využitím SweetAlert pluginu	54
5.5	Zobrazenie upozornenia s textovým polom s využitím SweetAlert .	55
5.6	Obrazovka pre zobrazenie zoznamu konzultačných hodín	62
5.7	Obrazovka pre zobrazenie konzultačnej hodiny	63
5.8	Registračná komponenta	64
5.9	Komponenta na zmenu dostupnosti lektora	64

Zoznam tabuliek

4.1	Atribúty databázovej tabuľky <i>address</i>	37
4.2	Atribúty databázovej tabuľky <i>answer</i>	39
4.3	Atribúty databázovej tabuľky <i>attachment</i>	39
4.4	Atribúty databázovej tabuľky <i>contact_info</i>	40
4.5	Atribúty databázovej tabuľky <i>homework</i>	40
4.6	Atribúty databázovej tabuľky <i>homework_has_attachment</i>	40
4.7	Atribúty databázovej tabuľky <i>homework_has_question</i>	41
4.8	Atribúty databázovej tabuľky <i>homework_state</i>	42
4.9	Záznamy v databázovej tabuľke <i>homework_state</i>	42
4.10	Atribúty databázovej tabuľky <i>lecturer</i>	42
4.11	Atribúty databázovej tabuľky <i>lecturer_availability</i>	42
4.12	Atribúty databázovej tabuľky <i>lesson</i>	43
4.13	Atribúty databázovej tabuľky <i>lesson_has_attachment</i>	43
4.14	Atribúty databázovej tabuľky <i>lesson_has_question</i>	44
4.15	Atribúty databázovej tabuľky <i>lesson_state</i>	44
4.16	Záznamy v databázovej tabuľke <i>lesson_state</i>	44
4.17	Atribúty databázovej tabuľky <i>payment_record</i>	45
4.18	Atribúty databázovej tabuľky <i>question</i>	45
4.19	Atribúty databázovej tabuľky <i>question_has_attachment</i>	46
4.20	Atribúty databázovej tabuľky <i>question_has_topic</i>	46
4.21	Atribúty databázovej tabuľky <i>single_lecture</i>	46
4.22	Atribúty databázovej tabuľky <i>single_lecture_price</i>	47
4.23	Atribúty databázovej tabuľky <i>subject</i>	47
4.24	Atribúty databázovej tabuľky <i>subject_price</i>	48
4.25	Atribúty databázovej tabuľky <i>topic</i>	48
4.26	Atribúty databázovej tabuľky <i>topic_has_single_lecture</i>	49
4.27	Atribúty databázovej tabuľky <i>user</i>	49

Úvod

Pravdepodobne každý žiak či študent už pri učení niekedy narazil na problém, ktorý sa mu nedarilo vyriešiť. Vtedy takýto človek hľadá riešenie buď v knihách alebo v informáciách zverejnených na internete. Lenže ani to niekedy nestačí a preto je vhodné rozmýšľať nad doučovaním.

Technológie sa však už dávno posunuli výrazne dopredu, a tak v súčasnosti nie je problém nechať sa doučovať z pohodlia domova, prostredníctvom internetu. Online doučovanie však môže byť veľmi užitočné aj pre učiteľov. Tí si môžu takouto formou začať privyrábať a pritom nemusí byť problém, že sú od svojho študenta vzdialení aj tisíce kilometrov.

Doučovanie online sa oprávnene stáva trendom a s ním vzrastá aj požiadavka na podporu doučovania rozsiahlymi informačnými systémami, ktoré môžu výrazne pomôcť študentom, lektorom aj celým školám.

Tému som si zvolil, pretože informačné systémy sú dnes veľmi dôležitou súčasťou každodenného života, mnohokrát až nevyhnutnosťou pri spracovaní údajov najrôznejšieho charakteru. Nejde pritom len o podporu komplexných biznis procesov. Často nájdú informačné systémy využitie aj pri bežných činnostiach, čo rozhodne platí aj pre doučovanie. A napriek tomu, že online doučovanie sa stáva postupne celosvetovým trendom, v Českej republike a na Slovensku zatiaľ nie je tak rozšírené.

Ciel' práce

Cielom rešeršnej časti bakalárskej práce bude analýza existujúcich riešení v oblasti online doučovacích informačných systémov a nástrojov na podporu individuálneho vzdelávania. Okrem toho bude cieľom teoretickej časti analýza požiadaviek zo strany zadávateľa, pričom ide predovšetkým o proces správy študentov a lektorov, proces plánovania a správy lekcií, proces počítania cien a fakturácie a proces správy študijných materiálov.

Cielom praktickej časti práce bude návrh a implementácia samotnej webovej aplikácie, tiež otestovanie tejto aplikácie. Okrem toho bude cieľom praktickej časti popísať a analyzovať konkrétne riešenia použité pri implementácii najdôležitejších alebo najproblematickejších častí vývoja aplikácie.

Úvod do problematiky

Na začiatok je vhodné definovať si základné pojmy súvisiace s problematikou budúceho informačného systému. V tejto kapitole tak sú zadané pojmy „informačný systém“ a „online doučovanie“, pretože sú predmetom celej práce. Kapitola sa ďalej zaoberá súčasným stavom problematiky na trhu systémov podporujúcich online doučovanie a individuálne vzdelávanie.

2.1 Základné pojmy

Informačný systém

V súčasnosti je celý svet preplnený informáciami najrôznejšieho charakteru, to platí určite aj v prípade internetu. S nárastom množstva informácií ale vzrastá aj požiadavka na prehľadnosť a spracovanie týchto informácií tak, aby cieľovej skupine používateľov boli tieto informácie prístupné čo najrýchlejšie, v jednoduchšej a ľahko čitateľnej forme. Práve kvôli tomu vznikli informačné systémy. Tie sa postupne uplatňujú v každej rovine života, či už ide o oblasť zdravotníctva, bankovníctva alebo vzdelávania.

Podľa [1] môžeme informačný systém (IS) definovať ako „súbor ľudí, technických prostriedkov a metód (programov) zabezpečujúcich zber, prenos, spracovanie a uchovanie dát za účelom prezentácie informácií pre potreby používateľov činných v systéme riadenia“. To je však definícia z pohľadu využívania technológií a dát. Na informačné systémy je nutné pozerať sa aj z perspektívy podniku, pretože informačné systémy by mali podporovať vznik zisku daného podniku a tým pádom aj zvyšovanie celkovej hodnoty podniku.

Online doučovanie

Pod pojmom „doučovanie“ si asi každý predstaví istý spôsob vzdelávania, v ktorom si študent dodatočne doplní vedomosti formou vyučovacích hodín so svojím učiteľom. Pojem „online doučovanie“ je však pomerne nový a nie

je presne špecifikovaný. Môžeme si pod tým predstaviť formu doučovania, pri ktorej učiteľ a študent nie sú počas vyučovacej hodiny priamo prítomní na jednom mieste, ale komunikujú spolu napríklad prostredníctvom videohovoru. Tak rozumieme tento pojem aj pre prípad tejto práce. Niektoré zdroje však uvádzajú, že „online doučovanie“ je takou formou vzdelávania, pri ktorej nemusí byť prítomný lektor. Ide tak o istú formu e-learningu, tým sa však táto práca nezaoberá.

2.2 Súčasný stav riešenia problematiky

V súčasnosti sa tematikou individuálneho online doučovania zaoberá množstvo webových portálov. Väčšina z nich je však dostupná len v anglickom jazyku a i tak obsahujú množstvo nedostatkov, respektíve nezodpovedajú požiadavkam zadávateľa.

Analýza konkurenčných aplikácií sa zameriava hlavne na tie webové aplikácie, ktoré sú zdarma pri registrácii. Za zmienku tak stoja anglické webové portály TutorHub a BuddySchool alebo český projekt Mathematicator.

2.2.1 TutorHub

TutorHub [2] je britský doučovací systém, ktorý funguje už od roku 2010.

Základ portálu je tvorený vyhľadávaním a filtrovaním lektorov. Každý lektor má stanovenú svoju cenu na hodinu a je možné ho kontaktovať. Súčasťou aplikácie je možnosť interaktívnej komunikácie medzi lektorom a študentom prostredníctvom správ a ukladanie tejto konverzácie. Okrem toho je tiež možné v rámci jednej hodiny zdieľať študijných materiálov.

Čo sa týka platieb, je TutorHub tzv. Pay As You Go službou, čo v tomto prípade znamená, že študent si dopĺňa do svojho účtu kredity, ktoré neskôr využíva na platbu za rezerváciu lekcí.

Bonusom aplikácie je tzv. Q&A hub, ktorý funguje ako fórum, v ktorom študenti zadávajú svoje otázky z najrôznejších tém, učitelia na ne môžu odpovedať a diskutovať svoje riešenia spolu so študentami.

2.2.2 BuddySchool

BuddySchool [3] je ďalší nástroj na podporu individuálneho vzdelávania. Založený bol v roku 2007.

Myšlienka tohto portálu je trochu iná v porovnaní s vyššie spomenutou aplikáciou. Všetky lekcie lektorov sú tentokrát rozdelené do predmetov, tém a podtém, pričom jeden lektor má možnosť vytvoriť viacero lekcí pre rôzne predmety a témy.

Aplikácia však neumožňuje priame prepojenie medzi lektorom a študentom. Ak chce študent požiadať o lekciiu, má možnosť buď upozorniť na seba „šŕuchnutím“ (v tom prípade sa lektor študentovi sám ozve) alebo má študent

možnosť priamo zarezervovať si hodinu na špecifikovaný čas daný harmonogramom lektora. Ten si lektor nastavuje priamo vo svojom profile.

Princíp platieb je takmer totožný s aplikáciou TutorHub a znova prebieha prostredníctvom kreditov.

2.2.3 Mathematicator

Mathematicator [4] je český webový portál, ktorý slúži okrem iného aj na podporu online doučovania.

V porovnaní s predchádzajúcimi nástrojmi je špecializovaný na matematiku a logické hlavolamy. Študenti majú znova možnosť zaregistrovať sa na konkrétnu lekciu, avšak kontaktná výuka vždy prebieha s jediným lektorom – autorom projektu. Ten má pevne stanovenú cenu za hodinu doučovania a je možné ho kontaktovať prostredníctvom emailu, a tak si dohodnúť lekciu.

Platby prebiehajú prevodom peňazí na účet pred prebehnutím danej hodiny. Je však možné využiť aj osobné doučovanie, kedy platba prebieha až po prebehnutí lekcie. Mathematicator tak nie je čistým nástrojom, ak hovoríme o online doučovaní, ale akousi hybridnou aplikáciou.

Bonusom tejto aplikácie sú videotutoriály, ktoré sa zaoberajú nielen matematickou teóriou, ale aj príkladmi na témy z najrôznejších oblastí matematiky.

2.2.4 Zhodnotenie súčasného stavu

Zadávatel plánuje mať v systéme zaregistrovaných svojich zamestnancov (lektorov), ktorí budú poskytovať svoje vedomosti študentom. Je tak potrebné mať prehľad o svojich zamestnancoch a o ich vyučovaných hodinách. Študenti si pritom budú platiť za vyučovacie hodiny lektorov, čo je potrebné tiež evidovať. Detailnejšia špecifikácia požiadaviek je však uvedená v kapitole 3, týkajúcej sa analýzy.

Z vyššie uvedeného je tak jasné, prečo uvedené existujúce systémy nespĺňajú požiadavky a prečo je nutné vyvinúť úplne nový informačný systém na mieru. Spomenutými aplikáciami je ale možné sa v menšej miere inšpirovať.

Vzhľadom na to, že študenti si budú za konzultačné hodiny platiť, systém platieb v kreditoch je pravdepodobne najjednoduchšou a najrozumnejšou variantou pri informačných systémoch ako je ten, ktorý je predmetom tejto práce. Takáto forma platieb je pre používateľov systému najrozumiteľnejšia a správca systému si pritom môže zvoliť kurz medzi hodnotou kreditu a skutočnou menou. Štandardne býva jeden kredit rovný jednotke meny, teda napríklad 1 kredit = 1 česká koruna. Správca systému má však voľnosť pri voľbe tohto kurzu, čo môže byť výhodou pri pôsobení na rozličných trhoch. Napríklad je možné, že aplikácia bude mať zaregistrovaných českých aj slovenských používateľov zároveň. Je tak nutné, aby nedošlo k nedorozumeniu spôsobenému používaním rozličných mien.

2. ÚVOD DO PROBLEMATIKY

Zo spomenutých aplikácií je ďalej potrebné vyzdvihnúť princíp rezervácií na konzultačné hodiny v aplikácii BuddySchool. Výhodou takéhoto princípu je ten, že si lektor sám dokáže určiť, kedy chce byť k dispozícii pre doučovanie a študent si rovnako vie vytvoriť rezerváciu na čas, ktorý mu najlepšie vyhovuje. Táto voľnosť pri výbere časov doučovania môže byť ďalšou výhodou pre používateľov.

Naopak, najväčšou nevýhodou spomenutých aplikácií, konkrétne v aplikácii TutorHub, môže byť to, že študenti si platia za čas lektora, nie za čas lekcie lektora. V takomto prípade je nutné, aby učiteľ vyučoval len jednu konkrétnu tému a v prípade, že by učiteľ chcel doučovať z viacerých tém, musel by si vytvoriť nový používateľský účet. To môže byť veľmi obmedzujúce pre lektorov a môže ich to odradiť od používania systému. Druhou možnosťou je, že si lektor všetky informácie o všetkých vyučovaných témach zadá do svojho profilu. To však môže byť nakoniec neprehľadné a vyhľadávanie lekcí veľmi chaotické. Lepšou možnosťou by tak bolo, aby si učitelia mohli vytvárať viacero lekcí z rozličných predmetov a tém.

Analýza a špecifikácia požiadaviek

Ako uvádza [5, s. 78], požiadavka je „špecifikáciou toho, čo by malo byť implementované“. Požiadavky by tak mali byť základom každého informačného systému.

V analýze požiadaviek sa zaoberáme tým, čo by systém mal dokázať. Bez toho, aby sme sa zamýšľali nad tým, akým spôsobom to zaistíme. To je až cieľom návrhu systému, ktorým sa táto práca zaoberá v kapitole 4. Cieľom analýzy požiadaviek je vyjasniť si zadanie so zákazníkom a tým určiť hranice systému vrátane všetkých obmedzení, ktoré sú na systém kladené. Okrem toho by mal byť cieľom analýzy požiadaviek aj odhad rozsahu vytváraného systému, pritom ide nielen o odhad času, ale aj o odhad pracnosti. Je to dôležité, pretože rozsah môže byť jediným parametrom ceny budúceho projektu počas rozhodovania o tom, či sa do projektu ako dodávateľ pustíme.

Množina všetkých takto definovaných požiadaviek na systém sa často nazýva špecifikácia systémových požiadaviek (SRS – System Requirements Specification). A vzhľadom na to, že SRS tvorí často základ dokumentácie k dodávanému softvéru a je tiež podkladom pre akceptačné testovanie, existujú určité odporúčenia pre to, ako má táto špecifikácia vyzeráť. Ďalší dôvod, prečo je žiadúce písať SRS správne je ten, že týmto procesom sa snažíme efektívne podporovať aj nasledujúce činnosti softvérového procesu ako je návrh, samotná implementácia či testovanie. Napríklad podľa normy IEEE 830 [6] by mali mať požiadavky nasledujúce vlastnosti:

- **Korektnosť:** SRS by mala správne popisovať správanie systému. Je nevhodné popisovať požiadavky, ktoré je nemožné splniť alebo požiadavky, ktorých vykonanie je veľmi nepravdepodobné.
- **Jednoznačnosť:** Požiadavky by mali mať jasný význam a nemali by byť predmetom rôznych interpretácií. Je tak vhodné zvoliť taký jazyk, aby všetky zodpovedajúce strany rozumeli všetkým požiadavkam.

- **Kompletnosť:** Dokument popisujúci požiadavky by mal úplne popísať očakávané chovanie systému a sadu funkcií.
- **Konzistentnosť:** Požiadavky by si nemali odporovať.
- **Hodnosť:** Požiadavky by mali byť ohodnotené prioritou.
- **Overiteľnosť:** Požiadavky by mali byť otestovateľné. Ak nie je možné povedať, či je požiadavka splnená alebo nie, je špecifikácia popísaná zle.
- **Modifikovateľnosť:** Požiadavky by malo byť možné upraviť počas samotného vývoja.
- **Trasovateľnosť:** Požiadavky by mali byť označené tak, aby bolo v dokumentácii jednoduché nájsť jednotlivé požiadavky a nájsť súvislosti medzi nimi.

3.1 Funkčné a nefunkčné požiadavky

Existuje množstvo spôsobov ako rozdeliť požiadavky. Základná kategorizácia delí požiadavky na funkčné a nefunkčné. Funkčné požiadavky sú všetky tie požiadavky, ktoré určujú aké funkcie bude systém ponúkať a ako sa bude systém správať. Nefunkčné požiadavky sú také požiadavky, ktoré majú určitú dodatočnú vlastnosť systému a stanovujú jeho obmedzujúce podmienky.

V nasledujúcich sekciách sa nachádza popis funkčných aj nefunkčných požiadaviek kladených na informačný systém na podporu online doučovania.

3.1.1 Funkčné požiadavky

F1 – Správa študentov a lektorov

Systém bude umožňovať správu jednotlivých používateľov – študentov a lektorov. Systém umožní používateľom zaregistrovať sa do systému a prihlásiť sa do aplikácie. Skutočná registrácia do systému bude ale podmienená potvrdením používateľského účtu administrátorom. Po vytvorení a aktivácii používateľského účtu lektora sa navyše vytvorí aj kalendár na účte Google. Pre každého zaregistrovaného lektora bude teda na požadovanom Google účte vytvorený samostatný kalendár.

Každý zaregistrovaný používateľ pritom bude mať v systéme používateľský profil, ktorý bude možné zobrazovať inými používateľmi alebo bude možné upraviť si vlastný profil. Systém dokáže evidovať základné informácie o používateľovi – jeho používateľské meno, celé meno, pohlavie, vek a niektoré ďalšie osobné informácie. Je však možné evidovať aj informácie o kontaktných údajoch na rôznych komunikačných portáloch a tiež adresu používateľa. V rámci správy vlastného profilu bude navyše možné zmeniť si registračnú emailovú adresu a heslo do systému.

