



Tvorba softvérového systému v tíme

Automatizovaná podpora predmetu z oblasti programovania

Dokumentácia k projektu



Tím č. 11

Pedagogický vedúci projektu: RNDr. Valéria Šimáková

Bc. Tomáš Lauro, 1. ročník, Softvérové inžinierstvo
Bc. Michal Lulčo, 1. ročník, Softvérové inžinierstvo
Bc. Tomáš Majer, 1. ročník, Softvérové inžinierstvo
Bc. Vladimír Martinka, 1. ročník, Softvérové inžinierstvo
Bc. Michal Pajbach, 1. ročník, Softvérové inžinierstvo
Bc. Michal Sokolský, 1. ročník, Softvérové inžinierstvo

Dátum vypracovania: 14.12. 2009

Kontakt na tím: extremexpresstp@gmail.com

História vývoja dokumentu

Dátum zmeny	Verzia dokument	Opis	Autor
24.10.	0.01	Vytvorenie základnej kostry dokumentu, prebratie a doplnenie názvov kapitol analýzy	M. Pajbach
24.10.	0.02	Rozšírenie kostry analýzy o komentáre k jednotlivým častiam	V.Martinka
24.10-25.10	0.03	Pridanie adaptívnych výučbových systémov, analýza bakalárskej práce Dávida Chalupu, úvod ku generovaniu testov	V.Martinka
25.10	0.04	Analýza bakalárskej práce Michala Noskoviča	M. Sokolský
24.10-25.10	0.05	Triologické učenie, analýza cieľového prostredia	M. Pajbach
25.10-26.10	0.06	Popis typov otázok	M. Pajbach
28.10.	0.07	Pridanie analýzy webtestera	T. Lauro
29.10.	0.08	História e-learningu	M. Pajbach
30.10	0.09	Dátový model, dátový model pre používateľov	T. Majer
30.10	0.10	Revízia dokumentu, úprava štýlov	T. Lauro
30.10.-1.11.	0.11	Analýza tvorby a automatického opravovania zadaní	M. Pajbach
31.10.	0.12	Doplnenie analýzy automatického generovania testov, popis Yonbanu	M. Pajbach
31.10.	0.13	Vytvorenie návrhu, architektúra systému	M. Lulčo, M. Sokolský
31.10	0.14	Dátový model pre zadania	T. Majer
31.10-1.11.	0.15	Vytvorenie prípadov použitia UC07-UC-17	T. Lauro
31.10.-1.11.	0.16	Vytvorenie šablóny pre tabuľky prípadov použitia, vytvorenie prípadov použitia UC01-06,18-19, charakteristika rolí	M. Pajbach
31.10.-1.11.	0.17	Dátový model pre obrázky, dátový model pre otázky	T. Majer

Dátum zmeny	Verzia dokument	Opis	Autor
1.11	0.18	Vytvorenie prípadov použitia UC20-22	V.Martinka
1.11	0.19	Elektronické skriptá, doplnená analýza DŠA, prípady použitia pre administrátora, pridaný všeobecný úvod, porovnanie existujúcich e-learningových systémov	V.Martinka
1.11.	0.20	Základné vetvy e-learningových systémov	M. Pajbach
1.11.	0.21	Vytvorenie popisu dátového modelu, revízia použitých prostriedkov	T. Majer
1.11.-2.11.	0.22	Doplnenie rozpracovaných kapitol v analýze, opravené chyby, úprava	M. Pajbach
1.11.-2.11.	0.23	Diagram prípadov použitia	V.Martinka
1.11.-2.11.	0.24	Úpravené niektoré kapitoly, oprava štýlov a gramatických chýb	T. Lauro
2.11.	0.25	Úprava niektorých kapitol, použitá notácia	V.Martinka
2.11.	0.26	Kontrola chýb, revízia	M. Lulčo
2.11.	1.00	Finálna úprava, kontrola chýb	M. Sokolský
21.11.	1.01	Úprava hlavičiek a číslovania	T. Lauro
12.12.	1.02	Zmenený obsah a číslovanie	T. Lauro
12.12-14.12	1.4	Dokumentácia k prototypu	V. Martinka, M. Pajbach
13.12-14.12	1.7	Používateľská príručka	V. Martinka, M. Pajbach
13.12-14.12	2.0	Revízia dokumentácie	V. Martinka, M. Pajbach, T. Lauro

Obsah

0	ÚVOD	I
	0.1 Účel a rozsah dokumentu.....	i
	0.2 Prehľad dokumentu.....	i
	0.3 Odkazy a zdroje.....	i
	0.4 Použité skratky.....	ii
	0.5 Použitá notácia.....	iii
1	ANALÝZA	1-1
	1.1 E-learning.....	1-1
	1.2 Porovnanie existujúcich e-learningových prostredí.....	1-5
	1.3 Analýza existujúcich prác.....	1-6
	1.4 Analýza automatického generovania testov.....	1-9
	1.5 Analýza tvorby a automatického opravovania zadaní.....	1-14
	1.6 Elektronické skriptá.....	1-15
	1.7 Analýza cieľového prostredia (predmet DŠA).....	1-16
2	ŠPECIFIKÁCIA	2-1
	2.1 Funkcionálne požiadavky.....	2-1
	2.2 Iné požiadavky.....	2-1
	2.3 Charakteristika rolí v systéme.....	2-1
	2.4 Diagram prípadov použitia.....	2-2
	2.5 Prípady použitia študenta.....	2-3
	2.6 Prípady použitia pedagóga.....	2-6
	2.7 Prípady použitia administrátora.....	2-10
3	NÁVRH	3-1
	3.1 Dátový model.....	3-1
	3.2 Architektúra systému.....	3-5
4	PROTOTYP	4-1
	4.1 Ciele prototypovania.....	4-1
	4.2 Vybrané časti systému na prototypovanie.....	4-1
	4.3 Scenár použitia systému.....	4-2
	4.4 Aplikačná vrstva.....	4-2
	4.5 Prezentačná vrstva.....	4-3
	4.6 Zhrnutie.....	4-4
5	POUŽÍVATEĽSKÁ PRÍRUČKA	5-1
	5.1 Inštalácia.....	5-1
	5.2 Použitie.....	5-3

0 Úvod

Projekt vznikol v rámci predmetu Tímový projekt na Slovenskej technickej univerzite – Fakulte informatiky a informačných technológií v akademickom roku 2009/2010. Bol vypracovaný tímom Extreme Xpress pod vedením RNDr. Valérie Šimákovéj. Cieľom projektu je vytvorenie podporného prostriedku pre výučbu predmetu DŠA, ktorý sa vyučuje na Fakulte informatiky a informačných technológií STU.

0.1 Účel a rozsah dokumentu

Dokument slúži ako technická dokumentácia k tímovému projektu Automatizovaná podpora predmetu z oblasti programovania. Je určený predovšetkým pre účely predmetu Tímový projekt, avšak môže slúžiť aj ako referenčný materiál pre projekty s podobným zameraním. Pokrýva analýzu cieľovej oblasti, špecifikáciu a návrh systému a jeho súčasťou bude aj opis implementácie s používateľskou príručkou.

0.2 Prehľad dokumentu

Prvá časť dokumentácie obsahuje analýzu problémovej oblasti relevantnej k nášmu projektu. Obsahuje stručný úvod do elektronického vzdelávania, jeho históriu, a taktiež niektoré zaujímavé trendy, ktoré by bolo možné v rámci nášho projektu použiť.

Dalej porovnáva niektoré e-learningové systémy používané v praxi a tiež bakalárske práce a tímové projekty vytvorené v rámci našej fakulty, ktoré majú podobné zameranie.

V časti analýzy sú rozobraté jednotlivé e-learningové komponenty, ktoré plánujeme zahrnúť do nášho projektu a nakoniec je uvedený prehľad súčasného stavu v predmete DŠA, pre ktorý systém vyvíjame.

Druhú časť dokumentácie tvorí špecifikácia požiadaviek na systém. Obsahuje funkcionálne a iné požiadavky, ktoré by projekt mal spĺňať. Sú tu špecifikované používateľské roly a k nim priradené prípady použitia pre jednotlivé typy používateľov.

Tretia časť pozostáva z konkrétneho návrhu riešenia projektu. Jej hlavná časť opisuje dátový model údajov. Napokon je popísaný návrh architektúry vytváraného systému a opis jednotlivých častí.

V štvrtej sekcii sa nachádza dokumentácia k prototypu, ktorý bol vytvorený koncom zimného semestra 2009/2010.

Nakoniec v piatej kapitole je priložená používateľská príručka k prototypu.

0.3 Odkazy a zdroje

- [1] BABIČ F., FURDÍK K., PARALIČ J., WAGNER J.: *Podpora procesov tvorby nových znalostí*. [online]. [použ. 2009-10-25]. Dostupný z: <http://web.tuke.sk/feit/cit/furdik/publik/Datakon07_babic-furdik-paralic-wagner_2007_PoZnaT.pdf>
- [2] *Knowledge Practices Laboratory*. [online]. [použ. 2009-10-24]. Dostupný z: <<http://www.kp-lab.org/>>
- [3] BABIČ F., PARALIČ J.: *Knowledge practices Laboratory (KP-Lab) Project*. In: *Znalosti 2007*. ISBN 978-80-248-1279-3
- [4] *Akademický informačný systém*. [online]. [použ. 2009-10-25]. Dostupný z: <<https://is.stuba.sk/>>
- [5] *Moodle FIIT STU*. [online]. [použ. 2009-10-25]. Dostupný z: <<http://moodle.fiit.stuba.sk/>>
- [6] Kopecký, K.: *Základy e-Learningu. Interaktívny CD-ROM*. Univerzita Palackého

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

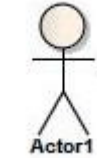

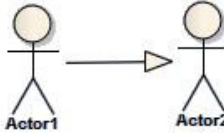
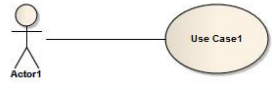
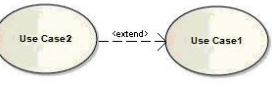
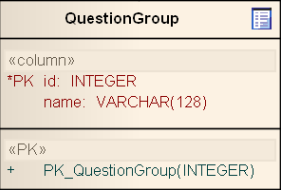


- v Olomouci, Olomouc, 2005. [online]. [použ. 2009-10-29]. Dostupný z: <<http://www.net-university.cz/data/cdrom/>>
- [7] PAL, A.: *Typy a štruktúry údajov*. Diplomová práca. Dostupný z: <<http://www2.fiit.stuba.sk/~simakova/Diplomovka/index.htm>>
- [8] *Glossary of education-related terms (A-C)*. [online]. [použ. 2009-11-01]. Dostupný z: <[http://prod.palette.tudor.lu/FormaHeticTest/bayfac/download.php/Glossaryofeducation-relatedterms\(A-C\).doc](http://prod.palette.tudor.lu/FormaHeticTest/bayfac/download.php/Glossaryofeducation-relatedterms(A-C).doc)>
- [9] *E-Learning*. [online]. [použ. 2009-11-01]. In: Wikipedia. Dostupný z: <<http://en.wikipedia.org/wiki/E-learning>>
- [10] Dostupný z: <<http://www.slis.kent.edu/~mzeng/Metadata-course.htm>>
- [11] ŠIMKO M.: *Domain Concepts Relationships Discovery*. Dostupný z: <<http://www2.fiit.stuba.sk/~bielik/courses/projects/simko08.html>>
- [12] ŠIMKO M., BIELIKOVÁ M.: *Automatizované získavanie metadát výučbového obsahu a vzťahov medzi konceptmi*. In: *Znalosti 2009*. ISBN 978-80-227-3015-0
- [13] *Moodle - A Free, Open Source Course Management System for Online Learning*. Dostupný z: <<http://moodle.org/>>
- [14] *Univerzitní informační systém*. Dostupný z: <http://is.mendelu.cz/dok_server/slozka.pl?id=15863;download=3965;lang=sk>
- [15] *What Is Wiki*. Dostupný z: <<http://wiki.org/wiki.cgi?WhatIsWiki>>
- [16] NOSKOVIČ, M.: *Návrh virtuálneho výučbového laboratória*. Bakalárska práca. STU v Bratislave FIIT, 2009.
- [17] CHALUPA, D.: *Návrh virtuálneho výučbového laboratória*. Bakalárska práca. STU v Bratislave FIIT, 2009.
- [18] BYSTRÍČAN M., DEMČÁK J., FILLO P., FRIDRICH P., KILÍK M., OLÁH M.: *Odovzdávanie, kontrola a hodnotenie zadaní*. Tímový projekt. STU v Bratislave FIIT, 2009.
- [19] *Online judge*. [online]. [použ. 2009-10-30]. In: Wikipedia. Dostupný z: <http://en.wikipedia.org/wiki/Online_judge>
- [20] *Sphere Online Judge*. [online]. [použ. 2009-10-30]. Dostupný z: <<http://www.spoj.pl/>>
- [21] *ProFIIT 2009*. [online]. [použ. 2009-10-30]. Dostupný z: <<http://profiit.fiit.stuba.sk/>>
- [22] *CakePHP CookBook*. Dostupný z <<http://book.cakephp.org/>>

0.4 Použité skratky

AIS	<i>Akademický informačný systém STU</i>
CSV	<i>Comma – separated values</i>
DŠA	<i>Dátové štruktúry a algoritmy</i>
IP	<i>Internet protocol</i>
LCMS	<i>Learning content management system</i>
LDAP	<i>Lightweight Directory Access Protocol</i>
LMS	<i>Learning management system</i>
MVC	<i>Model View Controller</i>
WWW	<i>World Wide Web</i>

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

0.5 Použitá notácia

	<p>Hráč</p>
	<p>Prípád použitia</p>
	<p>Hráč Actor1 je konkretizácia hráča Actor2 Hráč Actor2 je generalizácia hráča Actor1</p>
	<p>Hráč používa...</p>
	<p>Prípád použitia 1 rozširuje prípad použitia 2</p>
	<p>Entita</p>
	<p>Vzt'ah 1:N</p>
	<p>Vzt'ah M:N</p>

1 Analýza

V časti analýza sa zameriame na vývoj, históriu a ciele e-learningu. Následne podrobnejšie rozoberieme jednotlivé druhy e-learningových systémov a technológií v nich používaných. Zároveň zosumarizujeme poznatky získané v rámci bakalárskych a tímových projektov, ktoré boli vypracované na fakulte.

Uvedieme aj analýzu automatického generovania a opravovania testov a zadaní študentov, pretože touto oblasťou sa budeme v projekte zaoberať.

V závere uvedieme stručnú analýzu predmetu Dátové štruktúry a algoritmy, pre ktorý tento projekt vypracovávame.

1.1 E-learning

Klasické vzdelávanie založené na kontakte medzi študentmi a vyučujúcimi a štúdiu učebných materiálov bolo s nástupom informačných technológií doplnené o nové formy, ktoré súhrnne nazývame elektronické vzdelávanie alebo e-learning. Ďalšie možnosti elektronického vzdelávania sa objavili rozšírením počítačových sietí vo vzdelávacom procese, najmä rozšírením Internetu. Samotný pojem e-learningu sa začal prakticky používať až v deväťdesiatych rokoch, aj keď o rôznych formách elektronického vzdelávania možno hovoriť už skôr.

História e-learningu

Vysokoškolské prostredie bolo z hľadiska presadzovania vzdelávania podporovaného výpočtovou technikou priekopnícke a experimentovať sa začalo už koncom šesťdesiatych rokov s tzv. výučbovými automatmi. Tieto výučbové systémy sa nachádzali aj v bývalom Československu. Prvé pokusy ale neboli príliš úspešné.

Až s príchodom osobných počítačov sa objavujú prvé úspešné systémy, napr. počítačom vyhodnocované testy. Ďalšie pokroky prichádzajú s nástupom multimediálnych počítačov a technológií ako cd-rom, resp. veľkokapacitných nosičov dát. Tieto technológie umožnili pútavý a interaktívny spôsob výučby. [6]

Pojem, ako aj rozvoj e-learningu je silno zviazaný so vznikom technológie WWW. Jeho obsah a prevedenie využívali možnosti tejto technológie. Bol navrhnutý tak, aby učiaci sa získavali svoje vedomosti prostredníctvom online obsahu, interakcií, úloh, či simulácií.

V posledných piatich rokoch hovoríme o nástupe nových webových technológií, tzv. Webe 2.0, ktorý sa vyznačuje najmä zvýšeným podielom používateľskej interakcie a väčším podielom používateľov na samotnej tvorbe obsahu. Toto odráža aj vznik novej generácie e-learningu, tzv. E-Learning 2.0, ktorý využíva princípy Webu 2.0. [9]

Ciele e-learningu

Hlavným cieľom e-learningu je zjednotiť všetko čo s ním súvisí na jedno miesto – do jedného elektronického systému, ku ktorému budú mať všetci účastníci jednoduchý prístup. Taktiež by mal využívať možnosti, ktoré mu elektronické prostredie poskytuje na vytváranie efektívnejších učebných metód a materiálov za účelom podpory budovania znalostnej spoločnosti.

Použitie e-learningu v praxi

Najväčšou výhodou elektronického vzdelávania je podstatné zníženie zaťaženia pedagógov, ktorým sa uľahčujú úlohy tvorby, manažmentu a opravovania testov, pridelenia a kontroly zadaní, evidencie hodnotení študentov a mnohých iných činností, ktorým by za normálnych okolností museli venovať niekoľkonásobne viac času. Študenti majú sústredené všetky aktivity v rámci predmetu vo vzdelávacom systéme, čím sa aj im zjednodušuje práca.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Nevýhodou e-learningu je predpoklad, že pre efektívne fungovanie má každý študent a pedagóg prístup k počítaču s internetovým pripojením. Ak nemá, môže byť hendikepovaný.

Triologické učenie

Jedna z vetiev, ktorá sa v súčasnosti začína v e-learningu presadzovať, je založená na teórii tzv. trialogického učenia. Tradičné metódy učenia sú založené buď na individuálnom získavaní poznatkov (monologické učenie), alebo na skupinovom učení, resp. projektovej výučbe (dialogické učenie). Obe koncepcie sú veľmi využívané e-learningovými systémami. [1] [3]

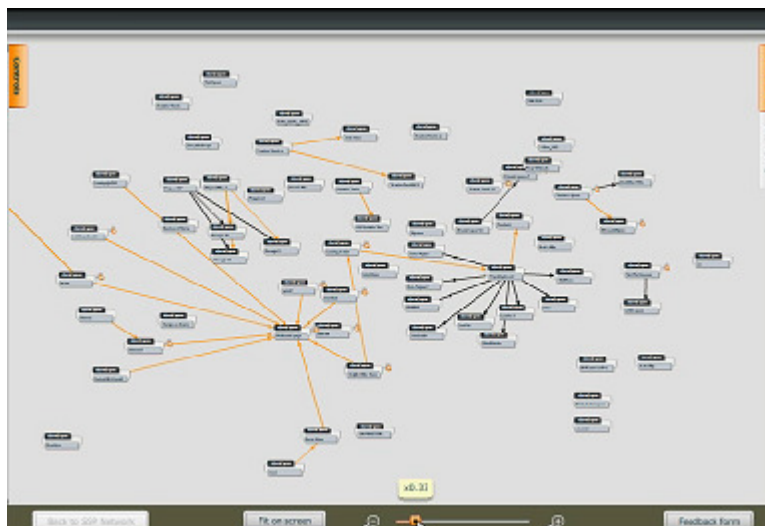
Triologické učenie predstavuje kolaboratívny spôsob učenia, vzájomná interakcia medzi aktérmi je sprostredkovaná okrem foriem uplatňovaných v monologickom a dialogickom učení aj zdieľanými objektmi aktivít (môžu to byť konceptuálne artefakty, produkty, činnosti).

KP-Lab (Knowledge Practices Laboratory) je projekt, ktorý implementuje tieto myšlienky. Zahŕňa súbor teórii, nástrojov a praktických modelov. Jedným z nástrojov je Knowledge Practices Environment (KPE). Jeho základom je zdieľaný priestor, ktorý reprezentuje súbor znalostí (ontológie) prostredníctvom prehľadného používateľského rozhrania. Objekty v ňom možno zoskupovať, presúvať, zväčšovať a zmenšovať náhľad, zjednocuje rôzne spôsoby organizovania informácií rôznych účastníkov. Tieto informácie možno filtrovať. Môžeme si zvoliť, aké informácie chceme vidieť. Znalosti v systéme možno zobrazovať najrôznejšími spôsobmi, podľa toho aký aspekt nás najviac zaujíma. Ak nás zaujíma časové hľadisko, napr. kedy boli zadávané a riešené jednotlivé úlohy, môžeme využiť Gantov diagram. Ten nám zobrazí ich časový priebeh v jednotlivých fázach. Gantov diagram je užitočný pri hľadaní cieľov a zadávaní úloh v procese vzdelávania. Pomocou grafickej reprezentácie znalostí v priestore možno znalosti preskupovať tak, aby boli čo najužšie sémanticky prepojené. [1] [2] [3]

Významnou súčasťou kolaboratívneho procesu je taktiež komunikácia reprezentovaná v zdieľanom priestore v rozličných formách. Jednou z metód je tzv. tagovanie obsahu. Tagy predstavujú kľúčové slová viažuce sa k nejakej informácii. Iným typom sú znalostné tagy. Tie predstavuje metainformácia, ktorá reprezentuje znalosti v obsahu, médiách, alebo dátach. Tagovanie je mocným nástrojom pri reprezentácii znalostí, umožňuje rýchlejšiu orientáciu v učebných textoch, nielen z hľadiska možnosti napovedať o čom je text (napr. výučbový materiál obsiahnutý v e-learningovom systéme), ale aj z hľadiska pomoci pri identifikácii tematicky podobných textov (na základe tagov). [1] [2] [3]

Samozrejmými funkciami v zdieľanom priestore sú možnosť komentovania jednotlivých položiek, ich usporiadanie, vytváranie odkazov medzi položkami, či sémantické vyhľadávanie. Účastníkom kolaboratívneho procesu môžu byť pridelované roly a zodpovednosti za úkony. Aj tie sú reprezentované ako znalosti a možno ich rôznymi metódami vizualizovať. Systém obsahuje aj vstavanú wiki, "to-do-list" aktivít, kolaboratívny nástroj pre vizuálne modelovanie a ďalšie nástroje podporujúce kolaboratívnu prácu. [2]

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.



