

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

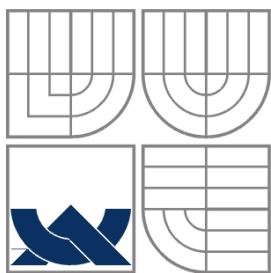
E-LEARNINGOVÝ KURZ PRE VÝUKU ALGORITMOV

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

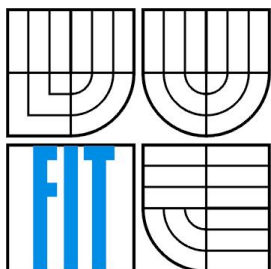
AUTOR PRÁCE
AUTHOR

Bc. BORIS HROZÁNY

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

E-LEARNINGOVÝ KURZ PRO VÝUKU ALGORITMŮ

E-LEARNING COURSE FOR ALGORITHMS TEACHING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. BORIS HROZÁNY

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ŠÁRKA KVĚTOŇOVÁ

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2006/2007

Zadání diplomové práce

Řešitel: **Hrozány Boris, Bc.**

Obor: Informační systémy

Téma: **E-learningový kurz pro výuku Algoritmů**

Kategorie: Softwarové inženýrství

Pokyny:

1. Seznamte se s požadavky kladenými na webovou aplikaci, která umožní on-line přístup k výukovým materiálům zvoleného předmětu.
2. Seznamte se s databázovým systémem MySQL a skriptovacími jazyky HTML, PHP, JavaScript a s možnostmi jejich využití při implementaci systému.
3. Podrobně analyzujte požadavky na e-learningový systém specializovaný na výuku Algoritmů pro zahraniční studenty v angličtině. Součástí by měla být také databáze testových otázek a vhodný systém jejich hodnocení.
4. Systém realizujte a proveďte odpovídající testy funkčnosti a ovladatelnosti systému.
5. Zhodnoťte dosažené výsledky a diskutujte další možný rozvoj systému.

Literatura:

- Welling, L., Thomsonová, L.: PHP a MySQL - rozvoj webových aplikací. Softpress, 2003, 720 s. ISBN 8086497607
- Kosek, J.: HTML, tvorba dokonalých www stránek. Praha: Grada Publishing, 1998, 291 s. ISBN 80-7169-608-0
- DeLisle, M.: PHPMyAdmin - efektivní správa MySQL. Brno: Zoner Press, 270 s. ISBN 8086815099
- W. Horton, K. Horton, Learning Tools and Technologies. Wiley 2003. ISBN: 0-471-44458-8
- Bates, A. W., Poole, G.: Effective Teaching with Technology in Higher Education: Foundations for Success. Wiley, 2003. ISBN: 0-7879-6034-9

Při obhajobě semestrální části diplomového projektu je požadováno:

- Body 1-3.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Květoňová Šárka, Ing., UIFS FIT VUT**

Datum zadání: 28. února 2006

Datum odevzdání: 22. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 86 Brno, Božetechova 2



doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Bc. Boris Hrozány**
Id studenta: 49597
Bytem: Horné Orešany 139, 919 03 Horné Orešany
Narozen: 06. 08. 1984, Trnava
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
diplomová práce

Název VŠKP: E-learningový kurz pro výuku Algoritmů
Vedoucí/školitel VŠKP: Květoňová Šárka, Ing.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....

Autor

Abstrakt

Cieľom práce je návrh a realizácia E-learningového systému. Tento systém bude zameraný na výuku kurzu Algoritmy v angličtine. Vďaka tejto skutočnosti sa budú môcť zúčastniť kurzu aj zahraniční študenti. V úvode sú zhrnuté základné koncepty E-learningových systémov: konceptuálny aj detailný návrh E-learningového systému, vývoj systému, návrh testovania systému a údržby a vylepšenie systému. Teoretická časť projektu sa venuje vybraným aspektom z oblasti dátových štruktúr, vyhľadávacích a radiacich algoritmov. Práca sa zaoberá vývojom E-learningového systému od úplného začiatku až po činnosti spojené s jeho dodaním. Aplikácia umožňuje správu kurzov a užívateľov v podobe rozhrania administrátora. Ďalšia časť systému slúži na kontrolu výsledkov študentov. Študentská časť aplikácie umožňuje štúdium kurzu Algoritmy a testovanie vedomostí z tejto oblasti. Výsledky testov sú v systéme spracovávané do štatistík.

Kľúčové slová

E-learning, E-learningový systém, online vzdelávanie, algoritmus, abstraktná dátová štruktúra, štúdium, test, kurz

Abstract

The aim of this work is to design and implement E-learning system. This system will be targeted on the Algorithms course education in English. Thanks to this fact, foreign students will be able to attend the course, too. The introduction part offers an overview of basic conceptions related to E-learning systems: conceptual and detailed design of E-learning system, system development, designing system testing and maintenance and system improving. Theoretical part of this project is focused on selected topics from data structures, searching and sorting algorithms. The work is dealing with the development of E-learning system from the entire beginning to activities related with its release. Through the interface of administrator the application enables administration of courses and users. The other part of the system serves for checking of students' results. Student part of the application makes the Algorithms course materials available and provides convenient means of knowledge testing. The results of tests are processed in a form of statistics within the system.

Keywords

E-learning, E-learning system, online education, algorithm, abstract data structure, study, test, course

Citácia

Boris Hrozány: E-learningový kurz pre výuku Algoritmov, diplomová práca, Brno, FIT VUT v Brně, 2007

E-learningový kurz pre výuku Algoritmov

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením Ing. Šárky Květoňovej.

Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Boris Hrozány
10.5.2007

Pod'akovanie

Chcel by som sa poďakovať Ing. Šárke Květoňovej za poskytnutie odbornej pomoci a usmernenie pri tvorbe diplomovej práce.

© Boris Hrozány, 2007.

Táto práca vznikla ako školské dielo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práca je chránená autorským zákonom a jej použitie bez udelenia oprávnenia autorom je nezákonné, s výnimkou zákonom definovaných prípadov.

Obsah

Obsah	1
1 Úvod	3
2 E-learning	5
2.1 Čo je E-learning	5
2.2 Rôzne druhy E-learningu	6
2.2.1 Online vzdelávanie	6
2.2.2 Online školenie	7
2.2.3 Spravovanie vedomostí	7
2.2.4 Elektronická podpora prevedenia	8
2.3 Návrh E-learningového systému	8
2.3.1 Výber správneho formátu	9
2.3.2 Konceptuálny návrh	10
2.3.3 Detailný návrh	13
2.3.4 Zaistenie kvality systému	15
2.4 Vývoj a údržba E-learningového systému	16
2.4.1 Efektívna online komunikácia	16
2.4.2 Technické prehliadky a testovanie	19
2.4.3 Podpora a údržba E-learningového systému	20
3 Existujúce E-learningové systémy	22
3.1 Moodle	22
3.1.1 Teoretické východiská Moodlu	22
3.1.2 Vlastnosti Moodlu	23
3.2 Mindflash	24
4 Prehľad kurzu Algoritmy	25
4.1 Abstraktné dátové štruktúry	25
4.1.1 Zoznam	25
4.1.2 Zásobník	27
4.1.3 Fronta	27
4.1.4 Strom	28
4.2 Vyhľadávacie algoritmy	29
4.2.1 Sekvenčné vyhľadávanie	29
4.2.2 Nesequenčné vyhľadávanie v zoradenom poli	30
4.2.3 Binárne vyhľadávacie stromy	30
4.2.4 Tabuľky s rozptýlenými položkami	31

4.3	Radiace algoritmy	31
4.3.1	Radenie na princípe výberu	32
4.3.2	Radenie na princípe vkladania	32
4.3.3	Radenie na princípe rozdelenia	33
4.3.4	Radenie na princípe zlučovania.....	33
5	Návrh E-learningového systému	34
5.1	Výber formátu.....	34
5.2	Konceptuálny návrh.....	35
5.2.1	Návrh systému.....	35
5.2.2	Návrh kurzu.....	38
5.3	Návrh databázy	39
5.3.1	ER diagram.....	39
5.3.2	Popis tabuliek	41
6	Implementácia	44
6.1	Použité programovacie technológie.....	44
6.1.1	PHP	44
6.1.2	MySQL.....	45
6.1.3	JavaScript	45
6.1.4	Eclipse	46
6.2	Implementácia systému	47
6.2.1	System	48
6.2.2	Administrátor	49
6.2.3	Supervisor.....	50
6.2.4	Študent.....	51
7	Záver.....	57
7.1	Zhodnotenie výsledkov.....	57
7.2	Ďalšie možnosti vývoja	57
	Literatúra.....	59
	Zoznam príloh.....	60
	Príloha 1 Ukážka E-learningového systému Moodle.....	61

1 Úvod

E-learning je pomerne nová forma vzdelávania, ktorej popularita stále rastie. Na rozdiel od klasického vzdelávania študent už nemusí prezenčne navštevovať kurzy, všetky informácie sú mu dostupné pomocou počítača. E-learningové kurzy sú prispôsobené tak, aby nadobudnutie nových vedomostí a informácií bolo efektívne a rýchle. V závislosti na druhu informácii, ktoré má E-learningový kurz naučiť, existuje veľa foriem E-learningu. Niektoré sú zamerané na systematické vzdelávanie podobné klasickému vzdelávaniu, iné na čo najrýchlejšie získanie požadovaných informácií pre okamžité použitie.