Používateľom s rolou lektora systém navyiac umožní upraviť si svoju dostupnosť, teda upraviť časy, v ktorých chce byť lektor dostupný pre doučovanie. Používatelia s privilégiami administrátorov budú môcť okrem toho explicitne zaregistrovať používateľa, aktivovať účet po registrácii iného používateľa, pridávať a odoberať administrátorské práva iným lektorom a tiež dopĺňať kredity na účet iných používateľov.

F2 – Správa lekcí

Systém umožní lektorom vytvárať lekcie. Lekciou rozumieme akýsi súhrn všetkého toho, čo sa študent môže dozvedieť na konzultačných hodinách danej lekcie. Pod konzultačnou hodinou potom rozumieme konkrétnu interakciu medzi študentom a lektorom. Výsledkom registrácie na lekcii je konzultačná hodina.

Systém umožní lektorom pri vytváraní lekcie určiť základný popis danej lekcie, teda toho, čo sa študent na hodinách lekcie môže naučiť. Tento popis si lektor dodatočne môže upraviť. V rámci editácie bude možné lekcii urobiť neaktívnu tak, aby sa nezobrazovala vo vyhľadávaní. Medzi jednotlivými lekciami sa teda bude dať vyhľadávať podľa rôznych kritérií. Bude tiež možné odfiltrovať si len tie lekcie, ktoré lektor sám vyučuje.

Po vytvorení lekcie, za predpokladu, že je lekcii aktívna, systém umožní študentovi zaregistrovať sa na lekcii lektora. Študent si sám na základe dostupnosti lektora vyberie čas, kedy chce mať s lektorom konzultačnú hodinu. Podmienkou registrácie je, že má dostatočné množstvo kreditov na svojom účte. Spolu s vytvorením konzultačnej hodiny sa vytvorí aj akcia v aplikácii Google Calendar.

F3 – Správa konzultačných hodín

Systém po zaregistrovaní študenta na danú lekcii vytvorí konzultačnú hodinu. Tento priestor bude dostupný len pre autora danej lekcie a študenta, ktorý je na hodinu zaregistrovaný. S vytvorením konzultačnej hodiny v systéme sa vytvorí aj udalosť v aplikácii Google kalendár.

V rámci konzultačnej hodiny bude možné meniť stav danej hodiny lektorom alebo bude možné zrušiť hodinu. Študent však môže hodinu zrušiť len s predstihom 24 hodín.

Systém ďalej umožní lektorom vypísať domácu úlohu týkajúcu sa danej hodiny. Súčasťou zadania budú inštrukcie k domácej úlohe, možné je tiež pripojiť k domácej úlohe súbor (ako doplnok k inštrukciám). Bude možné nahrať buď nový súbor alebo pripojiť existujúci súbor z „adresára“ lektora. Viac informácií o správe súborov je uvedených v požiadavke F7.

Navyše, systém dovolí lektorovi k domácej úlohe pripojiť aj príklad alebo úlohu z databázy, pričom je možné študentovi nechať zobraziť riešenie príkladu a prípadne nechať zobraziť postup riešenia (ak je k dispozícii). Viac o príkladoch

3. ANALÝZA A ŠPECIFIKÁCIA POŽIADAVIEK

je možné dočítať sa v informáciách k požiadavke F5. Takto zadanú domácu úlohu môže študent vyriešiť. Po vyriešení študent informuje systém o tom, že je úloha vyriešená. Príklady je však možné zadávať aj mimo domácu úlohu. Lektor môže jednoducho pripojiť k hodine príklad, ktorý študent priamo počas konzultačnej hodiny rieši a diskutuje svoje riešenie so svojim lektorom. Príklady sa budú zobrazovať aj s využitím systému LaTeX.

Systém by mal okrem toho používateľom dovoliť zobrazovať všetky prebehnuté alebo budúce konzultačné hodiny a filtrovať ich podľa stavu hodiny.

F4 – Správa predmetov a tém

V systéme evidujeme lekcie lektorov. Každá takáto lekcia je však naviazaná na nejaké predmety a témy a preto je nutné evidovať v systéme aj tieto informácie. Je to z dôvodu efektívnejšieho vyhľadávania. Navyše platí, že daná lekcia, ktorú lektor vytvára, by mala mať predvolenú cenu na 60 minút konzultačnej hodiny. V systéme je teda možné vytvárať strom predmetov a tém. Každý predmet bude obsahovať množstvo tém a témy zároveň môžu obsahovať podtémy.

F5 – Správa príkladov

Systém umožní lektorovi vytvárať databázu príkladov, z ktorých lektori čerpajú pri zadávaní domácich úloh. Aj príklady budú viazané na predmety a témy kvôli vyhľadávaniu. Okrem toho bude možné pri vytváraní príkladu zadať popis zadania príkladu, jeho obtiažnosť, riešenie a prípadne postup riešenia. K príkladu bude navyše možné pripojiť aj súbory. Buď bude možné nahráť nový súbor z disku alebo bude možné pripojiť existujúci súbor z „adresára“ zadávateľa príkladu.

Systém ďalej umožní spomínané vyhľadávanie príkladov podľa rôznych kritérií – priamo podľa identifikačného čísla príkladu, ďalej podľa tém príkladu alebo podľa obtiažnosti.

F6 – Počítanie cien a zobrazovanie transakcií

Systém eviduje všetky prebehnuté platby. Na základe vonkajšieho podnetu je možné, aby lektor doplnil študentovi na používateľský profil určité množstvo kreditov (dané požiadavkou F1). Systém však musí toto zvýšenie kreditov evidovať. Ďalej systém musí evidovať registráciu študentov na lekcie, ktoré sú platené. Nakoniec, všetky informácie o platbách týkajúce sa daného používateľa musí systém tomuto používateľovi zobrazovať.

F7 – Správa študijných materiálov

Systém umožní používateľom spravovať si študijné materiály vo forme súborov. Nezáleží pritom na formáte súboru. Systém dovolí používateľom nahráť nový

súbor a takýto súbor je neskôr možné pripojiť ku konzultačnej hodine, k domácej úlohe alebo k príkladu. Používateľovi bude zoznam študijných materiálov zobrazený a bude možné filtrovať v ňom.

F8 – Prepojenie študenta a lektora prostredníctvom videohovoru

Systém by mal umožňovať čo najrýchlejšie prepojiť študenta a lektora formou videoprenosu, napríklad pomocou programu Skype. Vlastný videoprenos však nebude súčasťou aplikácie.

3.1.2 Nefunkčné požiadavky

N1 – Dostupnosť systému na webe

Hlavnou nefunkčnou požiadavkou na systém je dostupnosť systému ako webovej aplikácie prostredníctvom webového prehliadača. Podporované by malo byť zobrazenie vo všetkých novších a populárnejších webových prehliadačoch – Google Chrome aspoň vo verzii 48, Mozilla Firefox vo verzii aspoň 46.0, Opera vo verzii aspoň 35, Microsoft Edge vo verzii aspoň 11 a Safari pre Mac vo verzii aspoň 9.

N2 – Rozšíriteľnosť a modifikovateľnosť

Výber architektúry a návrh aplikácie by mal byť podmienený ľahkou rozšíriteľnosťou o nové funkcie a prípadnou úpravou existujúcej sady funkcií.

3.2 Modelovanie prípadov použitia

Prípady použitia (anglicky *usecase*, skratkou UC) sa používajú na presné špecifikovanie funkčných požiadaviek, často vo forme postupnosti krokov. Štandardne platí, že jedna funkčná požiadavka sa rozpadne na niekoľko prípadov použitia. Jednotlivé prípady použitia je dôležité vytvárať, pretože sú základom pre tvorbu používateľskej príručky a tiež tvoria podklad pre akceptačné testovanie podľa scenárov.

Model prípadu použitia sa skladá:

- zo zoznamu účastníkov,
- z diagramu prípadov použitia,
- zo zoznamu prípadov použitia.

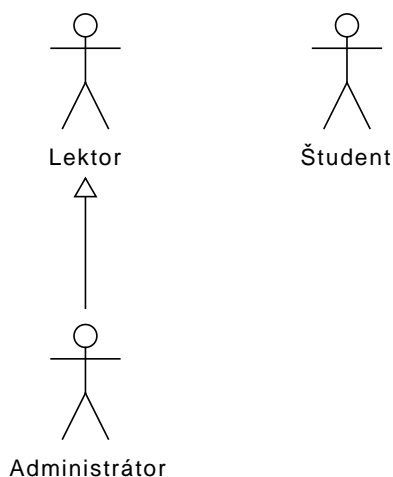
Pri detailnej špecifikácii požiadaviek na systém pomocou prípadov použitia využívame tzv. diagram prípadov použitia. Jeho úlohou je zobrazovať práva

jednotlivých aktérov na vykonávanie rôznych funkcií v systéme. Diagram prípadov použitia patrí medzi diagramy chovania definované v jazyku UML.

Jednotlivé prípady použitia potom obsahujú rôzne scenáre – úspešné, neúspešné alebo alternatívne. Vzhľadom na pomerne veľké množstvo prípadov použitia uvádzam v práci scenáre len ku niektorým, netriviálnym alebo problematickým prípadom použitia.

3.2.1 Účastníci

V systéme figurujú tri hlavné role používateľov, ktorí budú systém využívať – študent, lektor a administrátor.



Obr. 3.1: Účastníci

Študent

Hlavným cieľom študenta je vyhľadávanie lekcí lektorov a registrácia na tieto lekcie. Výsledkom registrácie je konzultačná hodina. V rámci nej študent môže s učiteľom diskutovať svoje otázky a problémy a riešiť domáce úlohy zadané lektorom.

Lektor

Hlavným cieľom lektora je tvorba nových lekcí, na ktoré sa môžu študenti zaregistrovať. Lektor si vytvára lekcii podľa svojich vlastných predstáv a po vzniknutí konzultačnej hodiny rieši so študentom problémy. Lektor tiež môže ku konzultačnej hodine pripojiť domácu úlohu, ktorú potom rieši študent. Navyše,

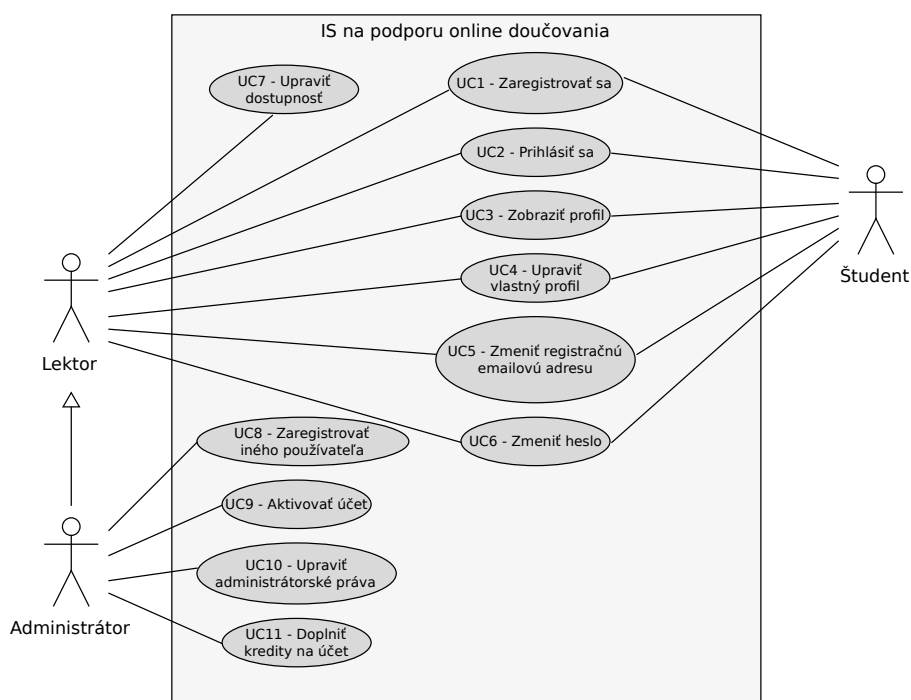
lektor môže dopĺňať databázu príkladov, z ktorej čerpajú ostatní učitelia pri zadávaní domácich úloh.

Administrátor

Ako je vidieť na obrázku 3.1, administrátor dedí práva lektora. Okrem toho v systéme spravuje jednotlivých používateľov. Môže vytvárať nových používateľov, potvrdzovať žiadosti o registráciu používateľského účtu, môže upravovať administrátorské práva iným lektorom a môže tiež doplniť kredity na účet iného používateľa, ktorý o to požiadal. Okrem toho môže administrátor vytvárať štruktúru stromu predmetov a tém.

3.2.2 Prípady použitia pre správu študentov a lektorov

Diagram prípadov použitia pre funkčnú požiadavku F1 – správu študentov a lektorov, je vykreslený na obrázku 3.2.



Obr. 3.2: UML diagram prípadov použitia pre správu študentov a lektorov

UC1 – Zaregistrovať sa

Prípad použitia umožňuje študentovi alebo lektorovi zaregistrovať sa do systému tak, aby v systéme figuroval v tejto roli.

UC2 – Prihlásiť sa

Prípád použitia umožňuje študentovi alebo lektorovi prihlásiť sa do systému za predpokladu, že má vytvorený používateľský účet a bol mu aktivovaný.

UC3 – Zobrazíť profil

Prípád použitia umožňuje študentovi alebo lektorovi zobrazíť používateľský profil.

Hlavný scenár: Zobrazenie vlastného profilu

1. Prípád použitia začína po kliknutí na avatar s menom používateľa v hlavnom menu aplikácie, keď si chce používateľ zobrazíť svoj profil.
2. Systém okamžite zobrazí informácie o používateľovi.

Alternatívny scenár č. 1: Zobrazenie vlastného verejného profilu

1. Alternatívny scenár začína v bode 2 hlavného scenára v prípade, že chce používateľ zobrazíť svoj verejný profil, teda len tie informácie, ktoré sa zobrazujú všetkým ostatným používateľom.
2. Používateľ klikne na tlačítko „Zobrazíť verejný profil“.
3. Systém používateľovi zobrazí len obmedzené množstvo informácií.

Alternatívny scenár č. 2: Zobrazenie verejného profilu iného používateľa

1. Prípád použitia začína vtedy, keď sa používateľ rozhodne zobrazíť si verejný profil iného používateľa.
2. Používateľ, ktorý si chce zobrazíť iný profil, klikne na „Viac informácií“ o danom používateľovi.
3. Systém prihlásenému používateľovi okamžite zobrazí obmedzené informácie o vybranom používateľovi.

UC4 – Upraviť vlastný profil

Prípád použitia umožňuje používateľom upraviť si vlastný profil.

Hlavný scenár: Úprava vlastného profilu

1. Prípád použitia začína po zobrazení používateľského profilu a po kliknutí na tlačítko „Editovať profil“, keď si chce používateľ upraviť svoj profil.

2. Systém zobrazí používateľovi niekoľko formulárov na úpravu používateľských údajov – formulár na úpravu základných osobných údajov, formulár na úpravu adresy a formulár na úpravu kontaktných informácií.
3. Používateľ upraví niektoré svoje osobné údaje – meno, priezvisko, dátum narodenia, prípadne ďalší popis.
4. Používateľ klikne na „Uložiť“.
5. Systém skontroluje vložené údaje. Ak sú údaje správne, systém upraví profil používateľa.

Alternatívny scenár č. 1: Úprava adresy

1. Alternatívny scenár začína v bode 2 hlavného scenára, keď si chce používateľ upraviť adresu.
2. Používateľ upraví svoju štrukturovanú adresu – názov ulice, číslo domu, poštové smerovacie číslo, mesto, krajinu.
3. Používateľ klikne na „Uložiť“.
4. Systém skontroluje vložené údaje. Ak sú údaje správne, systém upraví adresu používateľa.

Alternatívny scenár č. 2: Zmazanie adresy

1. Alternatívny scenár začína v bode 2 hlavného scenára, keď chce používateľ zmazať adresu z evidencie.
2. Používateľ klikne na „Zmazať adresu“.
3. Systém zmaže adresu prihláseného používateľa zo svojej evidencie.

Alternatívny scenár č. 3: Úprava kontaktných údajov

1. Alternatívny scenár začína v bode 2 hlavného scenára, keď si chce používateľ upraviť kontaktné informácie.
2. Používateľ upraví svoje kontaktné údaje, napr. kontaktnú emailovú adresu, používateľské meno na Skype, na Facebooku a prípadne ďalšie kontakty. Používateľ môže tiež upraviť zobrazovanie týchto informácií na verejnom profile.
3. Používateľ klikne na „Uložiť“.
4. Systém upraví kontaktné informácie prihláseného používateľa.

UC5 – Zmeniť registračnú emailovú adresu

Prípád použitia umožňuje používateľovi zmeniť emailovú adresu, ktorú použil pri registrácii do systému.

UC6 – Zmeniť heslo

Prípád použitia umožňuje používateľovi zmeniť heslo, ktorým sa prihlasuje do systému.

UC7 – Upraviť dostupnosť lektora

Prípád použitia umožňuje lektorovi upraviť dostupnosť, tzn. časy, v ktorých chce byť k dispozícii študentom kvôli doučovaniu.

Hlavný scenár: Úprava dostupnosti

1. Prípád použitia začína, ak sa lektor rozhodne zmeniť svoju dostupnosť v kalendári.
2. Systém zobrazí lektorovi kalendár jeho dostupnosti.
3. Lektor si v kalendári vyberie tie časy, v ktorých chce, resp. nechce byť k dispozícii pre doučovanie.
4. Systém pri každej voľbe času túto voľbu uloží.
5. Po vybratí všetkých časov prípad použitia končí.

UC8 – Zaregistrovať iného používateľa

Prípád použitia umožňuje administrátorovi zaregistrovať nového používateľa, pričom priamo tento účet aj aktivuje. V prípade, že nový používateľ je lektor, systém vytvorí na požadovanom Google účte samostatný kalendár s názvom používateľského účtu lektora.

UC9 – Aktivovať účet

Prípád použitia umožňuje administrátorovi aktivovať účet nového používateľa, ktorý vyplnil registračný formulár. Rovnako ako v prípade použitia UC8, za predpokladu, že nový používateľ je lektor, systém vytvorí na požadovanom Google účte samostatný kalendár s názvom používateľského účtu lektora.