Obr. 1: Knowledge Practices Enviroment.

Adaptívne výučbové systémy

V poslednej dobe sa do povedomia čoraz viac dostávajú adaptívne výučbové systémy. Za normálnych okolností majú všetci študenti k dispozícii rovnako organizované študijné materiály. Adaptívne výučbové systémy umožňujú prispôbiť učebné osnovy pre každého študenta na báze jeho aktuálnych vedomostí. [12]

Základom využitia adaptívneho systému je dôkladné zmapovanie korešpondujúcej doménovej oblasti a údajov o znalostiach študentov.

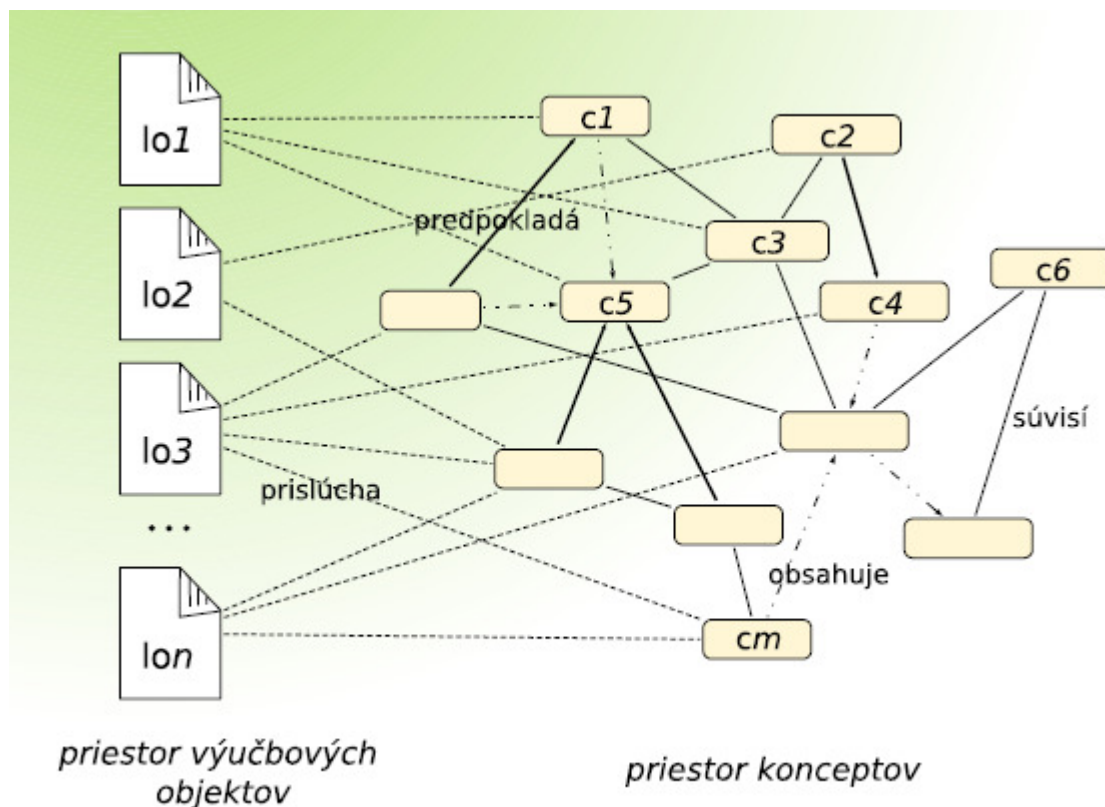
Každý výučbový kurz sa týka nejakých okruhov znalostí. Štandardne sú rozdelené na pomerne málo väčších sekcií postavených okolo vedúceho konceptu, na ktorého ovládnutie je v danej časti kurzu kladený najväčší dôraz (napr. Konečné automaty).

Avšak pri aplikovaní zložitejších výučbových metód nám tieto informácie o častiach kurzu nestačia, je užitočné napríklad vedieť, že isté znalosti je možné vysvetľovať až v prípade, že študenti ovládajú pojmy, s ktorými sa v aktuálnej látke pracuje (napr. Orientované grafy sa viažu na Konečné automaty), prípadne že študenti sa ľahšie orientujú v oblasti, ak ovládajú nejaký podobný koncept.

Precízne vyjadrenie kritických konceptov danej doménovej oblasti (doménová oblasť v našom prípade sú všetky možné znalosti, ktoré je možné získať v rámci predmetu DŠA) a vzťahov medzi nimi je teda bezpodmienečne nutné pre aplikáciu adaptívneho výučbového systému. Tento krok je však často pomerne náročný. Doménová oblasť kurzu často obsahuje desiatky až stovky konceptov a medzi jednotlivými konceptmi sa môžu navyše nachádzať rôzne typy ohodnotených relácií, takže dôveryhodné zmapovanie väčšej oblasti by mohlo stáť priveľa úsilia.

Zaujímavý prístup k tomuto problému ponúka softvérová podpora získavania metadát výučbového obsahu a vzťahu medzi konceptmi, ktorá bola analyzovaná v diplomovej práci Mariána Šimka [11]. Táto práca navrhuje softvérové riešenie, ktoré pomáha zmapovať koncepty doménovej oblasti a vzťahov medzi nimi na báze analýzy textu.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.



Obr. 2: lon – učebné materiály, cm – koncepty[11].

Zmapovaná doménová oblasť otvára nové možnosti skvalitnenia výučby. Veľakrát sú pre lektorov kurzu omnoho prehľadnejšie vzťahy medzi jednotlivými pojmami, možnosť vyhľadávať viac špecifické materiály súvisiace s konkrétnym pojmom a mnohé iné.

Adaptívna výučba sa dá aplikovať aj na generovanie elektronických testov v e-learningových systémoch. Metadáta – koncepty môžu byť tiež súčasťou jednotlivých otázok v rámci databázy daného systému. Na základe týchto údajov je možné zistiť okruhy, ktorých sa otázky týkajú a aké vedomosti u študenta predpokladajú. Okrem toho by bolo generovanie testov možné viac špecifikovať – obmedziť otázky v teste výhradne na koncepty, ktoré už boli prebrané, a tak dynamicky generovať testy podľa momentálnej situácie. Navyše podľa úspešnosti študentov na jednotlivých testoch sa dajú vytvoriť konkrétne štatistiky o vedomostiach študentov.

Ak máme okrem doménového modelu prístupné aj údaje o znalostiach študentov, máme všetky potrebné predpoklady na vytvorenie adaptívneho výučbového systému. Takéto systémy môžu mať rôzne zaujímavé funkcionality:

- prehľadná kategorizácia študijných materiálov,
- možnosť vyhľadávať poznatky ku konkrétnejším pojmom, a taktiež získanie informácií o vzťahoch na iné pojmy,
- vytvoriť prispôbený obsah kurzu pre individuálnych študentov,
- prispôbené testovanie – na základe toho čo študent na priebežných testoch nevedel je možné vygenerovať cvičný test z kritických okruhov, prípadne ponúknuť konkrétne študijné materiály k danej téme,
- štatistiky o znalostiach študentov na základe výsledkov testov. Výhoda je čo najčastejšie testovanie, vo vlastnom záujme aj nepovinné testy.

Koncept adaptívnych výučbových systémov je ešte stále mladý a pomerne nepreskúmaný. Implementácia predpokladov pre adaptívnosť do e-learningového systému je zložitý proces, takže nie je isté, či ho bude možné integrovať v rámci nášho projektu.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Avšak sú predpoklady, že v budúcnosti sa práve tento typ e-learningu bude používať, a to nielen na vysokých školách, keďže v dnešnej dobe sa veľký dôraz kladie na budovanie znalostnej spoločnosti a každý nový koncept podporného prostriedku pre výučbu je vítaný. Preto by sme pri tvorbe nášho systému mali mať minimálne na pamäti možnosť rozšíriť náš systém v tomto smere.

1.2 Porovnanie existujúcich e-learningových prostredí

Základné vetvy e-learningových systémov

Rozoznávame dve hlavné vetvy e-learningu. Jednou vetvou je tzv. Computer-Based Learning (CBL), ktorá pracuje s počítačmi ako s kľúčovým prvkom výučby, teda vyučovanie prebieha prostredníctvom nich. Zovšeobecnením CBL je tzv. Technology-Enhanced Learning (TEL), teda spôsob výučby obohatený o technológie vo všeobecnosti. Technológia sa v tomto prípade berie ako dôležitý prvok výučby. [8][9]

Druhou vetvou je Computer-Based Training (CBT), resp. jeho dnes už rozšírenejšia forma Web-Based Training (WBT). Predstavuje klasický spôsob e-learningových aplikácií s výučbovými textami, úlohami atď. Najmodernejšie prístupy zdôrazňujú najmä kolaboratívnosť vo výučbovom procese. Do CBT zaraďujeme akékoľvek výučbové texty alebo počítačové programy, ktoré pomáhajú v procese výučby. Môže ísť aj o dokumenty uložené v počítači, či tematické počítačové hry. Existujú však aj špeciálne softvérové e-learningové systémy, tzv. LMS a LCMS systémy, ktoré sú vytvorené špeciálne pre výučbový proces. Sú teda určené nielen pre samotnú výučbu, ale aj pre spravovanie činností s ňou súvisiacich, ako je správa študentov a hodnotení študentov alebo správa zadaní a celých kurzov. LCMS je v podstate súčasť LMS, špeciálne určená na správu obsahu, t.j. vytváranie obsahu, editovanie obsahu a pod. [9]

Pre úplnosť spomenieme aj pojem zmiešaného vyučovania, tzv. Blended Learning, ktoré vyjadruje prelínanie spôsobov výučby, typicky práve klasickej výučby s e-learningom. [6]

Moodle

Moodle je voľne šíriteľný open source e-learningový systém zameraný na vytváranie elektronických kurzov. Je veľmi dobre prispôsobený aj na pojmá veľkého množstva študentov, ale je použiteľný aj pre malé kurzy. Pre svoju prispôsobivosť bol od prvej verzie z roku 2002 prijatý s pozitívnymi ohlasmi. Podporuje aj plne elektronické kurzy, avšak je použiteľný aj ako podporný prostriedok pri kombinovaných kurzoch.

Je založený na technológii PHP a je až prekvapivo systémovo nenáročný a kompatibilný. Ďalšou jeho silnou stránkou je bezpečnosť – kontrola správnosti formulárov, validovanie dát a pod. Taktiež obsahuje modul na kontrolu plagiátorstva. Nevýhodou je absencia testovania programových zadaní a pomerne jednoduchý modul na generovanie testov. [13]

Akademický informačný systém

Akademický informačný systém je momentálne používaný na našej univerzite. Bol vyvinutý na Mendelovej univerzite v Brne a je používaný na viacerých univerzitách na Slovensku a v Českej republike. Tento produkt neslúži iba ako vzdelávací systém, ale poskytuje automatizáciu viacerých aspektov akademickej administratívy, napríklad štipendiá, registráciu predmetov, rozvrhy a pod.

Z hľadiska funkcionality e-learningu umožňuje zverejňovanie študijných materiálov prostredníctvom dokumentového servera, odovzdávanie súborov a elektronické testy. Hodnotenia testov a zadaní sú priamo pridelené príslušným študentom. Na druhej strane nepodporuje kontrolu plagiarizmu, testovanie programových zadaní a navyše typy testových otázok v elektronických testoch sú pomerne obmedzené.

Keďže sa jedná o komerčný produkt nie je jednoduché prispôbiť ho vlastným požiadavkám, a tak sú chýbajúce funkcionality dopĺňované prostredníctvom iných e-learningových systémov. [14]

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Wiki

Wiki je softvér, ktorý umožňuje používateľom voľne vytvárať a upravovať webové stránky iba za pomoci webového prehliadača. Je postavená na koncepte otvorených úprav, v ktorom má každý možnosť upraviť jej obsah. Charakteristická pre wiki je práve otvorenosť – podporuje používateľov v tom, aby zdieľali svoje názory a skúsenosti s ostatnými. Taktiež podporuje vzájomné prepojenie stránok citáciou alebo odkazom.

Z povahy wiki vyplýva hlavná nevýhoda, ktorou je veľká miera nedôveryhodnosti voči jej obsahu. Nakoľko je editovanie stránok prakticky anonymné, nie je možné zistiť či je editujúca osoba dostatočne odborne spôsobilá vzhľadom k danej téme. Wiki sama o sebe nie je príliš použiteľná priamo na správu učebných materiálov v rámci e-learningu, avšak môže byť veľmi užitočná pre študentov, ktorí si prostredníctvom nej môžu neformálne vymieňať vedomosti ohľadom daného kurzu. [16]

1.3 Analýza existujúcich prác

Webtester

Portál vznikol v rámci tímového projektu. Slúži ako podporný prostriedok pre automatické vyhodnocovanie zadaní študentov. Pod pojmom zadanie sa rozumie program vytvorený na základe zverejneného zadania. Pôvodné určenie projektu bolo pre predmet Procedurálne programovanie, neskôr bolo rozšírené aj na predmet Tvorba efektívnych algoritmov a programov.

Portál je vytvorený pomocou programovacieho jazyka Ruby a rámca Ruby on Rails. Do systému sa používatelia prihlasujú pomocou údajov z AIS. Prihlasovanie je realizované pomocou protokolu LDAP. V systéme sú definované nasledujúce tri typy používateľov a k nim prislúchajúca funkcionality.

- Študent
 - **Odobznanie zadania**
 - **Kontrola správnosti odovzdaného zadania** – ak je v programe chyba, systém vypíše rozdiel medzi kontrolnými výstupmi a výstupmi z testovaného programu. Vypisuje aj orientačný zdroj chyby, napríklad chybu kompilácie.
 - **Stiahnutie odovzdaných zadaní** – zadanie, ktoré študent raz odovzdal je možné znovu stiahnuť.
 - **Prihlásenie sa na cvičenie** – umožní študentovi prihlásiť sa v systéme do cvičenia, ktoré navštevuje.
- Pedagóg
 - **Kontrola automatizovaných výsledkov**
 - **Pridelenie bodov študentom**
 - **Pridávanie poznámok k hodnoteniam**
 - **Testovacie odovzdanie** – pedagóg má možnosť otestovať odovzdanie predtým, než finálne zverejní zadanie.
 - **Usporiadanie zadaní**
 - **Vytváranie cvičení**
 - **Pridelovanie študentov k cvičeniam** – pedagóg môže študentov priradiť k cvičeniu manuálne a nie na základe kľúča.
 - **Pridelovanie zápočtových zadaní k cvičeniam**
 - **Zverejňovanie zadaní**
- Administrátor
 - **Správa používateľov**
 - **Vytváranie zadaní a testov**

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Odvzdávanie zadaní

Študent odovzdá vypracované zadanie (program) do systému a ten ho vyhodnotí. Zadanie je možné odovzdávať v dvoch „módoch“. Odovzdanie klasického zadania a odovzdanie zápočtového zadania, teda zadania písaného v rámci zápočtovej písomky. Študentovi sa po odovzdaní zadania zobrazí správa o tom, ktoré testy prešli, prípadne ak neprešli, kde nastala chyba. Ak je test definovaný ako skrytý, podrobnosti o ňom sa nezobrazia.

Kontrola automatizovaných výsledkov

Pedagóg môže automatické hodnotenie systému upravovať, ale musí to patrične zdôvodniť. Toto zdôvodnenie sa ukladá do databázy, aby sa zabránilo nepodloženému meneniu výsledkov.

Vytváranie cvičení

Keďže predmet Procedurálne programovanie je organizovaný po cvičeniach, je pedagógovi daná možnosť vytvoriť cvičenia v systéme. Cvičenia slúžia na zjednotenie študentov, aby si pedagóg mohol vyfiltrovať iba študentov z jeho cvičenia. Pridelovanie študentov do cvičení sa môže organizovať na základe kľúčov, ako v systéme moodle, alebo manuálne pedagógom. Pri písaní zápočtových testov sa jednotlivé zadania pridelia cvičeniu s konkrétnymi študentmi. Ostatní študenti k zadaniam nemajú prístup. Prístup k systému v čase písania zápočtového testu majú iba počítače s IP adresou, ktorá bola definovaná pri jeho vytvorení.

Vytváranie zadaní pedagógom

Pedagóg vytvorí zadanie a následne ho môže zverejniť. V rámci vytvárania zadania je určená množina testov, ktorá je vykonávaná pri odovzdávaní študentových zadaní. V rámci jedného vytvoreného zadania môže byť definovaných viacero normálnych, aj skrytých testov.

Skryté testy

Boli vytvorené pre väčšiu kontrolu odovzdaných zadaní. Skryté testy budú kontrolovať zadanie, avšak ich detailný výpis sa nezobrazí. Zobrazí sa len, či daný test prešiel alebo nie. Pomáhajú zabrániť, aby študenti nemohli vytvoriť program, ktorý by zobrazoval len očakávané výstupy podľa vstupov.

Podporované jazyky

V súčasnej verzii sú podporované nasledujúce jazyky: C, C++, C#, JAVA. Funguje teda testovanie programov, ktoré sú napísané v týchto jazykoch. Spúšťanie programov je realizované pomocou knižnice Skynet. Tá implementuje prostredie Map/Reduce v jazyku ruby. Knižnica slúži na paralelné nezávislé vyhodnocovanie testov. Každé testovanie programu beží pod používateľom s obmedzenými právami, na čo je použité prostredie chroot na linuxovom serveri.

Export výsledkov

Výsledky sa exportujú do formátu CVS, kam sa zaradia údaje, ktoré sa dajú vyfiltrovať. Pôvodná myšlienka tímu bola exportovať výsledky priamo do systému AIS, avšak nezískali povolenie na prístup.

Štatistiky

V systéme sú dostupné štatistiky odovzdávania zadaní, ktoré slúžia na vytvorenie prehľadu o aktivite študentov.

Záver

V našom projekte sa venujeme najmä vytváraniu testov a ich automatizovanému vyhodnocovaniu. Aj v testoch sa však budú vytvárať krátke programy a bude potrebné ich vyhodnocovanie. Práve z tohto dôvodu vidíme zmysel v použití výsledkov tímového projektu. Tému vyhodnocovania

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

programov spracovali podrobne, umožnili pedagógom rozdeliť študentov do cvičení a značne im tak zjednodušiť prácu. Z ich riešenia je preto vhodné vychádzať a prebrať prvky vyhodnocovania do nášho projektu.

Yonban

Yonban umožňuje spravovanie zadávania, registrácie a pridelovania tém bakalárskych, diplomových a tímových projektov. Systém sa vyvíjal niekoľko rokov v rámci tímových projektov. Nechýba ani funkcionálna na vyplnenie používateľského profilu (napr. možno uviesť kontaktný e-mail), či vkladanie a prezeranie posudkov.

Návrh virtuálneho výučbového laboratória

Bakalárska práca Dávida Chalupu

Bakalárska práca Dávida Chalupu [17] sa zaoberá návrhom systému na podporu výučby, ktorý by bol schopný spravovať učebné materiály, generovať testy a zadania pre študentov a evidovať výsledky jednotlivých študentov. Principiálne sa jedná o prototyp komplexného e-learningového systému. Má podobné vlastnosti ako systém, ktorý plánujeme vytvoriť v rámci nášho projektu.

Analytická časť sa zaoberá hlavne štúdiom e-learningového prostredia – históriou a trendmi v tejto oblasti. Obsahuje všeobecný úvod do e-learningu a spomína vývojové modely a architektúry, ktoré sa bežne používajú pri tvorbe takýchto softvérových produktov. Taktiež porovnáva už existujúce, v praxi používané systémy, ako napríklad Wiki, AIS a Moodle.

Z návrhu systému je zaujímavá kategorizácia programových zadaní predmetu DŠA – s pomocou tohto rozdelenia by bolo možné upraviť systém pridelovania zadaní tak, aby študenti dostávali zadania s príbuznými témami.

V testovacom module rozoznáva tri druhy testov: semestrálne testy, skúška a „neoficiálne“ priebežné testy na pochopenie učiva. Tu sa naskytá otázka, či je možné generovať všetky tri druhy testov nad rovnakou databázou otázok. Riešenie tejto otázky by sme mali jednoznačne zahrnúť do nášho projektu.

Je načrtnutá aj škála hodnotenia náročnosti otázok, avšak nie je bližšie konkretizovaná. Taktiež je navrhnutý pomerne komplexný dátový model systému, ktorý dobre odzrkadľuje danú problematiku.