V súčasnosti existuje veľa E-learningových systémov, od malých, určených na vzdelávanie zamestnancov vo firmách, až po celosvetové systémy dostupné komukoľvek cez internet. Takisto existuje mnoho systémov, ktoré umožňujú rýchle a efektívne vytváranie vlastných kurzov.

Cieľom mojej práce je vytvorenie E-learningového systému, ktorý bude umožňovať správu kurzov a užívateľov a štúdium kurzov. Súčasťou práce je tiež navrhnutie a vytvorenie kurzu Algoritmy. Kurz bude celý vytvorený v anglickom jazyku. Vďaka tomu bude môcť byť použitý univerzálne pre vzdelávanie všetkých študentov, aj zahraničných.

Druhá kapitola bude venovaná E-learningu. Najprv v nej popíšem, čo je E-learning a aké rôzne druhy E-learningu existujú. Každý druh bude charakterizovaný z hľadiska jeho využitia, na čo je vhodný a ako ho správne použiť. V ďalšej časti popíšem návrh E-learningového systému od voľby správneho formátu systému na základe identifikácie požiadaviek zákazníka, cez návrh kostry systému, ktorá určuje hlavné vlastnosti a správanie systému, až po detailný návrh dotvárajúci systém do najmenších podrobností tak, aby splnil všetky očakávania zákazníka a budúcich študentov. Jednotlivé časti návrhu budú podrobne popísané a vysvetlené. Posledná časť kapitoly sa týka samotného vývoja navrhnutého systému. Budú v nej popísané postupy tvorby systému, čoho sa pridržať a čomu sa pri jeho tvorbe vyhnúť. Súčasťou tejto časti je popis testovania implementovaného systému a rady týkajúce sa údržby systému a prípadných vylepšení systému.

Tretia kapitola priblíži existujúce E-learningové systémy. Z voľne šíriteľných systémov bude popísaný Moodle, ktorý patrí medzi najviac rozšírené systémy a zástupcom komerčných systémov bude Mindflash. Budú popísané základné vlastnosti týchto systémov a spôsob, akým umožňujú užívateľom vytvárať vlastné E-learningové kurzy.

Ďalšia kapitola je venovaná stručnému popisu obsahu E-learningového kurzu Algoritmy, ktorý budem implementovať. Budú popísané najčastejšie používané abstraktné dátové štruktúry a ich využitie. Ďalšia časť tejto kapitoly je venovaná popisu niektorých vyhľadávacích algoritmov a ich stručnému zhodnoteniu. Posledná časť sa venuje vybraným radiacim algoritmom. Stručne popisuje princíp ich fungovania a ich vlastnosti.

Piata kapitola priblíži analýzu a návrh E-learningového systému. Bude popísaný výber formátu E-learningového systému a konceptuálny návrh systému. Budú popísané a v diagramoch prípadov použitia znázornené všetky užívateľské prístupy systému. Následne bude popísaný návrh kurzy Algoritmy, jeho vzhľad a spôsob reprezentácie informácií. Na záver bude pre systém navrhnutá databáza. Budú popísané všetky tabuľky a význam hodnôt uložených v nich.

Posledná kapitola bude venovaná implementácii navrhnutého systému. Najprv budú popísané technológie použité pri vývoji systému a vývojové prostredie. Následne bude popísaný spôsob implementácie časti systému spoločnej pre všetky druhy užívateľov. V ďalšej časti budú podrobnejšie popísané funkcie jednotlivých užívateľov systému. Na záver kapitoly bude popísané samotné vzdelávanie, spôsob sprístupňovania informácií študentovi a testovanie študentov.

Diplomová práca navesuje na semestrálny projekt, v ktorom som naštudoval a spracoval problematiku E-learningu a v vytvárania E-learningových kurzov.

2 E-learning

E-learning je pomerne novou formou vzdelávania, o ktorej účelnosti a budúcnosti sa vedú mnohé dohady. Niektorí odborníci si myslia, že je účelnejšia ako klasické vzdelávanie v triede a niektorí si dokonca myslia, že E-learning úplne nahradí klasické vzdelávanie. Štúdiá ukazujú, že E-learning je práve tak účelný ako klasické vzdelávanie v triede, ale učí viac efektívne. Doba učenia sa v priemere zredukuje na jednu tretinu. A ako ukazuje vývoj E-learningu, E-learning funguje najlepšie ak je prekladaný s klasickým vyučovaním.

E-learning má veľa spoločného s klasickým vzdelávaním, ale je tu niekoľko rozdielov, ktoré sa týkajú pripravovania materiálov na výuku. Učitelia zvyknutí navrhovať materiály pre knižnú podobu alebo výuku v triede musia vytvoriť nové postupy, vhodné pre vzdelávanie pomocou počítača. V niektorých oblastiach sa počítač zdá byť príliš obmedzujúci, ale na druhú stranu otvára celú škálu nových možností učenia.

2.1 Čo je E-learning

V 90-tych rokoch dvadsiateho storočia sa rapídne zvyšuje používanie počítačov v domácnostiach a v práci, čo otvára celú škálu nových možností vzdelávania. Trendom sa stáva vzdelávanie za podpory počítačov (CBT – Computer Based Learning). K vzdelávaniu pomocou počítača prispieva používanie CD-ROM, ktoré sa stáva vhodnou možnosťou prenosu takeého množstva dát, akým sú multimedialne encyklopédie. O niekoľko rokov neskôr vznikajú prvé aplikácie priamo určené a prispôbené na efektívne vzdelávanie ľudí.

E-learning je ešte stále v prudkom vývoji a je veľmi ťažké ho presne definovať. Predstavy o E-learningu sa u jednotlivých organizácií líšia. Každá považuje za E-learning niečo iné. Jedna z častých a jednoduchších definícií E-learningu hovorí, že E-learning je vzdelávanie za podpory počítačov. Iné organizácie definujú E-learning ako poskytovanie informácií, školiacich materiálov a vzdelávacieho obsahu pomocou rôznych foriem elektronických médií. Ďalšia definícia hovorí, že E-learning je vzdelávací proces, využívajúci informačné a komunikačné technológie [1].

Tak ako klasické vzdelávanie je rozdelené na kurzy, tak aj materiály pre E-learning učitelia zvyčajne rozdeľujú do kurzov. Je to dané už tým, že samotné systémy sú navrhnuté na používanie kurzov. Má to viac významov a nie všetky sú dôležité pre samotné vzdelávanie. Kurzy slúžia napríklad ako jednotka služby, za ktorú si študenti platia. Kurz je ďalej jednotkou pre meranie efektívnosti výuky, väčšina vzdelávacích organizácií odhaduje efektívnosť ich práce podľa odhadu efektívnosti jednotlivých kurzov.

Výuka v triedach predstavuje formálny typ učenia, lebo má presne stanovené ciele a má prinášať vopred stanovené výsledky. Niektoré formy E-learningu spadajú do tejto kategórie, ale

d'alsie formy idu d'alej a poskytujú vzdelávanie vtedy a na takom mieste, ako to vyhovuje študentovi. Takéto vzdelávanie, nemá stanovené ciele. Ciele si stanovuje každý študent sám, podľa svojich vlastných potrieb. Takéto formy vzdelávania sa nazývajú neformálne typy.

2.2 Rôzne druhy E-learningu

E-learning sa rozdeľuje na štyri hlavné druhy. Online vzdelávanie a online školenie reprezentujú formálne vzdelávanie a spravovanie vedomostí a elektronická podpora prevedenia reprezentujú neformálne vzdelávanie.

Formálne vzdelávanie je zámerové. Vzdelávacie materiály začínajú špecifikovaním cieľov, stanovených tvorcom kurzu. Efektívnosť učenia je odhadnutá na základe toho, do akej miery študenti splnili stanovené ciele. Tento typ vzdelávania stavia vyučujúceho do pozície človeka, ktorý riadi čo sa študenti budú učiť a kontroluje ich správanie, do akej miery plnia plán štúdia.

Neformálne vzdelávanie nie je riadené žiadnym inštruktorom ani počítačom. Študenti sa sami rozhodujú, čo sa potrebujú naučiť, hľadajú si materiály a študujú ich až pokiaľ nezvládnu danú problematiku. Takéto vzdelávanie nie je riadené zámerom, nezačína stanovenými cieľmi, záleží len na podnete študenta a jeho efektívnosť sa líši podľa potrieb študenta.

2.2.1 Online vzdelávanie

Online vzdelávanie je veľmi podobné klasickému vzdelávaniu v triede, s jedinou zmenou, keď rolu učiteľa plní počítač. Delí sa na synchronne a asynchronne.

2.2.1.1 Synchronne online vzdelávanie

Je podobné vzdelávaniu v triede, len študenti a inštruktor sú od seba geograficky vzdialení, ale pracujú v tom istom čase, od čoho je aj odvodený názov synchronne vzdelávanie. Teda inštruktor je v priamom kontakte so študentmi.

Kontakt môže byť uskutočnený pomocou chatu, obsah ktorého vidia všetci pripojení k danému kurzu. Chat sa zvyčajne skladá z dvoch častí. V prevej časti inštruktor vysvetľuje látku a nasleduje konverzácia so študentmi o prebratej látke.

Najčastejšou formou je broadcast inštruktorovho prednesu, ktorý vidia všetci účastníci. Študenti môžu klásť inštruktorovi dotazy ústnou cestou, alebo písaním otázok.