UC10 – Upraviť administrátorské práva

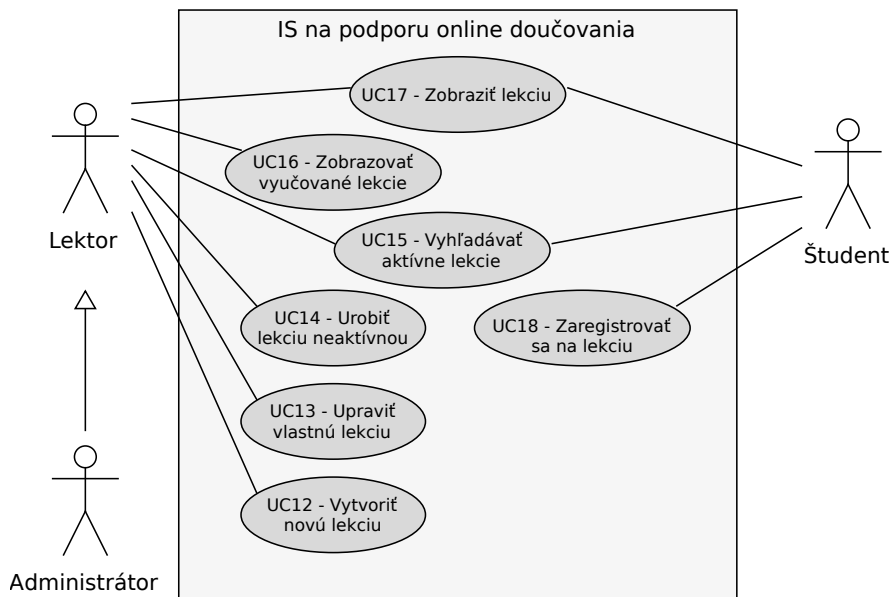
Prípád použitia umožňuje administrátorovi upravovať administrátorské práva iným lektorom a administrátorom.

UC11 – Doplniť kredity na účet

Prípad použitia umožňuje administrátorovi zvýšiť množstvo kreditov na účte iného používateľa systému. Musí sa pritom jednať o študenta.

3.2.3 Prípady použitia pre správu lekcí

Diagram prípadov použitia pre funkčnú požiadavku F2 – správu lekcí, je vykreslený na obrázku 3.3.



Obr. 3.3: UML diagram prípadov použitia pre správu lekcí

UC12 – Vytvoriť novú lekciiu

Prípad použitia umožňuje lektorovi vytvoriť novú lekciiu, na ktorú sa potom môže študent zaregistrovať.

Hlavný scenár: Tvorba novej lekciiu

1. Prípad použitia začína, keď sa lektor rozhodne vytvoriť novú lekciiu.
2. Systém zobrazí lektorovi formulár na pridanie novej lekciiu.
3. Lektor zadá do formulára požadované údaje – názov lekciiu, popis lekciiu (všetko to, čo sa študent môže naučiť, ak sa na túto lekciiu zaregistruje) a akých predmetov a tém sa lekciiu týka.

3. ANALÝZA A ŠPECIFIKÁCIA POŽIADAVIEK

4. Používateľ klikne na „Vytvoriť lekciiu“.
5. Systém skontroluje vložené údaje a ak sú vložené správne, systém vytvorí novú lekciiu, ktorá sa začne zobrazovať vo vyhľadávaní.

UC13 – Upraviť vlastnú lekciiu

Prípád použitia umožňuje lektorovi upraviť vlastnú lekciiu – popis lekciiu a témy, ktorých sa lekciiu týka. Nie je však možné zmeniť názov lekciiu alebo predmet lekciiu.

UC14 – Urobiť lekciiu neaktívnu

Prípád použitia umožní lektorovi označiť lekciiu ako aktívnu v prípade, že chce, aby sa študentom zobrazovala vo vyhľadávaní. Je tiež možné označiť lekciiu ako neaktívnu v prípade, že sa lektor rozhodne, že sa lekciiu nemá zobrazovať vo vyhľadávaní a študenti sa na ňu nemôžu registrovať.

UC15 – Vyhľadávať aktívne lekciiu

Prípád použitia umožňuje používateľom vyhľadávať aktívne lekciiu podľa predmetov a tém, ktorých sa lekciiu týka a tiež podľa ceny lekciiu. Zároveň bude možné výsledky vyhľadávania zoradiť podľa ceny alebo názvu lekciiu.

UC16 – Zobrazovať vyučované lekciiu

Prípád použitia dovoľí lektorovi zobraziť zoznam všetkých lekciiu, ktoré sú ním vyučované.

UC17 – Zobrazíť lekciiu

Prípád použitia umožní používateľovi zobraziť informácie o požadovanej lekciiu.

Hlavný scenár: Zobrazenie lekciiu

1. Prípád použitia začína, keď sa používateľ rozhodne vyhľadať si lekciiu, na ktorú sa zaregistruje. V prípade, že ho nejaká lekciiu zaujala, klikne na ňu, aby si zobrazil viac informácií.
2. Systém zobrazí informácie o požadovanej lekciiu – názov lekciiu, popis lekciiu, témy, ktorých sa lekciiu týka a cenu lekciiu. Okrem toho systém zobrazí používateľovi aj stručné údaje o lektorovi lekciiu. V prípade, že sa chce používateľ dozvedieť viac o lektorovi, klikne na „Viac o lektorovi“, vid' alternatívny scenár č. 2 prípadu použitia UC3.

3. V prípade, že si lekcii prehliada jej autor (lektor), má k dispozícii tlačítko na editovanie lekcii (viď prípad použitia UC13) a tlačítko na urobenie lekcii neaktívnou (viď prípad použitia UC14).
4. V prípade, že si lekcii prehliada študent, systém mu zobrazí aj registračný formulár.

UC18 – Zaregistrovať sa na lekcii

Prípad použitia umožňuje študentovi zaregistrovať sa na lekcii, na konkrétny čas, ktorý si sám zvolí. Podmienkou je, že lektor je v daný čas k dispozícii pre doučovanie. Výsledkom je konzultačná hodina v stave „naplánovaná“.

Hlavný scenár: Registrácia na lekcii

1. Prípad použitia začína, keď sa študent rozhodne zaregistrovať sa na zvolenú lekcii. Študent najskôr klikne na zvolenú lekcii, na ktorú sa chce zaregistrovať a systém mu zobrazí informácie o lekcii (viď prípad použitia UC17).
2. Systém zobrazí študentovi registračný formulár alebo registračnú komponentu.
3. Študent si v registračnej komponente zvolí, na ktoré časy sa chce zaregistrovať. Časy sa študentovi zobrazujú po polhodinách. V prípade, že si študent zvolí nesúvislý čas, je to chyba a systém študentovi túto chybu zobrazí.
4. Študent klikne na „Rezervovať čas“.
5. V prípade, že má študent na účte dostatočné množstvo kreditov, systém študenta zaregistruje na vybraný čas a v príslušnom Google kalendári vytvorí novú akciu. Zároveň systém odpočíta cenu hodiny z účtu študenta. V opačnom prípade systém zobrazí používateľovi informácie o neúspechu pri registrácii.

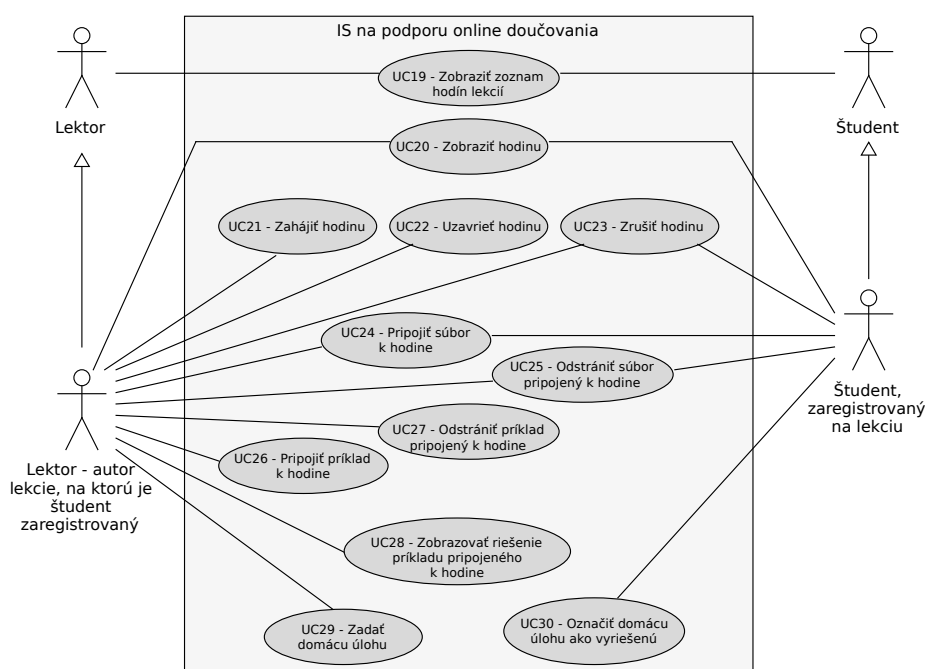
3.2.4 Prípady použitia pre správu konzultačných hodín

Diagram prípadov použitia pre funkčnú požiadavku F3 – správu konzultačných hodín, je vykreslený na obrázku 3.4.

UC19 – Zobrazíť zoznam hodín lekcii

Prípad použitia umožňuje lektorovi zobrazíť zoznam konzultačných hodín, na ktoré sa zaregistroval a lektorovi zoznam hodín lekcii, ktoré vyučoval alebo bude vyučovať. Prípad použitia dovoľuje používateľovi tiež filtrovať medzi týmito hodinami.

3. ANALÝZA A ŠPECIFIKÁCIA POŽIADAVIEK



Obr. 3.4: UML diagram prípadov použitia pre správu konzultačných hodín

UC20 – Zobrazíť hodinu

Prípad použitia umožňuje lektorovi a študentovi zobrazíť hodinu lekcie, ktorú vypísal tento lektor a na ktorú sa tento študent zaregistroval.

Hlavný scenár: Zobrazenie konzultačnej hodiny

1. Prípad použitia začína, keď sa používateľ rozhodne zobrazíť si hodinu lekcie počas konzultačnej hodiny v čase, ktorý si študent zvolil pri registrácii.
2. Systém zobrazí používateľovi informácie o zadanej hodine:
 - informácie o lekcii, z ktorej bola hodina vytvorená,
 - registrovaný čas hodiny,
 - stav konzultačnej hodiny,
 - skutočný čas konzultačnej hodiny v prípade, že hodina bola zahájená,
 - stručné informácie o lektorovi (ak je prihlásený používateľ študent) alebo o študentovi (ak je prihlásený používateľ lektor) s možnosťou spojenia v aplikácii Skype,

- priradené študijné materiály,
- priradené príklady a úlohy,
- informácie o zadanej domácej úlohe.

UC21 – Zahájiť hodinu

Prípad použitia umožňuje lektorovi zmeniť stav konzultačnej hodiny z „naplánovaná“ na „zahájená“. Spolu s tým sa začne počítať skutočné trvanie hodiny, až kým lektor neukončí hodinu, viď UC22.

UC22 – Uzavrieť hodinu

Prípad použitia umožňuje lektorovi zmeniť stav konzultačnej hodiny zo „zahájená“ na „uzavretá“. Spolu s tým sa ukončí počítanie skutočného trvania konzultačnej hodiny.

UC23 – Zrušiť hodinu

Prípad použitia umožňuje lektorovi alebo študentovi zmeniť stav konzultačnej hodiny z „naplánovaná“ na „zrušená“. Lektor môže konzultačnú hodinu zrušiť kedykoľvek s možnosťou prídania poznámky ku zrušeniu (ako dôvod jej zrušenia). Hodinu môže zrušiť aj študent, avšak jedine s predstihom aspoň 24 hodín.

Po zrušení konzultačnej hodiny sa študentovi pripočítajú naspäť kredity, ktoré zaplatil za platbu za túto hodinu.

UC24 – Pripojiť súbor k hodine

Prípad použitia dovoľuje lektorovi alebo študentovi pripojiť k hodine súbor. Môže ísť o študijné materiály, prílohy k zadaniam alebo riešenia domácich úloh.

Hlavný scenár: Pripojenie existujúceho súboru k hodine

1. Prípad použitia začína, keď sa používateľ rozhodne pripojiť k hodine existujúci súbor zo svojho profilu, teda súbor, ktorý už do systému niekedy v minulosti nahral.
2. Používateľ klikne na „Pripojiť existujúci súbor“.
3. Systém zobrazí používateľovi zoznam všetkých súborov, ktoré v minulosti do systému sám nahral.
4. Používateľ nájde v zozname požadovaný súbor, ktorý chce pripojiť k hodine a klikne na „Vybrať súbor“.

3. ANALÝZA A ŠPECIFIKÁCIA POŽIADAVIEK

5. Systém skontroluje, či už daný súbor nie je súčasťou danej hodiny. V prípade, že naozaj nie je, systém pripojí vybraný súbor k tejto hodine.

Alternatívny scenár: Pripojenie nového súboru k hodine

1. Prípád použitia začína, keď sa používateľ rozhodne pripojiť k hodine nový súbor, ktorý ešte nebol nahraný do systému.
2. Používateľ klikne na „Nahrať nový súbor“.
3. Systém zobrazí používateľovi okno, v ktorom môže vybrať nový súbor z disku.
4. Používateľ nájde požadovaný súbor, ktorý chce pripojiť k hodine a klikne na „Vybrať súbor“.
5. Systém pripojí vybraný súbor k danej hodine a zároveň tento súbor uloží k profilu prihláseného používateľa.

UC25 – Odstrániť súbor pripojený k hodine

Prípád použitia umožňuje používateľovi odstrániť súbor pripojený k hodine. Je možné odstrániť len vlastné súbory pripojené k hodine. Používateľovi by malo byť pri požadovanom súbore dostupné tlačítko na zmazanie súboru.

UC26 – Pripojiť príklad k hodine

Prípád použitia umožňuje lektorovi pripojiť ku konzultačnej hodine príklad z databázy. Implicitne sa študentovi nezobrazuje riešenie takto pripojeného príkladu a ani postup jeho riešenia.

UC27 – Odstrániť príklad pripojený k hodine

Prípád použitia umožňuje lektorovi odstrániť príklad pripojený ku konzultačnej hodine. Lektorovi by malo byť pri požadovanom príklade k dispozícii tlačítko na zmazanie príkladu.

UC28 – Zobrazovať riešenie príkladu pripojeného k hodine

Prípád použitia umožňuje lektorovi nechať zobrazovať riešenie príkladu alebo postup riešenia príkladu pripojeného k hodine (za predpokladu, že postup riešenia bol uvedený pri zadávaní príkladu do databázy). V prípade, že zadanie príkladu, jeho časť alebo riešenie príkladu sú napísané v systéme LaTeX, systém musí lektorovi zobraziť príklad prostredníctvom systému LaTeX.

UC29 – Zadať domácu úlohu

Prípad použitia umožňuje lektorovi zadať domácu úlohu súvisiacu s konkrétnou konzultačnou hodinou.

Hlavný scenár: Zadanie domácej úlohy

1. Prípad použitia začína, keď sa lektor rozhodne zadať študentovi domácu úlohu. Domácu úlohu je možné zadať ku konzultačnej hodine priamo po zobrazení hodiny. Lektorovi sa tak zobrazí niekoľko možností, ako upraviť domácu úlohu podľa svojich predstáv.
2. Lektor môže upraviť inštrukcie domácej úlohy (implicitne ide o prázdne pole)
 - a) Lektor klikne na „Pridať inštrukcie úlohy“.
 - b) Systém lektorovi zobrazí pole, do ktorého môže napísať inštrukcie domácej úlohy.
 - c) Lektor zadá inštrukcie domácej úlohy.
 - d) Lektor klikne na tlačítko „Upraviť inštrukcie“.
 - e) Systém uloží lektorovu voľbu a uloží popis inštrukcií.
3. Lektor môže k domácej úlohe pripojiť existujúci alebo nový súbor podobným spôsobom, ako to bolo možné v prípade použitia UC24.
4. Lektor môže k domácej úlohe pripojiť príklad z databázy.
 - a) Systém lektorovi zobrazí aj tlačítko na pridanie príkladu k domácej úlohe.
 - b) Lektor klikne na toto tlačítko.
 - c) Systém lektorovi zobrazí databázu príkladov, pričom pri každom príklade je možnosť pripojiť tento príklad k domácej úlohe.
 - d) Lektor klikne na „Pripojiť príklad k domácej úlohe“.
 - e) Systém pripojí vybraný príklad k domácej úlohe a obnoví zobrazenie konzultačnej hodiny.
 - f) V prípade, že lektor chce, aby sa riešenie príkladu alebo postup riešenia zobrazovali študentovi, môže kliknúť na príslušné tlačítko. Systém následne umožní študentovi zobraziť riešenie príkladu alebo postup jeho riešenia.
5. V prípade, že sa lektorovi niečo nezdá, môže ešte stále upraviť inštrukcie domácej úlohy, môže zrušiť pripojenie súboru k domácej úlohe podobne ako v prípade použitia UC25 a môže tiež odstrániť príklad pripojený k domácej úlohe podobným spôsobom, ako v prípade použitia UC27.

3. ANALÝZA A ŠPECIFIKÁCIA POŽIADAVIEK

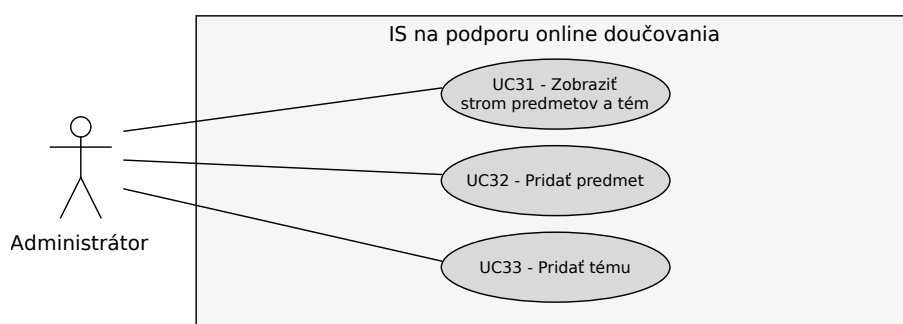
6. V prípade, že lektor uzná, že domáca úloha je zadaná správne, klikne na „Zadať domácu úlohu“. Až tak sa domáca úloha zobrazuje študentovi (a ten ju môže neskôr vyriešiť). Domácu úlohu už nie je možné ďalej upravovať lektorom.