Výsledný systém je implementovaný najmä s pomocou technológií PHP a SQL. Poskytuje všetky vyššie uvedené funkcionality: správu študijných materiálov, generovanie a opravu testov a výber zadaní. Podobnosť témy, a taktiež nášmu tímu blízke použité technológie, robia túto prácu veľmi zaujímavým náhľadom do oblasti e-learningu, ktorým sa pri tvorbe nášho produktu dá inšpirovať.

Bakalárska práca Michala Noskoviča

Úlohou práce bolo navrhnuť štruktúru virtuálneho výučbového laboratória, ktoré by malo poskytnúť možnosť individuálneho štúdia, tréningu vedomostí a testovania. [16] Hlavnou časťou je tvorba zadaní a testovanie.

Riešenie je implementovaná v jazyku PHP s využitím MySQL databázy. Jadrom aplikácie je implementácia samotnej otázky a na nej realizovaných funkcií. Používateľ má možnosť využiť dva typy otázok - otázky, ktoré majú len jednu správnu odpoveď alebo otázky s viacerými správnymi odpoveďami. Prvá skupina je reprezentovaná ako radiobutton, druhá skupinou checkboxov.

Jednotlivé otázky je možné spájať do testov, ktoré sú neskôr ponúknuté používateľovi na precvičenie, prípadne preskúšanie vedomostí. Každá otázka má zadanú náročnosť. V závislosti od nej je možné určiť počet otázok s jednotlivými stupňami náročnosti, ktoré sa v teste budú vyskytovať.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

1.4 Analýza automatického generovania testov

S rozšírením informačných technológií do oblasti školstva sa pomerne rýchlo dostala do povedomia myšlienka testovania študentov pomocou počítačov. Najjednoduchšou formou takéhoto testovania sú práve elektronické testy. Hlavnou výhodou týchto testov je podstatné zníženie zaťaženia lektorov a tiež istá miera objektivity – každý študent má rovnaké podmienky a oprava testov nezávisí od úsudku človeka. Práve možnosť automatizovať väčšinu rutinných činností spojených s prípravou, vykonávaním a opravou testov, posunula elektronické skúšanie do popredia a umožnila jeho čoraz častejšie použitie v praxi.

Východiská

- cieľ je prácu maximálne automatizovať
- existuje veľa externých programov na generovanie testov, avšak my potrebujeme špecifické kritériá na generovanie, preto je lepšie si implementovať vlastný
- testy musia byť rôznorodé
- náročnosť testov by mala byť vyvážená
- rôzne typy a varianty otázok
- náročnosť – každá otázka ohodnotená náročnosťou – statická (používateľský vstup) a dynamická (podľa nazbieraných dát úspešnosti od študentov)
- možnosť štatistík – úspešnosť zodpovedania jednotlivých otázok
- testy musia byť rôznorodé
- šablóny – možnosť uloženia šablóny generovaných otázok

Nevýhody elektronických testov

Hlavné výhody sme už naznačili. Hlavnou a v podstate jedinou významnou nevýhodou je skutočnosť, že rôzne formy testovania sú obmedzené na strane opravovania, ako aj vyplňania testu. Príkladmi môžu byť úlohy, pri ktorých sa kreslia diagramy (akým spôsobom by ich študenti kreslili a ako by sa to vyhodnocovalo) alebo úlohy, v ktorých treba písať zdrojový kód programu (veľmi ťažké vyhodnocovanie správnosti). Určite si je možné predstaviť aspoň čiastočne funkčné riešenia týchto problémov. Ale umožniť takú úroveň posudzovania ako v týchto prípadoch dosahuje človek je ťažko predstaviteľné a vyžadovalo by si veľmi pokročilé metódy umelej inteligencie, ktoré zrejme dnes ešte nie sú na takejto úrovni.

Túto nevýhodu možno ľahko kompenzovať tým, že časť otázok v teste bude určených na samostatné vypracovanie a opravujúci ich má možnosť ohodnotiť samostatne.

Medzi menej významné nedostatky možno zaradiť nedostupnosť technických prostriedkov. Táto skutočnosť sa však našej fakulty týka len okrajovo, keďže zabezpečenie výpočtovou technikou je dostatočné. Problémom je zrejme len dostupnosť počítačov pre veľkú skupinu študentov v rovnakom čase. Študenti nemôžu písať test naraz a tak niektorí môžu byť potenciálne zvýhodnení – napr. tým, že sa dozvedia informácie o otázkach od študentov, ktorí test písali skôr.

Typy otázok

- **Testová otázka s výberom 1 z N správnych možností**

Pri tomto type otázok sa využíva ovládací prvok typu prepínacie tlačidlo (radio button), ktoré umožňuje výber jedinej správnej odpovede.

typu A: Za správnu odpoveď je X bodov, za nesprávnu 0 bodov.

typu B: Za správnu odpoveď je X bodov, za nesprávnu -Y bodov.

typu C: Ako typu B + je možnosť odpovedať "neviem" za 0 bodov.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

- **Testová otázka s výberom M z N správnych možností**

Pri tomto type otázok sa využíva ovládací prvok typu zaškrtačacie tlačidlo (check box), ktoré umožňuje výber viacerých (aj žiadnej) správnych odpovedí.

typu A: Za správnu odpoveď je X bodov, za nesprávnu 0 bodov.

typu B: Za správnu odpoveď je X bodov, za nesprávnu -Y bodov.

typu C: Ako typu B + je možnosť odpovedať "neviem" za 0 bodov, v tomto prípade sa ku každej podotázke dávajú prepínacie tlačidlá (áno/nie/neviem).

- **Zreťazená testová otázka typu 1 z N dĺžky M**

Pri tomto type otázok sa využíva ovládací prvok typu prepínacie tlačidlo (radio button), pričom v každom z M po sebe idúcich stĺpcoch je pod sebou uložených N prepínacích tlačidiel. Správnu odpoveď za X bodov predstavuje správne prepojenie všetkých elementov. Alternatívna (a asi vhodnejšia reprezentácia) je reprezentácia pomocou výberového vysúvacieho ovládacieho prvku (selectov). Veľkosť N každého stĺpca môže byť rôzna.

Koreň stromu má	<input checked="" type="radio"/>	32	<input type="radio"/>	žltých potomkov	<input type="radio"/>
Koreň strom nemá	<input type="radio"/>	26	<input type="radio"/>	červených potomkov	<input checked="" type="radio"/>
Ľavý potomok koreňa stromu má	<input type="radio"/>	45	<input checked="" type="radio"/>	modrých potomkov	<input type="radio"/>
Pravý potomok koreňa stromu má	<input type="radio"/>	11	<input type="radio"/>	čiernych potomkov	<input type="radio"/>

Obr. 3: Zreťazená testová otázka.

- **Otázka s písanou číselnou odpoveďou s toleranciou T**

Pri tomto type otázok sa využíva ovládací prvok textové pole (input box), do ktorého sa napíše číselná odpoveď, pričom môže byť nastavená tolerancia T pri detekcii správnej odpovedi.

- **Otázka s písanou odpoveďou vo forme textového reťazca**

Pri tomto type otázok sa využíva ovládací prvok textové pole (input box), do ktorého sa napíše slovná odpoveď. Takto zadávané odpovede musia byť krátke, jednoduché a jednoznačné.

- **Testová otázka s výberom 1 z N správnych možností rozšírená o písanú odpoveď**

Pri tomto type otázok sa využíva ovládací prvok typu prepínacie tlačidlo (radio button), ktoré umožňuje výber jedinej správnej odpovede. Okrem toho obsahuje aj možnosť „iná“, ku ktorej je priradené textové pole s možnosťou napísať správnu odpoveď.

typu A: Za správnu odpoveď je X bodov, za nesprávnu 0 bodov.

typu B: Za správnu odpoveď je X bodov, za nesprávnu -Y bodov.

typu C: Ako typu B + je možnosť odpovedať "neviem" za 0 bodov.

- **Text obsahujúci viacero textových polí na vyplňanie**

Pri tomto type otázok sa pomedzi normálny text vloží niekoľko textových polí, ktoré možno vyplňať ako pri otázke s písanou odpoveďou vo forme textového reťazca.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

- **Prirad'ovacia otázka**

Pri tomto type otázok sa prirad'uje N objektov z jednej množiny údajov k M objektom z inej množiny údajov. V každom z N riadkov je vypísaná jedna položka z prvej množiny s prislúchajúcim selectom.

typu A: M a N majú rovnakú mohutnosť, je medzi nimi bijekcia.

typu B: M môže mať menšiu mohutnosť ako N, teda je možné priradiť ten istý prvok z množiny M viacerým prvkom z množiny N.

- **Zorad'ovacia otázka**

Veľmi podobné prirad'ovacej otázke typu A, množina M obsahuje čísla 1..n reprezentujúce poradie, teda údajom v množine N sa vyznačí ich usporiadanie.

- **Zužovacia zorad'ovacia otázka**

Pri tomto type otázok sa využíva ovládací viacriadkový select. Pomocou presunových tlačidiel sa do ľavého zoznamu dĺžky M priradia v správnom poradí odpovede dĺžky N, pričom $M \leq N$.



Obr. 4: Zužovacia zorad'ovacia tabuľka.

- **Maticová (tabuľková) otázka**

Tento typ je vhodný na grafové úlohy.

Príklad:

Bola by zadaná grafová úloha, v ktorej by sme mali hrany a vrcholy a výsledok by dospel k nejakému grafu. Do buniek tabuľky by sa vyplňali reťazce spôsobom, ktorý by určovala legenda.

Napríklad AXX, kde A znamená, že hrana je neorientovaná a XX je číselné ohodnotenie (príklad 05, 42)

BXX (ohodnotená z x do y)

DXX (ohodnotená z y do x)

CX-XX (ohodnotená z x do y, aj y do x a príslušné ohodnotenia)

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

	A	B	C	D	E	F
1			A25			
2				A30	B40	
3			C4-50			
4						

Obr. 5: Maticová otázka.

- **Otvorená otázka**

Otvorené otázky sú otázky, pri ktorých treba uplatniť tvorivý prístup, napríklad načrtnúť opis riešenia, napísať zdrojový kód. Tieto otázky nemožno automaticky vyhodnotiť, je nutné ich vyhodnocovať „ručne“.

typu A: K otázke je poskytnutý ovládací prvok textová oblasť (textarea).

typu B: Nie je poskytnutá textová oblasť, napríklad ak je nutné v odpovedi kresliť diagramy.

Vytváranie testov

Tvorba testov pozostáva z dvoch procesov. Jedným je pridávanie otázok do databázy. Pri tomto procese sa vyberie niektorý z implementovaných typov otázok a napíše sa v príslušnom tvare otázka so zoznamom odpovedí. Toto sa vykoná buď prostredníctvom používateľského rozhrania za pomoci formulárov, alebo priamym importom do databázy. Preferovaný je spôsob pomocou používateľského rozhrania, keďže otázky vo finálnej fáze vytvárajú vyučujúci predmetu. Pri každej otázke si môžeme všimnúť niekoľko charakteristík:

- spôsob vyhodnocovania – či sa berú do úvahy len správne možnosti alebo sa aj strháva za nesprávne odpovede, či je poskytnutá možnosť neviem. Spôsob vyhodnocovania závisí aj od typu.
- bodové ohodnotenie otázky,
- tematické zaradenie otázky – do akého tematického okruhu otázka patrí,
- váha otázky – určená statickou alebo dynamickou náročnosťou.

Vo všeobecnosti náročnosť otázky môže a nemusí súvisieť s výškou bodového ohodnotenia. V prípade, že súvisí, je možné náročnosť otázky jednoducho odstupňovať jej zaradením do skupiny otázok za viac bodov. Tento prístup je ale pomerne obmedzujúci, niekedy môžeme chcieť ťažšiu otázku za nízke bodové ohodnotenie a ľahšiu otázku za vyššie bodové ohodnotenie, ak napríklad ťažšia otázka nie je tak kľúčová. Preto je vhodné bodové ohodnotenie a váhu náročnosti oddeliť. Najvhodnejší prístup je zrejme vôbec nepriradovať otázke pevné bodové ohodnotenie, ale to určovať až pri generovaní testu na základe zadanej hodnoty.

Váhu náročnosti možno opísať slovne alebo číselne na určitej stupnici. Oba prístupy sú v podstate ekvivalentné. Pre vnútornú reprezentáciu je vhodnejšia číselná reprezentácia, pre tvorcu testu môže byť názornejší slovný popis. (príklad – „ľahká“, „stredná“, „ťažká“; 1,2,3). Túto váhu určuje tvorca otázky na základe subjektívneho odborného posúdenia. Hovoríme o nej tiež ako o staticky určenej náročnosti.

Pohľad na náročnosť je vždy subjektívny. Ak má tvorca otázky dlhodobé skúsenosti s testovanou oblasťou a samotným testovaním študentov v tejto oblasti, je veľký predpoklad, že náročnosť otázky odhadne správne. Presnejším ukazovateľom náročnosti by však mohla byť štatistická váha. Vyjadrovala by podiel úspešných odpovedí z celkových odpovedí na danú otázku. Mohla by odhaliť nielen náročnosť otázky, ale aj prípadné nedostatky vo výučbovom procese (napr.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

že danej oblasti nebola venovaná dostatočná pozornosť). Táto náročnosť sa po každom vypracovaní testu vo všeobecnosti mení, hovoríme o dynamickej náročnosti.

Druhým procesom je samotná tvorba testu z dostupnej databázy otázok. Takisto ho možno vykonať prostredníctvom používateľského rozhrania alebo priamou manipuláciou s databázou. Pri vytváraní testu sa stanoví, aké množstvo otázok s určitými parametrami sa má vygenerovať. Generovanie prebieha náhodne vždy z tej podmnožiny otázok v databáze, ktorá spĺňa zadané parametre. Príkladom môžu byť nasledujúce parametre:

Tematický okruh	Bodové ohodnotenie	Náročnosť – váha	Počet otázok
-----------------	--------------------	------------------	--------------

Tieto konkrétne parametre vyjadrujú koľko otázok s danou náročnosťou a z daného tematického okruhu sa má vytvoriť a aké majú mať bodové ohodnotenie. Pri bodovom ohodnotení treba zdôrazniť, že na rozdiel od tematického okruhu a náročnosti nemusí byť pevným atribútom otázky, ale zadávaným (teda nemusí ovplyvňovať generovanú množinu otázok).

Iná skupina parametrov by mohla zahŕňať aj konkrétny typ otázky, alebo ľubovoľný ďalší zmysluplný parameter. Príklad:

Tematický okruh	Bodové ohodnotenie	Náročnosť - váha	Typ otázky	Počet otázok
-----------------	--------------------	------------------	------------	--------------

Súhrn takýchto zoskupení parametrov je podkladom pre vygenerovanie testu. Rozvrhnutie tohto súhrnu môže byť charakteristické pre určitý typ testu (semestrálny test, skúškový test, tematické testy pre jednotlivé cvičenia). Preto je nepochybne vhodné mať možnosť si toto rozvrhnutie uložiť ako šablónu testu. Pri ďalšom generovaní testov tak nie je potrebné znova vyplňať všetky údaje, ale len načítať uloženú šablónu.

Najdôležitejším výstupom sú samotné výsledky – vyhodnotenie testu. Tie sa taktiež ukladajú do databázy v príslušnej tabuľke študenta. Pre ich ďalšie spracovanie (perspektívne napr. do AIS) je vhodné ich vyexportovať do vhodného formátu, akým je CSV formát, prípadne iné. Dáta v tomto formáte by mal byť schopný cieľový systém importovať a zahrnúť ho ako súčasť vyhodnocovania predmetu. Motiváciou je, aby vyučujúci nemuseli ručne prepisovať výsledky testov. Za samotnú realizáciu importu však zodpovedá cieľový systém.

Zhrnutie základnej funkcionality

V rámci analýzy sme identifikovali určitú minimálnu funkcionality, ktorú by sme chceli do systému implementovať už v prototype:

- práca nad jednou spoločnou databázou otázok,
- automatické opravovanie,
- export výsledkov do XLS / CSV pre potreby ďalšieho spracovania.

Zhrnutie doplnkovej funkcionality

Za doplnkovú funkcionality považujeme tie funkcie testovania, ktoré nie sú úplne nevyhnutné pre základnú činnosť testovania, ale obohacujú možnosti systému. Niektoré z nich môžu byť implementované už v prototype:

- vyvážená náročnosť,
- monitorovanie percenta úspešnosti pre otázky,
- tzv. relatívna náročnosť testu, na základe úspešnosti odhadnúť znalosti študentov – štatistiky,

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

- možnosť testy generovať dynamicky a jednoducho – zadanie času, bodov, okruhov, príp. počet otázok a systém vygeneruje test,
- extra dáta uchovávané v rámci otázky: koncepty, náročnosti (statická, dynamická).

1.5 Analýza tvorby a automatického opravovania zadaní

Automatizované vyhodnocovanie programov

Klasickým spôsobom vyhodnocovania programových zadaní sú špeciálne testovače zdrojových kódov, v angličtine nazývané *online judges*, ktoré sa využívajú aj pri programátorských súťažiach, ako sú ACM ICPC, TopCoder, lokálnych univerzitných súťažiach (možno spomenúť ProFIIT konaný na našej fakulte) a mnoho ďalších. Významnejšie súťaže väčšinou využívajú vlastné testovače, iné súťaže používajú testovače, ktoré poskytujú práve organizácie s vlastnými testovačmi. Jednotlivé testovače sa mierne líšia vo funkciách, výkonnosti, bezpečnostných prvkoch a najmä v podpore programovacích jazykoch. Veľmi prepracovaný testovač, ktorý v súčasnosti podporuje až 39 programovacích jazykov, je Sphere online judge prevádzkovaný na <http://www.spoj.pl/>. Samotná služba má v databáze vyše 5000 problémov a tiež je príkladom služby, ktorá poskytuje možnosť prevádzkovať vlastné súťaže. [19][20][21]

Spôsob práce je nasledovný – súťažiaci dostane text zadania problému a zadaný formát vstupu a výstupu programu, príp. aj vzorové ukážky vstupov a príslušných výstupov. Presné dodržanie ich formátu je dôležité, keďže kontrola spočíva na porovnávaní výstupov programu súťažiaceho so správnym výstupom umiestneným na serveri k príslušnému programu, ktorý samozrejme nie je prístupný. Zvyčajne nie sú prístupné ani testovacie vstupy, to však záleží od konkrétnej súťaže. Keď používateľ napíše zdrojový kód, uloží ho ako čistý text (väčšinou sa tiež vyžaduje, aby bola uvedená aj príslušná prípona, najmä ak testovač umožňuje testovať viac programovacích jazykov, a tak podľa prípony môže zvoliť správny jazyk) a nahrá ho do testovača. Ten ho sa ho pokúsi príslušným kompilátorom preložiť a ak sa to podarí, porovná výstup programu. Podľa výsledku tejto činnosti oznámi testovač správu o výsledku vyhodnotenia. Tieto správy sa mierne líšia v závislosti od konkrétneho testovača, ale v princípe sú veľmi podobné. Príklady a významy možných správ, ktoré používa súťaž ProFIIT sú uvedené v nasledujúcej tabuľke.

Správa	Význam
Chyba pri kompilovaní	Nepodarilo sa preložiť zaslaný zdrojový kód.
Chyba pri behu programu	Program bol úspešne preložený, ale počas výpočtu sa zrútil (napríklad kvôli deleniu nulou alebo zlému prístupu do pamäte).
Vypršanie časového limitu	Program bol úspešne preložený a spustený, ale neskončil počas požadovaného časového limitu. Buď sa v ňom vyskytol nekonečný cyklus, alebo bol použitý málo efektívny algoritmus.
Zlá odpoveď	Program sa začal vykonávať, skončil a vytvoril nejaké (poprípade aj žiadne) výsledky, ale tieto výsledky nie sú správne.
Prezentačná chyba	Program skončil a vypísal nejaké výsledky, ktoré sa zdajú byť v poriadku. Ich tvar však nie je presne taký, ako bol požadovaný v zadaní. Často ide o preklep, vynechanú medzeru vo výstupe a podobne.
Akceptovaný	Program je uznaný za správny. Daný problém sa považuje za úspešne vyriešený.
Volanie nepovolenej funkcie	Súťažiaci sa pokúsili použiť niektorú z nepovolených funkcií programovacieho jazyka.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Porušenie pravidiel	Táto odpoveď sa použije, keď rozhodcovia zistia nejaké iné porušenie pravidiel alebo nesplnenie zadania.
---------------------	--

Tab. 1: Správy zo súťaže ProFIIT.

Takéto systémy samozrejme nie sú obmedzené len na programátorské súťaže. Sú veľmi dobrým nástrojom aj pre programovanie zadaní. Tento prístup sa využíva aj na našej fakulte, napr. v rámci predmetu Tvorba efektívnych algoritmov a programov. Na predmete DŠA sa takýto systém zatiaľ nepoužíva, no myslíme si, že vzhľadom na charakter predmetu by bolo použitie takéhoto systému veľmi vhodné.

Zadania na predmete DŠA a ich modifikácia

Na predmete DŠA sa počas semestra robia tri väčšie doma vypracovávané zadania a menšie zadania vypracovávané na každom cvičení. Ich stručný opis a spôsob ich hodnotenia uvádzame v kapitole 1.7 Analýza cieľového prostredia (predmet DŠA).