2.2.1.2 Asynchronne online vzdelávanie

Je forma vzdelávania, pri ktorej sú inštruktor a študent od seba oddelení geograficky aj časom, nedochádza teda k priamemu kontaktu inštruktora so študentom. Kvôli tomu, že komunikácia neprebíha priamo v jednom čase, nazýva sa táto forma učenia asynchronna.

Materiály na vzdelávanie sú rozdelené do kurzov. Typicky sa skladajú z vysvetľovacej časti, ktorá má študenta naučiť danú látku, a z cvičení, v ktorých si študent overí nadobudnuté vedomosti.

Študenti sú v kontakte s počítačom, ktorý ich smeruje kurzami. Sami si ovplyvňujú rýchlosť vzdelávania, tak aby im vyhovovala. Kurzy zvyčajne obsahujú otázky alebo testy, na ktorých si študent môže overiť dosiahnuté vedomosti.

Spoločným znakom oboch foriem vzdelávania je, že kurzy sú dopĺňané učebnicami, materiálmi na čítanie, cvičeniami a zaujímavými odkazmi. Študenti sa venujú tejto forme vzdelávania podobným spôsobom ako pri klasickom učení v triede, teda niekoľko hodín týždenne počas niekoľkých týždňov alebo mesiacov. Medzi jednotlivými kurzami dostávajú študenti domáce úlohy alebo materiály na prečítanie.

Keďže existuje veľa rôznych druhov študentov od začiatočníkov, cez pokročilých až po expertov, je aj online vzdelávanie prispôbené týmto kategóriám. Začiatočnicke kurzy nechávajú študentom priestor a kľud na rozvoj svojich schopností. Oproti tomu, univerzity poskytujú kurzy na vysokej úrovni, ktorých výsledkom certifikát online vzdelávania.

2.2.2 Online školenie

Podobne ako pri online vzdelávaní je zámerom rozvoj vedomostí a schopností študenta. Rozdiel je však v tom, že u online školenia sa očakáva okamžité použitie nadobudnutých vedomostí. Online školenie sa takisto delí na synchrónne a asynchrónne.

Hlavným rozdielom vo výuke oproti online vzdelávaniu je, že študenti berú kurzy online školenia s minimálnymi prerušeniami a len zriedkavo dostávajú domáce úlohy. Napríklad študent môže prejsť online tréningovým kurzom v priebehu dvoch hodín. Častejšou formou online školenia je asynchrónne školenie, pretože študent môže potrebovať vedomosti kedykoľvek a tento spôsob mu to umožňuje.

Kurzy zvyčajne ponúkajú aj diskusné skupiny, kde majú študenti možnosť medzi sebou diskutovať o problematike. Dáva to študentom pocit štúdia v kolektíve. Obdobne niektoré organizácie ponúkajú možnosť kontaktovať experta v danom obore, ktorý študentom zodpovie prípadné dotazy.

Online školenia sú považované za formálne, pretože kurzy majú explicitne stanovené ciele a často obsahujú testy, aby si študenti mohli overiť, či zvládli daný obsah. Online školenia sú obzvlášť vhodné pre učenie ako pracovať s aplikáciami alebo školenia o dodržiavaní bezpečnosti pri ochrane informácií.

2.2.3 Spravovanie vedomostí

Spravovanie vedomostí je zbieranie, odkladanie, transformovanie a šírenie informácií v organizácii s cieľom zvýšenia efektívnosti a inovácie. Cieľom je teda poskytnúť prístup k informáciám ako napríklad k firemnej politike, k procedúram alebo k neformálnym dokumentom.

Informácie sa ukladajú do databáz nazývaných vedomostné základne. Študenti sú pri učení v priamom kontakte s týmito databázami. Prístup k vedomostným základniam je možný kedykoľvek, podľa potrieb ľudí. Študovanie informácií je možné prirovnať k hľadaniu informácií v encyklopédii. Spravovanie vedomostí tiež zahŕňa online diskusie a chaty, kde si účastníci vymieňajú vedomosti.

Spravovanie vedomostí vo veľkom využívajú poradenské firmy. Klienti, ktorí žiadajú tieto firmy o pomoc majú často podobné požiadavky, alebo aspoň ich časti. Firma potom na základe skúseností s predchádzajúcimi zákazníkmi upravuje vedomostnú základňu. Znovu používa osvedčené zdroje informácií a nahrádza tie, ktoré sa neosvedčili. Zabráňuje tak opakovaniu chýb a skvalitňuje poskytované služby.

2.2.4 Elektronická podpora prevedenia

Poskytuje užívateľom pracovného prostredia pomoc vo forme informácií, školenia, navádzania a sledovania potrebného pre správne používanie daného prostredia. Niekedy dokonca podpora môže riešiť úlohy namiesto zamestnancov. Ide čiastočne o online pomoc, o online návod, databázu, aplikačný program a expertný systém. Môže byť realizovaná ako samostatný program alebo ako časť existujúceho programu.

Cieľom učenia pomocou podpory prevedenia je samotné prevedenie. Ak systém prevedie nejakú úlohu namiesto užívateľa, užívateľ sa to naučí a nabudúce bude úlohu schopný splniť sám. Neučí ho prečo a čo spraviť, ale učí ho ako to spraviť. Preto je učenie náhodné a teda môže ale aj nemusí vyvinúť špecifické schopnosti. Keďže užívateľ nie je v kontakte s inštruktorom, keď prebieha podpora prevedenia, spadá tento druh učenia do kategórie asynchrónneho učenia.

Elektronická podpora prevedenia nachádza využitie v mnohých oblastiach života. Napríklad ak doktor chce predpísať lieky pacientovi, musí si overiť, či by nereagovali s liekmi, ktoré už pacient berie. To by trvalo príliš dlho a preto namiesto toho do systému obsahujúceho históriu liekov pacienta zadá liek, ktorý chce predpísať a systém ho upozorní na prípadné nežiadane reakcie s inými liekmi.

2.3 Návrh E-learningového systému

Je najdôležitejšou časťou vývoja E-learningového systému. V tejto fáze sa na základe analýzy požiadaviek vzdelávacej organizácie alebo firmy musí rozhodnúť, aká forma E-learningu je najviac vyhovujúca. Systém sa musí premyslieť a navrhnuť od najvšeobecnejších vlastností až po najmenšie detaily, tak aby bol vhodný pre výuku zadaných informácií.

2.3.1 Výber správneho formátu

Najjednoduchšou a najrýchlejšou možnosťou je vývoj tradičného návodu. Ide iba o transformáciu učebníc do online podoby. Táto cesta je však jednou z najmenej efektívnych z hľadiska vzdelávania.

Pre správny výber formátu je najlepšie zamyslieť sa nad niekoľko veľmi dôležitými bodmi:

1. použitie obsahu E-learningu: ak obsah učenia je jeden celok, ktorého štúdium je dlhodobejšie, je vhodné požiť systém, ktorý bude študenta navigovať od začiatku až po koniec. A naopak, ak sa obsah učenia skladá z viacerých uzavretých celkov, použije sa systém, ktorý umožní študentovi rýchlo nájsť tú časť materiálov, ktorú sa potrebuje naučiť
2. formálne alebo neformálne učenie: formálne učenie poskytuje lepšie možnosti v určovaní smeru štúdia a kontroly dosiahnutých výsledkov. Študent je navigovaný materiálmi a priebežne testovaný, čo sa naučil. Neformálne učenie je vhodné na rýchle vyhľadanie potrebných vedomostí, efektívne ich odprezentuje a dá študentovi možnosť okamžite ich využiť. Ciele si stanoví každý študent sám a záleží iba na ňom, či si overí dosiahnuté vedomosti.
3. predchádzajúce vedomosti študentov: v závislosti na vedomostiach študentov je možné rozdeliť systém na formálnu časť, ktorá bude zvyšovať už nadobudnuté vedomosti a neformálnu časť, ktorá umožní začiatočníkom rýchlejší postup a dobehnutie skúsených študentov.
4. skúsenosti vývojárov: z hľadiska efektívneho využitia prostriedkov je vhodné zväžiť skúsenosti s vývojom jednotlivých druhov E-learningu. Vývoj úplne nového systému je vždy nákladnejší ako vývoj systému, s ktorým vývojový tím už má skúsenosti.
5. technologická platforma: v niektorých prípadoch je výber možných formátov E-learningu obmedzený technologickou platformou organizácie vyvíjajúcej systém, alebo študentov. Napríklad organizácia nevlastní nástroj na vývoj animácií, alebo študent nemá prístup k programu, ktorý by mu animácie prehrával.

Aby bol E-learningový systém skutočne dobrý, nestačí zvoliť len správny formát učenia. Úspešnosť E-learningového systému z veľkej časti závisí aj na splnení očakávaní študentov. Od online výuky študenti očakávajú, že bude mať vopred stanovené ciele a bude navigovať študenta tak, aby dané ciele boli splnené. A naopak od online návodu študenti očakávajú, že ich bude priamo navigovať k bodu ich záujmu a neočakávajú, že sa ich systém bude pokúšať niečo systematicky naučiť. Očakávania študentov sa dajú rozdeliť do niekoľkých základných kategórií:

1. cesta akou sa dostanú k žiadaným informáciám: v manuáloch študent očakáva priamy odkaz na hľadanú informáciu a od výuky očakáva sprevádzanie informáciami
2. typ dostupných informácií: od online výuky sa očakáva, že bude obsahovať aj príklady a cvičenia a nápoveda by mala obsahovať len čisté inštrukcie
3. formát informácií: nápoveda krok za krokom popisuje funkcie systému a katalóg obsahuje zoradenie a špecifikáciu produktov.