UC30 – Označiť domácu úlohu ako vyriešenú

Prípád použitia umožňuje študentovi označiť domácu úlohu ako vyriešenú za predpokladu, že lektor zadal domácu úlohu k danej konzultačnej hodine.

3.2.5 Prípady použitia pre správu predmetov a tém

Diagram prípadov použitia pre funkčnú požiadavku F4 – správu predmetov a tém, je vykreslený na obrázku 3.5.



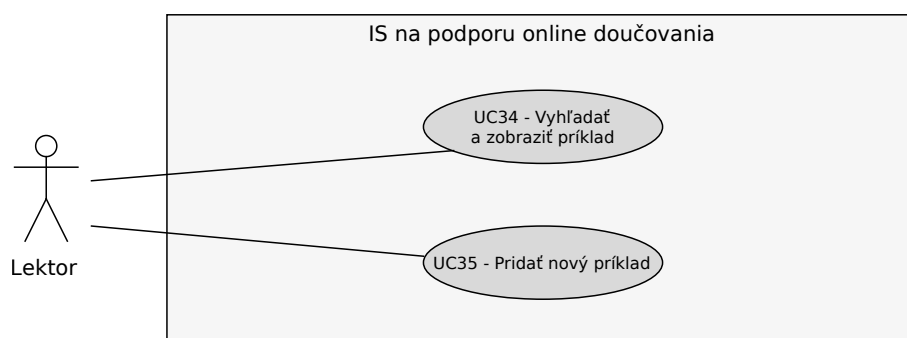
Obr. 3.5: UML diagram prípadov použitia pre správu predmetov a tém

UC31 – Zobrazíť strom predmetov a tém

Prípád použitia umožňuje administrátorovi zobrazíť strom predmetov a tém. Na najvyššej úrovni stromu sa budú nachádzať evidované predmety. Predmety zároveň môžu obsahovať témy a témy môžu obsahovať podtémy. Systém zároveň pri každom predmete zobrazuje predvolenú cenu na 60 minút konzultačnej hodiny lektora.

UC32 – Pridať predmet

Prípád použitia dovoľuje administrátorovi pridať predmet do stromu predmetov a tém. Administrátor určuje názov predmetu (musí byť unikátny) a predvolenú cenu predmetu na 60 minút konzultačnej hodiny lektora.



Obr. 3.6: UML diagram prípadov použitia pre správu príkladov

UC33 – Pridať tému

Prípad použitia umožňuje administrátorovi pridať tému do stromu predmetov a tém. Administrátor môže tému priradiť ako tému predmetu alebo môže tému priradiť pod inú tému.

3.2.6 Prípady použitia pre správu príkladov

Diagram prípadov použitia pre funkčnú požiadavku F5 – správu príkladov, je vykreslený na obrázku 3.6.

UC34 – Vyhľadať a zobrazí príklad

Prípad použitia umožňuje lektorovi vyhľadávať príklady uložené v databáze. Príklady spĺňajúce podmienky filtra sú zároveň kompletne zobrazené lektorovi. V prípade, že zadanie príkladu, jeho časť alebo riešenie príkladu sú napísané v systéme LaTeX, systém musí lektorovi zobraziť príklad prostredníctvom nejakej knižnice, ktorá umožňuje spracovávať takýto text.

Hlavný scenár: Vyhľadanie a zobrazenie príkladu

1. Prípad použitia začína, keď sa lektor rozhodne vyhľadať si príklad, ktorý chce pripojiť k požadovej konzultačnej hodine alebo k domácej úlohe.
2. Systém používateľovi zobrazí formulár na vyhľadávanie medzi príkladmi.
3. Lektor zadá do formulára požadované možnosti. Lektor môže filtrovať príklady podľa požadovaných tém, podľa obtiažnosti a výsledky môže zároveň zoradiť podľa poradového čísla alebo obtiažnosti. V prípade potreby je možné zadať aj konkrétne poradové číslo príkladu.
4. Lektor klikne na „Vyhľadať príklady“.

3. ANALÝZA A ŠPECIFIKÁCIA POŽIADAVIEK

5. Systém lektorovi zobrazí zoznam všetkých splňajúcich príkladov a úloh. Pre každý príklad sa lektorovi zobrazia:
 - identifikačné číslo príkladu,
 - zadanie príkladu,
 - obtiažnosť príkladu,
 - predmet a témy príkladu,
 - priložené súbory.
6. V prípade, že si lektor chce zobraziť riešenie niektorého príkladu, klikne pri tomto príklade na tlačítko „Zobrazíť riešenie“. V prípade potreby a v prípade, že bol vyplnený postup riešenia príkladu, si lektor môže zobrazíť aj tento postup.

UC35 – Pridať nový príklad

Prípád použitia umožňuje lektorovi pridať do databázy nový príklad. V prípade potreby môže zadanie príkladu a postup riešenia príkladu využívať systém LaTeX.

Hlavný scenár: Pridanie príkladu

1. Prípád použitia začína, keď sa lektor rozhodne pridať do databázy nový príklad, ktorý chce pripojiť k požadovej konzultačnej hodine alebo k domácej úlohe.
2. Systém používateľovi zobrazí formulár na pridanie nového príkladu.
3. Lektor musí do formulára vložiť zadanie príkladu, obtiažnosť príkladu, predmety a témy, ktorých sa lekcia týka, ďalej tiež riešenie príkladu a v prípade potreby aj postup riešenia príkladu.
4. Systém lektorovi musí umožňovať zobrazíť si zadanie príkladu alebo postup riešenia príkladu aktívne počas ich písania, pretože je možné písať zadanie a postup riešenia prostredníctvom systému LaTeX. Lektor by tak mal mať možnosť skontrolovať si, že príklad sa bude zobrazovať správne.
5. Lektor môže k príkladu priradiť aj súbory podobným spôsobom, ako to bolo možné v prípade použitia UC24.
6. Ak je lektor spokojný so zadaným príkladom, klikne na „Uložiť príklad“. Tým sa príklad uloží do databázy a zobrazuje sa vo vyhľadávaní.

3.2.7 Prípady použitia pre počítanie cien a zobrazovanie transakcií

Počítanie cien a manipulácia s kreditmi používateľa budú v aplikácii reprezentované niekoľkými prípadmi použitia, ktoré už boli vysvetlené a zasahujú do iných funkčných požiadaviek. Jedná sa o prípady použitia UC11 – Doplniť kredity na účet, UC18 – Zaregistrovať sa na lekciiu a UC23 – Zrušiť hodinu.

Je však možné doplniť k nim potrebu zobrazovania transakcií. Vzhľadom na jediný skutočný prípad použitia pre túto funkčnú požiadavku neuvádzam diagram prípadov použitia.

UC36 – Zobrazíť zoznam transakcií

Prípad použitia umožňuje študentovi zobrazíť si zoznam všetkých transakcií, ktoré s ním súvisia. Jedná sa o navýšenie množstva kreditov na účet študenta, záznamy o registrácii na lekciiu a tiež záznamy o vrátení kreditov v prípade predčasného zrušenia hodiny. Systém musí študentovi zobrazovať dátum transakcií a tiež hodnotu transakcie v kreditoch.

3.2.8 Prípady použitia pre správu študijných materiálov

Správa študijných materiálov bude v systéme reprezentovaná prostredníctvom súborov. Aj táto funkčná požiadavka však zasahuje do iných funkčných požiadaviek. Súbor je totiž možné priradovať ku konzultačnej hodine (viď prípad použitia UC24), k domácej úlohe (viď prípad použitia UC29) a tiež k príkladu (viď prípad použitia UC35). K týmto prípadom použitia je možné pridať možnosť zobrazenia študijných materiálov nahraných daným používateľom a možnosť nahráť nový súbor.

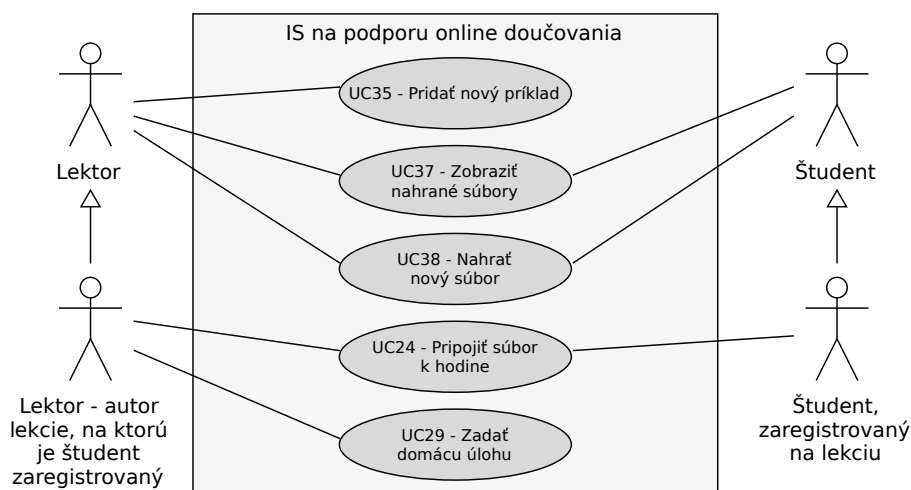
Diagram prípadov použitia pre funkčnú požiadavku F7 – správa študijných materiálov, je vykreslený na obrázku 3.7.

UC37 – Zobrazíť nahrané súbory

Prípad použitia umožňuje používateľovi zobrazíť si zoznam všetkých doposiaľ nahraných súborov a nechať si tieto súbory zoradiť podľa dátumu nahratia alebo podľa názvu súboru.

UC38 – Nahrať nový súbor

Prípad použitia umožňuje používateľovi nahráť nový súbor, ktorý sa používateľovi bude zobrazovať na profile a môže ho neskôr pripojiť ku konzultačnej hodine, k domácej úlohe alebo k zadaniu príkladu.



Obr. 3.7: UML diagram prípadov použitia pre správu študijných materiálov

3.2.9 Prípad použitia pre prepojenie študenta a lektora prostredníctvom videohovoru

Ako bolo uvedené, funkčná požiadavka F8 vyžaduje prepojiť študenta a lektora prostredníctvom videohovoru. Nie je však nutné, aby bol vlastný videoprenos súčasťou aplikácie.

Vzhľadom na jediný prípad použitia neuvádzam diagram prípadov použitia pre túto funkčnú požiadavku.

UC39 – Prepojiť študenta a lektora prostredníctvom videohovoru

Prípad použitia umožňuje študentovi počas konzultačnej hodiny zavolať svojmu lektorovi prostredníctvom aplikácie Skype. Rovnako tak umožňuje lektorovi zavolať študenta. Predpokladom je, že si prihlásený používateľ zobrazuje prebiehajúcu konzultačnú hodinu.

Hlavný scenár: Zavolanie lektora študentom

1. Prípad použitia začína, keď sa študent rozhodne so začiatkom konzultačnej hodiny zavolať svojmu lektorovi na Skype. Študent vidí ako súčasť zobrazenia konzultačnej hodiny stručné informácie o lektorovi. V prípade, že lektor má na profile nastavené svoje používateľské meno v aplikácii Skype a tiež to, aby sa mu táto informácia zobrazovala verejne, systém zobrazí študentovi tlačítko „Zavolať na Skype“.
2. Študent klikne na tlačítko „Zavolať na Skype“.

3. Systém sa pokúsi používateľa prepnúť do aplikácie Skype a zároveň spustí videohovor.

Alternatívny scenár rieši zavolanie študenta lektorom a je veľmi podobný hlavnému scenáru, avšak v tomto prípade sa snaží lektor o spojenie so študentom.

Softvérová architektúra a návrh

Definované už boli všetky požiadavky kladené na informačný systém, teda všetko to, čo má systém dokázať. Nebolo však definované to, akým spôsobom bude splnenie týchto požiadaviek zaistené. To je cieľom návrhu systému v rámci procesu tvorby softvéru.

Ďalej je cieľom návrhu systému zvoliť si vhodný implementačný jazyk. V prípade rozsiahlejších informačných systémov však ide takmer výlučne o voľbu niektorého z objektovo-orientovaných jazykov. Okrem toho sa zaoberáme v návrhu tiež voľbou spôsobu ukladania dát, či už ide o voľbu databázových systémov alebo o možnosť ukladania súborov.

Pred samotným návrhom systému je však potrebné zaoberať sa aj architektúrou budúceho systému, teda tým, ako zostavíme aplikáciu, čo sa týka softvérových tried (ak predpokladáme objektový návrh). Zamýšľame sa nad tým, ako tieto triedy usporiadame do softvérových balíčkov a ako usporiadame tieto balíčky do vrstiev a väčších podsystémov. Výber softvérovej architektúry tiež výrazne závisí na tom, že chceme efektívne realizovať nefunkčné požiadavky.

4.1 Voľba implementačného jazyka a frameworkov

Jednou z nefunkčných požiadaviek definovaných v kapitole 3 o analýze bola dostupnosť aplikácie prostredníctvom webového prehliadača. Po konzultáciách s vedúcim práce a po uvážení na vlastných schopnostiach bolo rozhodnuté, že práca bude naprogramovaná v jazyku PHP s využitím Nette frameworku vo verzii 2.4. Samozrejmosťou je tak použitie jazyka HTML.

Dôvodom, prečo si zvoliť Nette framework, môže byť niekoľko. Jeden z nich je ten, že Nette poskytuje širokú podporu pre vývoj webových aplikácií s využitím MVC architektúry. Podrobné informácie ohľadom architektúry informačného systému na podporu online doučovania sú uvedené v sekcii 4.2: Softvérová architektúra.

Okrem toho Nette podporuje prácu s JavaScriptom využitím technológie AJAX. AJAX (Asynchronous Javascript + XML) je označenie pre „sadu

technológií určenú k vývoju interaktívnych webových aplikácií, ktoré menia obsah svojich stránok bez nutnosti ich znovunačítania“ [7, s. 18]. AJAX tak funguje ako akési prepojenie medzi klientom a serverom pri spracovaní obsahu. Napriek tomu, že požadovaná webová aplikácia bude primárne bežať na PHP, viaceré funkcie systému môžu mať interaktívny charakter a preto je nutné, aby aplikácia fungovala z časti aj na strane klienta. Kvôli tomu budú niektoré časti systému využívať spomínaný jazyk JavaScript a ajaxové spracovanie, ktoré je súčasťou Nette.

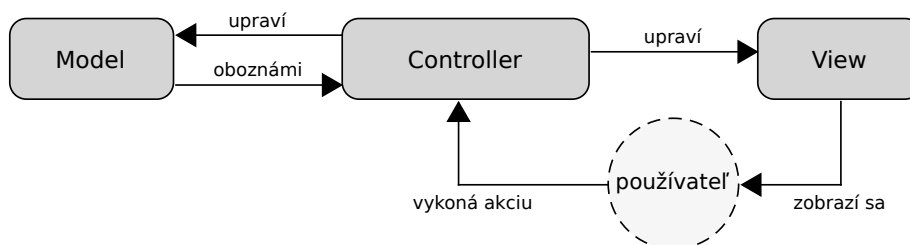
Nette je zároveň z veľkej časti založený na použití komponent a pretože je tento framework open source a má v Českej republike a na Slovensku založenú širokú komunitu programátorov, je množstvo programovateľných komponent verejne dostupných. To môže výrazne zjednodušovať proces implementácie systému.

Aj keď primárnym cieľom práce nie je tvorba dizajnu webových stránok, bolo rozhodnuté, že kostra aplikácie bude využívať front-endový framework Bootstrap [8] a to z toho dôvodu, že tento framework obsahuje množstvo preddefinovaných šablón pre typografiu, formuláre, navigáciu a iné webové komponenty, čo výrazne uľahčuje realizáciu dizajnu webových stránok. To je dôležité okrem iného aj kvôli nevyhnutnej údržbe webových aplikácií.

Navyše, Bootstrap výrazne uľahčuje tvorbu responzívnych a používateľsky prívetivých webstránok.

4.2 Softvérová architektúra

Ako bolo spomenuté v predchádzajúcej sekcii, Nette framework poskytuje podporu pre budovanie aplikácií s využitím MVC architektúry. MVC (Model-View-Controller) je softvérová architektúra, ktorá „vznikla z potreby oddeliť pri aplikáciách s grafickým rozhraním kód obsluhy (Controller) od kódu aplikáčnej logiky (Model) a od kódu zobrazujúceho dáta (View)“ [9]. Kód aplikácií napísaných s využitím MVC architektúry je tak prehľadný a uľahčuje budúci vývoj. To boli hlavné dôvody, prečo bola zvolená práve táto architektúra.



Obr. 4.1: Diagram popisujúci spoluprácu vrstiev MVC architektúry

Ako je možné vidieť na obrázku 4.1, úlohou vrstvy Controller je teda spracovávať požiadavky používateľov. Každá akcia používateľa by mala viesť na akciu kontroléru a táto akcia potom volá zodpovedajúcu aplikačnú logiku (tzn. Model). Úlohou modelovej vrstvy je potom spracovávať dáta a poskytovať ich naspäť vrstve Controller. Štandardne sa jedná o úpravu alebo získanie hodnôt z databázy. Po spracovaní požiadavky kontroléru je úlohou vrstvy View poskytnúť získané a spracované dáta používateľovi takým spôsobom, aby používateľ týmito informáciami porozumel a mohol vykonávať ďalšie akcie.

4.2.1 Softvérová architektúra v Nette

Vzhľadom na charakter požadovanej webovej aplikácie bude mať modelová vrstva na starosti hlavne prístup do databázy. V Nette je tento prístup realizovaný prostredníctvom tzv. Selection API [10], ktoré umožňuje nad databázou vykonávať štandardné operácie, ako je výber hodnôt z databázy a obmedzovanie tohto výberu, spájanie tabuliek, vloženie hodnôt do databázy alebo úprava hodnôt v databáze. Je však možné dotazovať sa do databázy aj pomocou „čistých“ SQL príkazov podobným spôsobom, ako je to možné v jazyku PHP bez využitia Nette.