Zadania pre domáce vypracovanie v súčasnej podobe nie sú vhodné pre automatické spracovanie. Väčšina z nich neobsahuje presný formát vstupu a výstupu, niektoré ho čiastočne obsahujú, alebo ho možno veľmi ľahko transformovať do takejto podoby.

Tematicky je prvá séria zadaní zameraná na zásobníky, rady a spájané zoznamy, druhá na binárne stromy, binárne vyhľadávacie stromy, prioritné fronty, Hufmannove kódy a množiny. Tretia séria je zameraná na grafy, 2-3 stromy a hashovanie.

Je zrejmé, že pre niektoré typy úloh nie je prístup automatizovaného vyhodnocovania programov možný alebo vhodný. Napríklad pri hashovacích tabuľkách alebo stromoch totiž nie je dôležitý výstup, ale vnútorná reprezentácia. Pri úlohách, v ktorých je dôležitý výstup (napr. prevody do poľskej, alebo reverznej poľskej notácie výrazov), by však prerobenie zadaní bolo zrejme vhodné. Študent by mal k dispozícii testovacie vstupy, na ktorých by si mohol overiť funkčnosť programu. Inú sadu testovacích vstupov by mal k dispozícii vyučujúci. Jej použitím by sa mohol ušetriť čas, ktorý sa zvykne pri prezentovaní programu stráviť. Kontrola kódu je však stále nutná pre zhodnotenie, či boli správne použité dátové štruktúry.

Vo všeobecnosti možno povedať, že tento spôsob je vhodný pre voľné algoritmické úlohy, v ktorých nie je predpísané použitie dátovej štruktúry a menej vhodné pre algoritmické úlohy, v ktorých sa požaduje určitá dátová štruktúra, prípadne ich úlohou je priama implementácia dátovej štruktúry. To je pri predmete DŠA problematické. Prevládajú totiž úlohy, v ktorých je použitá štruktúra dôležitá. Tento systém hodnotenia je teda výhodný, ale na predmete DŠA použiteľný len čiastočne.

Nájsť vhodný spôsob pre kontrolu štruktúry programu je veľmi náročnou úlohou a efektívna, univerzálna implementácia pre tento účel zrejme neexistuje. Možným riešením je k určitým zadaniam presne nadefinovať hlavičky operácií v konkrétnom programovacom jazyku a k týmto operáciám vytvoriť testovacie prípady, cez ktoré musia študentmi implementované operácie úspešne prejsť. Implementáciu takéhoto prístupu, prípadne ďalších možnosti vyhodnocovania zadaní v našom projekte ešte zvážime.

Dobrym prístupom pre uľahčenie kontroly zadaní je predostrieť pedagógovi odovzdaný zdrojový kód v určitej upravenej forme jeho vhodnou vizualizáciou (ako farebné zvýraznenie určitých častí kódu, napr. na ktorom riadku sa vyskytlo pri kompilácii varovanie, či chyba, alebo jednoduché zvýraznenie syntaxe).

1.6 Elektronické skriptá

Študijné materiály sa oddávna preferovali v papierovej forme, či už sa jednalo o skriptá, učebnice alebo iné zdroje. S touto formou materiálov sa však spájajú niektoré výrazné nevýhody. Pri veľkom

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

počte študentov v odbore je nutné, aby mal každý ročník tlačené materiály prístupné. To nie je vždy zo strany vzdelávacej inštitúcie jednoduché zabezpečiť. Zisky z predaja skript sú väčšinou malé a je potrebná nemalá počiatočná investícia na tlač takýchto materiálov.

V posledných rokoch sa však objavil trend, ktorý potláča práve tlačenú formu študijných materiálov. S nástupom informačných technológií a ich integráciou na školách a univerzitách sa čoraz viac pôvodne tlačených študijných materiálov dostáva do elektronickej podoby. Elektronické študijné materiály majú oproti svojim tlačeným verziám viacero výhod. Je to predovšetkým jednoduchý prístup k nim a ich jednoduché šírenie. K jednému takto zverejnenému dokumentu môže mať prístup prakticky ľubovoľný počet študentov a dokument je im sprístupnený všetkým v rovnakej kvalite, nakoľko pri tlačených materiáloch sa musia študenti často uspokojiť s niekoľko rokov starými skriptami ktoré niekedy dokonca nie je jednoduché ani získať.

Hlavný problém elektronickej študijných materiálov je samozrejme v ich forme. Tým, že sú dostupné iba v elektronickej podobe, na ich prezeranie je nutný prístup k počítačom a internetovému pripojeniu, takže ich nasadenie sa preferuje skôr na technicky zameraných odboroch.

Ak sú však predpoklady na zavedenie tejto formy publikácie študijných materiálov splnené, výhody jednoznačne zatieňujú ich nedostatky. Okrem jednoduchých učebných textov môžu elektronické materiály obsahovať rôzne animované obrázky a grafy, ktoré umožňujú pomocou vizualizácie ľahšie pochopiť niektoré koncepty. Okrem toho web rozhranie ponúka široké spektrum multimediálnych prvkov, ktoré napomáhajú k atraktívnejšej forme výkladu látky. V neposlednom rade, keďže informatika je oblasť stále podliehajúca pomerne častým zmenám, nie je nezvyčajné, že sa prispôbujú aj učebné osnovy. V prípade tlačených materiálov by to znamenalo nutnosť dotlače aktualizovaných skript, avšak pri ich elektronickej podobe je ich úprava a následné distribúcia podstatne jednoduchšia a lacnejšia.

V dnešnej spoločnosti je vlastníctvo počítača už pomaly nutnosťou a taktiež sa zvyšuje počet domácností s pripojením na internet. Z tejto perspektívy vyzerá budúcnosť elektronickej študijných materiálov pomerne priaznivo a v budúcnosti by sme sa mohli dočkať jej širšieho uplatnenia vo všetkých oblastiach školstva.

Náš systém by mal určite študentom poskytovať nejakú formu študijných materiálov v elektronickej podobe, nakoľko by bolo v systéme obsiahnuté všetko potrebné k danému predmetu. Taktiež sa výborne dopĺňajú s elektronickejmi testami.

1.7 Analýza cieľového prostredia (predmet DŠA)

Stručný opis predmetu

Nami navrhovaný e-learningový systém je určený pre výučbu predmetu Dátové štruktúry a algoritmy (DŠA), ktorý sa vyučuje v 3. semestri bakalárskeho štúdia v študijných programoch Informatika a Počítačové a komunikačné systémy a siete. Predmet je v oboch týchto programoch povinný, jeho obsah predstavuje štandard vo výučbe univerzitných informatických odborov a možno ho považovať za jeden z nosných predmetov bakalárskeho štúdia.

Obsahovo nadväzuje na predmet Procedurálne programovanie vyučovanom v 1. semestri, kde sa študenti oboznámia so základmi programovania v jazyku C a základnými princípmi procedurálneho programovania. Dobré zvládnutie tohto predmetu je predpokladom predmetu DŠA. V samotnom predmete DŠA si študenti prehľadajú znalosti s prácou s údajovými typmi a algoritmami na nich aplikovanými.

Stručná osnova predmetu:

1. Základné pojmy: Štruktúra údajov, údajové typy, špecifikácia, reprezentácia a implementácia údajového typu, zapuzdrenie. Implementovaný typ, implementujúci typ.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

2. Spájaný zoznam, jednosmerne spájaný, obojsmerne spájaný zoznam.
3. Zásobník, front, algoritmus transformujúci výraz do postfixovej notácie.
4. Reťazec, pole (vektor).
5. Stromy: všeobecný strom, binárny strom, binárny vyhľadávací strom (vrátane červeno-čiernych).
6. Prioritný front, halda, binárna halda. Huffmanove kódy.
7. Množina - implementácia pomocou vektora, spájaného zoznamu a stromu (vrátane vyvážených 2-3, AVL a binárnych vyhľadávacích).
8. Grafy: orientovaný graf, algoritmy hľadania najkratšej cesty, tranzitívny uzáver. neorientovaný graf: minimálna kostra grafu.
9. Tabuľka, tabuľka s rozptýlenými prvkami: otvorené rozptýlenie/adresovanie, vonkajšie zreťazenie, vnútorné zreťazenie.
10. Usporiadanie. metódy vnútorného usporadúvania: vkladaním, výmenou, výberom, Shellovo, rýchle, zlučovaním, distributívne, radixové.
11. Usporiadanie. metódy vonkajšieho usporadúvania: zlučovaním.
12. Správnosť a ďalšie vlastnosti programov.

Tab. 2: Osnova predmetu DŠA.

Analýza terajšieho spôsob hodnotenia predmetu

V minulosti hodnotenie predmetu pozostávalo z niekoľkých (štyroch, neskôr troch) doma vypracovávaných zadaní a jedného semestrálneho testu a záverečnej skúšky. Podmienkou bolo získať určitý počet bodov počas semestra. Minimálny počet bodov za skúšku nebol stanovený, bolo potrebné len dosiahnuť taký celkový počet, aký je potrebný na úspešné ukončenie predmetu.

Systém, v ktorom boli v menšej miere stanovené minimálne bodové zisky síce umožňoval flexibilnejšie rozvrhnutie času na prípravu, no mnoho študentov nedokázalo z tejto flexibility ťažiť a neboli si schopní túto prípravu vhodne rozvrhnúť. Nechávali si ju na poslednú chvíľu a nestihli sa požadované vedomosti poriadne naučiť. Preto sa postupne zaviedol systém, v ktorom je z každej úlohy (či už zadania, alebo testu) nutné získať určitý minimálny počet bodov, čo študentov núti k systematickejšej príprave.

V akademickom roku 2009/2010 sa okrem toho zmenila štruktúra zadaní, znížilo sa bodové ohodnotenie za doma vypracované zadania a zaviedli sa dvojbodové zadania vypracovávané na každom cvičení. Ich ohodnotenie pozostáva zo slovného opisu navrhnutého algoritmu (0,5 bodu), samotnej implementácie (1 bod) a odhadu zložitosti (0,5 bodu). Ich náročnosť je aj vzhľadom k časovej tiesni pomerne vysoká. Sú však silným motivujúcim faktorom práve preto, že si vyžadujú oveľa intenzívnejšiu prípravu.

Samozrejme študentom, ktorí nemajú hlbšie skúsenosti so zložitejšími algoritmickými úlohami (a takých môže byť v 2. ročníku ešte pomerne dosť), zrejme nebudú postačovať znalosti z predmetu Procedurálne programovanie na vysoké bodové zisky. Práve týmto študentov takéto testy prospejú najviac, keďže ich nútia sa naučiť dobre programovať väčšmi ako klasické zadania, na ktoré majú oveľa viac času a ktoré nemusia riešiť vždy samostatne (čo nie je nutne nežiaduce, napr. pri konzultácii so spolužiakmi si môžu objasniť určité nejasnosti, na druhej strane možno zvládnu predmet bez dostatočných nadobudnutých zručností samostatne programovať). Ak v nich aj nebudú príliš úspešní, stále si môžu stratené body vynahradiť na zadaniach riešených doma. Súčasnú podmienku absolvovania sú nasledovné.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

<p>Môžno získať až 100 bodov: zadania (max. 40 bodov: na cvičeniach 20 a doma 20) priebežné testy : prvý T1 (max. 5 bodov) druhý T2 (max. 15 bodov) záverečná skúška (max. 40 bodov)</p> <p>Podmienky absolvovania predmetu: Zápočet: A získať minimálne 15 bodov zo zadaní A získať minimálne 2 body z T1 a 5 bodov z T2</p> <p>Predmet: získať zápočet A získať minimálne 18 bodov zo skúšky A získať minimálne 56 bodov spolu.</p>

Tab. 3: Hodnotenie predmetu DŠA.

Možné modifikácie výučby s ohľadom na navrhovaný e-learningový systém

Náš e-learningový systém by chcel reflektovať uvedené skutočnosti. Jedným z hlavných cieľov nášho systému je zefektívnenie priebežnej prípravy študentov. Predpokladom je, že študenti budú mať možnosť vykonávať cvičné testy, ktoré budú mať rovnakú formu ako priebežné zápočtové a skúškové testy. Primárnou úlohou by bolo, aby testy obsahovali otázky, ktoré napomôžu k aktuálne riešeným úlohám, resp. ktoré sú viazané k aktuálne preberanej téme predmetu. Určitý podiel otázok v cvičných testoch by sa potenciálne mohol vyskytnúť aj v regulárnych testoch. Tento podiel musí byť menšinový, inak by študenti mohli dospieť k tomu, že by sa na skúšku pripravovali len z týchto testov. Na druhej strane pri vedomí, že sa nejaký malý počet otázok môže vyskytnúť na regulárnom teste, budú pristupovať k vypracovávaniu cvičných testov zodpovednejšie, no nezanedbajú ani ostatné výučbové zdroje. Významom cvičných testov by bolo aj to, aby sa študenti mohli dokonale oboznámiť s formou testu a nemali by vznikať nedorozumenia o spôsobe vypracovávania otázok. Cvičné testy by neboli priamou súčasťou hodnotenia, slúžili by ako príprava.

Druhým podporným prostriedkom pre študentov by boli už skôr spomínané overovacie sady vstupov a výstupov pre zadania. Tieto by odhalili logické chyby v programe, ktoré by študenti pri náhodnom skúšaní na vlastných vstupoch nemuseli odhaliť.

Pri tvorbe nášho systému by sme však mali mať na pamäti, že spôsob výučby predmetu DŠA sa môže v priebehu rokov zmeniť. Momentálne je systém hodnotenia nastavený tak, že študenti majú každé cvičenie malý test. Avšak malo by byť možné nastaviť výsledný systém tak, aby sa dali vygenerovať napríklad dva väčšie semestrálne testy, prípadne sa inak prispôbiť možným zmenám v organizácii predmetu.

2 Špecifikácia

2.1 Funkcionálne požiadavky

Funkcionálne požiadavky na vyvíjaný systém nám vyplynuli z komunikácie so zákazníkom a následne z analýzy, ktorú sme vykonali na základe stretnutí. Zotriedili sme ich podľa dôležitosti nasledujúco:

- Vysoká priorita
 - automatické generovanie testov
 - testovanie študentov
 - automatické vyhodnocovanie testov
 - odovzdávanie zadaní študentov
 - automatické vyhodnocovanie zadaní
- Stredná priorita
 - autentifikácia cez systém AIS
 - exportovanie výsledkov do AIS
 - zaradovanie študentov do skupín (cvičení)
- Nízka priorita
 - evidencia študentov
 - úprava používateľského profilu

2.2 Iné požiadavky

Medzi ďalšie požiadavky patria najmä tie, ktoré sa týkajú bezpečnosti systému, spoľahlivosti, výkonnosti a správnosti v ňom uložených údajov.

- **Bezpečnosť:**
V systéme budú študenti vypracovávať testy, je preto potrebné dbať na bezpečnosť spracovania údajov. Systém nesmie byť zneužíteľný zo strany študentov, ako ani zo strany externých používateľov.
- **Správnosť údajov:**
Na študentovo výsledné hodnotenie bude mať priamy vplyv automatické vyhodnocovanie zadaní a testov. V žiadnom prípade preto nesmie nastať chyba, kvôli ktorej by mal študent v systéme nesprávne pridelené body za zadanie alebo za test.
- **Výkonnosť:**
Systém musí byť dostatočne výkonný na to, aby dokázal spracovať testy a zobraziť čiastočné alebo úplné výsledky študentom hneď po jeho ukončení.
- **Spoľahlivosť:**
Spoľahlivosť je veľmi dôležitá, keďže je neprípustné, aby systém zlyhal počas písania testu a tak znehodnotil snaženie študentov.

2.3 Charakteristika rolí v systéme

V našom systéme budú nasledujúce roly:

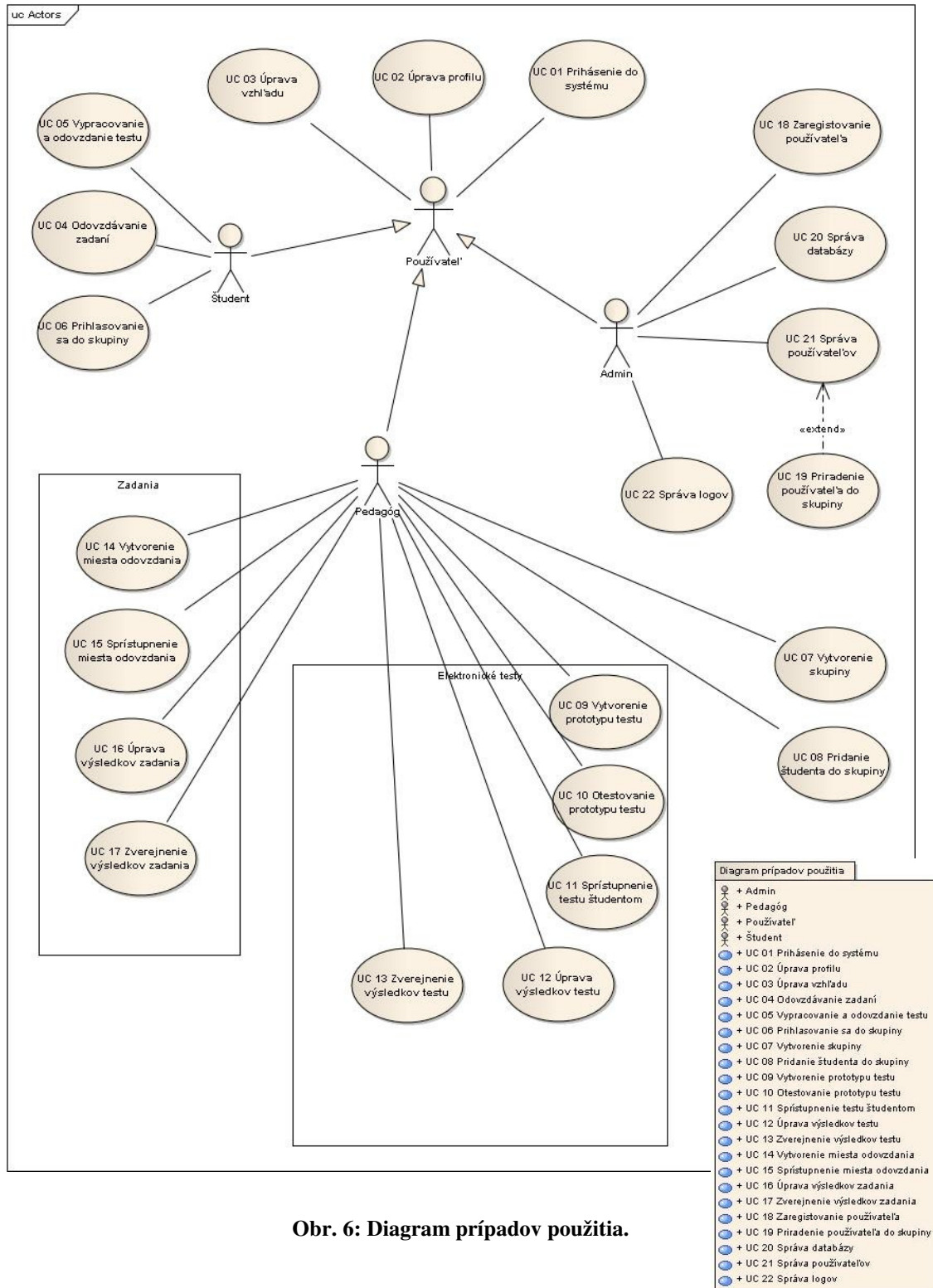
- **Používateľ** – základný nezaradený používateľ bez dodatočných práv, typicky čerstvo prihlásený študent s AIS kontom v systéme, ktorý ešte nie je priradený ku konkrétnemu systému
- **Študent** – študent, ktorému už prislúchajú určité práva v rámci výučby ako odovzdávanie zadaní, vypracovávanie testov, a pod.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

- **Pedagóg** – osoba, ktorá má pod správou tvorbu testov, zadaní a ich priradovanie k nej prislúchajúcim študentom
- **Administrátor** – osoba, ktorá spravuje systém a prislúchajú jej všetky práva

2.4 Diagram prípadov použitia

Na nasledovnom obrázku je zobrazený diagram prípadov použitia:



Obr. 6: Diagram prípadov použitia.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

2.5 Prípady použitia študenta

V nasledujúcich troch podkapitolách uvádzame podrobný popis prípadov použitia, ktoré sú znázornené na diagrame prípadov použitia v predchádzajúcej podkapitole. V tejto podkapitole uvádzame prípady použitia študenta, pričom niektoré z nich sa ale týkajú aj iných rolí.