4. štýl textu: manuál je písaný jednoduchým písmom a od online demonštrácií sa očakáva, že budú farebnejšie a príjemnejšie na pohľad
5. vzhľad plochy: výukové materiály sú prispôsobené tak, aby sa zmestili na jednu obrazovku aby užívateľ nemusel scrollovať. Na druhú stranu články webových magazínov môžu rozsahom presahovať jednu obrazovku, lebo sa očakáva, že študent článok aj tak dočíta do konca.
6. organizácia materiálov: od online výuky sa očakáva, že začne stanovením cieľov, za ktorými budú nasledovať študijné materiály, cvičenia a zhrnutia. Na druhú stranu referencie sú zoradené v abecednom poradí.

Výber vhodného formátu a spôsobu akým bude obsah vzdelávania podaný je teda dôležitým faktorom, určujúcim kvalitu výsledného systému.

2.3.2 Konceptuálny návrh

Vývoj konceptuálneho návrhu je najkomplexnejšou časťou celého návrhu. Je to návrh všeobecnej stratégie prezentovania materiálov vo vzdelávacom programe. Zahŕňa štrukturovanie obsahu, jeho pripravovanie a stratégiu prezentovania informácií.

2.3.2.1 Štrukturovanie obsahu

Štrukturovanie obsahu sa skladá z troch základných častí. Prvou je zoradenie informácií, podľa toho, ktorá informácia sa zobrazí ako prvá, ktorá po nej a ktorá nakoniec. Ďalšou časťou je rozdelenie obsahu na sekcie, prípadne podsekcie. Poslednou časťou je zváženie elementov, ktoré sú časťou každého E-learningového programu, ale nespádajú do obsahu materiálov. Sem patria napríklad titulné strany, menu a slovníky.

Každá publikovaná práca, či už online, alebo v knižnej podobe má určitú povinnú štruktúru. Začína úvodnými elementmi a končí záverečnými elementmi. Medzi úvodné elementy E-learningového programu patrí titulná strana, hlavné menu, príhovor a inštrukcie. Záverečnými elementmi sú index, slovník, prípadná skúška.

Rovnako aj každá lekcia by mala obsahovať určité úvodné a záverečné elementy. Úvod každej lekcie by mal obsahovať úvodnú obrazovku, na ktorej sa bude nachádzať názov lekcie, ciele, ktoré chceme dosiahnuť, potrebné prerekvizície a čas, ktorý lekcia zaberie. Chýbať by nemalo ani vymenovanie tém, ktorých sa lekcia týka. Záverečnými elementmi lekcie sú zhrnutie popisujúce prebraté témy a najdôležitejšie body, prípadné preskúšania prebratej látky vo forme testu alebo otázok a zoznam dodatočných zdrojov, ktoré umožnia študentom nájsť ďalšie informácie o prebratých témach.

Prvým krokom pri vývoji všeobecnej štruktúry obsahu je vytvorenie schémy pre kurzy a lekcie. Veľká časť tejto práce je v tomto štádiu už spravená vďaka rozdeleniu obsahu na sekcie a podsekcie a je stanovená logická následnosť jednotlivých sekcií. Študenti však často ignorujú takéto roztriedenie a chcú priamo nájsť informácie, ktoré ich zaujímajú. Preto je treba prezentovať štruktúru obsahu tak,

aby sa v nej dalo ľahko orientovať. To vo väčšine prípadov neznamená prepracovanie cieľov obsahu, ale len zviditeľnenie štruktúry obsahu aj pre študentov. Existuje päť základných schém pre štruktúrovanie obsahu [2]. Sú to:

1. kategórie – pop, rock, klasika
2. čas – reálne časové usporiadanie, alebo logické časové usporiadanie napríklad v procedúre
3. lokácia – v závislosti na mieste alebo druhu veci
4. abeceda – slovník, encyklopédia
5. následnosť – od najhoršieho k najlepšiemu, od najmenej po najviac

Pri rozdeľovaní obsahu na časti, treba brať v úvahu aj to, koľko času majú študenti na učenie. Výskumy ukázali, že študenti venujú online učeniu v približne 5 až 30 minút. Toto rozmedzie sa samozrejme líši podľa individuálnych vlastností študentov. Je ale nereálne očakávať, že študenti budú ochotní venovať online štúdiu toľko času ako klasickému štúdiu v triede. Z veľkej miery to je ovplyvnené aj tým, že pri práci za počítačom robia viac vecí naraz a nevenujú sa len štúdiu. Rozptyľuje ich príliš veľa činností ako sú napríklad surfovanie po webe alebo čítanie nových prijatých emailov. Doba strávená štúdiom je teda iba časť z celkovej doby strávenej za počítačom.

Výukové časti, by teda mali mať obmedzený obsah, aby bolo možné nastudovať ich v priebehu najviac pol hodiny. To ale neznamená, že je potreba zredukovať množstvo preberanej látky. Stačí len rozdeliť E-learningový program na krátke časti, segmenty, z ktorých každý bude obsahovať jeden z hlavných cieľov, ktorý treba splniť. V prípade, že je obsah E-learningového programu príliš komplexný, môže byť splnenie hlavného cieľa rozdelené do viac segmentov, každý, však musí obsahovať splnenie aspoň jedného podcieľa. V prípade, že segmenty na seba úzko navesujú, je možné navrhnuť program tak, aby študenti preberali jednotlivé časti v sekvencii. Ak však jednotlivé segmenty na seba nenavesujú, je lepšie navrhnuť program tak, aby si študenti mohli sami zvoliť poradie preberania častí.

Po tom, čo sme rozdelili obsah E-learningového programu na časti, v ktorých bude vzdelávanie prebiehať je treba prispôbiť niekoľko ďalších vlastností. Prvou z nich je obtiažnosť kurzu. Je treba identifikovať tie časti preberanej látky, ktoré robia študentom najväčšie problémy. To sa najlepšie zistí na základe zhliadnutia výsledkov testov a cvičení nachádzajúcich sa na konci lekcie. Tieto časti je treba upravovať až dokiaľ s nimi študenti budú mať problémy. Najlepšou cestou je zvolenie inej alternatívy vysvetlenia problematiky. Je tiež možné inú postupy ako vizuálne pomôcky.

Ďalšou z vlastností je poskytnutie širších materiálov pre študentov so záujmom preniknúť hlbšie do problematiky. To možno dosiahnuť napríklad zahrnutím cvičení, ktoré presahujú obsah prebratej lekcie. Inou možnosťou je použitie odkazov na miesta, kde študenti môžu pokračovať v štúdiu danej problematiky.

2.3.2.2 Volba stratégie prezentovania informácií

V niektorých prípadoch je možné zvoliť jednotnú stratégiu prezentovania pre všetky materiály. Často sa však stáva, že jednotlivé sekcie a podsekcie musia byť prezentované inými stratégiami, lebo ich obsah je tak rozdielny, že žiadna stratégia by nemohla vyhovovať všetkým sekciam.

Lekcia môže byť vedená klasickým postupom, kde inštruktor prednáša informácie študentom cez online prezentáciu. Online prezentácia môže byť pasáž na čítanie, video alebo audio nahrávka. V niektorých prípadoch môže byť obsah prezentácie prepísaný do textovej podoby, aby študenti mohli sledovať látku zároveň s inštruktorom. Študenti následne diskutujú o prebratej látke na chate, alebo v študijných skupinách. Takáto stratégia výuky je vhodná najmä pre akademické kurzy a kurzy, ktorých súčasťou je mimo iného aj interakcia medzi študentmi.

Ďalšou možnou voľbou stratégie je zdokonaľovacie učenie. Lekcia začína vysvetlením istého postupu a následnou praktickou demonštráciou. V ďalšom kroku študenti trénujú daný postup, až dokiaľ ho nezdokonalia na najvyššiu úroveň. Táto stratégia je vhodná napríklad pre nováčikov na naučenie technických schopností, hlavne ak študenti musia vedieť aplikovať dané schopnosti predpísaným spôsobom. Využiť sa teda dá napríklad pri výuke inštalácií alebo riešenia problémov. Stratégia je tiež vhodná na učenie schopností, ktoré si nevyžadujú interakciu s ďalšími študentmi.

Tretou stratégiou je odhaľovacie učenie, u ktorého sa študenti najprv stretnú s problémom, ktorý ich postaví do reálnej situácie. Riešením problému študenti odhaľujú kľúčové body učenia. Následný rozbor problému potvrdí a posilní zistené body učenia. Problém a jeho riešenie môžu mať niekoľko podôb, najčastejšou je simulácia. Odhaľovacie učenie je vhodné pre rozvíjanie schopností, pri ktorých študenti musia robiť rozhodnutia. Sem patria napríklad manažérske schopnosti, alebo riešenie pokročilých problémov, ktoré nie sú popísané v príručkách.

V niektorých oblastiach výuky je vhodné použiť zmiešanú stratégiu, ktorá kombinuje postupy vyššie spomenutých stratégií. Najčastejšie sa stáva, že časť látky je prezentovaná online a časť v klasickej triede. Ďalej jedna časť online výuky je navrhnutá pre formálne učenie a druhá pre neformálne učenie. Takáto forma výuky je vhodná pre vysvetlenie a naučenie komplexných informácií.