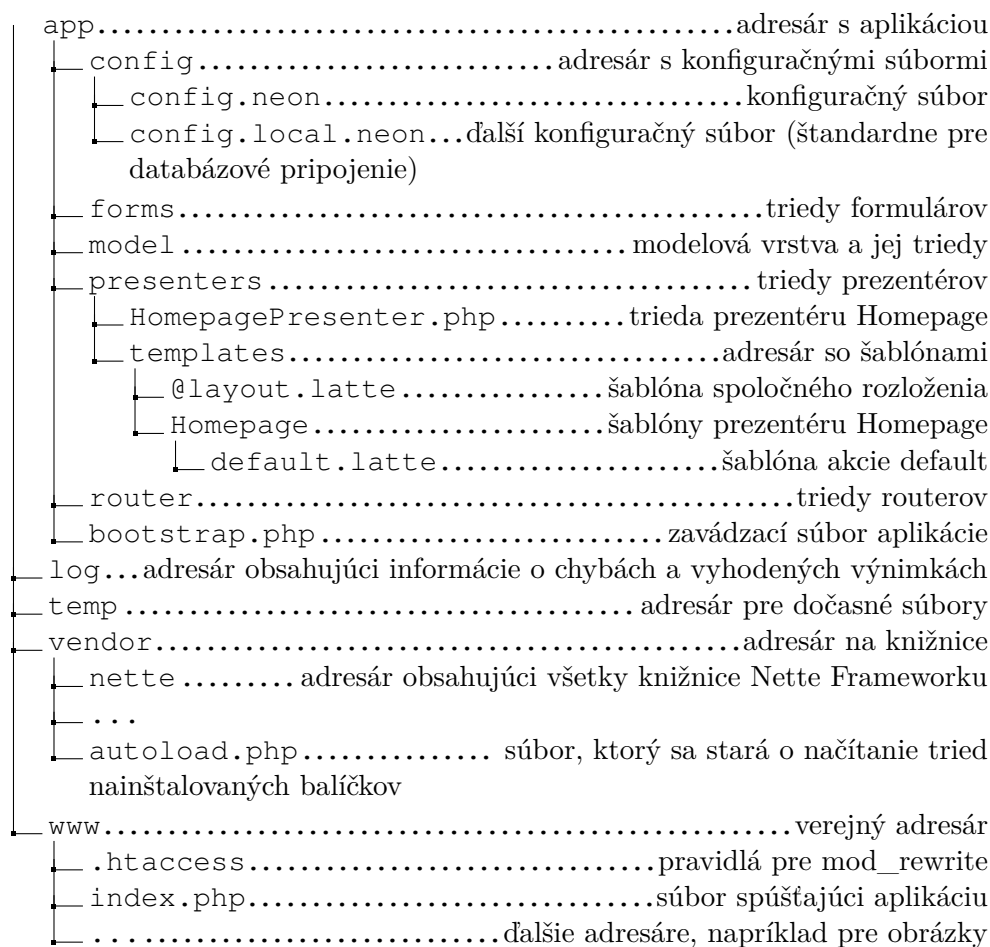
Obdobou triedy vrstvy Controller je v Nette tzv. prezentér. Niekedy sa podľa toho označuje takáto architektúra ako MVP (Model-View-Presenter) a priamo vychádza z architektúry MVC. Prezentér je v Nette objekt, ktorý na základe prichádzajúcich požiadaviek spúšťa zodpovedajúce akcie a podľa ich výsledkov vykresľuje šablóny.

Vrstva View v Nette je realizovaná prostredníctvom šablónovacieho systému Latte [11]. Každý prezentér, každá akcia prezentéru, ktorá vedie na zobrazenie stránky (a nie napríklad na presmerovanie) a tiež každá komponenta má v Nette definovanú svoju šablónu. Prezentér pritom posiela dáta do šablóny a jej úlohou je potom vykresliť požadované dáta a formátovať ich podľa potreby. Šablónovací systém Latte má preddefinovaných množstvo makier, ktoré umožňujú spracovať dáta, ktoré boli preposlané prezentérom. V prípade potreby je dokonca možné dodefinovať vlastné makrá.

Doporučená adresárová štruktúra Nette projektu využívajúceho MVC architektúru [9] je zobrazená na obrázku 4.2.

4.3 Ukladanie dát

Nasledujúce strany sú venované spôsobu uloženia dát. Sekcia pojednáva o voľbe databázy, tvorbe databázového modelu, ale aj o spôsobe ukladania fyzických súborov.



Obr. 4.2: Doporučená adresárová štruktúra Nette projektu

4.3.1 Voľba spôsobu uloženia dát

Na ukladanie textových dát bola zvolená relačná databáza MySQL a to hlavne z toho dôvodu, že práca s databázou MySQL je priamo podporovaná jazykom PHP a navyše, takmer všetky populárne a dostupné webhostingové služby ponúkajú už vo svojom základom cenníku podporu pre MySQL.

Vzhľadom na niektoré funkčné požiadavky na informačný systém je však nutné aj ukladanie súborov. To bude riešené priamym nahraním súborov používateľmi na server, na ktorom bude bežať samotná aplikácia. Keďže každý súbor bude nahraný nejakým konkrétnym používateľom a len tento používateľ bude mať k takýmto súborom priamy prístup, pre každý aktivovaný používateľský účet bude nutné vytvoriť vlastný adresár, do ktorého sa nahrané súbory budú ukladať.

4.3.2 Databázový model

Na nasledujúcich stranách je popísaný databázový model informačného systému na podporu online doučovania. Popísané sú všetky atribúty entít a zobrazené sú všetky asociácie medzi entitami. Databázový model je zameraný na zvolený systém riadenia bázy dát MySQL. Aj kvôli tomu bol navrhnutý v špecializovanom grafickom nástroji MySQL Workbench, ktorý umožňuje export modelu do tzv. create skriptu, teda skriptu na vytvorenie samotnej databáze.

Databázový model je zobrazený na obrázku 4.3. Popis jednotlivých databázových tabuliek vrátane atribútov je následne uvedený v tabuľkách. Databázové tabuľky sú uvedené v alfabetickej poradí. Atribúty tabuliek obsahujú aj popis toho, či sa jedná o povinný atribút (P) – znak A (áno) ak sa jedná o povinný atribút alebo znak N (nie) v prípade, že sa jedná o nepovinný atribút.

Databázová tabuľka *address*

Databázová tabuľka *address* reprezentuje adresu používateľa zaregistrovaného v systéme. Jednotlivé atribúty databázovej tabuľky sú zobrazené a popísané v tabuľke 4.1.

Tabuľka 4.1: Atribúty databázovej tabuľky *address*

P	Názov atribútu	Dátový typ	Popis
A	street_name	VARCHAR(63)	Názov ulice.
A	street_number	VARCHAR(15)	Číslo domu.
A	zip	VARCHAR(7)	Poštové smerovacie číslo.
A	town	VARCHAR(63)	Názov mesta.
A	country	VARCHAR(63)	Názov krajiny.
A	user_userID	INT	Identifikátor používateľa, ktorému patrí adresa. Cudzí kľúč tabuľky <i>user</i> .

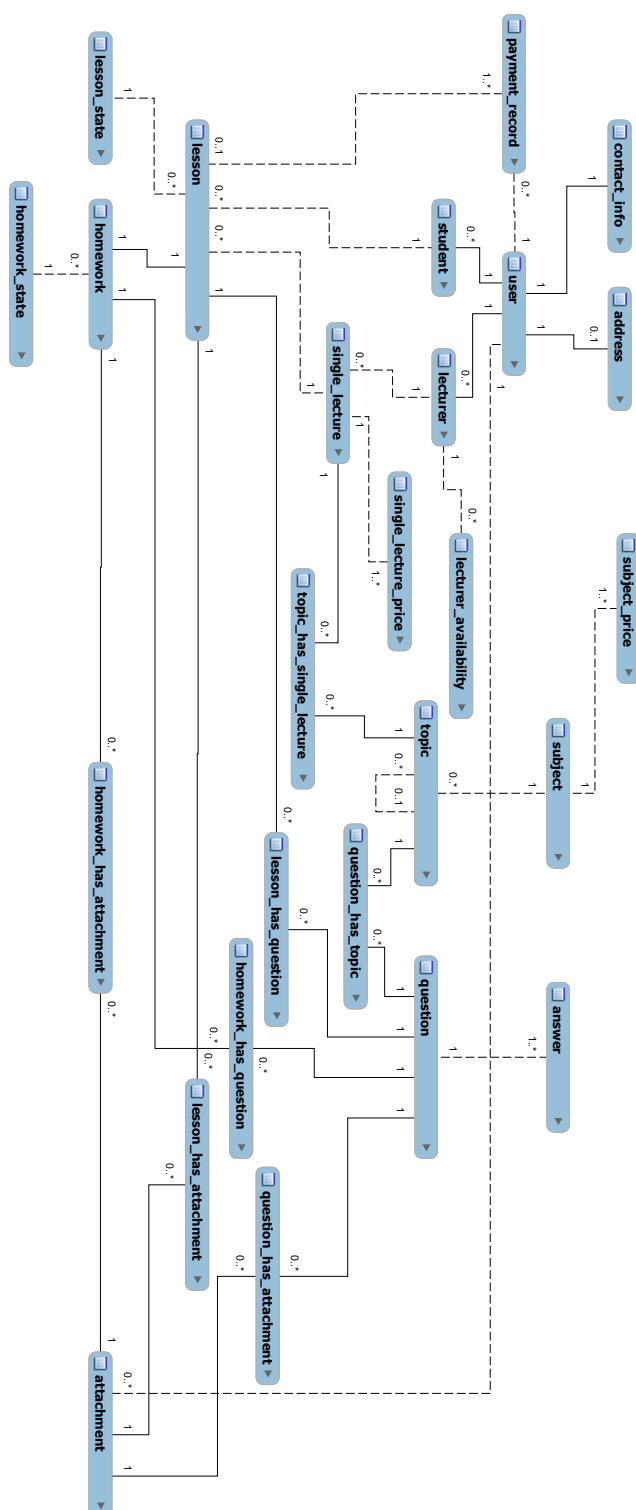
Databázová tabuľka *answer*

Databázová tabuľka *answer* reprezentuje odpoveď na otázku (príklad alebo úlohu) danú tabuľkou *question*. Jednotlivé atribúty databázovej tabuľky *answer* sú zobrazené a popísané v tabuľke 4.2.

Databázová tabuľka *attachment*

Databázová tabuľka *attachment* reprezentuje v systéme súbor používateľa, resp. prílohu domácej úlohy, konzultačnej hodiny alebo príkladu. Jednotlivé atribúty databázovej tabuľky *attachment* sú zobrazené a popísané v tabuľke 4.3.

4. SOFTVÉROVÁ ARCHITEKTÚRA A NÁVRH



Obr. 4.3: Databázový model

Tabuľka 4.2: Atribúty databázovej tabuľky *answer*

P	Názov atribútu	Dátový typ	Popis
A	answerID	INT	Identifikátor odpovede.
A	answer	VARCHAR(255)	Odpoveď na otázku, riešenie úlohy.
A	is_correct	BOOLEAN	Indikátor toho, či je odpoveď na otázku správna alebo nie.
A	question_questionID	INT	Identifikátor otázky, ku ktorej patrí odpoveď. Cudzí kľúč tabuľky <i>question</i> .

Tabuľka 4.3: Atribúty databázovej tabuľky *attachment*

P	Názov atribútu	Dátový typ	Popis
A	attachmentID	INT	Identifikátor súboru.
A	attachment_name	VARCHAR(511)	Názov súboru, ktorý je identický s názvom fyzického súboru.
A	date_uploaded	DATETIME	Dátum nahrania súboru.
A	user_userID	INT	Identifikátor používateľa, ktorý nahral súbor do systému. Cudzí kľúč tabuľky <i>user</i> .

Databázová tabuľka *contact_info*

Databázová tabuľka *contact_info* reprezentuje kontaktné informácie o danom používateľovi zaregistrovanom v systéme. Jednotlivé atribúty databázovej tabuľky *contact_info* sú zobrazené a popísané v tabuľke 4.4.

Databázová tabuľka *homework*

Databázová tabuľka *homework* reprezentuje domácu úlohu zadanú na konzultačnej hodine. Jednotlivé atribúty databázovej tabuľky *homework* sú zobrazené a popísané v tabuľke 4.5.

Databázová tabuľka *homework_has_attachment*

Tabuľka *homework_has_attachment* slúži na dekomponovanie vzťahu M:N medzi tabuľkami *homework* a *attachment*. Databázová tabuľka obsahuje záznamy o tom, že k domácej úlohe bol pripojený súbor. Jednotlivé atribúty databázovej tabuľky *homework_has_attachment* sú zobrazené a popísané v tabuľke 4.6.

4. SOFTVÉROVÁ ARCHITEKTÚRA A NÁVRH

Tabuľka 4.4: Atribúty databázovej tabuľky *contact_info*

P	Názov atribútu	Dátový typ	Popis
N	email	VARCHAR(127)	Kontaktná emailová adresa používateľa.
A	is_email_visible	BOOLEAN	Indikátor toho, že je kontaktná emailová adresa viditeľná ostatným používateľom.
N	skype	VARCHAR(127)	Používateľské meno v aplikácii Skype.
A	is_skype_visible	BOOLEAN	Indikátor toho, že je používateľské meno v aplikácii Skype viditeľné ostatným používateľom.
N	facebook	VARCHAR(127)	URL adresa alebo meno používateľa v aplikácii Facebook.
A	is_facebook_visible	BOOLEAN	Indikátor toho, že je kontakt na Facebook viditeľný ostatným používateľom.
N	others	TEXT	Ostatné kontaktné údaje.
A	user_userID	INT	Identifikátor používateľa. Cudzí kľúč tabuľky <i>user</i> .

Tabuľka 4.5: Atribúty databázovej tabuľky *homework*

P	Názov atribútu	Dátový typ	Popis
N	instructions	TEXT	Inštrukcie domácej úlohy.
A	lesson_lessonID	INT	Identifikátor hodiny, na ktorej bola zadaná domáca úloha. Cudzí kľúč tabuľky <i>lesson</i> .
A	homework_state_homework_state_ID	INT	Identifikátor stavu domácej úlohy. Cudzí kľúč tabuľky <i>homework_state</i> .

Tabuľka 4.6: Atribúty databázovej tabuľky *homework_has_attachment*

P	Názov atribútu	Dátový typ	Popis
A	homework_lesson_lessonID	INT	Identifikátor konzultačnej hodiny, ku ktorej patrí domáca úloha. Cudzí kľúč tabuľky <i>lesson</i> .
A	attachment_attachmentID	INT	Identifikátor prílohy. Cudzí kľúč tabuľky <i>attachment</i> .

Databázová tabuľka *homework_has_question*

Tabuľka *homework_has_question* slúži na dekomponovanie vzťahu M:N medzi tabuľkami *homework* a *question*. Databázová tabuľka obsahuje záznamy o tom, že k domácej úlohe bol pripojený príklad a tiež to, akým spôsobom bol pripojený. Jednotlivé atribúty databázovej tabuľky *homework_has_question* sú zobrazené a popísané v tabuľke 4.7.

Tabuľka 4.7: Atribúty databázovej tabuľky *homework_has_question*

P	Názov atribútu	Dátový typ	Popis
A	homework_lesson_lessonID	INT	Identifikátor konzultačnej hodiny, ku ktorej patrí príklad. Cudzí kľúč tabuľky <i>lesson</i> .
A	question_questionID	INT	Identifikátor otázky. Cudzí kľúč tabuľky <i>question</i> .
A	is_answer_visible	BOOLEAN	Indikátor, že sa pri príklade zobrazuje aj jeho riešenie.
A	are_solution_steps_visible	BOOLEAN	Indikátor toho, že sa pri príklade zobrazuje aj postup jeho riešenia.

Databázová tabuľka *homework_state*

Databázová tabuľka *homework_state* reprezentuje stav domácej úlohy. Jednotlivé atribúty databázovej tabuľky *homework_state* sú zobrazené a popísané v tabuľke 4.8. Záznamy v tejto databázovej tabuľke (tabuľka 4.9) budú pevne dané a nebude možné ich meniť.

Databázová tabuľka *lecturer*

Databázová tabuľka *lecturer* reprezentuje lektora. Jednotlivé atribúty databázovej tabuľky *lecturer* sú zobrazené a popísané v tabuľke 4.10.

Databázová tabuľka *lecturer_availability*

Databázová tabuľka *lecturer_availability* reprezentuje čas, v ktorom je lektor k dispozícii študentom kvôli doučovaniu. Jednotlivé atribúty databázovej tabuľky *lecturer_availability* sú zobrazené a popísané v tabuľke 4.11.

Databázová tabuľka *lesson*

Databázová tabuľka *lesson* reprezentuje konzultačnú hodinu lektora so študentom. Jednotlivé atribúty databázovej tabuľky *lesson* sú zobrazené a popísané v tabuľke 4.12.

4. SOFTVÉROVÁ ARCHITEKTÚRA A NÁVRH

Tabuľka 4.8: Atribúty databázovej tabuľky *homework_state*

P	Názov atribútu	Dátový typ	Popis
A	homework_state_ID	INT	Identifikátor stavu domácej úlohy, primárny kľúč tabuľky.
A	homework_state_name	VARCHAR(31)	Pomenovanie stavu domácej úlohy.

Tabuľka 4.9: Záznamy v databázovej tabuľke *homework_state*

homework_state_ID	homework_state_name	Popis
1	unconfirmed	Úloha zatiaľ nebola zadaná lektorom.
2	posted and unsolved	Úloha bola zadaná lektorom, ale zatiaľ nebola vyriešená.
3	solved	Úloha bola vyriešená študentom.

Tabuľka 4.10: Atribúty databázovej tabuľky *lecturer*

P	Názov atribútu	Dátový typ	Popis
A	isAdmin	BOOLEAN	Indikátor toho, či má lektor navyše aj administrátorské práva.
N	googleCalendarID	VARCHAR(127)	Identifikátor príslušného kalendára v aplikácii Google Calendar.
A	user_userID	INT	Identifikátor lektora. Cudzí kľúč tabuľky <i>user</i> .