Prihlásenie sa do systému	
Identifikátor	UC01
Názov	Prihlásenie sa do systému
Opis	Používateľ sa prihlási do systému, aby mohol vykonávať aktivity v systéme, ku ktorým má oprávnenie.
Priorita	Vysoká
Vstupné podmienky	Používateľ má konto v AIS, pomocou LDAP sú overené jeho prihlasovacie údaje. ALEBO Má špeciálne konto, ktoré nie je viazané na AIS konto.
Výstupné podmienky	Používateľ je prihlásený a môže pracovať v rámci jemu dostupnej funkcionality systému.
Používatelia	Používateľ, študent, pedagóg, administrátor
Kroky používateľa	Popis krokov:
1.	Cez prihlasovací formulár zadá svoje meno a heslo a stlačí tlačidlo prihlásiť.
2.	Overí sa, či ide o AIS alebo iného používateľa.
3a.	Ak ide o AIS používateľa, prebehne jeho LDAP autentifikácia.
3b.	Ak nejde o AIS používateľa, prebehne autentifikácia porovnaním zadaného mena a hesla s údajmi v databáze.
4.	Ak ide o prvé prihlásenie AIS používateľa, jeho konto v e-learningovom systéme ešte nie je vytvorené, automaticky sa vytvorí jeho konto s daným loginom, pričom ale ešte nie je zaradený do žiadnej skupiny ani nemá žiadnu rolu (študent, pedagóg, administrátor), takže má len veľmi obmedzené práva.
4a.	Používateľ je úspešne autentifikovaný a prihlásený do systému
4b.	Autentifikácia neprebehla úspešne, používateľ nezadal správne údaje. Používateľ môže ísť na krok 1. a opätovne sa pokúsiť prihlásiť.

Tab. 4: UC01 Prihlásenie sa do systému.

Úprava profilu	
Identifikátor	UC02
Názov	Úprava profilu
Opis	Umožňuje používateľovi zmeniť si vybrané informácie o sebe, ktoré chce zverejniť.
Priorita	Stredná
Vstupné podmienky	Používateľ je prihlásený v systéme.
Výstupné podmienky	Sú uložené zmeny používateľa v jeho profile.
Používatelia	Študent, pedagóg, administrátor
Kroky používateľa	Popis krokov:
1.	Zvolí v menu sekciu pre úpravu profilu.
2.	Cez príslušné rozhranie upraví formulár so svojimi údajmi a potvrdí

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

	tlačidlom.
3.	Môže prebehnúť kontrola niektorých polí (napr. formát e-mailu), ak bolo pole vyplnené nesprávne, musí ho používateľ zadať znova.
4.	Používateľ môže údaje ďalej upravovať, alebo môže rozhranie pre úpravu profilu opustiť.

Tab. 5: UC02 Úprava profilu.

Úprava vzhľadu	
Identifikátor	UC03
Názov	Úprava vzhľadu
Opis	Umožňuje používateľovi prispôbiť si vzhľad, napr. veľkosť písma, farba pozadia, atď.
Priorita	Nízka
Vstupné podmienky	Používateľ je prihlásený v systéme.
Výstupné podmienky	Sú uložené zmeny používateľa v nastaveniach vzhľadu.
Používatelia	Študent, pedagóg, administrátor
Kroky administrátora	Popis krokov:
1.	Zvolí v menu sekciu pre úpravu vzhľadu.
2.	V rozhraní pre úpravu vzhľadu zachová alebo zmení príslušné nastavenia a potvrdí zmeny.
3.	Používateľ môže vzhľad ďalej upravovať, alebo môže rozhranie pre úpravu vzhľadu opustiť.

Tab. 6: UC03 Úprava vzhľadu.

Odobovanie zadaní	
Identifikátor	UC04
Názov	Odobovanie zadaní
Opis	Umožňuje študentovi odovzdať vypracované zadanie.
Priorita	Vysoká
Vstupné podmienky	Študent je prihlásený v systéme. Študent vypracoval zadanie na základe textu zadania a má pripravené súbory vypracovaného zadania.
Výstupné podmienky	Študentovo zadanie je odovzdané a skompilované (ak prebehla kompilácia úspešne) a pedagóg ho môže skontrolovať.
Používatelia	Študent
Kroky administrátora	Popis krokov:
1.	Študent cez rozhrania odovzdá príslušné súbory na odovzdanie, ktoré prípadne musia mať určenú formu (v tom prípade sú v rozhraní uvedené pokyny).
2.	Študent voliteľne napíše do príslušnej položky formulára poznámky pre pedagóga k danému zadaniu.
3.	Študentove súbory (a prípadne poznámky) sú prichystané na odovzdanie.
4.	Študent si môže odovzdané súbory ešte skontrolovať ich otvorením.
5.	Študent potvrdí odovzdanie súborov a jeho akcia je už nezvratná.
Poznámky	Tento prípad použitia je všeobecný, zadaní bude viac typov, tu

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

	predpokladáme doma vypracovávané zadania, ktorých zdrojový kód sa odovzdáva a kompiluje na serveri. Takisto v rámci špecifikácie ešte niektoré kroky nechávame otvorené, ako otázky študentovej kontrolnej kompilácie v rámci e-learningového systému, kompilácia už pri odovzdávaní, alebo až pri kontrole pedagógom, atď.
--	---

Tab. 7: UC04 Odovzdávanie zadaní.

Vypracovanie a odovzdanie testu	
Identifikátor	UC05
Názov	Vypracovanie a odovzdanie testu
Opis	Umožňuje študentovi vypracovať test na overenie svojich vedomostí.
Priorita	Vysoká
Vstupné podmienky	Študent je prihlásený v systéme. Pedagóg študentovi sprístupnil test.
Výstupné podmienky	Výsledky testu a študentove odpovede sú zaznamenané v databáze a prípadne je študentovi zobrazené bodové ohodnotenie a správne riešenia (v závislosti od typu alebo nastavenia testu).
Používatelia	Študent
Kroky administrátora	Popis krokov:
1.	Študent si zvolí príslušný test na vypracovanie.
2.	Prečíta si pokyny a začne vyplňať jednotlivé úlohy vo vygenerovanom teste v určitom časovom limite (ak je stanovený).
3.	Študent odovzdá test.
4.	Podľa nastavenia testu (resp. podľa toho o aký test ide) sa mu zobrazia vyhodnotenú informácie o teste.
5.	Opäť v závislosti od konkrétneho testu je študentovi test znepřístupnený, alebo si test môže spraviť znova.
Poznámky	Tento prípad použitia je všeobecný, zahŕňa rôzne varianty testov, nie sú tu teda podrobne popísané kroky pre jednotlivé typy testov. Špecifikácia je čiastočná, treba zvážiť možnosť opätovného vygenerovania toho istého testu (keďže sa generuje náhodne na základe určených parametrov), takisto treba zvážiť konkrétne možnosti prezerania už vypracovaných testov študentmi.

Tab. 8: UC05 Vypracovanie a odovzdanie testu.

Prihlasovanie sa do skupiny	
Identifikátor	UC06
Názov	Prihlasovanie sa do skupín
Opis	Umožňuje sa používateľovi prihlásiť do určitej skupiny, napr. do skupiny určitého cvičenia.
Priorita	Vysoká
Vstupné podmienky	Používateľ je prihlásený v systéme. Používateľ má k dispozícii kľúč k prihláseniu do skupiny. Používateľ ešte nie je zaradený do žiadnej skupiny.
Výstupné podmienky	Používateľ je zaradený do skupiny a prípadne je zmenená jeho používateľská rola. Používateľovi je znepřístupnené rozhranie pre zadávanie kľúča, keďže

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

	už je členom skupiny.
Používatelia	Používateľ, študent, pedagóg, administrátor
Kroky administrátora	Popis krokov:
1.	Používateľ zadá do položky formulára pre prihlasovací kľúč pridelený kľúč a odošle ho.
2.	Overí sa, či je kľúč správny.
3a.	Ak kľúč je správny, používateľ je zaradený do skupiny a zároveň, ak je s členstvom v skupine viazaná aj určitá rola, ku ktorej zatiaľ nie je priradený, je priradený k tejto skupine (napr. čerstvo prihlásený používateľ sa prihlási na určité cvičenie, čím sa zároveň zaradí do role študenta)..
3b.	Ak kľúč nie je správny, používateľ nie je zaradený do skupiny, pre opätovný pokus o zaradenie pokračuje krokom 1.
Poznámky	Druhým spôsobom zaradenia sa do určitej skupiny je, že toto zaradenie vykoná administrátor, tento spôsob je nutný pri zmene skupiny.

Tab. 9: UC06 Prihlasovanie sa do skupín.

2.6 Prípady použitia pedagóga

V tejto podkapitole uvádzame prípady použitia týkajúce sa pedagóga.

Vytvorenie skupiny	
Identifikátor	UC07
Názov	Vytvorenie skupiny
Opis	Pedagóg sa prihlási do systému a chce vytvoriť skupinu (cvičenie), do ktorého priradí študentov, ktorých vyučuje.
Priorita	Vysoká
Vstupné podmienky	Pedagóg je prihlásený v systéme.
Výstupné podmienky	V systéme vznikne skupina, ku ktorej majú prístup študenti a môžu sa do nej prihlásiť.
Používatelia	Pedagóg
Kroky používateľa	Popis krokov:
1.	Pedagóg zadá cez formulár názov skupiny a jej detaily.
2.	Do formuláru zadá prihlasovací kľúč, pomocou ktorého sa študenti budú prihlasovať do skupiny.
3a.	Uloží vytvorenú skupinu do systému a zverejní ju pre študentov.
3b.	Uloží vytvorenú skupinu do systému, ale nezverejní ju pre študentov.

Tab. 40: UC07 Vytvorenie skupiny.

Pridanie študenta do skupiny	
Identifikátor	UC08
Názov	Pridanie študenta do skupiny
Opis	Pedagóg pridáva do svojej skupiny, teda cvičenia, študenta, ktorý sa do nej nemôže prihlásiť pomocou kľúča.
Priorita	Vysoká
Vstupné podmienky	Pedagóg je prihlásený v systéme, má vytvorenú cieľovú skupinu a študent má v systéme aktívne konto.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Výstupné podmienky	Študent je zaradený v pedagogovej skupine.
Používatelia	Pedagóg
Kroky používateľa	Popis krokov:
1.	Pedagóg si v zozname skupín zvolí tú, s ktorou chce pracovať.
2a.	Vo zvolenej skupine si zo zoznamu vyberie študenta, ktorého chce zaradiť do skupiny.
2b.	Vo zvolenej skupine si pomocou mena alebo priezviska vyhľadá študenta, ktorého chce zaradiť do skupiny.
3.	Uloží svoj výber a študent je priradený do skupiny.

Tab. 51: UC08 Priradenie študenta do skupiny.

Vytvorenie prototypu testu	
Identifikátor	UC09
Názov	Vytvorenie prototypu testu
Opis	Pedagóg pridá do systému prototyp testu, na základe ktorého si neskôr jednoducho vygeneruje skutočný test pre študenta.
Priorita	Vysoká
Vstupné podmienky	Pedagóg je prihlásený, v systéme sú vytvorené otázky a sú zaradené v okruhoch.
Výstupné podmienky	Je vygenerovaný prototyp testov.
Používatelia	Pedagóg
Kroky používateľa	Popis krokov:
1.	Pedagóg vyplní názov a detaily prototypu testu.
2.	Vyberie jednotlivé okruhy, ktoré sa majú v teste nachádzať.
3.	Zvolí koľko otázok sa má nachádzať v jednotlivých okruhoch.
4.	Zvolí bodové hodnotenie otázok.
5.	Uloží prototyp testu.

Tab. 62: UC09 Vytvorenie prototypu testov.

Otestovanie prototypu testu	
Identifikátor	UC10
Názov	Otestovanie prototypu testu
Opis	Pedagóg si môže otestovať vygenerovaný prototyp, aby sa presvedčil o jeho správnosti.
Priorita	Vysoká
Vstupné podmienky	Pedagóg je prihlásený, v systéme je vytvorená šablóna testu.
Výstupné podmienky	
Používatelia	Pedagóg
Kroky používateľa	Popis krokov:
1.	Pedagóg si vygeneruje test.
2.	Vyplní test.
3.	Odošle test na spracovanie.
4a.	Prezrie si svoje odpovede a vyhodnotenie testu.
4b.	Nemá záujem prezrieť si odpovede a pokračuje na vyhodnotenie testu.
5a.	Vypracovaný prototyp nie je dobrý, chce ho upraviť.
5b.	Prototyp je v poriadku a test je ukončený.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Tab. 73: UC10 Otestovanie prototypu testu.

Sprístupnenie testu študentom	
Identifikátor	UC11
Názov	Sprístupnenie testu študentom
Opis	Pedagóg sprístupní test študentom, teda na základe prototypu sa im vygenerujú testy.
Priorita	Vysoká
Vstupné podmienky	Pedagóg je prihlásený, v systéme je vytvorený prototyp testu.
Výstupné podmienky	Študentom je sprístupnená možnosť vyplnenia testu.
Používatelia	Pedagóg
Kroky používateľa	Popis krokov:
1.	Pedagóg si vyberie príslušný prototyp, podľa ktorého chce generovať testy.
2.	Vyberie si študentov alebo skupinu, ktorej sa má test zobrazit'.
3.	Vyberie možnosť zverejnenia testu študentom.

Tab. 84: UC11 Sprístupnenie testu študentom.

Úprava výsledkov testu	
Identifikátor	UC12
Názov	Úprava výsledkov testu
Opis	Pedagóg si uvedomí chybné zadanú odpoveď v teste alebo ho na ňu upozornia študenti a on chce upraviť bodové hodnotenie.
Priorita	Vysoká
Vstupné podmienky	Pedagóg je prihlásený, študenti už vypracovali konkrétny test.
Výstupné podmienky	Sú upravené body v systéme.
Používatelia	Pedagóg
Kroky používateľa	Popis krokov:
1.	Pedagóg si vyberie test, ktorého výsledky si chce prezrieť.
2.	Zobrazí si zoznam študentov a výsledkov z testu.
3.	Vyberie si študenta, ktorého bodové hodnotenie chce upraviť.
4.	Upraví mu bodové hodnotenie.
5.	Zadá do systému dôvod, pre ktorý bolo hodnotenie zmenené.
6a.	Uloží zmeny a skončí.
6b.	Neuloží zmeny.

Tab. 95: UC12 Úprava výsledkov testu.

Zverejnenie výsledkov testu	
Identifikátor	UC13
Názov	Zverejnenie výsledkov testu
Opis	Pedagóg chce zverejniť výsledky testu a finalizovať ich.
Priorita	Vysoká
Vstupné podmienky	Pedagóg je prihlásený, študenti už vypracovali konkrétny test.
Výstupné podmienky	Body za test už nie je možné upravovať.
Používatelia	Pedagóg
Kroky používateľa	Popis krokov:

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

1.	Pedagóg si vyberie test, ktorého výsledky chce zverejniť.
2.	Zverejní výsledky testu.

Tab. 106: UC13 Zverejnenie výsledkov testu.

Vytvorenie miesta odovzdania	
Identifikátor	UC14
Názov	Vytvorenie miesta odovzdania
Opis	Pedagóg vytvorí priestor, do ktorého možno odovzdať zadania.
Priorita	Vysoká
Vstupné podmienky	Pedagóg je prihlásený.
Výstupné podmienky	Je vytvorené a sprístupnené miesto, kam môže študent odovzdať zadanie.
Používatelia	Pedagóg
Kroky používateľa	Popis krokov:
1.	Pedagóg vyberie názov a opis miesta odovzdania.
2.	Určí čas, do ktorého je potrebné zadanie odovzdať.
3.	Zadá kontrolné vstupy a výstupy na základe ktorých sa testuje zadanie.
4.	Vyberie študentov alebo skupiny študentov, ktorí môžu odovzdávať zadanie.
5a.	Sprístupní miesto odovzdania študentom.
5b.	Miesto odovzdania iba uloží, ale nesprístupní ho.

Tab. 117: UC14 Vytvorenie miesta odovzdania.

Sprístupnenie miesta odovzdania	
Identifikátor	UC15
Názov	Sprístupnenie miesta odovzdania
Opis	Pedagóg zverejní miesto odovzdania študentom a tí doň môžu vkladať zadania.
Priorita	Vysoká
Vstupné podmienky	Pedagóg je prihlásený.
Výstupné podmienky	Je sprístupnené miesto, kam môže študent odovzdať zadanie.
Používatelia	Pedagóg
Kroky používateľa	Popis krokov:
1.	Pedagóg vyberie miesto odovzdania, ktoré chce sprístupniť.
2.	Sprístupní miesto odovzdania.

Tab. 128: UC15 Sprístupnenie miesta odovzdania.

Úprava výsledkov zadania	
Identifikátor	UC16
Názov	Úprava výsledkov zadania
Opis	Pedagóg môže napríklad na základe výborne vypracovaného zadania zmeniť body za zadanie, teda dať študentovi bonusové body.
Priorita	Vysoká
Vstupné podmienky	Pedagóg je prihlásený, študenti už odovzdali zadanie a tie boli otestované systémom a vyhodnotené.
Výstupné podmienky	Sú upravené body v systéme.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Používatelia	Pedagóg
Kroky používateľa	Popis krokov:
1.	Pedagóg si vyberie zadanie, ktorého výsledky si chce prezrieť.
2.	Zobrazí si zoznam študentov a výsledkov zo zadania.
3.	Vyberie si študenta, ktorého bodové hodnotenie chce upraviť.
4.	Upraví mu bodové hodnotenie.
5.	Zadá do systému dôvod, pre ktorý bolo hodnotenie zmenené.
6a.	Uloží zmeny a skončí.
6b.	Neuloží zmeny.

Tab. 139: UC16 Úprava výsledkov zadania.

Zverejnenie výsledkov zadania	
Identifikátor	UC17
Názov	Zverejnenie výsledkov zadania
Opis	Pedagóg chce zverejniť body za zadanie a finalizovať ich.
Priorita	Vysoká
Vstupné podmienky	Pedagóg je prihlásený, študenti už odovzdali konkrétne zadanie.
Výstupné podmienky	Body za zadanie už nie je možné upravovať.
Používatelia	Pedagóg
Kroky používateľa	Popis krokov:
1.	Pedagóg si vyberie zadanie, ktorého výsledky chce zverejniť.
2.	Zverejní výsledky zadania.

Tab. 20: UC17 Zverejnenie výsledkov zadania.

2.7 Prípady použitia administrátora

V tejto podkapitole uvádzame prípady použitia týkajúce sa administrátora.

Zaregistrovanie používateľa	
Identifikátor	UC18
Názov	Zaregistrovanie používateľa
Opis	Umožňuje administrátorovi pridať do systému nového používateľa.
Priorita	Vysoká
Vstupné podmienky	V databáze existuje administrátorské konto a administrátor je prihlásený do systému. Registrácia sa týka špeciálnych kont používateľov, ktorých konto nebude viazané na AIS (tí, ktorí majú AIS konto sú automaticky registrovaní pri prvom prihlásení).
Výstupné podmienky	Používateľ je uložený v databáze a je registrovaný.
Používatelia	Administrátor
Kroky administrátora	Popis krokov:
1.	Prihlási sa do systému a vstúpi do administrátorského rozhrania.
2.	Cez administrátorské rozhranie vyplní príslušný formulár s údajmi používateľa.
3.	Vyplní jeho login a heslo a prípadne ďalšie údaje ako rola, či skupina, v databáze sa zaznačí, že nejde o AIS používateľa.
4.	Odošle formulár.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

5a.	Ak prebehne všetko bez problémov, všetky parametre sú korektne vyplnené a nevyskytli sa ani iné problémy, je používateľ pridaný.
5b.	Pridanie bolo neúspešné kvôli zlému vyplneniu formulára, administrátor sa vracia na krok 2.
5c.	Pridanie bolo neúspešné z iného neočakávaného dôvodu, napr. problémy s databázou, treba zistiť príčinu problémov a odstrániť ich, následne sa opätovne pokúsiť pridať používateľa.

Tab. 214: UC18 Zaregistrovanie používateľa.

Priradenie používateľa do skupiny	
Identifikátor	UC19
Názov	Priradenie používateľa do skupiny
Opis	Umožňuje administrátorovi priradiť používateľa do určitej používateľskej skupiny.
Rozširuje prípad použitia:	UC21 Správa používateľov
Priorita	Vysoká
Vstupné podmienky	V databáze existuje administrátorské konto a administrátor je prihlásený do systému.
Výstupné podmienky	Používateľská skupina daného používateľa je zmenená.
Používatelia	Administrátor
Kroky administrátora	Popis krokov:
1.	Prihlási sa do systému a vstúpi do administrátorského rozhrania.
2.	Cez administrátorské rozhranie vyberie príslušného používateľa, ktorého chce zaradiť do skupiny, alebo chce zmeniť jeho zaradenie do skupiny.
3.	Zmení zaradenie používateľa do skupiny a ak pripadá k danej skupine aj príslušná rola, zmení sa aj rola.
4a.	Zaradenie bolo úspešné, administrátor môže vykonávať ďalšie činnosti.
4b.	Zaradenie bolo neúspešné z neočakávaného dôvodu, napr. problémy s databázou, treba zistiť príčinu problémov a odstrániť ich, následne sa opätovne pokúsiť zaradiť používateľa do skupiny.

Tab. 22: UC19 Priradenie používateľa do skupiny.