Poslednou možnou stratégiou je učenie bez inštrukcií. Ide o typické neformálne učenie, kde si študent sám určuje čo sa chce naučiť a čo ho nezaujíma. Súčasťou tejto stratégie je poskytovanie vhodne upravených prehľadov materiálov, aby bolo vyhľadanie potrebných informácií rýchle a jednoduché. Ak je cieľom vytvoriť niekoľko previazaných E-learningových programov, mala by stratégia zabezpečiť prepojenie týchto programov. Najlepšou možnosťou ako to dosiahnuť je vytvoriť spoločnú domovskú stránku, ktorá rozdelí obsah do kategórií a bude obsahovať odkazy na jednotlivé kategórie.

2.3.3 Detailný návrh

Detailný návrh rozširuje konceptuálny návrh o špecifické plány prezentovania obsahu vzdelávacieho systému. Plány zahŕňajú prototypovú sekciu, všeobecný návrh vzhľadu obrazovky a ďalšie časti systému. Súčasťou detailného návrhu je tiež prevedenie testov použiteľnosti a schopnosti naučiť informácie na prototypu.

2.3.3.1 Príprava prototypu

Pokiaľ vytvárame prvý E-learningový systém, alebo prvý E-learningový systém daného typu, potrebujeme odozvu v každej fáze návrhu systému. Pre sponzorov a užívateľov je však veľmi ťažké reagovať na plán daného systému, radšej by reagovali na hotový systém. Aby sme sa vyhli nepríjemnému cyklu, kedy pre vývoj systému potrebujeme odozvu a zároveň na to aby sme dostali odozvu potrebujeme dokončený systém, je vhodné vytvoriť prototyp daného systému, ktorý bude prezentovať vzhľad a spôsob fungovania E-learningového systému. Po vzhliadnutí prototypu sú sponzori a užívatelia schopní poskytnúť potrebnú odozvu ohľadom všeobecného fungovania systému, jeho designu a orientovania sa v ňom.

Na výber máme niekoľko druhov prototypov:

1. Plne pracujúci prototyp, ktorý sa skladá z plne vyvinutej a naprogramovanej lekcie, obsahuje audio a video sekvencie. Aby sa znížili náklady na vývoj takéhoto prototypu, majú sekvencie iba zjednodušenú formu.
2. Papierový prototyp je vzhľadom veľmi podobný tomu ako by mal vyzeráť finálny produkt, ale nie je ešte naprogramovaný. Napríklad ak systém obsahuje test a študenti majú dostávať odozvu hneď ako odpovedajú na otázky v teste, prototyp iba povie ako, čo by sa stalo, keď študent odpovie na otázku. Ne zobrazí konkrétnu odpoveď.
3. Čiastočne pracujúci prototyp zahŕňa niektoré časti chýbajúce v papierovom prototypu, ale nie je taký rozsiahly ako plne pracujúci prototyp.

Ktorý prototyp si vybrať záleží na viacerých faktoroch. Čím rozsiahlejší a kompletnejší prototyp bude, tým bude odozva komplexnejšia. Ale s rozsahom prototypu rastú aj náklady a čas na jeho vývoj. Najčastejšou voľbou je čiastočne pracujúci prototyp, ktorý je v strede medzi potrebou odozvy a potrebou znížiť náklady.

2.3.3.2 Prezentovanie obsahu

Ďalšou otázkou, ktorú si musíme položiť je ako prezentovať obsah vzdelávania. U konceptuálneho návrhu sme si zvolili všeobecný postup, ktorý nám udáva všeobecný postup udalostí. V tejto časti je potrebné si zvoliť ako presne podať danú informáciu. K dispozícii máme hneď niekoľko možných postupov ako napríklad rozprávanie príbehu, použitie príkladov, definícií, obrázkov alebo kombinácií jednotlivých postupov. V nasledujúcej časti budú jednotlivé postupy podrobnejšie popísané.

Najprv je treba prejsť si postupne celý E-learningový systém a určiť, poradie splňovania jednotlivých cieľov učenia. Niektoré ciele sú tak rozsiahle, že zaberú viac obrazoviek, naopak iné sa môžu zmestiť aj viaceré na jednu obrazovku. Ku každému cieľu určíme akou formou bude podané jeho splnenie. Nestačí si vybrať len médium, ktorým bude obsah podaný, ale je potrebné určiť aj čo konkrétne bude daným médium podané. Tu sú popísané niektoré postupy prezentovania obsahu [2]:

1. Rozprávanie príbehu – podobne ako pri výuke v klasickej triede pomáha pochopiť teóriu demonštrovaním abstraktného konceptu v praxi. Poskytuje tiež možnosť zahrnutia študentov do procesu tým, že dostanú určitú rolu v príbehu. Napríklad budú musieť odpovedať na otázku ako by sa vysporiadali so situáciou, ktorá v príbehu nastala.
2. Demonštrácia obsahu – niekedy je vhodné študentom ukázať koncept v reálnej situácii. Napríklad niektorým študentom pomôže ak vidia vykonanie nejakej akcie priamo v praxi, než sa začnú učiť ako danú akciu vykonať.
3. Uvádzanie príkladov – podobne ako rozprávanie príbehov aj príklady pomáhajú študentom lepšie pochopiť abstraktný koncept. Napríklad ak učíme študentov používať programovací jazyk, je vhodné uviesť príklady použitia daného jazyka. Študenti si ho pozrú, zanalyzujú a pokúsia sa v ňom spraviť drobné zmeny, aby dosiahli výsledok aký potrebujú. Je dobré uvádzať príklady aj správneho aj nesprávneho použitia. Príklady nesprávneho použitia študentov naučia rozpoznávať chyby a vyvarovať sa im.
4. Kladenie otázok – E-learningový systém je schopný prijímať podnety od užívateľa a reagovať na ne. Preto je niekedy vhodné klásť užívateľovi otázky a na základe odpovedí sa prispôbiť užívateľovi tak, aby lepšie vyhovoval jeho potrebám.
5. Vizualizácia – niektoré časti sú lepšie pochopiteľné ak sú vizuálne znázornené, ako keď sú iba slovne popísané. Vhodným príkladom je zobrazenie čísiel do grafu, ktorý dá študentom určite viac ako strohé prečítanie čísiel. Takisto si viac zapamätajú z obrázku, ktorý vidia, ako keby ho mali iba slovne popísaný.
6. Fotografie – fotografie poskytujú realistické zobrazenie obsahu a zobrazujú skutočný svet.
7. Animácia – v niektorých prípadoch je lepšie vidieť obsah ako pracuje. Napríklad najlepším spôsobom ako sa naučiť ako pracuje nejaký stroj, je vidieť priamo ako pracuje.
8. Rozprávanie – obsah vzdelávania je narozprávaný na audio nahrávku a ten istý obsah je často dostupný aj v papierovej podobe, aby si ho mohli študenti s pomalším sluchovým vnímaním v klude preštudovať. Táto forma je vhodná napríklad pre výuku jazykov.
9. Video – video umožňuje študentovi obsah vzdelávania zároveň počuť aj vidieť. Je veľmi podobné animácií, ale na rozdiel od nej zobrazuje skôr reálne situácie ako ilustrácie. Ide o spojenie niektorých predchádzajúcich postupov do jedného celku.
10. Zmiešaný postup – ide súčasné používanie viacerých postupov. Aj najjednoduchšie programy sa skladajú z textu a grafickej prezentácie a tie zložitejšie obsahujú animácie, fotografie, či video

sekvencie. Systémy používajú zmiešané postupy, lebo jednotlivé časti obsahu si vyžadujú rôzne formy prezentovania.

Pri výbere vhodného postupu je treba brať ohľad aj na obmedzujúce faktory ako je napríklad rozpočet. Vytvorenie E-learningového systému je najlacnejšie ak bude obsahovať iba textové pasáže. Vývoj animácií, video a audio nahrávok je často tak nákladný, že nie je možné ho uskutočniť. Ďalším obmedzujúcim faktorom je technické nároky na systém. Vo firmách je často zakázané prehrávanie audio a video nahrávok, lebo príliš preťažujú prenos po sieti. Takisto nie všetci užívatelia majú počítače, ktoré sú schopné zvládnuť technické nároky kladené systémom obsahujúcim videá.

2.3.4 Zaistenie kvality systému

V tomto štádiu návrhu už máme špecifikované čo a ako bude v kurze prezentované, návrh ale ešte nezahrňuje veľa technických aspektov vývoja obsahu. Práve táto časť návrhu zohráva dôležitú rolu pri zabezpečovaní kvality E-learningového systému.

V prvom rade musíme zabezpečiť konzistentnosť používania názvov, pomenovaní, inštrukcií, číslovania a odrážok. Nejednotnosť by mohla viesť k tomu, že by sa študent cítil zmätený. Štýl písania obsahu programu je nutné si zvoliť ešte pred začiatkom vývoja systému a je potrebné s ním oboznámiť všetkých vývojárov. Pri voľbe štýlu sa v prvom rade berie ohľad na to, či firma, ktorá si systém objednala už nemá svoj vlastný štýl, ktorý by očakávala aj od tohto systému.