Tabuľka 4.11: Atribúty databázovej tabuľky *lecturer_availability*

P	Názov atribútu	Dátový typ	Popis
A	lecturer_availability_ID	INT	Identifikátor záznamu o dostupnosti, primárny kľúč tabuľky.
A	time_from	DATETIME	Počiatkový čas dostupnosti lektora.
A	time_to	DATETIME	Koncový čas dostupnosti.
A	lecturer_user_userID	INT	Identifikátor lektora. Cudzí kľúč tabuľky <i>user</i> .

Tabuľka 4.12: Atribúty databázovej tabuľky *lesson*

P	Názov atribútu	Dátový typ	Popis
A	lessonID	INT	Identifikátor konzultačnej hodiny, primárny kľúč tabuľky.
A	reservation_time_from	DATETIME	Počiatkový čas rezervácie študenta.
A	reservation_time_to	DATETIME	Koncový čas rezervácie študenta.
N	real_starting_time	TIMESTAMP	Skutočný čas zahájenia hodiny.
N	real_ending_time	TIMESTAMP	Skutočný čas ukončenia hodiny.
N	notes_about_cancelling_lesson	VARCHAR(127)	Poznámky ku zrušeniu hodiny.
N	googleEventID	VARCHAR(127)	Identifikátor udalosti v aplikácii Google Calendar.
A	single_lecture_single_lecture_ID	INT	Identifikátor lekcie, z ktorej bola hodina vyrobená. Cudzí kľúč tabuľky <i>single_lecture</i> .
A	student_user_userID	INT	Identifikátor študenta. Cudzí kľúč tabuľky <i>user</i> .
A	lesson_state_lesson_state_ID	INT	Identifikátor stavu konzultačnej hodiny. Cudzí kľúč tabuľky <i>lesson_state</i> .

Databázová tabuľka *lesson_has_attachment*

Tabuľka *lesson_has_attachment* slúži na dekomponovanie vzťahu M:N medzi tabuľkami *lesson* a *attachment*. Databázová tabuľka obsahuje záznamy o tom, že ku konzultačnej hodine bol pripojený súbor. Jednotlivé atribúty databázovej tabuľky *lesson_has_attachment* sú zobrazené a popísané v tabuľke 4.13.

Tabuľka 4.13: Atribúty databázovej tabuľky *lesson_has_attachment*

P	Názov atribútu	Dátový typ	Popis
A	lesson_lessonID	INT	Identifikátor konzultačnej hodiny, ku ktorej patrí príloha. Cudzí kľúč tabuľky <i>lesson</i> .
A	attachment_attachmentID	INT	Identifikátor súboru, cudzí kľúč tabuľky <i>attachment</i> .

Databázová tabuľka *lesson_has_question*

Tabuľka *lesson_has_question* slúži na dekomponovanie vzťahu M:N medzi tabuľkami *lesson* a *question*. Databázová tabuľka obsahuje záznamy o tom, že ku konzultačnej hodine bol pripojený príklad a tiež to, akým spôsobom bol pripojený. Jednotlivé atribúty databázovej tabuľky *lesson_has_question* sú zobrazené a popísané v tabuľke 4.14.

Tabuľka 4.14: Atribúty databázovej tabuľky *lesson_has_question*

P	Názov atribútu	Dátový typ	Popis
A	lesson_lessonID	INT	Identifikátor konzultačnej hodiny, ku ktorej patrí príklad. Cudzí kľúč tabuľky <i>lesson</i> .
A	question_questionID	INT	Identifikátor otázky, resp. príkladu. Cudzí kľúč tabuľky <i>question</i> .
A	is_answer_visible	BOOLEAN	Indikátor toho, že sa pri príklade zobrazuje aj jeho riešenie.
A	are_solution_steps_visible	BOOLEAN	Indikátor toho, že sa pri príklade zobrazuje aj postup jeho riešenia.

Databázová tabuľka *lesson_state*

Databázová tabuľka *lesson_state* reprezentuje stav konzultačnej hodiny. Jednotlivé atribúty databázovej tabuľky *lesson_state* sú zobrazené a popísané v tabuľke 4.15. Záznamy v tejto databázovej tabuľke (tabuľka 4.16) budú pevne dané a nebude možné ich meniť.

Tabuľka 4.15: Atribúty databázovej tabuľky *lesson_state*

P	Názov atribútu	Dátový typ	Popis
A	lesson_state_ID	INT	Identifikátor stavu hodiny, primárny kľúč tabuľky.
A	lesson_state_name	VARCHAR(31)	Pomenovanie stavu hodiny.

Tabuľka 4.16: Záznamy v databázovej tabuľke *lesson_state*

lesson_state_ID	lesson_state_name	Popis
1	confirmed	Hodina bola zaplatená a naplánovaná.
2	in progress	Hodina prebieha.
3	successfully finished	Hodina úspešne skončila.
4	cancelled	Hodina bola zrušená.

Databázová tabuľka *payment_record*

Databázová tabuľka *payment_record* ukladá záznamy o platbách v systéme. Jednotlivé atribúty databázovej tabuľky *payment_record* sú zobrazené a popísané v tabuľke 4.17.

Tabuľka 4.17: Atribúty databázovej tabuľky *payment_record*

P	Názov atribútu	Dátový typ	Popis
A	payment_record_ID	INT	Identifikátor platobného záznamu, primárny kľúč tabuľky.
A	date_of_payment	TIMESTAMP	Čas platby.
A	credit	DOUBLE(8,2)	Množstvo kreditov, ktoré sú predmetom platby. Je to kladné číslo, ak sa jedná o prírastok na účte. V opačnom prípade je to záporné číslo.
N	lesson_lessonID	INT	Identifikátor konzultačnej hodiny, ktorej sa platba môže týkať. Cudzí kľúč tabuľky <i>lesson</i> .
A	user_userID	INT	Identifikátor používateľa. Cudzí kľúč tabuľky <i>user</i> .

Databázová tabuľka *question*

Databázová tabuľka *question* reprezentuje otázku, príklad alebo úlohu pripojenú ku konzultačnej úlohe alebo k domácej úlohe. Jednotlivé atribúty databázovej tabuľky *question* sú zobrazené a popísané v tabuľke 4.18.

Tabuľka 4.18: Atribúty databázovej tabuľky *question*

P	Názov atribútu	Dátový typ	Popis
A	questionID	INT	Identifikátor príkladu, primárny kľúč tabuľky.
A	is_confirmed	BOOLEAN	Indikátor toho, že príklad bol naozaj zadán a potvrdený.
A	question	TEXT	Textové inštrukcie príkladu.
A	difficultyLevel	INT	Obtiažnosť príkladu na stupnici od 1 (jednoduchá) až 5 (obtiažna).
N	solution_steps	TEXT	Textový popis riešenia príkladu.

Databázová tabuľka *question_has_attachment*

Tabuľka *question_has_attachment* slúži na dekomponovanie vzťahu M:N medzi tabuľkami *question* a *attachment*. Databázová tabuľka obsahuje záznamy o tom, že k príkladu bol pripojený súbor. Jednotlivé atribúty databázovej tabuľky *question_has_attachment* sú zobrazené a popísané v tabuľke 4.19.

Tabuľka 4.19: Atribúty databázovej tabuľky *question_has_attachment*

P	Názov atribútu	Dátový typ	Popis
A	question_questionID	INT	Identifikátor príkladu, ku ktorému patrí príloha. Cudzí kľúč tabuľky <i>question</i> .
A	attachment_attachmentID	INT	Identifikátor prílohy, cudzí kľúč tabuľky <i>attachment</i> .

Databázová tabuľka *question_has_topic*

Tabuľka *question_has_topic* slúži na dekomponovanie vzťahu M:N medzi tabuľkami *question* a *topic*. Databázová tabuľka obsahuje záznamy o tom, že príslušný príklad sa týka určitej témy. Jednotlivé atribúty databázovej tabuľky *question_has_topic* sú zobrazené a popísané v tabuľke 4.20.

Tabuľka 4.20: Atribúty databázovej tabuľky *question_has_topic*

P	Názov atribútu	Dátový typ	Popis
A	question_questionID	INT	Identifikátor príkladu, ktorý sa týka témy. Cudzí kľúč tabuľky <i>question</i> .
A	topic_topicID	INT	Identifikátor témy. Cudzí kľúč tabuľky <i>topic</i> .

Databázová tabuľka *single_lecture*

Databázová tabuľka *single_lecture* reprezentuje lekciiu zadanú učiteľom. Jednotlivé atribúty databázovej tabuľky *single_lecture* sú zobrazené a popísané v tabuľke 4.21.

Tabuľka 4.21: Atribúty databázovej tabuľky *single_lecture*

P	Názov atribútu	Dátový typ	Popis
A	single_lecture_ID	INT	Identifikátor lekcii, primárny kľúč tabuľky.
A	lecture_name	VARCHAR(63)	Názov lekcii.
N	description	TEXT	Textový popis lekcii.
A	isActive	BOOLEAN	Indikátor toho, že je lekcii aktívna.
A	lecturer_user_userID	INT	Identifikátor lektora. Cudzí kľúč tabuľky <i>user</i> .

Databázová tabuľka *single_lecture_price*

Databázová tabuľka *single_lecture_price* reprezentuje cenu lekcie zadanej učiteľom. Jednotlivé atribúty databázovej tabuľky *single_lecture_price* sú zobrazené a popísané v tabuľke 4.22.

Tabuľka 4.22: Atribúty databázovej tabuľky *single_lecture_price*

P	Názov atribútu	Dátový typ	Popis
A	single_lecture_price_ID	INT	Identifikátor ceny lekcie, primárny kľúč tabuľky.
A	price_per_hour	DOUBLE(8,2)	Čena lekcie na hodinu.
A	time_from	DATETIME	Čas, keď začala platiť táto cena.
N	time_to	DATETIME	Čas, v ktorom skončila platnosť tejto ceny. Jedná sa o hodnotu NULL v prípade, že je cena aktuálna a nebola nahradená inou.
A	single_lecture_single_lecture_ID	INT	Identifikátor lekcie. Cudzí kľúč tabuľky <i>single_lecture</i> .

Databázová tabuľka *student*

Databázová tabuľka *student* reprezentuje študenta. Tabuľka má jediný atribút „user_userID“ typu INT, ktorý je identifikátorom študenta. Jedná sa o cudzí kľúč tabuľky *user*. Tabuľka vznikla kvôli odlíšeniu používateľov v rolách študenta a lektora.

Databázová tabuľka *subject*

Databázová tabuľka *subject* reprezentuje predmet, napr. matematika, fyzika a pod. Jednotlivé atribúty databázovej tabuľky *subject* sú zobrazené a popísané v tabuľke 4.23.

Tabuľka 4.23: Atribúty databázovej tabuľky *subject*

P	Názov atribútu	Dátový typ	Popis
A	subjectID	INT	Identifikátor predmetu, primárny kľúč tabuľky.
A	subject_name	VARCHAR(63)	Názov predmetu.

Databázová tabuľka *subject_price*

Databázová tabuľka *subject_price* reprezentuje predvolenú cenu na lekcii daného predmetu. Jednotlivé atribúty databázovej tabuľky *subject_price* sú zobrazené a popísané v tabuľke 4.24.

Tabuľka 4.24: Atribúty databázovej tabuľky *subject_price*

P	Názov atribútu	Dátový typ	Popis
A	subject_price_ID	INT	Identifikátor predvolenej ceny predmetu, primárny kľúč tabuľky.
A	price	DOUBLE(8,2)	Predvolená cena.
A	time_from	DATETIME	Čas, keď začala platiť táto cena.
N	time_to	DATETIME	Čas, v ktorom skončila platnosť tejto ceny. Jedná sa o hodnotu NULL v prípade, že je cena aktuálna a nebola nahradená inou.
A	subject_subjectID	INT	Identifikátor predmetu, cudzí kľúč tabuľky <i>subject</i> .

Databázová tabuľka *topic*

Databázová tabuľka *topic* reprezentuje tému v rámci predmetu. Jednotlivé atribúty databázovej tabuľky *topic* sú zobrazené a popísané v tabuľke 4.25.

Tabuľka 4.25: Atribúty databázovej tabuľky *topic*

P	Názov atribútu	Dátový typ	Popis
A	topicID	INT	Identifikátor témy, primárny kľúč tabuľky.
A	topic_name	VARCHAR(63)	Názov témy.
A	subject_subjectID	INT	Identifikátor predmetu, pod ktorý téma patrí. Cudzí kľúč tabuľky <i>subject</i> .
N	suptopic_topicID	INT	Identifikátor nadtémy, cudzí kľúč tabuľky <i>topic</i> .

Databázová tabuľka *topic_has_single_lecture*

Tabuľka *topic_has_single_lecture* slúži na dekomponovanie vzťahu M:N medzi tabuľkami *single_lecture* a *topic*. Databázová tabuľka obsahuje záznamy o tom, že lekcii sa týka určitej témy. Jednotlivé atribúty databázovej tabuľky *topic_has_single_lecture* sú zobrazené a popísané v tabuľke 4.26.

Tabuľka 4.26: Atribúty databázovej tabuľky *topic_has_single_lecture*

P	Názov atribútu	Dátový typ	Popis
A	single_lecture_single_lecture_ID	INT	Identifikátor lekcie. Cudzí kľúč tabuľky <i>single_lecture</i> .
A	topic_topicID	INT	Identifikátor témy. Cudzí kľúč tabuľky <i>topic</i> .

Databázová tabuľka *user*

Databázová tabuľka *user* reprezentuje jeden používateľský účet v systéme. Jednotlivé atribúty databázovej tabuľky sú zobrazené a popísané v tabuľke 4.27.

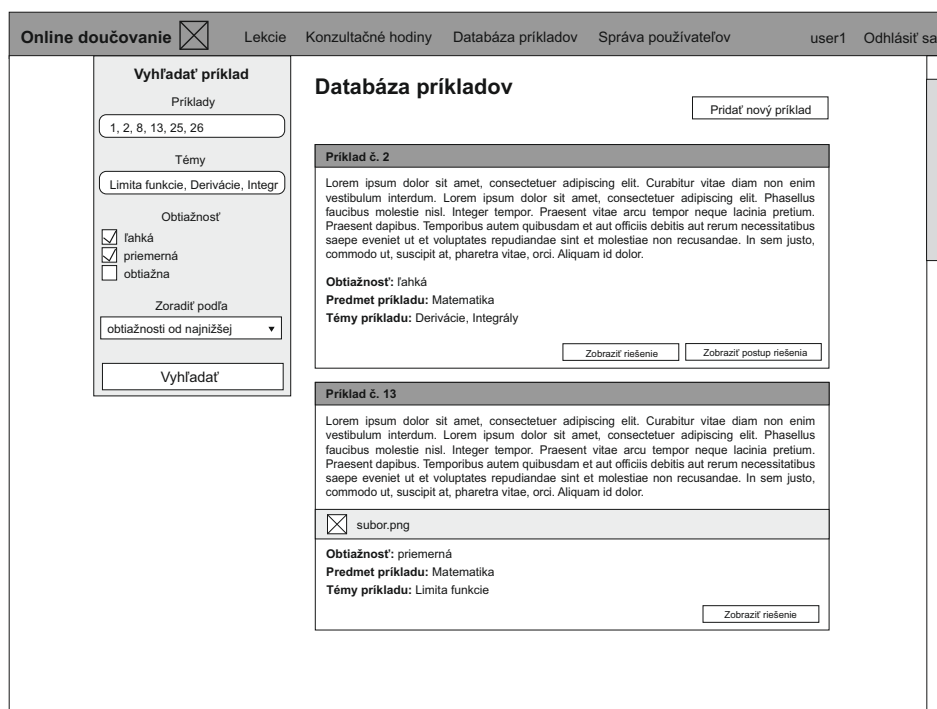
Tabuľka 4.27: Atribúty databázovej tabuľky *user*

P	Názov atribútu	Dátový typ	Popis
A	userID	INT	Identifikátor používateľa, primárny kľúč tabuľky <i>user</i> .
A	username	VARCHAR(63)	Používateľské meno.
A	password	VARCHAR(255)	Používateľské heslo.
A	email	VARCHAR(127)	Registračná emailová adresa používateľa.
A	first_name	VARCHAR(63)	Meno používateľa.
A	surname	VARCHAR(63)	Priezvisko používateľa.
A	is_active	BOOLEAN	Indikátor toho, či je používateľský účet aktívny alebo nie.
A	gender	VARCHAR(1)	Pohlavie používateľa (znak M v prípade muža, F v prípade ženy).
N	date_of_birth	DATE	Dátum narodenia používateľa.
N	description	TEXT	Doplňkový popis zadaný používateľom.
A	credit	DOUBLE(8,2)	Množstvo kreditov na účte používateľa.

4.4 Wireframe

Súčasťou návrhu systému býva často tzv. wireframe alebo „drôtený model“. Ide o náčrt webovej stránky, resp. jej obrazoviek, ktorý sa používa pri návrhu rozloženia webových komponent na stránke. Je podkladom pre ďalšie procesy vrátane návrhu dizajnu. Vzhľadom na to, že cieľom bakalárskej práce nie je tvorba dizajnu, uvádzam len príklad náčrtu pre zobrazenie databázy príkladov (obrázok 4.4).

4. SOFTVÉROVÁ ARCHITEKTÚRA A NÁVRH



Obr. 4.4: Wireframe pre vyhľadávanie a zobrazenie príkladov

Realizácia

5.1 Použité nástroje

V sekcii 4.1 bola zdôvodnená voľba implementačného jazyka (PHP a JavaScriptu) a dôvody použitia Nette frameworku a Bootstrap frameworku. Okrem týchto nástrojov boli však v práci použité i ďalšie nástroje priamo súvisiace s implementáciou aplikácie, resp. realizáciou konkrétnych prípadov použitia.

jQuery

jQuery [12] je knižnica pre jazyk JavaScript, ktorá sa zameriava na spoluprácu JavaScriptu a HTML. Knižnica umožňuje výber a úpravu elementov HTML stránky, jednoduchšiu prácu s efektami a animáciami a so svojim API rozhraním funguje naprieč väčšinou prehliadačov.

Hlavný dôvod, prečo bol v práci použitý jQuery je ten, že množstvo existujúcich pluginov je založených na využití tejto knižnice. Napríklad neskôr spomenutý plugin Bootstrap Datepicker taktiež využíva jQuery.