Správa databázy	
Identifikátor	UC20
Názov	Správa databázy
Opis	Umožňuje administrátorovi vykonávať základné operácie s databázou.
Priorita	Vysoká
Vstupné podmienky	V databáze existuje administrátorské konto a administrátor je prihlásený do systému.
Výstupné podmienky	Operácia vybraná administrátorom sa bezproblémovo vykoná. Konzistentnosť databázy je zachovaná.
Používatelia	Administrátor
Kroky administrátora	Popis krokov:
1.	Prihlási sa do systému a vstúpi do administrátorského rozhrania.
2.	Vyberie si cez administrátorské rozhranie operáciu, ktorú nad databázou chce vykonať: import, export, úprava.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

3a.	Vo formulári pre importovanie dát zadá cestu k súboru v niektorom z podporovaných formátov a taktiež kde do databázy sa má uložiť.
3b.	Vo formulári pre exportovanie dát zadá cestu a meno súboru do ktorého sa databáza exportuje.
3c.	Vo formulári pre úpravu databázy vyberie ktoré údaje v databáze chce upraviť a zadá ich nové hodnoty.
4.	Potvrdí danú operáciu a odošle vyplnený formulár systému.
5a.	Systém vykoná požadovanú operáciu a vytvorí záznam o vykonanej operácii do logu.
5b.	Nastane chyba pri vykonávaní operácie, program vyhodí chybové hlásenie a nastane návrat do bodu 2.

Tab. 23: UC20 Správa databázy.

Správa používateľov	
Identifikátor	UC21
Názov	Správa používateľov
Opis	Umožňuje administrátorovi prístup k informáciám o používateľoch a manipuláciu s heslami používateľov mimo AIS.
Priorita	Vysoká
Vstupné podmienky	V databáze existuje administrátorské konto a administrátor je prihlásený do systému.
Výstupné podmienky	Všetky používateľské kontá majú vyplnené povinné osobné údaje.
Používatelia	Administrátor
Kroky administrátora	Popis krokov:
1.	Prihlási sa do systému a vstúpi do administrátorského rozhrania.
2.	Pristúpi k rozhraniu pre správu používateľov.
3.	V rozhraní pre správu používateľov si vyberie niektorý z používateľských účtov.
4a.	Ak sa jedná o používateľa s kontom mimo AIS, sú dostupné a editovateľné všetky jeho údaje nachádzajúce sa v systéme vrátane hesla.
4b.	Ak sa jedná o používateľa s kontom v AIS, sú dostupné a editovateľné všetky jeho údaje nachádzajúce sa v systéme okrem hesla.
5a.	Ak boli vykonané v údajoch používateľa zmeny, pošle sa požiadavka na zmenu systému.
5b.	Ak neboli vykonané zmeny, návrat do bodu 2.
6a.	Požiadavka na zmenu používateľských údajov bola prijatá a vykonaná, systém vytvorí záznam o vykonanej operácii do logu.
6b.	Nastane chyba pri vykonávaní operácie, program zobrazí chybové hlásenie a nastane návrat do bodu 2.

Tab. 24: UC21 Správa používateľov.

Správa logov	
Identifikátor	UC22
Názov	Správa logov
Opis	Umožňuje administrátorovi prezerat' a pridavat' poznámky k logom.
Priorita	Stredná
Vstupné podmienky	V databáze existuje administrátorské konto a administrátor je prihlásený

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

	do systému.
Výstupné podmienky	Všetky informácie o dôležitých zmenách v systéme v logoch ostanú zachované.
Používatelia	Administrátor
Kroky administrátora	Popis krokov:
1.	Prihlási sa do systému a vstúpi do administrátorského rozhrania.
2.	Otvorí rozhranie pre správu logov.
3.	Má možnosť prezerat' si všetky záznamy o zmenách v systéme a databáze.
4a.	Skončí prezeranie logov bez zmeny.
4b.	Pridá poznámku ku konkrétnej zmene v systéme.

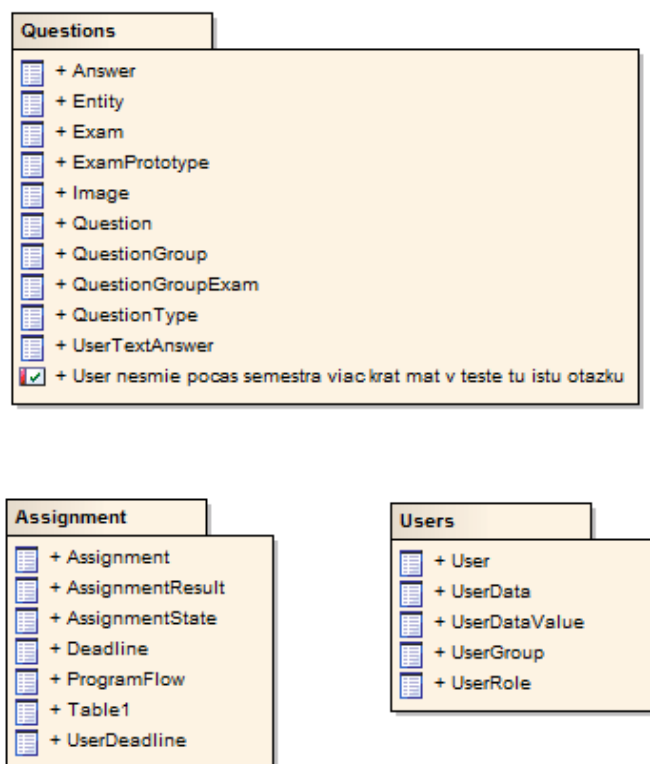
Tab. 25: UC22 Správa logov.

3 Návrh

3.1 Dátový model

Dátový model pre projekt pokrýva tri základné časti projektu. Prvá časť sa zaoberá generovaním testov. Pokrýva celú časť od generovania a priradzovania otázok pre jednotlivých používateľov. Druhá časť sa zaoberá odovzdávaním a validovaním zadaní. Tretia časť zobrazuje používateľov systému.

Dátový model je rozdelený do troch hlavných diagramov a jedného doplnujúceho. Každý z nich zobrazuje inú časť systému. Niektoré entity sú samozrejme spoločné a nachádzajú sa vo viacerých diagramoch.



Obr. 7: Dátový model.

Model pre otázky

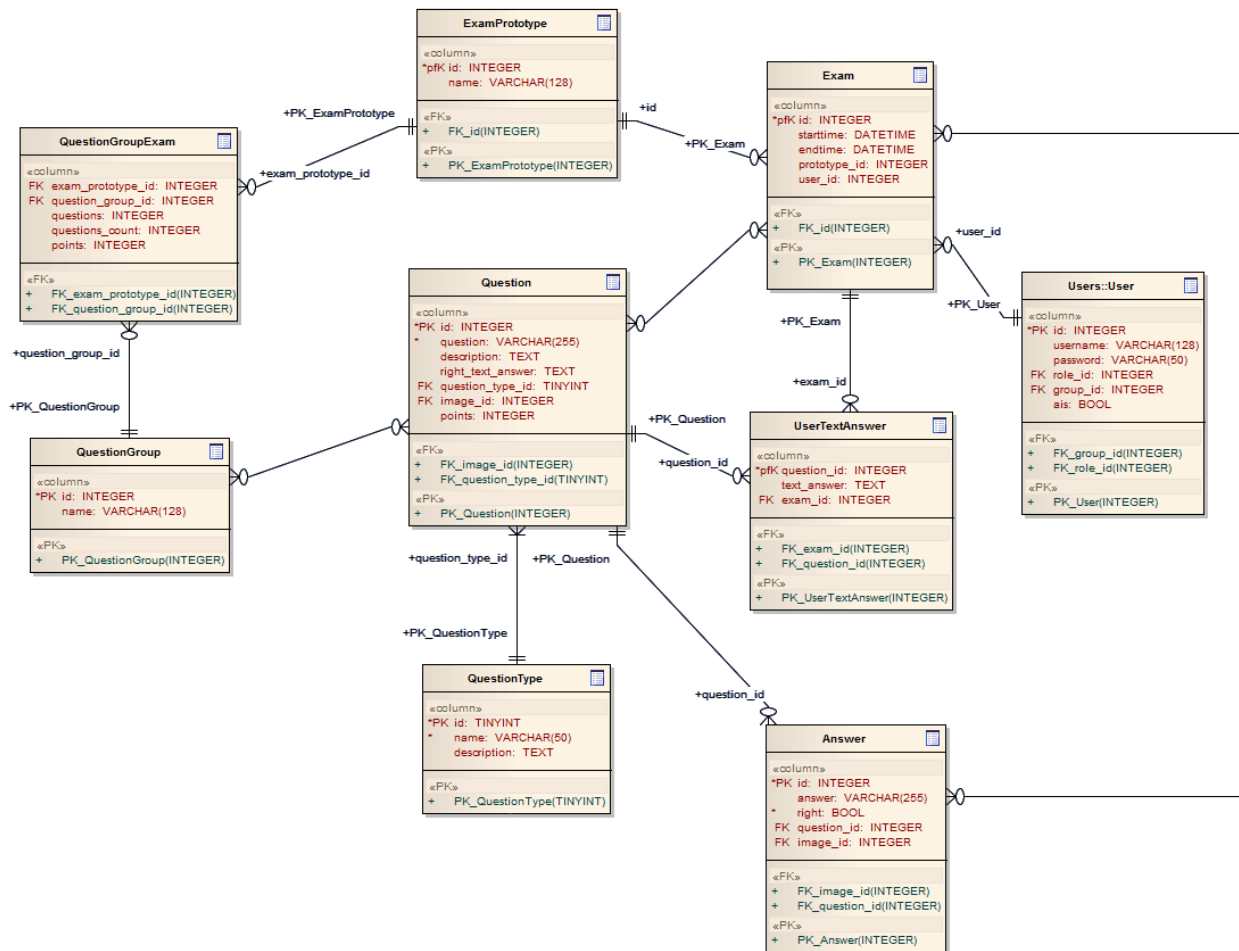
Diagram zobrazuje model k testovým otázkam. Ide o časť, kde sú uchovávané otázky a odpovede. Tieto otázky sú spájané do skupín a tiež obsahuje vzorové testy, ktoré vytvorili učitelia. Základnou entitou je tabuľka *Questions*. Ide o tabuľku ktorá uchováva testové otázky. Každá otázka musí byť nejakého typu, preto sa jej priraduje *QuestionType* podľa ktorého sa následne generuje daná otázka.

Pred vytvorením konkrétneho testu (entita *Exam*), ktorý je už naviazaný na konkrétneho používateľa(entita *User*), je potrebné vytvoriť takzvaný testový prototyp (entita *ExamPrototype*). Tento testový prototyp uchováva kritéria, podľa ktorých sa budú generovať testy pre používateľov. Čiže prototyp obsahuje informácie o tom, z ktorej skupiny otázok sa vyberie koľko otázok a za koľko bodov.

Každá odpoveď priradená k otázke obsahuje informáciu o tom, či je to odpoveď správna. Po vyplnení testu používateľom vzniknú väzby medzi testom a odpoveďami a následne podľa toho sa

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

dá identifikovať, ktoré odpovede boli správne a ktoré nie. Pre špeciálne textové otázky, kde je potrebné používateľom vyplniť nejaký súvislý text alebo číslo, sa odpoveď zaznamenáva do tabuľky *UserTextAnswer*. Takéto typy otázok nemajú priradené žiadne odpovede, ale správnu odpoveď si uchovávajú samé.

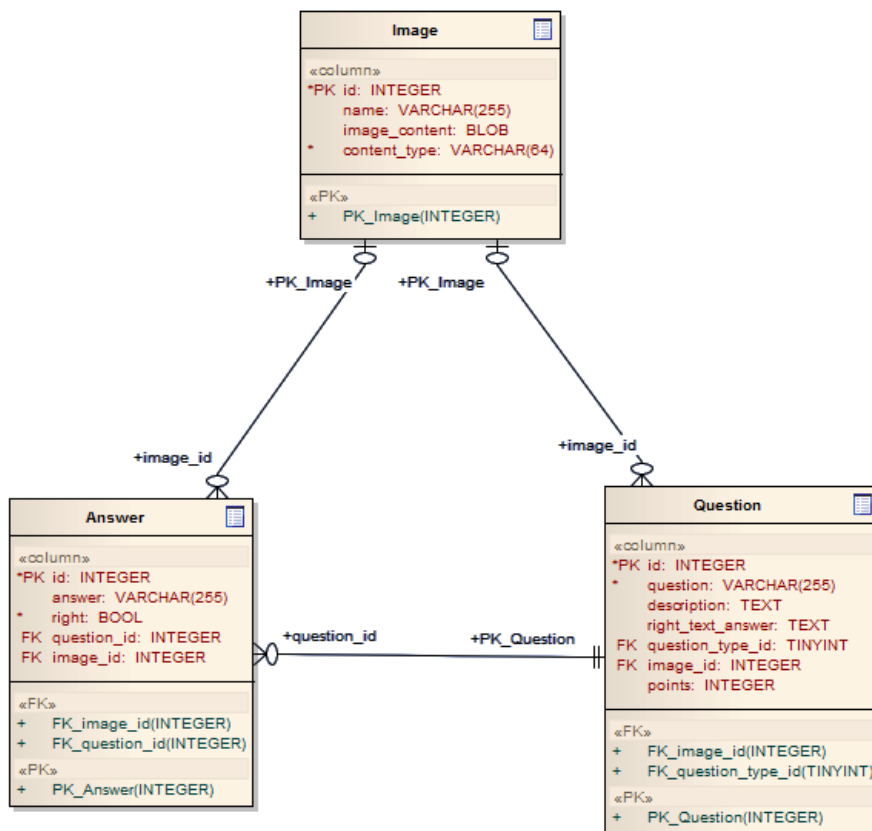


Obr. 8: Dátový model pre otázky.

Model pre obrázky

Veľmi jednoduchý diagram, ktorý dopĺňa funkcionality ku otázkam a odpovediam. Pridáva možnosť linkovať k nim obrázok (entita *Image*). Pri niektorých typoch otázok alebo odpovedí to môže byť užitočné.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

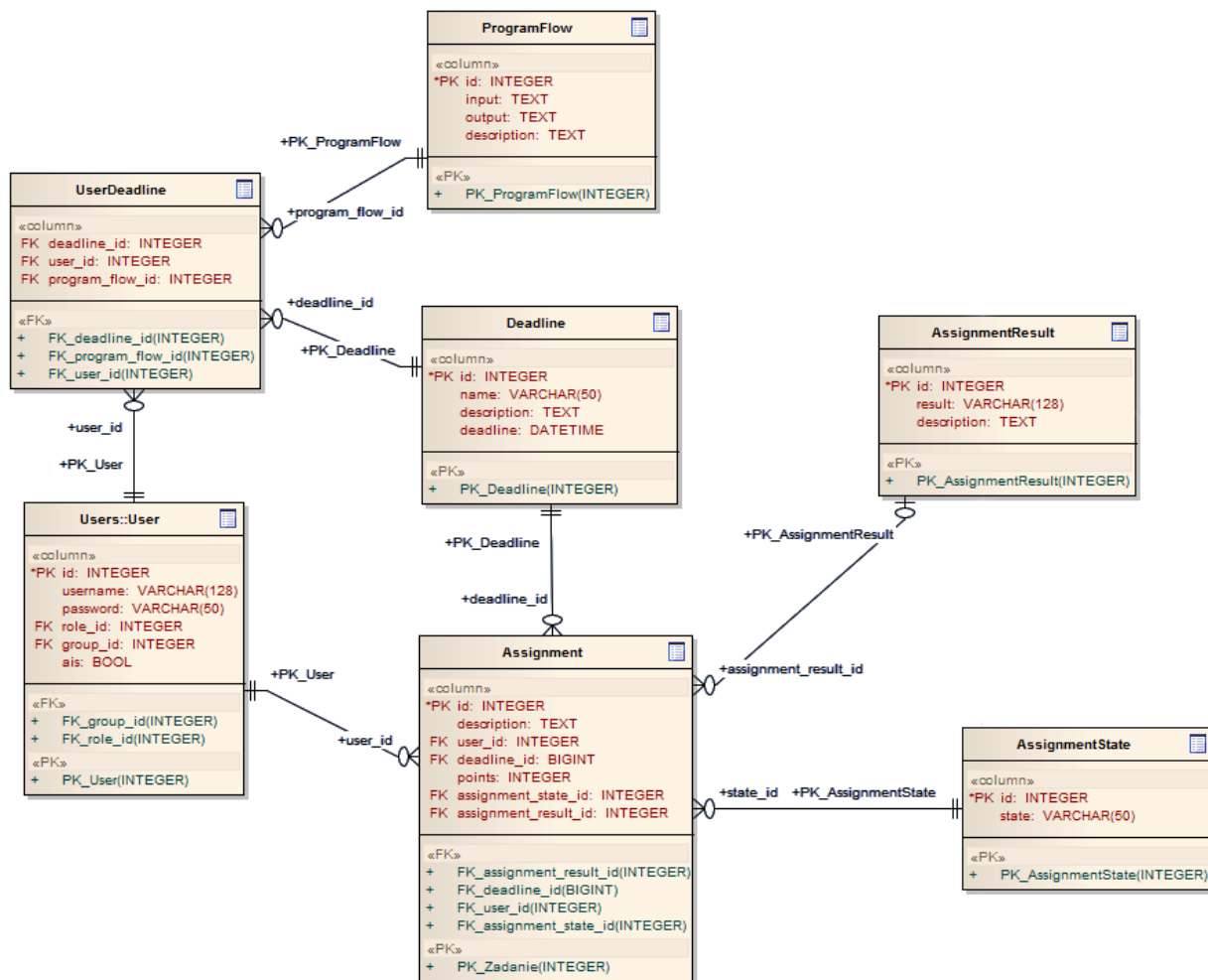


Obr. 9: Dátový model pre obrázky.

Model pre zadania

Pre jednotlivých používateľov môžu byť vytvorené v systéme takzvané časy pre odovzdanie zadaní. To pokrýva entita *Deadline*. Pri pridaní záznamu *Deadline* používateľovi je potrebné vybrať takzvaný *ProgramFlow*, ktorý predstavujú konkrétne zadanie, vstupy a požadované výstupy. Toto spojenie sa nachádza v tabuľke *UserDeadline*. Následne ak používateľ vloží svoje zadanie, vznikne záznam v tabuľke *Assignment*. Jeho zadanie sa automatické podľa priradeného *ProgramFlow* skontroluje a vyhodnotí. Zadanie sa nachádza vždy v nejakom stave - *AssignmentState* a jeho výsledok je vyjadrený priradením jedného z *AssignmentResult*.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

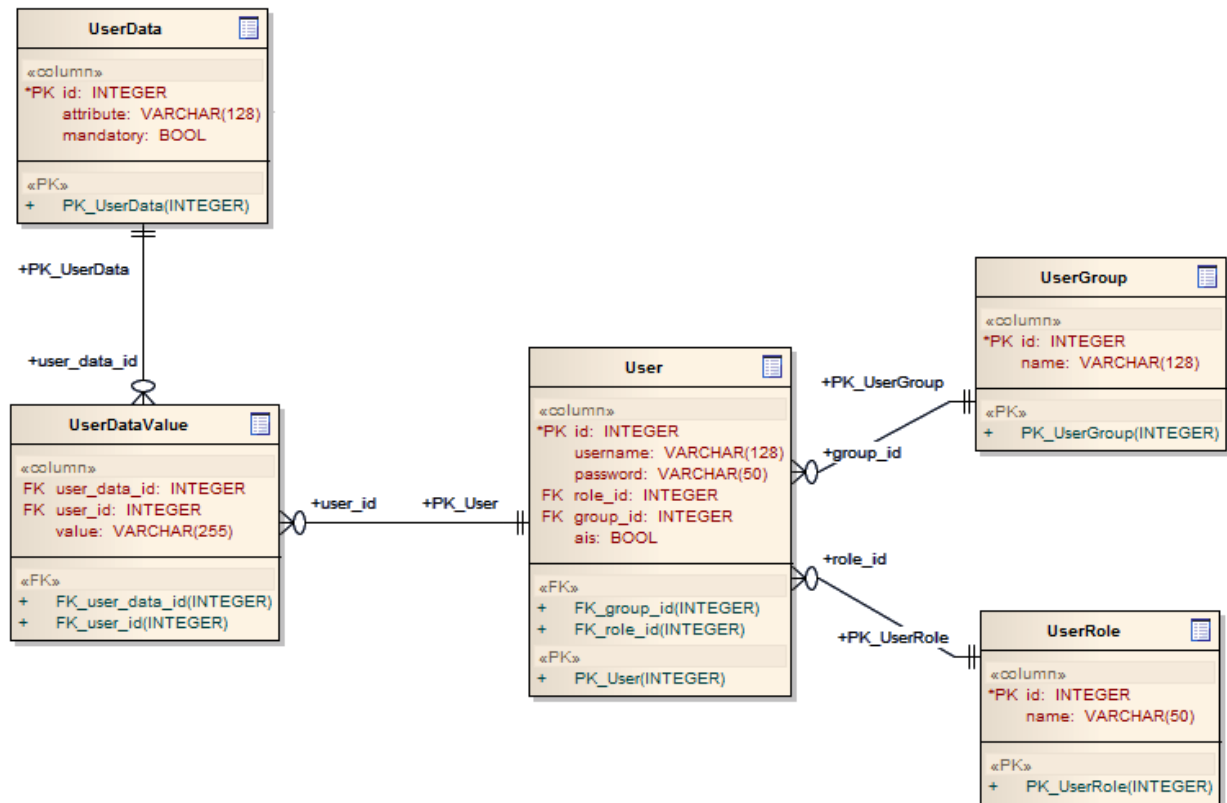


Obr. 10: Dátový model pre zadania.