Ďalej sa musíme dohodnúť na tom, aký software a prostredie bude použité na vývoj systému. Je treba zabezpečiť, že všetci vývojári budú môcť používať toto prostredie. Tým sa zabezpečí konzistentnosť kódu a nenastanú situácie, kedy software jednotlivých vývojárov nie sú navzájom kompatibilné. Umožní to tiež znovu pracovať na systéme použitím zvoleného softwaru aj po určitej dobe, keď napríklad nastanú požiadavky na úpravu systému.

Následne je potrebné stanoviť minimálnu konfiguráciu počítaču, ktorá umožní systém používať. Teda budú stanovené presné požiadavky na hardware ešte predtým ako sa systém začne vyvíjať. Každá vyvinutá časť systému bude hneď odskúšaná na počítači s minimálnou konfiguráciou, aby sa overilo, že užívateľ s minimálnou konfiguráciou nebude mať žiadne problémy s používaním programu.

Vytvorenie šablón zabezpečí, že jednotlivé obrazovky budú mať podobný vzhľad a zároveň urýchli vývoj systému. Šablóna je ako prázdny formulár, teda obsahuje časti, ktoré sú napevno dané a budú sa opakovať vo viacerých obrazovkách a prázdne miesta, do ktorých bude doplnený obsah danej lekcie. Šablóna tiež zabezpečí konzistentnosť štýlu zobrazenia textu. Pri tvorbe šablón je potrebné najprv stanoviť, pre ktoré obrazovky sa budú vytvárať šablóny. Najčastejšie sú to titulné strany, menu, záverečné prehľady a zhrnutia, kvízy, testy, ale môžu to byť aj obrazovky obsahujúce informácie, alebo audio a video sekvencie. Následne sa identifikujú spoločné vlastnosti jednotlivých

obrazoviek ako sú hlavičky a grafické elementy, ktoré sa vyskytujú vždy na tom istom mieste. Nakoniec sa určí typ a veľkosť použitého písma, formát odstavcov a iné vlastnosti danej obrazovky.

2.4 Vývoj a údržba E-learningového systému

Vývoj E-learningového systému je fáza, v ktorej sa vytvorí systém podľa návrhu vytvoreného v predchádzajúcich fázach. Návrh nám nešpecifikuje len to čo má byť vytvorené a ako to má vyzerat', ale stanovuje nám aj isté očakávania na systém, ktoré musíme splniť. Pri vývoji systému sa musíme držať vytvorených plánov. Hoci v niektorých situáciách sú zmeny do plánov nevyhnutné, nemôžeme ich robiť len na základe nášho rozhodnutia, je potrebné ich konzultovať s firmou, ktorá si systém objednala.

Systém sa vyvíja zásadne v niekoľkých stupňoch. Po vytvorení prvého stupňa systému, je táto časť predvedená objednávajúcej firme, ktorá vyjadrí svoj názor na daný systém. Na základe tejto odozvy sa zreviduje systém a pripraví sa druhý stupeň. Po vytvorení druhého stupňa systému, je opäť predvedený firme a následne podstúpi aj prvé testy. Ide najmä o technické testy, ale uskutočnia sa aj testy použiteľnosti priamo na budúcich užívateľoch.

Nasledujúce kapitoly popíšu ako efektívne vytvoriť obsah E-learningového systému a ako previesť prehliadky a testy použiteľnosti.

2.4.1 Efektívna online komunikácia

Online komunikácia sa v mnohom odlišuje od komunikácie pomocou papiera. Podľa prieskumov je online čítanie menej kvalitné, ako čítanie z papiera. Je to dané tým, že čitatelia často prečítajú prvé dve vety na obrazovke a zvyšok obrazovky iba preletia a idú na ďalšiu obrazovku. Takisto rýchlosť čítania z obrazovky je nižšia ako pri čítaní z papiera. Na druhú stranu online komunikácia poskytuje možnosti, ktoré nie sú možné na papieri:

1. **Obraz** – E-learning poskytuje silné vizuálne prostriedky, ktoré sú často veľmi vhodné pre lepšiu výuku.
2. **Interaktivita** – Komunikácia medzi študentom a počítačom má formu dialógu. Počítač kladie otázky, študent na ne odpovedá a následne dostáva odozvu od počítača.
3. **Bezprostrednosť** – Študent dostáva odozvu na požiadavku bezprostredne po jej zadaní a v súvislosti s najaktuálnejším obsahom.
4. **Dôvernosť** – Ukladá si informácie o študentovi, mieru jeho postupu a môže prispôbiť obsah študentovým záujmom.

Vývojári E-learningových systémov využívajú tieto rozdiely medzi klasickým vzdelávaním a E-learningom a robia viac než len píše texty. Na komunikáciu so študentom využívajú obrázky a snažia sa vyvíjať interaktívnu komunikáciu so študentom, pripomínajúcu dialóg.

2.4.1.1 Vizualná komunikácia

Hlavný dôvod, prečo sa v E-learningu používajú obrázky je, že obrázky sa dajú lepšie pochopiť a ich obsah sa v pamäti študenta udrží dlhšie ako obsah textu. Najlepším príkladom toho je zobrazovanie číselných hodnôt. Hoci text môže obsahovať čísla, nedokáže vyjadriť vzťah medzi nimi tak efektívne a tak ľahko ako graf. Na vyjadrenie vzťahu časti k celku je vhodné použiť koláčový graf, alebo na vyjadrenie vzťahu dát počas určitého časového úseku je vhodné použiť histogram.

Vizualizácia sa dá tiež použiť na vyjadrenie konceptu, či už konkrétneho, alebo abstraktného. Príkladom je popis vzhľadu vecí, toho kde sa veci nachádzajú alebo použitím animácie popis princípu ako veci fungujú. Vhodná je tiež na popis procedúr ako napríklad graf ukazujúci ako pracovať s počítačom alebo s faxom.

Grafickým zvýraznením niektorých textových pasáží sa dá dosiahnuť zvýšená pozornosť študenta na obsah danej pasáže, alebo odlišenie častí, ktoré sú potrebné, od častí, ktoré sú len zaujímavé. Na vyvolanie zvýšenej pozornosti je možné použiť ohraničený alebo vytieňovaný rámček, alebo kruhy a šípky navigujúce študenta k dôležitým pasážam.

Pri používaní vizualizácie je veľmi dôležité dodržiavať určité pravidlá, aby sa ich efektívnosť neznížila, alebo aby neboli skôr prekážkou ako pomôckou pri štúdiu [2]:

1. Používať vizualizáciu len tam, kde má praktický význam. Každá vizualizácia by mala napomáhať štúdiu tým, že zobrazuje niečo relevantné vzhľadom k obsahu lekcie. V opačnom prípade vizualizácie vyrušujú študenta a sťažujú proces učenia.
2. Odstrániť nepotrebné detaily z obrázkov. Tie zbytočne zaťažujú študenta a sťažujú mu pochopenie vecí, ktorá je dôležitá.
3. Na vysvetlenie technických informácií použiť radšej vytvorenú ilustráciu ako fotografiu. Ilustrácia obsahuje len tie časti, ktoré sú pre pochopenie aktuálnej látky potrebné a neobsahuje žiadne prebytočné detaily.
4. Umiestniť vizualizáciu hneď za text, v ktorom je spomenutá. Vyhnúť sa umiestňovaniu do vyskakovacích okien, aby sa študent vyvaroval strate kontextu použitia obrázku.
5. Popísať každú ilustráciu a každú jej dôležitú časť. Výstižný názov ilustrácie uľahčí jej pochopenie, tak ako aj pomenovanie najdôležitejších častí ilustrácie. U grafov je nevyhnutné pomenovať osy a jednotky merania.
6. Udržovať veľkosť súborov s grafikou tak malú ako je možné. Užívatelia E-learningového systému očakávajú, že obsah lekcie sa zobrazí okamžite a teda nebudú musieť čakať na nahrávanie veľkých súborov. Aby sme zmenšili veľkosť súboru je potrebné vyhnúť sa zbytočným detailom. Ak aj napriek tomu bude nahratie grafiky trvať dlhšiu dobu, je potrebné o tom informovať užívateľa.

Vizuálna komunikácia má veľa kladov, ale aj mnohé úskalia. Ak však dodržíme všetky pravidlá tvorby a používania vizualizácií, vytvoríme E-learningový systém, ktorý bude schopný učiť študentov efektívnejšie ako by to bolo možné pri klasickom vzdelávaní.

2.4.1.2 Interaktívna komunikácia

Online komunikácia je výmena informácií medzi študentom a počítačom, nastáva teda interakcia. V niektorých prípadoch je vývojár kurzu schopný pripraviť celú interakciu sám, ale inokedy iba navrhne interakciu a pri realizácii návrhu mu musí pomôcť programátor. Je úlohou vývojára kurzu uvedomiť si možnosti interakcie popri tom ako pripravuje obsah kurzu.

Interakciou sa rozumie možnosť študenta zadať systému informácie a dostať naspäť primeranú odozvu. Najčastejšou formou interakcie v E-learningových kurzoch je kladenie otázok, na ktoré študent odpovedá a následne dostáva odozvu. Otázky môžu slúžiť ako oddeľovač jednotlivých častí obsahu, alebo ako test na záver kurzu. Najčastejšími podobami otázok sú výber z možností, áno / nie, spojenie možností alebo vyplnenie prázdneho miesta. Menej častou podobou sú cvičenia, v ktorých študent premiestňuje časti celku do správneho poradia, aby napríklad dostal správne fungujúcu procedúru.