Ublaboo Datagrid

Ublaboo Datagrid [13] je plugin, ktorý má funkciu datagridu. Datagrid je komponenta, ktorá uľahčuje prácu s tabuľkovými dátami. Ak napríklad chceme pracovať s databázou, čo je v prípade informačného systému samozrejmosťou, potrebovali by sme dáta rozumne vypísať tak, aby im používateľ porozumel. Datagridy okrem zobrazovania dát umožňujú aj filtrovanie dát, radenie stĺpcov alebo vyhľadávanie podľa parametrov. Niektoré datagridy, vrátane Ublaboo Datagrid pluginu, umožňujú dáta v tabuľke aj priamo upravovať alebo vykonávať iné akcie nad jednotlivými riadkami tabuľky. Ublaboo Datagrid je komponentou vytvorenou priamo pre Nette.

Na obrázku 5.1 je zobrazený príklad takéhoto datagridu pre aktiváciu používateľských účtov administrátorom. V práci bola komponenta využitá

5. REALIZÁCIA

ďalej aj pri ďalších prípadoch použitia ako napr. na zobrazenie zoznamu konzultačných hodín, na zobrazenie transakcií alebo tiež na zobrazenie zoznamu lektorov kvôli zmene administrátorských práv.

Užívateľské meno	Email	Jméno	Příjmení	Role	Akce
julius123	julius123@gmail.com	Julius	Gál	student	Aktivovat účet Zrušit žádost
martins999	martins999@seznam.cz	Martin	Šeda	student	Aktivovat účet Zrušit žádost
simon.nemeth	simon.nemeth@gmail.com	Šimon	Németh	lektor	Aktivovat účet Zrušit žádost
marekabc	marekabc@post.cz	Marek	Svozil	student	Aktivovat účet Zrušit žádost
jaromirbartl	jaromirbartl@gmail.com	Jaromír	Bártl	lektor	Aktivovat účet Zrušit žádost
kami27	kami27@azet.sk	Kamila	Rejmanová	student	Aktivovat účet Zrušit žádost
zuzanafojtikova	zuzanafojtikova@post.cz	Zuzana	Fojtiková	lektor	Aktivovat účet Zrušit žádost
lauralaura	lauralaura@gmail.com	Laura	Altmanová	student	Aktivovat účet Zrušit žádost
vilma1	vilma1@seznam.cz	Vilma	Hrubešová	lektor	Aktivovat účet Zrušit žádost
ema.barankova	ema.barankova@gmail.com	Emílie	Baránková	lektor	Aktivovat účet Zrušit žádost

(Položky: 0 - 10 z 11)

◀ 1 2 ▶ 10 ▼

Obr. 5.1: Datagrid pre aktiváciu používateľských účtov

Ublaboo TreeView

Ublaboo TreeView [14] je ďalšou webovou komponentou vytvorenou priamo pre Nette framework. V tomto prípade slúži komponenta na zobrazenie stromovej štruktúry nejakých dát.



V práci je táto komponenta využitá na realizáciu zobrazovania štruktúry predmetov a tém. Tento príklad je viditeľný na obrázku 5.2.

Bootstrap Datepicker

Bootstrap Datepicker [15] je komponenta, ktorá slúži na interaktívny výber dátumu z kalendára. Pre svoju funkčnosť využíva javascriptovú knižnicu jQuery a Bootstrap framework. V systéme sa výber dátumov používa napr. pri zmene dátumu narodenia používateľa alebo pri dopĺňaní kreditov na účet študenta.

Select2

Select2 [16] je plugin, ktorý umožňuje prispôbiť si výberovú komponentu (HTML element `<select>`) podľa vlastných predstáv. Takto upravená komponenta umožňuje interaktívne vyhľadávanie, interaktívny výber a dokáže

	Předvolená cena	
> Fyzika	350,00 kreditů 	+ Přidat tému k předmětu
▼ Matematika	300,00 kreditů 	+ Přidat tému k předmětu
> Číselné posloupnosti		+ Přidat podtému
▼ Funkce		+ Přidat podtému
Exponenciální funkce		+ Přidat podtému
Goniometrické funkce		+ Přidat podtému
Konstantní a lineární funkce		+ Přidat podtému
Kvadratická funkce		+ Přidat podtému
Lineární lomená funkce		+ Přidat podtému
Logaritmická funkce		+ Přidat podtému
Mocninná funkce		+ Přidat podtému
Vlastnosti funkce		+ Přidat podtému
Kombinatorika		+ Přidat podtému
Komplexní čísla		+ Přidat podtému

Obr. 5.2: TreeView komponenta pre zobrazenie stromu predmetov a tém

čerpať z rôznych zdrojov dát. Select2 komponentu je navyše možné zobrazit aj v podobe komponent Bootstrap frameworku.

V práci je plugin Select2 použitý prakticky v každej časti kódu, kde používam výber z množiny hodnôt. Najviac sa však osvedčilo použiť plugin pri výbere viacerých hodnôt z množiny s využitím vyhľadávania v tejto množine hodnôt. Príklad použitia je možné vidieť na obrázku 5.3.

Bootstrap File Input

Bootstrap File Input [17] je plugin, ktorý je možné využiť pri nahrávaní súborov do systému. S využitím pluginu je možné okrem iného aj priamo spravovať nahrané súbory a zobrazovať ich. V systéme sú nahrané študijné materiály reprezentované ako obyčajné súbory, bez ohľadu na to, či je to obrázok, PDF súbor alebo iný formát súboru. Preto plugin používam v systéme len na úpravu formulára na nahranie súboru do systému.

Vyhledat lekce

Předměty

x Matematika

Témata

x Funkce x Číselné posloupnosti

Kom|

Matematika

Kombinatorika

Komplexní čísla

Seraille

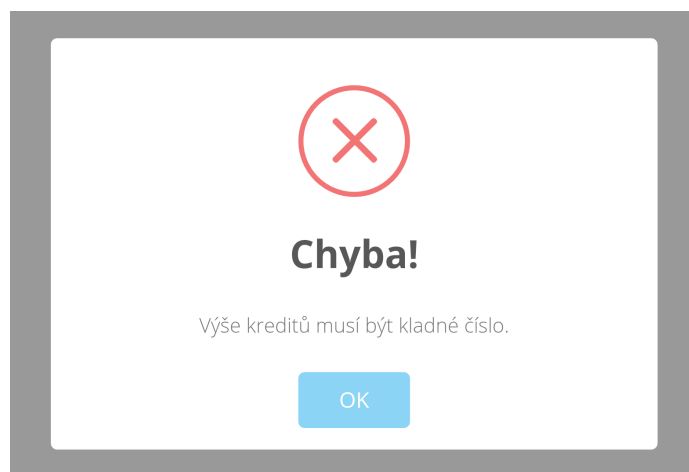
ceny od nejnižší

Vyhledat lekce

Obr. 5.3: Příklad interaktívnej Select2 komponenty

SweetAlert

SweetAlert [18] je plugin pre JavaScript, ktorý nahradzuje javascriptový príkaz `alert()`. Okrem toho plugin umožňuje v prípade potreby zobrazit s hláškou aj textové pole, ktorým je možné potvrdit určitú voľbu. Příklad použitia je uvedený na obrázkoch 5.4 a 5.5.



Obr. 5.4: Zobrazenie chybovej hlášky s využitím SweetAlert pluginu



Obr. 5.5: Zobrazenie upozornenia s textovým poľom s využitím SweetAlert

Google API

Google API je rozhranie poskytované firmou Google pre prístup k rôznym aplikáciám Google. V systéme je konkrétne využívané API pre aplikáciu Google Calendar. Prístup k API bol nakonfigurovaný dočasne pre konkrétny účet `onlinetutoringexample@gmail.com`. Systém tak s využitím API umožňuje vytvárať na tomto účte množstvo kalendárov a vytvárať v nich udalosti.

MathJax

MathJax je javascriptový engine, ktorý umožňuje zobrazovať text s využitím systému LaTeX. V systéme slúži štandardne na vypisovanie príkladov (ako matematické zápisy, rovnice a pod.), engine je však možné využiť v ľubovoľnej časti systému.

5.2 Ukážky kódu

5.2.1 Spolupráca vrstiev MVC

V sekcii 4.2.1 bola popísaná MVC architektúra v Nette frameworku a spôsob spolupráce jednotlivých vrstiev. V tejto sekcii uvádzam konkrétny príklad tejto spolupráce na prípade zobrazenia lekcie. Jedná sa len o jednoduché segmenty kódu, ale základ aplikácie funguje práve na tomto princípe.

Pre každý prezentér v Nette musí byť definovaná vlastná trieda. Ako už bolo spomenuté, triedy prezentérov sú uložené v adresári `/app/presenters`. Prezentér pre zobrazenie lekcí je nazvaný `LecturePresenter`. Každá akcia používateľa potom vedie na nejakú akciu prezentéru. V prípade, že chceme zobrazovať lekcie, je rozumné nazvať takúto akciu `show`. Každá akcia potom musí mať v triede prezentéru svoju metódu. Vo výsledku musí mať táto metóda názov `actionShow` a je možné volať ju s požadovanými parametrami. Napríklad v prípade, že chceme zobraziť lekcii s poradovým číslom 1 (číslo 1

5. REALIZÁCIA

je len umelým identifikátorom v príslušnej databázovej tabuľke), metóda bude vyzerat asi takto:

```
public function actionShow($lecture) {
    if (!$this->getUser()->isLoggedIn()) {
        $this->flashMessage("Na tuhle akci musíte být přihlášen.", '
            danger');
        $this->redirect('Sign:in');
    }

    if (!$lecture) {
        throw new Nette\Application\BadRequestException;
    }
}
```

Akcie prezentérov sú v Nette úzko spojené s URL a tzv. routingom. Napríklad ak do URL zadáme adresu `www.adresa_projektu.cz/www/lecture/show?lecture=1`, spustí sa práve akcia `show` prezentéru `LecturePresenter` a teda sa zavolá metóda `actionShow` s parametrom `lecture` rovným 1. Po skončení tejto metódy sa ešte zavolá druhá metóda, ktorá sa v tomto prípade bude nazývať `renderShow` a bude mať rovnaké parametre. Renderovacie metódy, na rozdiel od „action metód“, sa využívajú na prerušovanie.

```
public function renderShow($lecture)
{
    $lectureID = $lecture;
    $lecture = $this->lecturesManager->getLecture($lectureID);
    if (!$lecture) {
        throw new Nette\Application\BadRequestException;
    }

    $topics = $this->lecturesManager->getTopicsOfLecture($lectureID)
        ->fetchAll();
    $lecture->topics = $topics;

    // ... ďalšie posielanie dát do šablóny

    $this->template->lecture = $lecture;
}
```

Zavolanie tejto metódy spôsobí, že do šablóny odošleme dáta v premennej `$lecture`. Následne je nutné tieto dáta zobrazit v nejakej podobe. Pre každú akciu prezentéru musíme mať definovanú šablónu, ktorá dokáže tieto dáta vykreslit a zobrazit používateľovi. Šablóna je umiestnená v súbore `/app/presenters/templates/Lecture/show.latte`. V Latte šablóne

je využitý jazyk HTML a preddefinované makrá šablónovacieho systému Latte. Napríklad pre spracovanie dát z premennej je použitý nasledujúci jednoduchý kód, ktorý formátuje zodpovedajúci zoznam tém lekcie tak, aby boli oddelené čiarkami:

```
{* ... *}
{block content}
{* ... *}
<div>
  <strong>Témy lekcie: </strong>
  {php $len = count($lecture->topics)}
  {php $i = 0}
  {foreach $lecture->topics as $topic}
    {if $i == 0}
      {$topic->topic_name}
      {php $i++}
    {else}
      , {$topic->topic_name}
    {/if}
  {/foreach}
</div>
{* ... *}
{/block}
```

V metóde `renderShow` však bol použitý aj kód, ktorý smeroval do modelovej vrstvy: `$this->lecturesManager->getLecture($lectureID);`. Táto časť kódu by mala siahnuť do modelovej triedy `LectureManager` a opýtať sa databázy na informácie o lekcii s identifikátorom `lectureID`.

```
public function getLecture($lectureID) {

    $query = "
        SELECT *
        FROM " . self::TABLE_LECTURE . "
        JOIN " . self::TABLE_LECTURE_PRICE . "
        ON (" . self::COLUMN_LECTURE_ID . "
            = " . self::COLUMN_LECTURE_FK . "
            AND " . self::COLUMN_LECTURE_PRICE_TIME_TO . " IS NULL
            AND " . self::COLUMN_LECTURE_ID . " = ? )";

    return $this->database->query($query, $lectureID)->fetch();
}
```

5.2.2 Formuláre

Vzhľadom na charakter informačného systému a častú prácu s dátami, veľká časť aplikácie využíva formuláre na vkladanie dát do systému alebo vyhľadávanie.

5. REALIZÁCIA

V tejto sekcii je zobrazený príklad takého formulára na vyhľadávanie príkladov.

Formulár je reprezentovaný objektom triedy `SearchQuestionsForm`. Jedná sa vlastne o továreň na výrobu formulára. Rozhranie pre generovanú továreň je reprezentované triedou `ISearchQuestionsFormFactory`. Tento kód je uložený v súbore `/app/forms/SearchQuestionsForm.php`.

```
<?php

/* SearchQuestionsForm.php */

namespace App\Forms;

use App\Model;
use Nette\Application\UI;

class SearchQuestionsForm extends UI\Control
{
    private $topicsManager;

    public $onSuccess;

    public function __construct(Model\TopicsManager $topicsManager)
    {
        parent::__construct();
        $this->topicsManager = $topicsManager;
    }

    protected function createComponentForm()
    {
        $form = new UI\Form;

        $items = $this->topicsManager->getAllConfirmedQuestions(true);
        $form->addMultiSelect('questionIDs', 'Příklady', $items)
            ->setAttribute('class', 'form-control input-sm select2');

        $items = [];
        $subjects = $this->topicsManager->getAllSubjects();
        foreach ($subjects as $subject) {
            $subjectID = $subject->subjectID;
            $subject_name = $subject->subject_name;
            $topics = $this->topicsManager->getAllTopicsOfSubject(
                $subjectID);
            $topicsArray = [];
            foreach ($topics as $topicID => $topic) {
                $topicsArray[$topicID] = $topic['recursive_name'];
            }
            if (isset($topicsArray)) {
                natsort($topicsArray);
            }
            $items[$subject_name] = $topicsArray;
        }
    }
}
```



```

}
$form->addMultiSelect('topics', 'Témata', $items)
    >setAttribute('class','form-control input-sm select2');

$items = [1 => 'lehká', 2 => 'středně lehká',
    3 => 'průměrná', 4 => 'středně náročná', 5 => 'náročná'];
$form->addCheckboxList('difficulty','Obtížnost', $items);

$items = [ 1 => "por. čísla od nejnižšího",
    2 => "por. čísla od nejvyššího",
    3 => "obtížnosti od nejnižší",
    4 => "obtížnosti od nejvyšší"];
$form->addSelect('order',"Seřadit dle", $items)
    ->setAttribute('class','form-control input-sm select2')
    ->setDefaultValue(1);

$form->addSubmit('submit', 'Vyhledat příklady')
    ->setAttribute('class','btn btn-primary');

$form->onSuccess[] = function(UI\Form $form) {
    $values = $form->getHttpData();
    $this->onSuccess($values);
};

return $form;
}

public function render() {
    $this->template->render(__DIR__.'/searchQuestionsForm.latte');
}

}

interface ISearchQuestionsFormFactory
{
    /**
     * @return SearchQuestionsForm
     */
    function create();
}

```

Po zavolaní metody `create` sa vytvorí objekt formulára, ktorý obsahuje prvky na výber konkrétnych príkladov z databázy, na vyhľadávanie z množiny tém, vyhľadávanie podľa obtiažnosti a tiež element na radenie výsledkov. Nakoniec je do formulára pridané tlačítko na potvrdenie výberu, ktoré pri úspešnom odoslaní formulára zavolá funkciu `onSuccess`. Ako je možné vidieť z kódu, nielen akcia prezentéru, ale aj formulár a každá komponenta má definovanú svoju šablónu. Tá je v tomto prípade umiestnená v rovnakom adresári, v súbore `searchQuestionsForm.latte`. Súbor šablóny potom vyzerá približne

5. REALIZÁCIA

takto:

```
{form form}
  <form n:name=$form n:class="form-horizontal">
    <div class="form-group">
      {label questionIDs}
      {input questionIDs}
    </div>
    <div class="form-group">
      {label topics}
      {input topics}
    </div>
    <div class="form-group">
      {label difficulty}
      <div class="text-left difficultySelect">
        {input difficulty}
      </div>
    </div>
    <div class="form-group">
      {label order}
      {input order}
    </div>
    <div class="form-group">
      {label submit}
      {input submit}
    </div>
  </form>
{/form}
```

Z tohto príkladu je možné vidieť mapovanie PHP kódu na HTML elementy. Celý takto naprogramovaný formulár je však nutné ešte vykresliť. Napríklad ak chceme formulár zobraziť na nejakej obrazovke prezentéru `TopicsAndQuestionsManagerPresenter`, tak do triedy prezentéru vložíme metódu `createComponentSearchQuestionsForm`, ktorá sa postará o vytvorenie formulára v nejakej šablóne prezentéru.

```
protected function createComponentSearchQuestionsForm() {

    $form = $this->searchQuestionsFormFactory->create();
    $form->onSuccess[] = function($values) {
        // values su hodnoty ziskane z formulara
        $topics = "";
        if (array_key_exists("topics", $values)) {
            $topics = implode(",", $values['topics']);
        }
        $questionIDs = "";
        if (array_key_exists("questionIDs", $values)) {
            $questionIDs = implode(",", $values['questionIDs']);
        }
    }
}
```

```

    $difficultyLevels = "";
    if (array_key_exists("difficulty", $values)) {
        $difficultyLevels = implode(",", $values['difficulty']);
    }
    $order = $values['order'];
    $this->redirect("TopicsAndQuestionsManager:questions", 1,
        $questionIDs, $topics, $difficultyLevels, $order);
};

return $form;
}

```

V tele tejto metódy, okrem skutočného vytvorenia objektu formulára, definujeme aj udalosť pri úspešnom odoslaní formulára.