Model pre používateľov

Používatelia v systéme a ich väzby na ďalšie časti systému sú vyjadrené v diagrame Users. Každý používateľ má v systéme nejakú rolu podľa priradenia k *UserRole*. Každý používateľ patrí aj do nejakej skupiny *UserGroup*. Pre uchovávanie rôznych informácií o používateľoch sme zvolili zatiaľ všeobecný model, aby bolo možné tieto informácie pridávať dodatočne. To znamená, že v tabuľke *UserData* sa nachádza zoznam informácií, ktoré o používateľoch môžeme uchovávať. Konkrétne informácie sú uložené v tabuľke *UserDataValue*, ktorá sa vždy viaže na konkrétneho používateľa a konkrétny záznam *UserData*. Spolu s touto väzbou je uložená aj konkrétna hodnota.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

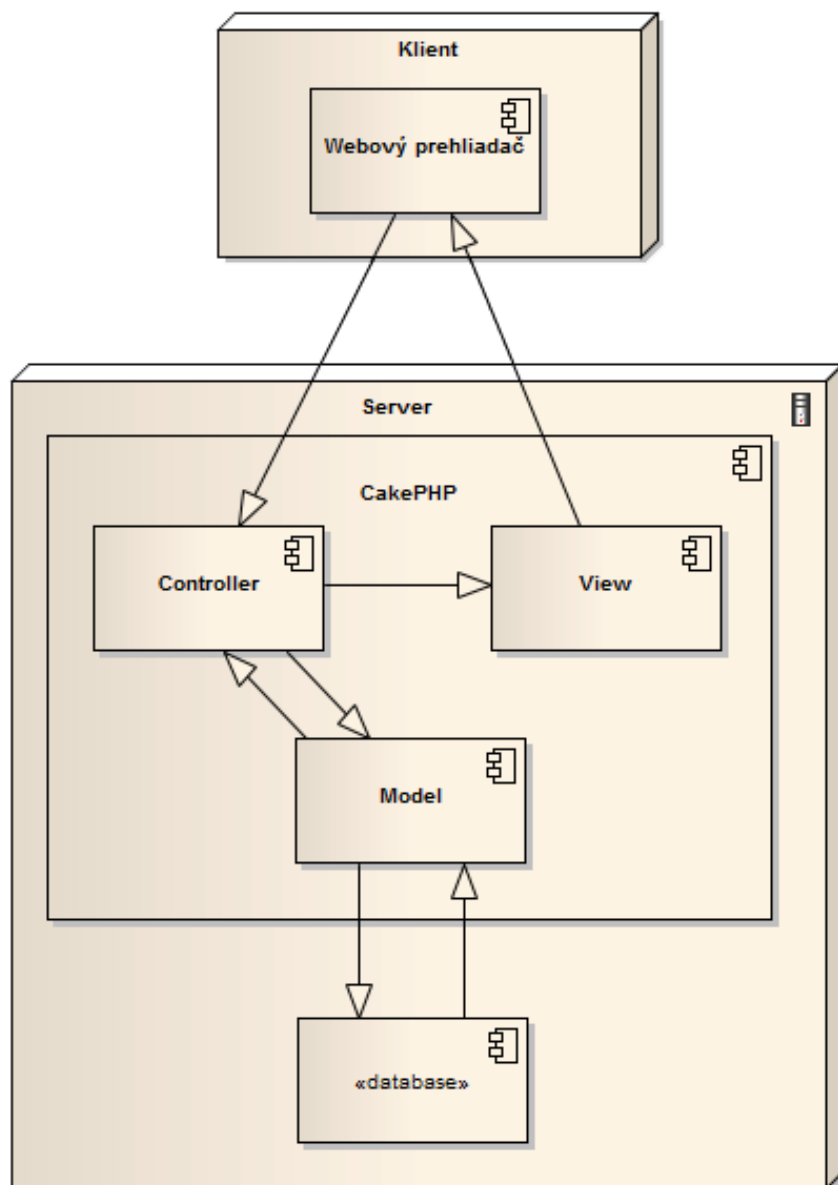


Obr. 11: Dátový model pre používateľov.

3.2 Architektúra systému

Rozhodli sme sa implementovať systém ako webovú aplikáciu. Systém bude založený na architektonickom vzore MVC (model-view-controller). Tento vzor pracuje na princípe oddelenia údajov aplikácie (model), spôsobu zobrazenia (view) a spôsobu práce s modelom (controller). Hlavnou výhodou použitia tohto modelu je oddelenie vzájomných závislostí jednotlivých komponentov. Pri zmene jedného nie je nutná úprava ostatných. Na jeho implementáciu použijeme prostredie CakePHP. Úplný návrh architektúry systému je na nasledujúcom obrázku.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.



Obr. 12: Návrh architektúry systému.

Klientská časť

Tvorí ju webový prehliadač klienta. Poskytuje používateľovi rozhranie na prácu so systémom.

Serverová časť

Skladá sa z dvoch častí – databázy a prostredia CakePHP (architektúra MVC). Stará sa o spracovanie požiadaviek zo strany používateľa a prezentáciu dát používateľovi.

Model

Model predstavuje abstrakciu databázy, s ktorou aplikácia pracuje. Každá trieda modelu reprezentuje jednu tabuľku databázy a jej vzťahy s inými tabuľkami. Jednotlivé triedy modelu určujú vzťahy medzi údajmi a zabezpečujú operácie nad nimi. Model predstavuje rozhranie pre čítanie, zápis, mazanie a úpravu údajov aplikácie.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

View

View vytvára výstup pre používateľa, prezentovaný časťou modelu aplikácie. Poskytuje používateľovi prostriedky pre prácu s ním. Nepriamo, cez controller, mu dovoľuje modifikovať model aplikácie. Používateľ pracuje len s touto časťou architektúry.

Controller

Controller spracováva požiadavky od používateľa. Slúži ako logický aparát celej aplikácie. Spolupracuje so všetkými súčasťami MVC architektúry. Využíva model na získanie prístupu k údajom aplikácie, na ich úpravu a pridávanie nových údajov. Po skončení činnosti je aktuálny stav znova prezentovaný používateľovi cez view. Controller tvorí hlavnú časť architektúry. Je to jediný komponent, ktorý zapisuje priamo do modelu a zabezpečuje správnosť aplikácie.

Databáza

Použijeme relačnú databázu MySQL.

4 Prototyp

Prvý prototyp nášho systému bol vytvorený v zimnom semestri akademického roku 2009/2010. Rozhodli sme sa pre model evolučného vývoja, takže prototyp bude použitý ako základ pre finálny systém.

4.1 Ciele prototypovania

Náš prototyp bol navrhnutý a implementovaný najmä za účelmi:

- Možnosti prezentácie a získania spätnej väzby od potenciálnych používateľov.
- Načrtnutie formy finálneho systému.
- Identifikovania problematických oblastí ďalšieho vývoja.
- Overenia správnosti navrhnutého systému.

V rámci prototypu budú implementované niektoré základné funkcionality, ktoré boli navrhnuté do finálneho systému. Hlavným cieľom prototypu je identifikovať nedostatky v návrhu, ktoré neboli na prvý pohľad zrejmé a tiež určiť, ktoré z plánovaných funkcionalít je možné alebo potrebné do projektu pridať. Prototyp by mal taktiež umožniť identifikáciu funkcií, na ktoré sa v návrhu buď pozabudlo, alebo vôbec neboli uvažované.

4.2 Vybrané časti systému na prototypovanie

Ako už bolo spomenuté, prototyp slúži ako základ pre ďalší vývoj systému. Do prototypu sme sa rozhodli zahrnúť základné funkcie, ktoré by mal náš systém používateľom poskytovať a to:

- Prihlásenie používateľov a ich autentifikácia cez LDAP.
- Vytváranie otázok a ich komponovanie do testov.
- Vypracovávanie a odovzdanie testov.
- Vyhodnocovanie testov a zverejnenie ich výsledkov.

Implementovaným funkciám zodpovedajú nasledovné prípady použitia z návrhu:

Používateľ

UC01 Prihlásenie sa do systému

Študent

UC05 Vypracovanie a odovzdanie testu

Pedagóg

UC09 Vytvorenie prototypu testu

UC11 Sprístupnenie testu študentom

UC12 Úprava výsledkov testu

UC13 Zverejnenie výsledkov testu

Administrátor

UC18 Zaregistrovanie používateľa

UC21 Správa používateľov

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

4.3 Scenár použitia systému

Bežné použitie prototypu by mohlo prebiehať nasledovne:

1. Učiteľ sa prihlási a zadá možnosť vygenerovať test z nejakých oblastí, určí parametre žiadaného testu a vytvorí prototyp testu.
2. Daný test sprístupní študentom.
3. Prihlási sa študent, vyberie si možnosť vypracovať sprístupnený test. Zobrazí sa mu príslušný test a časomiera.
4. Študent test vypracuje a odovzdá. Po odovzdaní sa mu zobrazí predbežný počet bodov, na koľko daný test vypracoval.
5. Pedagóg si prezrie vypracovaný test a v prípade potreby upraví hodnotenie v niektorých otázkach. Následne zverejní definitívne výsledky testu.
6. Študent si môže prezrieť definitívne hodnotenie vypracovaného testu.

4.4 Aplikačná vrstva

Aplikačnú vrstvu aplikácie tvoria kontolery a ich služby, resp. metódy, ktoré využívajú prístup k aplikačným dátam a vykonávajú nad nimi operácie. Aplikačná vrstva je teda vrstva, v ktorej je riešená aplikačná logika, t.j. funkcie, ktoré systém vykonáva. Aplikácia obsahuje služby všeobecné, ktoré sa týkajú všetkých používateľov, ako aj také, ktoré sa týkajú len určitej kategórie používateľov.

Prototyp by mal poskytovať nasledujúce služby:

Používateľ

Prihlásenie cez LDAP

Do systému sa môže prihlásiť každý, kto má vytvorené konto na doméne stuba.sk

Prezeranie si profilu

Každý používateľ si môže prezerať svoje osobné údaje

Študent

Vypracovanie a odoslanie testu

Študent má možnosť vypracovať zverejnený test a po vypracovaní ho odoslať na skontrolovanie. Odoslať môže každý test iba raz.

Prezeranie si výsledkov testu

Študent má možnosť prezerať si definitívne bodové ohodnotenia všetkých testov ktoré vypracoval

Prístup k študijným materiálom

Všetci študenti majú prístup k sekcii „materiály“ , obsahujúcej sťahovateľné študijné materiály a užitočné odkazy

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Pedagóg

Pridávanie a úprava otázok (obsahuje manažment odpovedí, aj textových)

Pedagóg má právo pridávať vlastné testové otázky, ktoré môžu byť následne začlenené do testov, a tiež môže editovať a mazať už existujúce otázky

Pridávanie a úprava skupín otázok

Taktiež má možnosť pridávať a upravovať skupiny otázok (okruhy) a následne do nich priradovať nové otázky.

Pridávanie a úprava obrázkov

Pedagóg má možnosť pridávať, upravovať a vymazávať obrázky, ktoré môžu byť použité pri zadaniach otázok.

Vytváranie a úprava prototypov testov

Nie je nutné vytvárať test odznova pri každom testovaní. Je možné vytvárať tzv. prototypy testov, ktoré je možné znovu použiť alebo upraviť podľa potreby.

Sprístupňovanie testov

Pedagóg môže pomocou touto funkciou umožniť študentom vypracovanie vybratého prototypu testu.

Úprava hodnotení vypracovaných testov

Hodnotenie každého vypracovaného testu môže pedagóg pred definitívnym zápisom výsledkov upraviť.

Zverejnenie hodnotení testov

Ak je pedagóg spokojný s konečným hodnotením testov, môže ich zverejniť, a tým výsledky sprístupniť príslušným študentom.

Administrátor

Označenie pedagógov s právami k manažmentu testov

Administrátor určuje, ktorí používatelia majú práva pedagógov a môžu manipulovať s testami a databázou otázok.

4.5 Prezentáčna vrstva

Prezentáčna vrstva poskytuje používateľom údaje a informácie, ktoré sú výstupom aplikácie, a tiež od nich prijíma vstupné údaje určené na spracovanie. Prezentáčnu vrstvu tvoria dynamické XHTML stránky, ktoré obsahujú formuláre na získanie používateľských dát, taktiež ju tvoria súbory CSS, ktoré ovplyvňujú štýl vzhľadu stránky. Generovanie týchto súčastí prezentáčnej vrstvy zabezpečujú v MVC frameworku CakePHP modely. Tie prevezmú dáta (z databázy, LDAP položky, ...), nad ktorými kontrolery vykonali operácie a prevedú ich do podoby vhodnej pre prezentáciu v klientovi (teda na príslušné XHTML+CSS stránky). Naša aplikácia predstavuje model tenkého klienta. Databáza ako aj väčšina aplikačného spracovania sa nachádza na serveri, aj keď to neplatí úplne, lebo využívame aj klientský JavaScript kód.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

4.6 Zhrnutie

Do prvého prototypu sa nám nepodarilo implementovať úplne všetky navrhnuté funkcionality, avšak môžeme s istotou povedať, že prototyp splnil svoj účel. Ilustruje, ako bude vyzeráť finálny systém, zviditeľňuje potenciálne ťažkosti a problémy, s ktorými sa môžeme stretnúť pri vývoji finálneho systému a tiež nám umožňuje získať cennú spätnú väzbu od používateľov. Vďaka nemu máme oveľa lepšiu a konkrétnejšiu predstavu, akým smerom pokračovať vo vývoji nášho systému, ako aj solídny základ, na ktorom môžeme pri ďalšom vývoji stavať.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

5 Používateľská príručka

5.1 Inštalácia

Aplikácia je dostupná zadaním URL adresy <http://labss2.fiit.stuba.sk/TeamProject/2009/team11is-si/appop/> v akomkoľvek webovom prehliadači, ktorý spĺňa súčasné W3C štandardy. Taktiež je potrebné mať aktivovaný JavaScript.

Inštalácia na vlastnom serveri a lokálnom počítači

V prípade nasadenie aplikácie na vlastnom serveri, prípadne lokálnom počítači, je potrebné vopred nainštalovať potrebné aplikačné vybavenie (ak toto nainštalované nie je) a umiestniť súbory aplikácie do zvoleného priečinku. Týmto aplikačným vybavením je webový server Apache, interpret jazyka PHP a databázový systém MySQL. Aplikácia bola vyvíjaná pomocou PHP interpreta verzie 5.3 a databázového systému MySQL 5.1 na webovom serveri Apache verzie 2.2. Je vhodné tiež použiť tieto, alebo novšie verzie. Samozrejme však aplikácia môže byť funkčná aj v starších verziách, ale táto skutočnosť nebola testovaná. Pri inštalácii týchto nástrojov sa riadime pokynmi v ich dokumentácii. Môžeme použiť aj balík nástrojov ako napr. XAMPP, ktorý zabezpečí súčasne inštaláciu všetkých týchto súčastí a poskytuje aj rozhranie na ich konfiguráciu, či spúšťanie.

Po úspešnej inštalácii týchto nástrojov sa treba presvedčiť, či sú správne nastavené konfiguračné súbory servera Apache a interpreta PHP.

V konfiguračnom súbore `httpd.conf` (umiestnenom v adresári `apache/conf`) by mali byť odkomentované nasledujúce riadky:

```
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
LoadModule ldap_module modules/mod_ldap.so
```

V konfiguračnom súbore `php.ini` (umiestnený v adresári `php`) by mali byť odkomentované nasledujúce riadky:

```
extension=php_ldap.dll
```

Celá aplikácia sa nachádza v adresári `appop` dostupnom na adrese <http://labss2.fiit.stuba.sk/TeamProject/2009/team11is-si/appop/>. Tento treba skopírovať do koreňového adresára webového servera (je pomenovaný v závislosti od nastavenia Apache servera, najčastejšie `www_root`, `www`, `htdocs`).

Konfigurácia databázy

Pre samotným rozbehnutím aplikácie je ešte nutné správne nakonfigurovať databázu. Cesta ku konfiguračnému súboru `database.php` je `appop/app/config/`. Predvolené sú nastavenia prislúchajúce k databáze umiestnenej na <http://www.tomaj.sk/> (dočasne, bude presunuté na server labss2).

Ak chceme pracovať nad inou databázou je potrebné tieto nastavenia zmeniť na vlastné nastavenia.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Súbor s SQL príkazmi, ktoré sa použijú na vygenerovanie databázy a jej tabuliek stiahneme príkazom

```
mysqldump -u LOGIN -pPASSWORD -h HOST DATABASE > subor.sql
```

Položky LOGIN, PASSWORD, HOST a DATABASE opíšeme podľa údajov v konfiguračnom súbore.

Pomocou príkazu

```
mysql -u USER -pPASSWORD -h HOST DATABASE < subor.sql
```

Položka HOST je adresa, resp. názov servera (na lokálnom počítači pravdepodobne localhost) a DATABASE ľubovoľné zvolené meno databázy. Následne zmeníme údaje konfiguračného súboru v súlade s nastavenou databázou.

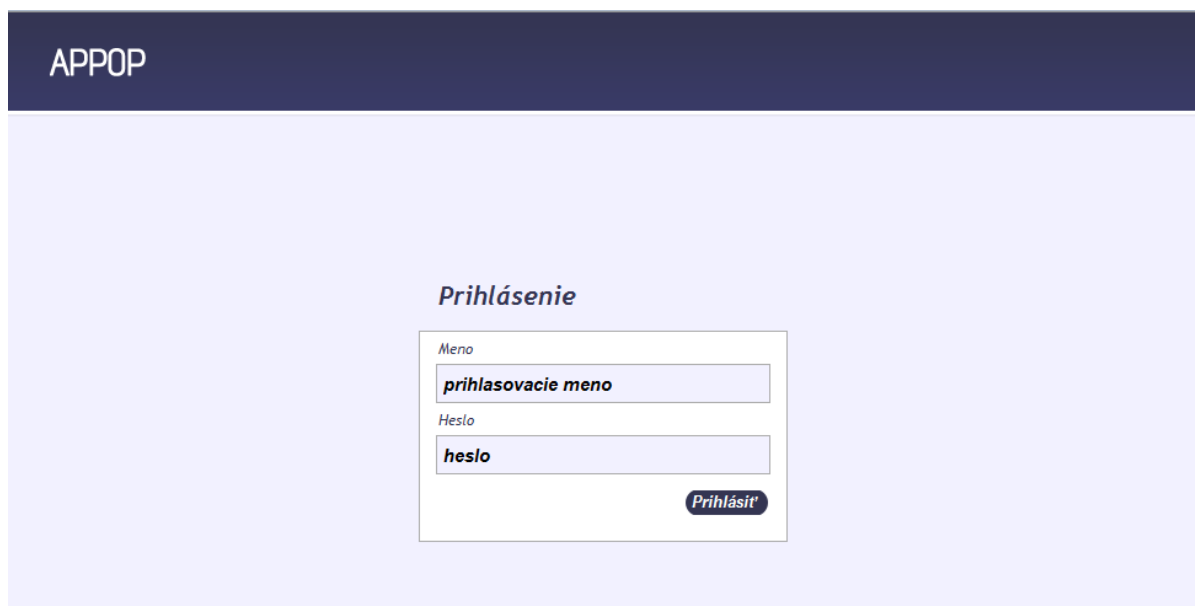
Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

5.2 Použitie

Prihlásenie

Pri prvom spustení prototypu sa každému používateľovi zobrazí prihlasovací formulár. Pre prihlásenie používateľ zadá svoje prihlasovacie meno a heslo ako do AIS.

Pri prvom prihlásení je nutné byť prihlásený v počítači, ktorý ja napojený na univerzitnú sieť STU. Nie je nutné byť fyzicky na počítači tejto siete, možno tiež využiť napr. vzdialený VPN prístup do univerzitnej siete. Pri ďalšom prihlásení sú už login a zašifrované heslo uložené v databáze a tak nie je nutné byť prihlásený v sieti STU.



The image shows a login form titled "Prihlásenie" (Login) for the APPOP system. The form is centered on a light blue background. It contains two input fields: "Meno" (Name) with the placeholder text "prihlasovacie meno" and "Heslo" (Password) with the placeholder text "heslo". A "Prihlásiť" (Login) button is located at the bottom right of the form.

Obr. 13: Formulár pre prihlasovanie do systému

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Rozhranie

Po prihlásení sa používateľovi zobrazí úvodná obrazovka systému APPOP. Náš systém používa jednotnú formu rozhrania vo všetkých moduloch. Toto zabezpečuje ľahkú orientáciu používateľov v systéme.

The screenshot shows the APPOP user interface. At the top left is the APPOP logo with 'Student' and a red box labeled '4'. The top navigation bar contains 'Testy', 'Zadania', and 'Materiály', with 'Testy' highlighted and a red box labeled '2'. On the top right, a dark blue box shows the user's name 'Bc. Vladimír Martinka' and options for 'Môj profil' and 'Odhlásiť', with a red box labeled '5'. The main content area is titled 'Zoznam otázok' (3) and shows 'Page 1 of 3, showing 10 records out of 27 total, starting on record 1, ending on 10'. Below this is a table with columns: 'Znenie otázky', 'Typ', 'Počet bodov', and 'Akcie'. The table contains six rows of questions related to C++ struct operations and pointer arithmetic. A left sidebar menu is also visible with various options like 'Zoznam testov', 'Nový test', 'Zoznam otázok', etc., with a red box labeled '2' next to the 'Testy' header.