Ďalším typom interakcie je možnosť študenta určiť si vlastnú cestu kurzom. Najčastejšími nástrojmi, ktoré toto umožňujú sú menu, odkazy a vyhľadávacie nástroje. Tie majú za úlohu rýchlo a jednoducho navigovať študenta k bodu jeho záujmu.

Nie všetky typy interakcie musia zahŕňať výmenu informácií medzi študentom a daným kurzom. Medzi takéto typy interakcie patria:

1. Sebazhodnotenie – študent pred začatím kurzu vyhodnotí svoje doterajšie vedomosti v danej oblasti a zistí, v čom má ešte medzery. Na základe toho si určí, ktorým oblastiam sa bude venovať.
2. Kooperujúce učenie – viac študentov spolupracuje na splnení online vzdelávacieho programu
3. Lov pokladov – online vzdelávací program naviguje študenta, kde na Internete má hľadať informácie

Tak ako pri vizuálnej komunikácii, tak aj pri interaktívnej komunikácii je potrebné dodržiavať isté pravidlá, aby interakcia bola efektívna. V prvom rade je treba zabezpečiť, aby interakcia bola primeraná. Tým sa myslí výmena informácií, ktoré priamo súvisia s obsahom kurzu. Ak sa interakcia netýka obsahu kurzu, odvedie pozornosť študenta od podstaty kurzu, čo môže viesť k strate záujmu študenta o daný obsah.

Ďalej je treba zabezpečiť primeranú odozvu na každý podnet od študenta. Ak študent odpovedá na otázky, mala by nasledovať okamžitá odozva hovoriaca, či bola odpoveď správna. Ak nebola správna, tak by mal študent dostať vysvetlenie prečo nie je správna. Následne by mal dostať inštrukcie ako má ďalej pokračovať.

2.4.2 Technické prehliadky a testovanie

Predtým ako sa E-learningový systém dostane k užívateľom, musí byť zaručené, že funguje správne a bez chýb. Prehliadky a testovanie sú najspôľahlivejším prostriedkom zabezpečenia týchto požiadaviek.

2.4.2.1 Technické prehliadky

Pri odhade úplnosti a presnosti online vzdelávacieho programu sa vývojári musia spoliehať na odozvu od klientov a marketingových špecialistov. Proces, pri ktorom vytvárajú tieto odhady sa volá prehliadka. Prehliadky sú obzvlášť dôležité v prípadoch, kedy chybné technické informácie môžu viesť k závažným situáciám, ako napríklad:

- Nesprávny obsah vyučovanej látky môže spôsobiť, že študenti budú nesprávne vykonávať svoju prácu, čo môže viesť k životunebezpečným situáciám.
- Vzdelávací program môže poskytovať nesprávne inštrukcie napríklad na ovládanie nového softwaru, čo spôsobí potenciálne problémy študentov a užívateľov daného softwaru.
- Odbornosť obsahu vzdelávania vysoko prevyšuje vedomosti vývojára kurzu.

Počas prehliadky sleduje každá skupina ľudí svoje vlastné kritériá, podľa ktorých systém hodnotí. Človek zodpovedný za projekt skontroluje, či je program presný. Marketingový špecialista zhodnotí, či program osloví skupinu užívateľov, pre ktorú je určený. Zákazník prehliadne program, aby zistil, či užívatelia budú schopní dosiahnuť stanovené ciele.

Prehliadky vzdelávacích programov typicky prebiehajú ako časť návrhu a po vytvorení prvých dvoch verzií. Vzdelávacie programy sú podrobené dvom typom technických prehliadok, formálnej a neformálnej.

Počas neformálnej technickej prehliadky vývojár požiada blízkych spolupracovníkov, aby prehliadli rannú verziu vzdelávacieho programu a zhodnotili, či sa ubera správny smerom. Na základe tohto zhodnotenia vývojár odchyť hlavné nedostatky programu a vylepší ich ešte predtým ako ho dá na formálnu prehliadku.

Pri formálnej technickej prehliadke vývojári požiadajú všetkých zainteresovaných ľudí v organizácii, aby vyjadrili názor na úplnosť a presnosť danej verzie. Aby sme sa vyhli možnosti, že ľudia zodpovedný za prehliadku nebudú vedieť ako prehliadať program a teda nebudú schopní podať vhodnú odozvu, je dobré ich usmerňovať počas postupu programom. Na rozdiel od neformálnej prehliadky je formálna prehliadka vopred naplánovaná. To nám zabezpečí, že správni ľudia budú mať možnosť prehliadnuť program a budú mať dostatok času na to, aby vyprodukovali kvalitnú odozvu. Po obdržaní pripomienok od prehliadačov je vhodné prediskutovať s nimi jednotlivé pripomienky. Je dobré sa vyjadriť, ktoré prijmem, a ktoré odmietame, prípadne požiadame o bližšiu špecifikáciu.

2.4.2.2 Technické testovanie

Testy majú overiť, že všetky odkazy a vetvenia v online vzdelávacom programe fungujú tak ako boli navrhnuté, že program beží bez chýb aj ak je spojený s inými komponentmi a softwarom, a že program je schopný zvládnuť také množstvo užívateľov, na aké bol navrhnutý. Testy prebiehajú v dobe, keď už je vývoj programu dokončený a je nepravdepodobné, že nastanú nejaké zmeny. Existuje niekoľko rôznych druhov testov, ktoré majú za úlohu otestovať rôzne časti systému.

1. Funkcionálne testovanie – overuje, že každý odkaz a každá vetva funguje tak ako bola navrhnutá.
 1. Každá možnosť na navigačnej lište v každej obrazovke prepne študenta tam kam má.
 2. Každá odpoveď na otázku vyvolá správnu odozvu.
 3. Každý odkaz v programe prepne študenta na žiadanú stránku.
 4. Všetky videá a audiá fungujú tak ako majú.

Funkcionálne testovanie môže byť prevedené rôznymi spôsobmi. Zatiaľ čo u malých programov ho prevedie jeden človek, u rozsiahlejších systémov sa testovaniu venuje niekoľko vývojárov kurzu. V niektorých prípadoch programátori napíšu postupy, ktoré automaticky kontrolujú tieto možnosti. Niektoré firmy dokonca predávajú software, ktorý automaticky skontroluje všetky odkazy.

2. Integračné testovanie – slúži na overenie, či bude online vzdelávací program fungovať, ak budú na počítači spustené iné aplikácie. Hlavnou úlohou je overiť, či online program spolupracuje so všetkými prehliadačmi, pre ktoré je navrhnutý. Integračné testovanie väčšinou prebieha manuálne.
3. Zavádzacie testovanie – overuje, či je online program schopný pracovať so stanoveným počtom zároveň prihlásených užívateľov. Zavádzacie testovanie je potrebné iba v prípade, že online program beží na serveri. Pomáha vývojárom určiť, koľko užívateľov môže byť naraz prihlásených než bude niekomu odmietnutý prístup.

Tieto testy určia, ktoré časti fungujú, a ktoré musia byť opravené, predtým ako bude program vypustený. Nájdeným chybám je potrebné priradiť prioritu a na jej základe ich opravovať vo zvolenom poradí.

2.4.3 Podpora a údržba E-learningového systému

Veľa vývojárov, začiatocníkov si myslí, že práca na systéme je dokončená po tom, ako je systém vyvinutý a otestovaný. Skutočnosť je však iná. Po zavedení systému do obehu, je nutné udržiavať systém a odstraňovať vzniknuté chyby. Činnosti, ktoré podpora a údržba E-learningového systému zahŕňa budú popísané v nasledujúcich odstavcoch.

V prvom rade je potrebné si naplánovať opravu technických nedostatkov. Aj napriek prevedeniu pokročilých testov, prevedených ešte pred vypusteným systémom, nejaké nedostatky sa objavia až počas každodenného používania systému. Pretože tieto nedostatky znekdialitňujú učenie,

musia byť opravené v primeranom čase. Najlepšie by bolo opraviť chyby ihneď, to však v reálnom svete nie je možné. Pravdepodobne bude potrebné začať nový projekt zameraný na odstránenie nedostatkov. Tým, že bude vopred naplánované prípadné nevyhnutné opravovanie systému, sa vyhneme situácii, kedy oprava systému spomalí prácu na inom projekte.

Ďalšou činnosťou je pomoc pri práci a pochopení obsahu E-learningového systému. Niektorí užívatelia nebudú rozumieť obsahu časti alebo celého kurzu. Iní budú potrebovať pomoc ako uplatniť obsah učenia na ich konkrétnu situáciu. A iným obsah kurzov nebude stačiť a budú sa zháňať po informáciách presahujúcich obzor kurzu. Preto je potrebné zabezpečiť poskytnutie pomoci študentom v podobe odborných rád a vysvetlení. Vhodnými formami komunikácie sú email, bezplatná telefónna linka, alebo online konzultácie napríklad pomocou chatu. Tak ako opravu chýb, aj pomoc pri práci je dobré vopred naplánovať a vyhradiť ľudí, ktorí sa o ňu budú starať.

Pokiaľ vopred vieme, že obsah vzdelávacieho programu je náchylný na zmeny, mali by sme to zahrnúť do plánov údržby systému. Napríklad ak sme vytvorili online vzdelávací program o nejakom novom produkte a vieme, že sa mu v priebehu niekoľkých mesiacov dostane vylepšení, je potrebné s tým počítať v plánoch údržby kurzu. Vylepšenia systému sa dajú rozdeliť do troch základných kategórií.