Vytvorený formulár potom stačí vykresliť v šablóne niektorej z akcií prezentéru `TopicsAndQuestionsManager` pomocou preddefinovaného Latte makra `{control searchQuestionsForm}`.

5.3 Ukážky obrazoviek aplikácie

Zoznam konzultačných hodín

Na obrázku 5.6 je zobrazená obrazovka pre zobrazenie zoznamu konzultačných hodín z pohľadu lektora. Na zobrazenie zoznamu bola použitá už spomenutá komponenta `Ublaboo Datagrid`. Zoznam je možné filtrovať podľa registrovaného času hodiny, názvu lekcie a tiež podľa stavu hodiny. Pri každej hodine je navyše informácia o tom, či na hodine bola zadaná domáca úloha a ak áno, tak tiež informácia o tom, či bola alebo nebola vyriešená študentom.

Zobrazenie konzultačnej hodiny

Na obrázku 5.7 je zobrazená obrazovka pre zobrazenie konkrétnej konzultačnej hodiny z pohľadu študenta. Je možné vidieť základné informácie o lekcii, z ktorej bola hodina vytvorená, tiež informácie o danej konzultačnej hodine – rezervovaný čas hodiny alebo stav hodiny. Ďalej je možné vidieť základné informácie o lektorovi s možnosťou zavolať mu na Skype, pripojené súbory a pripojený príklad. Vzhľadom na to, že bola hodina zahájená a stále beží, zobrazuje sa aj skutočne prebehnutý čas hodiny.

Registračná komponenta

Registračná komponenta má v systéme dve podoby. Na jednej strane slúži komponenta doslovne na registráciu študenta na vybranú lekciu. Na druhej strane slúži komponenta lektorovi na zmenu časov dostupnosti pre doučovanie. Obidva prípady sú zobrazené na obrázku 5.8, resp. 5.9.

5. REALIZÁCIA

Online doučování [Lekce](#) **Hodiny lekcí** [Správa tém a příkladů](#) [Správa uživatelů](#) [simon.nemeth](#) [Odhlásit se](#)

Hodiny lekcí

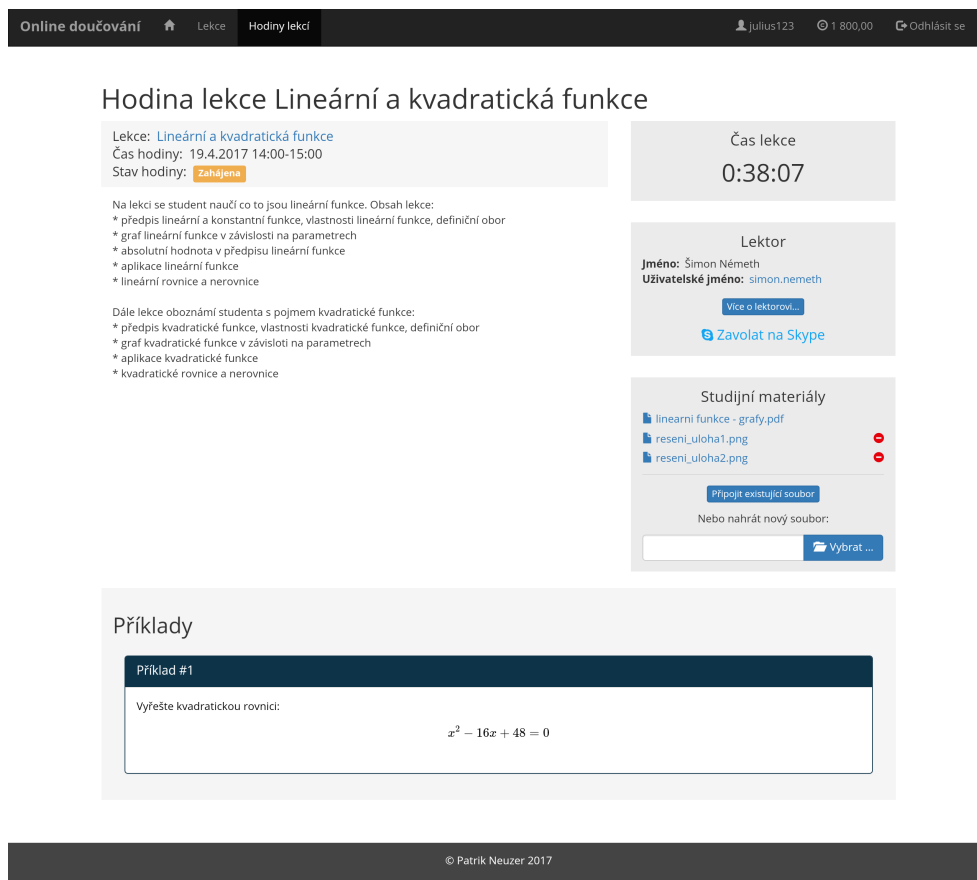
Datum a čas	Hodiny lekcí	Stav hodiny
		Zaplacená a naplánovaná
21.4.2017 15:00-16:00	<input type="checkbox"/> Hodina lecke Základy pravděpodobnosti	Zaplacená a naplánovaná
20.4.2017 19:00-20:30	<input type="checkbox"/> Hodina lecke Základy pravděpodobnosti	Zaplacená a naplánovaná
20.4.2017 15:30-17:00	<input type="checkbox"/> Hodina lecke Exponenciální a logaritmická funkce	Zaplacená a naplánovaná
19.4.2017 14:00-15:00	<input type="checkbox"/> Hodina lecke Lineární a kvadratická funkce	Zahájena
19.4.2017 13:00-14:00	<input type="checkbox"/> Hodina lecke Základy finanční matematiky	Úspěšně ukončena
19.4.2017 10:30-12:00	<input checked="" type="checkbox"/> Hodina lecke Komplexní čísla	Úspěšně ukončena
18.4.2017 19:30-20:30	<input type="checkbox"/> Hodina lecke Exponenciální a logaritmická funkce	Zrušena
18.4.2017 17:30-18:30	<input checked="" type="checkbox"/> Hodina lecke Základy pravděpodobnosti	Úspěšně ukončena
18.4.2017 16:00-17:30	<input checked="" type="checkbox"/> Hodina lecke Exponenciální a logaritmická funkce	Úspěšně ukončena
18.4.2017 11:30-12:30	<input type="checkbox"/> Hodina lecke Lineární a kvadratická funkce	Úspěšně ukončena

(Položky: 0 - 10 z 11) ← 1 2 → 10

© Patrik Neuzer 2017

Obr. 5.6: Obrazovka pre zobrazenie zoznamu konzultačných hodín

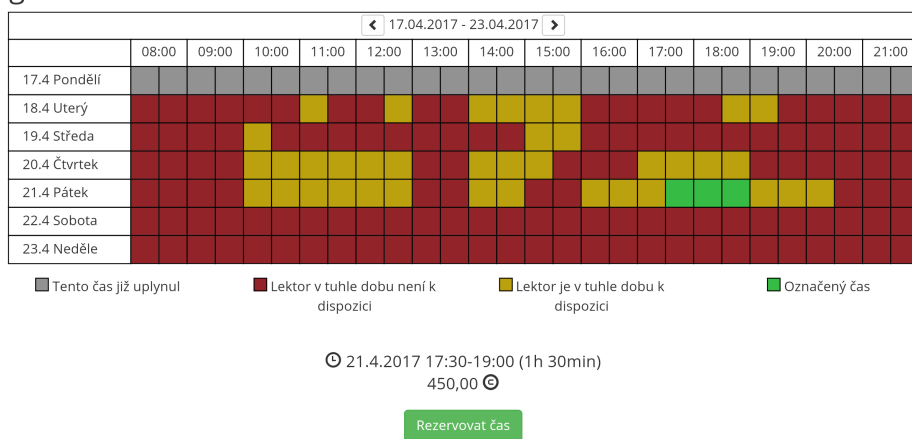
5.3. Ukážky obrazoviek aplikácie



Obr. 5.7: Obrazovka pre zobrazenie konzultačnej hodiny

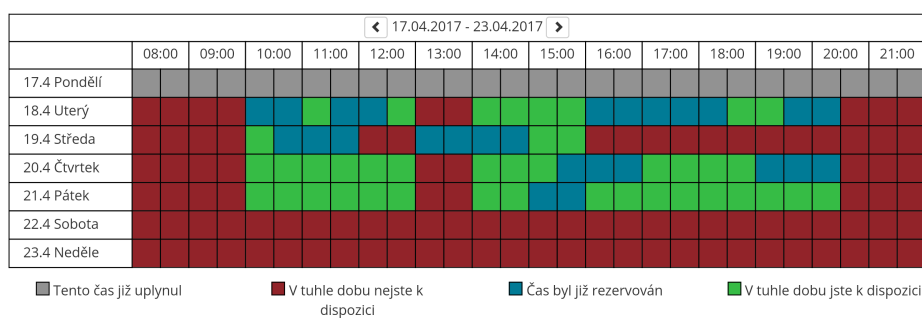
5. REALIZÁCIA

Registrovat se



Obr. 5.8: Registračná komponenta

Změna dostupnosti



Obr. 5.9: Komponenta na změnu dostupnosti lektora

Testovanie

Táto kapitola sa venuje procesu testovania aplikácie počas jej vývoja a tiež nástrojmi použitými na otestovanie výstupov.

6.1 Lokálne testovanie

Na lokálne testovanie bol použitý nástroj XAMPP. Jedná sa o open-source balík, ktorý umožňuje naprieč rôznymi operačnými systémami inštaláciu softvérového webového serveru Apache, tiež MySQL a PHP. Vzhľadom na to, že aplikácia bude nakoniec nasadená na reálny webový server využívajúci PHP a MySQL, bol XAMPP jasnou voľbou.

6.2 Testovanie prípadov použitia podľa scenárov

Hlavnú časť procesu testovania zabralo testovanie podľa jednotlivých prípadov použitia funkčných požiadaviek.

Najväčšie problémy boli zaznamenané v procese registrácie na lekciu (UC18), konkrétne s vytvorením registračnej komponenty. Problémom bolo korektné ajaxové spracovanie. Ďalšie väčšie problémy boli zaznamenané v procese konfigurácie Google účtu a prácou s Google kalendárom. Zvyšné problémy, súvisiace s viacerými prípadmi použitia, boli zaznamenané v modelovej vrstve pri práci s databázou.

Chyby zaznamenané počas testovania prípadov použitia boli samozrejme odstránené. Prípady použitia tak možno vyhlásiť za funkčné a funkčné požiadavky za splnené.

6.3 Používateľské testovanie

Používateľské testovanie je testovanie využitím reálnych používateľov. Do testovania boli zaradení dobrovoľníci, ktorých úlohou bolo vyskúšať si aplikáciu,

kompletne si ju preklikať a výsledky nahlásiť autorovi projektu.

Testovanie odhalilo viacero chýb. Najväčšia odhalená chyba súvisela s procesom zobrazovania konzultačnej hodiny. Problémom bolo zobrazovanie informácií na základe rôznych rolí používateľov, rôzneho stavu lekcie, priradených príkladov, priradených súborov, stavu domácej úlohy alebo priradených príkladov a súborov k domácej úlohe. Množstvo rôznych faktorov tak ovplyvnilo korektné zobrazenie informácií o hodine, čo testovanie podľa scenárov neodhalilo.

Zvyšné nahlásené chyby boli skôr menšieho rozsahu. Jednalo sa napríklad o nesprávne zobrazovanie celého mena používateľa alebo nesprávne zobrazovanie popisov lekcie. Iné nahlásené výsledky menšieho rozsahu boli dizajnového či estetického charakteru. Vzhľadom na to, že cieľom práce nie je tvorba dizajnu, tieto nedostatky boli buď opomenuté alebo boli upravené len v menšej miere.

Spomenuté chyby však boli odstránené a testovanie je tak možné prehlásiť za úspešné.

6.4 Testy vo webových prehliadačoch

Aplikácia bola navyše otestovaná vo viacerých webových prehliadačoch. Testovanie skúmalo beh aplikácie na webových prehliadačoch spomenutých v nefunkčnej požiadavke N1:

- Google Chrome vo verzii 48,
- Mozilla Firefox vo verzii aspoň 46.0,
- Opera vo verzii aspoň 35,
- Microsoft Edge vo verzii aspoň 11,
- Safari pre Mac vo verzii aspoň 9.

Výsledky testovania odhalili takmer bezproblémový beh v týchto prehliadačoch. Drobným problémom môže byť rozdielne vykresľovanie v rôznych prehliadačoch. Malé rozdiely vo vykresľovaní však nijak neobmedzujú funkčnosť aplikácie.

Testovanie ale prebiehalo aj na starších webových prehliadačoch. Výsledky testovania v starších prehliadačoch odhalili zopár chýb súvisiacich s funkčnosťou aplikácie. Hlavným problémom niektorých starších prehliadačov bola klikateľnosť registračnej komponenty a problematické zobrazenie textu spracovaného knižnicou MathJax a systémom LaTeX.

Vo všeobecnosti je však možné nefunkčnú požiadavku označiť za kompletne splnenú.

Záver

Cielom bakalárskej práce bola najskôr analýza existujúcich riešení v oblasti online doučovacích systémov a systémov na podporu online doučovania. Z analýzy však bolo jasné, že neexistuje vhodné riešenie a jediným správnym riešením je vytvoriť nový systém na mieru zadávateľa. Následne sa autor v práci venoval analýze požiadaviek zadávateľa, išlo predovšetkým o proces správy používateľov, proces plánovania a správy lekcí a proces počítania cien a fakturácie. Súčasťou analýzy požiadaviek bolo rozdelenie požiadaviek a následné rozloženie funkčných požiadaviek na prípady použitia.

Na základe analýzy požiadaviek bol navrhnutý nový informačný systém. V práci boli spomenuté hlavné dôvody voľby implementačného jazyka PHP a použitých frameworkov Nette a Bootstrap, dôvody výberu architektúry a popis architektúry súvisiacej priamo so zvoleným frameworkom. Súčasťou návrhu systému bol tiež návrh spôsobu ukladania dát, či už sa jednalo o uloženie textových dát v databáze alebo o spôsob uloženia súborov.

Okrem toho sa práca ďalej venovala procesu realizácie systému. Popísané boli ďalšie použité nástroje súvisiace s implementáciou webovej aplikácie. Súčasťou kapitoly boli tiež ukážky priamo z aplikácie, išlo o ukážky kódu a ukážky obrazoviek aplikácie. Záver práce sa venoval procesu testovania aplikácie.

Výsledkom bakalárskej práce je teda funkčný informačný systém na podporu online doučovania. Prínos práce smeruje hlavne k jeho používateľom – lektorom a študentom a samozrejme k zadávateľovi projektu, ktorému by mal systém priniesť ekonomický prospech.

Literatúra

- [1] Molnár, Z.; Juřenčák, B.; Riessler, P.; a i.: *Informační systém podniku*. Zlín: Univerzita Tomáše Bati ve Zlíně, Fakulta managementu a ekonomiky, prvé vydanie, 2001, ISBN 80-238-6525-0.
- [2] Tutorhub – Online tutoring, online tuition, online tutors [online]. 2017. Dostupné z: <https://tutorhub.com/>
- [3] BuddySchool: Online Tutoring Platform [online]. 2017. Dostupné z: <https://buddyschool.com/>
- [4] Mathematicator: Klub přátel matematiky [online]. 2017. Dostupné z: <http://mathematicator.com/>
- [5] Arlow, J.; Neustadt, I.: *UML 2 a unifikovaný proces vývoje aplikací*. Brno: Computer Press, a.s., prvé vydanie, 2007, ISBN 978-80-251-1503-9.
- [6] How do you write software requirements? What are software... [online]. 2011. Dostupné z: <https://hubtechinsider.wordpress.com/2011/07/28/how-do-you-write-software-requirements-what-are-software-requirements-what-is-a-software-requirement/>
- [7] Procházka, D.: *PHP 6 – začínáme programovat*. Praha: Grada Publishing a.s., prvé vydanie, 2012, ISBN 978-80-247-3899-4.
- [8] Bootstrap: The world's most popular mobile-first and responsive front-end framework [online]. 2017. Dostupné z: <http://getbootstrap.com/>
- [9] MVC aplikace & presentery [online]. 2017. Dostupné z: <https://doc.nette.org/cs/2.4/presenters>
- [10] Database: Selection [online]. 2017. Dostupné z: <https://latte.nette.org/cs/>

LITERATÚRA

- [11] Latte [online]. 2017. Dostupné z: <https://doc.nette.org/cs/2.4/presenters>
- [12] jQuery API Documentation [online]. 2017. Dostupné z: <https://api.jquery.com/>
- [13] Datagrid [online]. 2017. Dostupné z: <https://ublaboo.org/datagrid/>
- [14] Tree view [online]. 2017. Dostupné z: <https://ublaboo.org/datagrid/tree-view>
- [15] Bootstrap-datepicker [online]. 2017. Dostupné z: <https://bootstrap-datepicker.readthedocs.io/en/latest/index.html>
- [16] Select2 [online]. 2017. Dostupné z: <https://select2.github.io/>
- [17] Bootstrap File Input [online]. 2017. Dostupné z: <http://plugins.krajee.com/file-input>
- [18] SweetAlert [online]. 2017. Dostupné z: <http://t4t5.github.io/sweetalert/>

Zoznam použitých skratiek

- AJAX** Asynchronous Javascript and XML
- API** Application Programming Interface
- ČVUT** České vysoké učení technické
- HTML** HyperText Markup Language
- IEEE** Institute of Electrical and Electronics Engineers
- IS** Informačný systém
- MVC** Model-View-Controller
- MVP** Model-View-Presenter
- PDF** Portable Document Format
- PHP** PHP: Hypertext Preprocessor (rekurzívna skratka)
- SQL** Structured Query Language
- SRS** System Requirements Specification
- UC** Use Case
- UML** Unified Modeling Language
- URL** Uniform Resource Locator
- XML** eXtensible Markup Language

Obsah priloženého CD

	readme.txt.....	stručný popis obsahu CD
	src	
		impl.....zdrojové súbory informačného systému
		create_script.sql.....skript na vytvorenie databázy
	thesis	
		src adresár so zdrojovými súbormi bakalárskej práce vo formáte LaTeX
		BP_Neuzer_Patrik_2017.pdf.....text práce vo formáte PDF