Znenie otázky	Typ	Počet bodov	Akcie
Uvažujte operácie: void vloz(struct Zasobnik *z, struct Front *f); struct Zasobnik *vyber(struct Front *f); Je toto správna definícia operácií zásobníku, ktorého prvky sú fronty?	Klasická otázka	3	🔍 📄 ✖
Uvažujte operácie: void vloz(struct Front *f, struct Zasobnik *z); struct Front *vyber(struct Zasobnik *z); Je toto správna definícia operácií zásobníku, ktorého prvky sú fronty?	Klasická otázka	3	🔍 📄 ✖
Je možné implementovať spájaný zoznam pomocou jedného zásobníku?	Klasická otázka	3	🔍 📄 ✖
Je možné implementovať zásobník pomocou jedného spájaného zoznamu?	Klasická otázka	3	🔍 📄 ✖
Pre množinu hodnôt 7, 1, 3, 8, 4, a veľkosť tabuľky M=8, určite všetky perfektné (bez kolízií) hashovacie funkcie.	Klasická otázka	3	🔍 📄 ✖
Pre množinu hodnôt 5, 2, 7, 1, 6, a veľkosť tabuľky M=9, určite všetky perfektné (bez kolízií) hashovacie funkcie.	Klasická otázka	3	🔍 📄 ✖

Obr. 14: Prehľad používateľského rozhrania

1- Horná lišta

Na hornej lište sa nachádzajú kategórie funkcií, ktoré systém danému používateľovi ponúka. Kliknutím na jednu z kategórií sa v ľavej časti obrazovky zobrazí menu, korešpondujúce s vybranou kategóriou.

2- Ľavé menu

V ľavej časti obrazovky sa nachádza menu korešpondujúce s aktuálne vybranou kategóriou. Kliknutím na niektorú z položiek tohto menu sa zmení hlavná stredná sekcia obrazovky podľa vybratej položky. Aktuálna položka je zvýraznená bielym štvorčekom naľavo od jej popisu.

3- Hlavná sekcia

V tejto sekcii sa nachádza práve spracovávaný formulár. Môže sa jednať o zoznam prvkov, práve vypracovávaný test, formulár na pridávanie a editáciu komponentov a pod.

4- Logo systému

V ľavej vrchnej časti obrazovky sa nachádza logo systému, zároveň s kategóriou aktuálne prihláseného používateľa. Kliknutím na toto logo sa používateľ vráti úvodnú obrazovku.




5- Údaje aktuálne prihláseného používateľa

V pravej hornej časti obrazovky sa nachádza meno práve prihláseného používateľa, ako tiež možnosti prezrieť si svoj používateľský profil alebo odhlásiť sa zo systému.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Legenda

Počas práce so systémom APPOP narazí používateľ často na niektoré ikony, symbolizujúce akciu, ktorá sa ich kliknutím vykoná.

	Tlačidlo „Zobraz“. V hlavnej sekcii (3) sa zobrazia detaily položky v riadku zoznamu, v ktorom bolo na túto ikonu kliknuté.
	Tlačidlo „Uprav“. V hlavnej sekcii (3) sa zobrazí formulár pre úpravu položky v riadku zoznamu, v ktorom bolo na túto ikonu kliknuté
	Tlačidlo „Zmaž“. V prípade, že sa jedná o tlačidlo pri položke v zozname, používateľovi sa zjaví okno, v ktorom potvrdí, alebo odmietne zmazanie položky v riadku zoznamu, v ktorom bolo na túto ikonu kliknuté. Vo formulároch toto tlačidlo zmažáva prídavné atribúty (napr. prebytočné odpovede na otázku vo formulári na pridanie otázky)

Služby používateľom

V prototypy rozlišujeme tri typy používateľov:

Študent, Pedagóg, Admin

Typ používateľa, ktorý je práve prihlásený je možné vidieť pri logu v ľavej hornej časti obrazovky.

V prototypy majú zatiaľ všetci používatelia rovnaké práva.

Po prihlásení sa používateľovi zobrazí úvodná obrazovka.

Na hornej lište má používateľ na výber kategórie „Testy“, „Zadania“ a „Materiály“.

V ľavom menu je zobrazená uvítacia správa.

V prípade, že existujú nejaké nevypracované testy pre daného používateľa, zobrazí sa mu v hlavnej sekcii možnosť ich vypracovať.



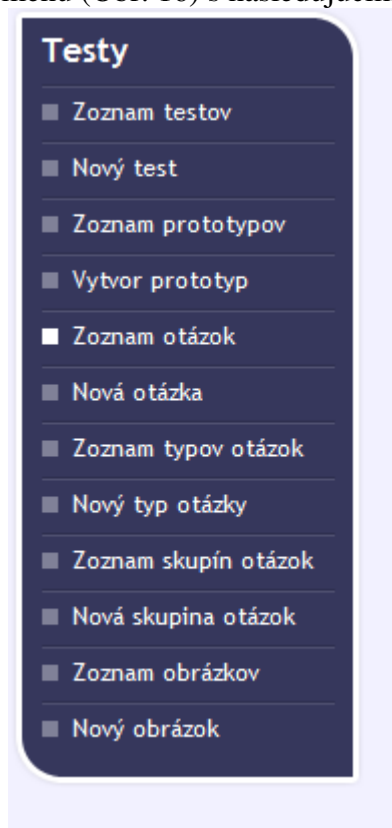
Obr. 15: Uvítacia obrazovka po prihlásení

Po kliknutí na niektorú z kategórií na hornej lište sa mu zobrazí ľavé menu korešpondujúce so zvolenou kategóriou:

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Testy

Kategória „Testy“ poskytuje ľavé menu (Obr. 16) s nasledujúcimi funkciami:



Obr. 16: Ľavé menu

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Zoznam testov

V hlavnej časti zobrazí zoznam všetkých testov v stránkach po desiatich. Tieto zobrazené testy je možné prezerat', upravovat' alebo mazat', pomocou tlačidiel v stĺpci „Actions“ . Kliknutím na ID nejakého testu je možné daný test skúšobne vypracovat'.

Id	Starttime	Endtime	Exam Prototype	User	Actions
6	2009-12-14 11:12:00	2009-12-14 11:12:00	malytestkat1	Vladimir	
3	2009-10-13 00:42:00	2010-12-13 00:42:00	vzor1	Vladimir	
4	2009-12-12 20:35:00	2009-12-12 20:35:00	vzor1	Vladimir	

Obr. 17: Tabuľka zoznamu testov

Nový test

Táto funkcia umožňuje pedagógovi vytvorit' inštanciu prototypu testu, a sprístupniť ju určitej skupine študentov.

Generovanie testov

Starttime
December 14 2009 11 42 am

Endtime
December 14 2009 11 42 am

Textový prototyp
(vyber prototyp)

Skupina
(vyber skupinu používateľov)

Uložiť

Obr. 18: Formulár generovania testov

Vo formulári možno zvoliť začiatočný a koncový čas, od ktorého, resp. do ktorého možno test vypracovat'. Taktiež sa vyberie na základe akého testového prototypu sa má test vygenerovat' a pre ktorú skupinu používateľov je určený.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Zoznam prototypov

Vypíše zoznam prototypov, ich *id* a mená. Prototypy tu možno prezerať, editovať a mazať kliknutím na príslušné ikony.

Id	Name	Akcie
63	vzor1	
65	sorty	
66	malytestkat1	

Obr. 19: Tabuľka zoznamu prototypov

Vytvor prototyp

Pomocou tohto formulára možno vytvoriť nový testový prototyp. Možno určiť jeho názov a počty otázok z jednotlivých kategórii. V prototypy možno zatiaľ vybrať určitý počet otázok danej kategórie. Pre pridanie ďalšieho kritéria, teda pre vytvorenie určitého počtu otázok iného typu, je potrebné zvoliť voľbu „Pridať kritéria“ v spodnej časti formulára. Táto funkcionlita nie je v prototypy implementovaná.

Nový Testový prototyp

Name

Kategória 1
(vyber kategóriu)

Počet otázok































Pridať kritéria

Obr. 20: Formulár pridávania testového prototypu

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Zoznam otázok

V hlavnej sekcii sa zobrazí zoznam všetkých otázok v databáze, použiteľných pri tvorbe testov. Tieto otázky sú zobrazené v zozname po desiatich a je možné si prezerať ich detaily, upravovať ich, prípadne mazať.

Zoznam otázok			
Page 1 of 3, showing 10 records out of 27 total, starting on record 1, ending on 10			
Znenie otázky	Typ	Počet bodov	Akcie
Uvažujte operácie: void vloz(struct Zasobnik *z, struct Front *f); struct Zasobnik *vyber(struct Front *f); Je toto správna definícia operácií zásobníku, ktorého prvky sú fronty?	Klasická otázka	3	  
Uvažujte operácie: void vloz(struct Front *f, struct Zasobnik *z); struct Front *vyber(struct Zasobnik *z); Je toto správna definícia operácií zásobníku, ktorého prvky sú fronty?	Viacnásobna	3	  
Je možné implementovať spájaný zoznam pomocou jedného zásobníku?	Textová otázka	3	  
Je možné implementovať zásobník pomocou jedného spájaného zoznamu?	Klasická otázka	3	  
Pre množinu hodnôt 7, 1, 3, 8, 4, a veľkosť tabuľky $M=8$, určite všetky perfektné (bez kolízií) hashovacie funkcie.	Klasická otázka	3	  
Pre množinu hodnôt 5, 2, 7, 1, 6, a veľkosť tabuľky $M=9$, určite všetky perfektné (bez kolízií) hashovacie funkcie.	Klasická otázka	3	  
Pre hashovaciu funkciu H a prirodzené čísla U a V platí, že ak $H(U)$ sa rovná $H(V)$, potom:	Klasická otázka	3	  
Pre hashovaciu funkciu H a prirodzené čísla U a V platí, že U sa rovná V , potom:	Klasická otázka	3	  
Uvažujte prioritný rad reprezentovaný polôm: {10, 6, ?, -4, 4, 6, 8, -8} s jedným neznámym prvkom (otáznik). Určte, ktorú z nasledujúcich hodnôt priority môže mať neznámy prvok:	Klasická otázka	3	  
Uvažujte prioritný rad reprezentovaný polôm: {10, 8, 6, ?, 4, 4, -4, -6, 4, -12} s jedným neznámym prvkom. Určte, ktorú z nasledujúcich hodnôt priority môže mať neznámy prvok:	Klasická otázka	3	  

<< predchádzajúca 1 | 2 | 3 ďalšia >>

Obr. 21: Tabuľka zoznamu otázok

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Nová otázka

Po zvolení tejto možnosti v ľavom menu sa zobrazí v hlavnej sekcii formulár pre pridávanie nových otázok do databázy. Formulár obsahuje nasledovné položky:

Znenie otázky

Tu sa zadá text otázky, ktorý sa zobrazí študentovi pri vypracúvaní testu, a podľa ktorého určí správnu odpoveď (odpovede).

Popis otázky

Toto je nepovinné pole, ktoré môže obsahovať poznámky k danej konkrétnej otázke (tieto sa nezobrazujú v teste).

Typ otázky

Tu sa určí, akého typu je daná otázka. Momentálne implementované sú: klasická otázka (jedna možná odpoveď), viacnásobná (viac možných odpovedí), textová otázka (testovaný študent píše priamo text).

Počet bodov za otázku

Do tohto poľa sa zadá bodové ohodnotenie danej otázky, toto pole musí byť vyplnené.

Skupiny otázok

Tu sa zaznačí skupina (alebo skupiny), do ktorej otázka patrí. Testy sú generované práve na základe týchto skupín.

Odpovede

Do tejto sekcie sa zadávajú možné odpovede na pridávanú otázku. Pre pridanie viacerých odpovedí sa používa tlačidlo „pridať odpoveď“, ktoré pridá pri každom stlačení riadok pre ďalšiu odpoveď. Pre vymazanie prebytočných odpovedí sa používa tlačidlo „Zmaž“, napravo od každej odpovede. Správne odpovede sa označujú zaškrtnutím checkboxu „Správna“ pri príslušnej odpovedi.

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Pridanie otázky

Znenie otázky

Popis

Typ otázky

Obrázok

Počet bodov

Skupiny otázok

Odpovede

Správna 

Pridať odpoveď

Obr. 22: Formulár pridávania otázok

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Zoznam typov otázok

Táto funkcia ľavého menu zobrazí v hlavnej sekcii všetky podporované typy otázok s ich popismi. Je možné ich prezerat', upravovat' a mazať.

Testy

- Zoznam testov
- Nový test
- Zoznam prototypov
- Vytvor prototyp
- Zoznam otázok
- Nová otázka
- **Zoznam typov otázok**
- Nový typ otázky
- Zoznam skupín otázok
- Nová skupina otázok
- Zoznam obrázkov
- Nový obrázok

Zoznam typov otázok

Page 1 of 1, showing 3 records out of 3 total, starting on record 1, ending on 3

Názov	Popis	Akcie
Klasická otázka	Šprávna je len jedna odpoveď	
Viacnásobna	Viac správnych odpovedí	
Textová otázka	Odpoveďou je text	

<< predchádzajúca ďalšia >>

Obr. 23: Tabuľka zoznamu typov otázok

Nový typ otázky

Táto funkcia nie je v prototype implementovaná.

Zoznam skupín otázok

Zobrazí kategórie otázok (učebné okruhy), na základe ktorých sa vytvárajú prototypy testov a neskôr generujú samotné testy. Kategórie je možné prezerat', upravovat' a mazať.

Testy

- Zoznam testov
- Nový test
- Zoznam prototypov
- Vytvor prototyp
- Zoznam otázok
- Nová otázka
- Zoznam typov otázok
- Nový typ otázky
- **Zoznam skupín otázok**
- Nová skupina otázok
- Zoznam obrázkov
- Nový obrázok

Zoznam skupín otázok

Page 1 of 1, showing 7 records out of 7 total, starting on record 1, ending on 7

Názov	Actions
Zásobník, front, spajanjý zoznam	
Binárne stromy	
Hashovanie	
Binárna halda, halda, prioritný front	
Vývážené stromy, AVL, 2-3 stromy	
Usporiadúvanie	
Grafové algoritmy	

<< predchádzajúca ďalšia >>

Obr. 24: Tabuľka zoznamu typov otázok

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Nová skupina otázok

Zobrazí v hlavnej sekcii formulár na pridanie novej skupiny otázok, ktorú je po vytvorení následne možné priradiť novým, alebo existujúcim otázkam.



Obr. 25: Formulár pridania skupín otázok

Zoznam obrázkov

Táto funkcia nie je v prototypu implementovaná.

Nový obrázok

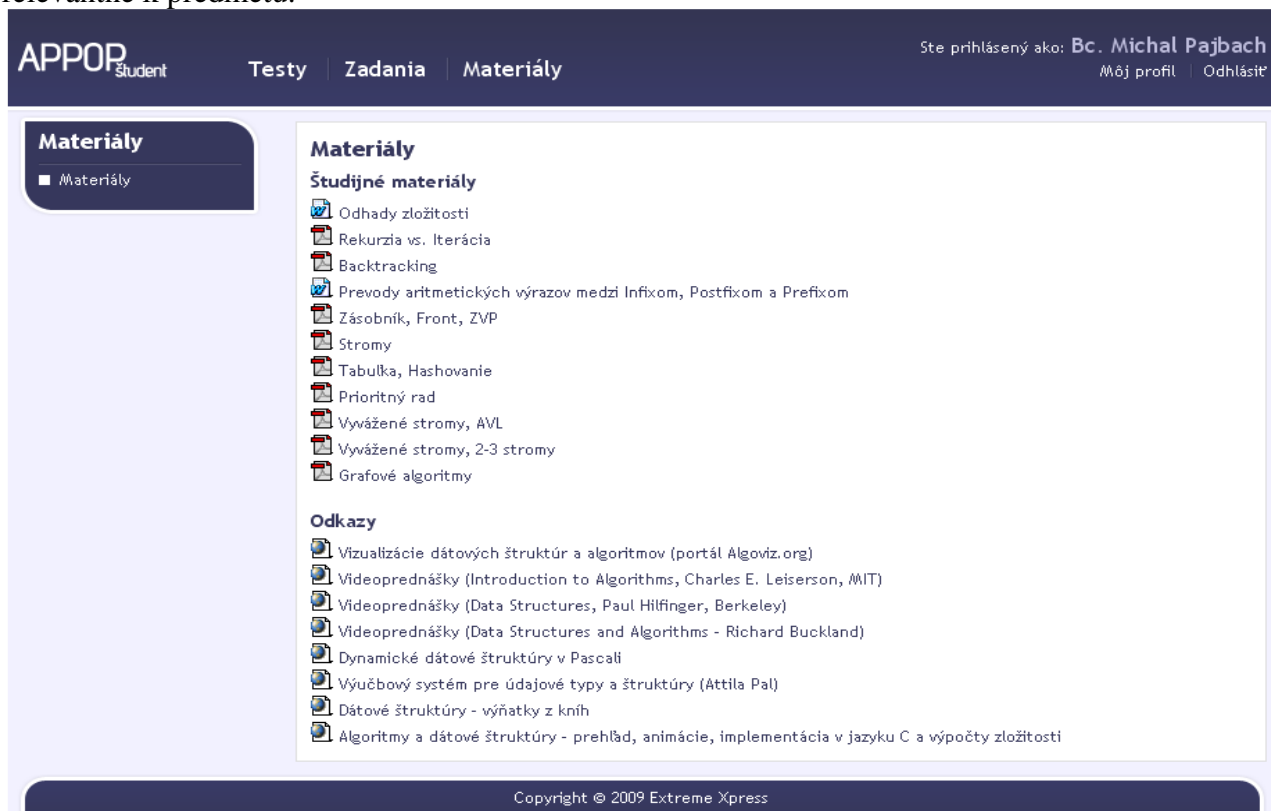
Táto funkcia nie je v prototypu implementovaná.

Zadania

Táto kategória v prototypu nie je implementovaná, a tak sa po jej zvolení nezobrazí ľavé menu.

Materiály

V kategórii „Materiály“ sa študentovi zobrazí v hlavnej časti obrazovky zoznam dokumentov k predmetu, ktoré je možné priamo stiahnuť, a tiež odkazy na stránky obsahujúce informácie relevantné k predmetu.



Obr. 26: Sekcia prehľadu materiálov

Error! Use the Home tab to apply Nadpis 1 to the text that you want to appear here.

Vypracúvanie testov

Aktuálne vypracúvaný test je zobrazený v hlavnej sekcii rozhrania. Používateľ na základe svojich vedomostí označí / vpíše pre každú otázku správnu odpoveď (odpovede) a po skončení daný test odošle tlačidlom „Odoslať“ v pravej dolnej časti testu. Po odoslaní sa používateľovi zobrazí opravený test, s označenými správnymi a nesprávnymi odpoveďami.

The screenshot shows the APPOR student interface. At the top, there is a navigation bar with 'APPOR študent', 'Testy', 'Zadania', and 'Materiály'. On the right, it says 'Ste prihlásený ako: Bc. Vladimír Martinka' with links for 'Môj profil' and 'Odhlásiť'. On the left, there is a sidebar with 'Akcie' and 'Testy'. The main content area is titled 'Test' and contains four questions:

- Otázka č. 1** Je možné implementovať spájaný zoznam pomocou jedného zásobníku? [3b]
- Otázka č. 2** Uvažujte operácie:
`void vlož(struct Front *f, struct Zasobnik *z);`
`struct Front *vyber(struct Zasobnik *z);`
Je toto správna definícia operácií zásobníku, ktorého prvky sú fronty? [3b]
 áno
 nie
- Otázka č. 3** Uvažujte operácie:
`void vlož(struct Zasobnik *z, struct Front *f);`
`struct Zasobnik *vyber(struct Front *f);`
Je toto správna definícia operácií zásobníku, ktorého prvky sú fronty? [3b]
 áno
 nie
- Otázka č. 4** V binárnej halde platí [3b]
 Ľavý podstrom obsahuje vždy prvky s nižšími hodnotami kľúčov ako pravý
 Ľavý podstrom môže, ale nemusí obsahovať prvky s nižšími hodnotami kľúčov ako pravý
 Ľavý podstrom nikdy neobsahuje prvky s nižšími hodnotami kľúčov ako pravý

Obr. 27: Formulár s vypracúvaným testom

The screenshot shows the graded version of the test. It includes feedback for the previous question and the current question:

nesprávne

Otázka č. 4 V binárnej halde platí [3b]
 Ľavý podstrom obsahuje vždy prvky s nižšími hodnotami kľúčov ako pravý
 Ľavý podstrom môže, ale nemusí obsahovať prvky s nižšími hodnotami kľúčov ako pravý
 Ľavý podstrom nikdy neobsahuje prvky s nižšími hodnotami kľúčov ako pravý

spravne

Otázka č. 5 Uvažujte prioritný rad reprezentovaný polom: {10, 8, 6, ?, 4, 4, -4, -6, 4, -12} s jedným neznámym prvkom. Určte, ktorú z nasledujúcich hodnôt priority môže mať neznámy prvok: [3b]
 -8
 0
 -4
 6
 9

nesprávne

Otázka č. 6 Pre hashovaciu funkciu H a prirodzené čísla U a V platí, že U sa rovná V, potom: [3b]
 Najväčší spoločný deliteľ U a V je väčší ako 1
 H(U) a H(V) sú rôzne
 H(U) a H(V) môžu byť rôzne
 Ani jedna z ostatných možností neplatí

spravne

Vyhodnotenie: 6 / 18

Obr. 28: Formulár s vyhodnoteným testom