1. Malé úpravy – zahŕňajú zmeny niektorých pasáží, zvyčajne ide len o zmenu niekoľkých slov, alebo vety, prípadne jedného odstavca alebo ilustrácie.
2. Stredné zmeny – zahŕňajú pridanie nových sekcií, alebo zmeny pasáží alebo grafiky.
3. Veľké zmeny – zahŕňajú komplexnú zmenu buď vizuálnej stránky programu, alebo obsahu kurzu.

Stredné a veľké zmeny sú celkom nové projekty a mali by byť podľa toho aj naplánované.

Počas plánovania údržby online vzdelávacieho programu je potrebné venovať pozornosť aj údržbe obsahu programu. Čím viac vieme o plánovaných zmenách vopred, tým lepšie vieme odhadnúť koľko času a prostriedkov nám to zaberie.

3 Existujúce E-learningové systémy

V súčasnosti existuje veľké množstvo nástrojov pre tvorbu E-learningových systémov a kurzov. V nasledujúcich podkapitolách sa zameriam na dva významné nástroje Mindflash a Moodle.

3.1 Moodle

Moodle je software na vytváranie internetových kurzov a stránok. Je navrhnutý tak, aby umožňoval učiteľom vytvárať online kurzy s možnosťou širokej interakcie so študentmi. Je to open-source software, šírený pod licenciou GNU. Moodle môže byť nainštalovaný na ktoromkoľvek počítači, ktorý podporuje PHP a SQL typ databáz, napríklad MySQL. Funguje na operačných systémoch Windows, MacOS a na viacerých druhoch linuxov [6].

Moodle je skratka z Modular Object-Oriented Dynamic Learning Environment, teda modulárne objektovo orientované dynamické vzdelávacie prostredie. U jeho zrodu stál Martin Dougiamas, ktorý sa dodnes podieľa na vývoji nových verzií. Prvá verzia vznikla v roku 2002 a bola určená pre prostredie menších tried na vysokých školách. V súčasnosti už Moodle nepoužívajú len vysoké školy, ale aj stredné a základné školy, neziskové organizácie, súkromné firmy alebo nezávislí učители. Šírenie Moodle a rastúci počet užívateľov umožňuje získavať stále viac ohlasov od ľudí z rôznych vzdelávacích prostredí.

Ukážky kurzov a testov vytvorených v systéme Moodle je možné nájsť v prílohe.

3.1.1 Teoretické východiská Moodle

Koncepcia a celý vývoj systému Moodle sú založené na smere v teórii učenia, ktorý sa nazýva sociálne konštrukcionistická pedagogika. Táto teória je založená na štyroch základných konceptoch, sú nimi konštruktivizmus, konštrukcionizmus, sociálny konštruktivizmus a kolektívne a samostatné chovanie [8].

Konštruktivizmus tvrdí, že ľudia si nové znalosti aktívne konštruujú, vytvárajú pri interakcii s okolím. Všetko čo čítame, vidíme, počujeme, cítime a čoho sa dotýkame je porovnávané s našimi znalosťami a pokiaľ to je s našim mentálnym svetom kompatibilné, môže sa to stať novým poznatkom. Znalosť sa upevňuje, pokiaľ ju úspešne použijeme v prostredí, v ktorom žijeme. Nie sme len pamäťová banka pasívne prijímajúca informácie a znalosti nám nemôžu byť predané len prečítaním niečoho alebo počúvaním niečoho. To neznamená, že sa nemôžeme nič naučiť čítaním prednášky, konštruktivizmus len zdôrazňuje, že pri tom prebieha interpretácie a nie len prenos informácií z jedného mozgu do druhého.

Konštrukcionizmus vychádza z toho, že učenie je obzvlášť efektívne, ak pri ňom tvoríme niečo pre ostatných. Môže to byť čokoľvek, od hovorenej vety až po zložitejšie výtvary, akými sú obraz,

dom alebo softwarový produkt. Napríklad tento text si môžeme prečítať niekoľkokrát a aj napriek tomu jeho obsah do zajtra zabudnúť. Ale ak by sme mali myšlienky, ktoré obsahuje vysvetliť vlastnými slovami niekomu inému, je zaručené, že by sme získali hlbšie porozumenie, ktoré by bolo lepšie začlenené do nášho vlastného myslenia. Preto je dobré si robiť poznámky na prednáške, aj ak ich nikdy nebudeme čítať.

Sociálny konštruktivizmus rozširuje vyššie uvedené myšlienky na sociálnu skupinu, kde sa vytvárajú veci spoločne a pre všetkých. Teda vzniká malá kultúra spoločných výtvorov so spoločnými významami. Ak je jedinec do takejto skupiny zaradený, nepretržite sa učí ako byť jej súčasťou. Veľmi jednoduchým príkladom môže byť ľubovoľný predmet, napríklad šálka. Dá sa použiť na všeličo možné, ale jej tvar jasne naznačuje existenciu akéhosi povedomia o skladovaní tekutín. Komplikovanejším príkladom môže byť elektronický kurz, tu už nielen “tvar“ softwarového nástroja naznačuje niečo o tom, ako by online kurzy mali fungovať, ale činnosť členov skupiny a texty, ktoré vytvorili, pomáhajú utvárať chovanie každého člena skupiny.

Kolektívne a samostatné chovanie sa hlbšie zaoberá motiváciou účastníkov diskusie. Za individuálne sa považuje chovanie, pri ktorom sa človek snaží zostať objektívny a vecný, háji svoje vlastné myšlienky a hľadá logické medzery v argumentoch protivníka. Vzťahové chovanie je empatickejší prístup, ktorý pripúšťa subjektívnejšie postoje, snaží sa počúvať a klásť otázky v snahe porozumieť názoru druhého. Cieľové chovanie vzniká vtedy, keď človek vníma obe tieto možnosti prístupu a je schopný si jednu z nich vybrať ako vhodnú pre danú situáciu. Rozumná miera vzťahového chovania je pri učení v skupine veľmi silným stimulom pre štúdium. Popri zblížovaní ľudí podporuje hlbšie preskúmavanie ich presvedčenia.

3.1.2 Vlastnosti Moodlu

Moodle je vhodný pre distančnú internetovú výuku aj ako doplnok klasickej výuky. Má jednoduché, efektívne, široko kompatibilné, technicky nenáročné a intuitívne užívateľské rozhranie. Obsahuje zoznam kurzov, ktorý obsahuje stručný popis kurzov a informáciu o ich návštevnosti študentmi. Kurzy je možné rozdeliť do kategórií, ktoré je možné prehľadávať.

Správu systému má na starosti administrátor systému, ktorý je zvolený pri inštalácii. Systém takisto poskytuje rozsiahlu správu užívateľov. Jej cieľom je obmedziť nutnosť zásahov administrátora na minimum a udržať pritom vysoký stupeň zabezpečenia. Každý užívateľ potrebuje pre celý systém iba jeden účet, ktorému je možné nastaviť rôzne práva pre rôzne účely. Vytváranie kurzov riadi administrátor, ktorí tiež určí jedného z užívateľov ako učiteľa daného kurzu. Učiteľ môže pre každý kurz nastaviť kľúč zápisu, aby do neho mali prístup len oprávnení študenti. Tento kľúč im oznámi osobne alebo súkromným emailom. Učiteľ má tiež možnosť pridávať a odoberať študentov ručne.

Moodle poskytuje široký výber činností v kurze. K dispozícii sú fóra, denníky, testy, materiály, hlasovanie, dotazníky, úlohy a chat. Systém poskytuje rôzne možnosti sledovania a zaznamenávania

činnosti užívateľov, poskytuje prehľad jeho príspevkov do fór, zápisov do denníku a dôb prístupu do jednotlivých kurzov.

Moodle je v súčasnosti najrozšírenejším voľne šíreným systémom pre tvorbu E-learningových kurzov. Existuje tiež komerčná variant tohto systému, ktorá obsahuje ďalšie rozšírenia a funkcie, ktoré sa dajú využiť pri vzdelávaní na tej najvyššej úrovni.

3.2 Mindflash

Systém Mindflash je zástupcom komerčných E-learningových systémov. Bol navrhnutý, tak aby umožnil organizáciám akejkoľvek veľkosti vytvárať online kurzy rýchlo. K dispozícii je zadarmo skúšobná verzia systému, v ktorej je možné si vytvoriť vlastný kurz a skúsiť si ho. Mindflash je hosťiteľský systém, nie je teda potrebné nič inštalovať a nevznikajú dodatočné náklady spojené s hardwarovou podporou systému alebo údržbou systému. Takisto je zaručené, že všetci klienti budú pracovať na najnovšej verzii, ktorá je spätne kompatibilná s predchádzajúcimi verziami. Systém je kombináciou nástrojov na vytváranie kurzov, výstavbu a riadenie kurzov a riadenie nahrávania kurzov [9].

Umožňuje vytvorenie kurzov v rôznych podobách. Na vytvorenie kurzu je možné použiť súbory Wordu, PowerPointu, Flash, HTML, pdf, audio, video a iné. Súbory sa jednoducho nahrajú do systému a ten sa postará o ich správnu interpretáciu. Zabezpečenie kurzu proti neoprávnenému používaniu je implementované prihlasovaním pomocou loginu a hesla.

Mindflash umožňuje prispôbiť vzhľad kurzu podľa požiadaviek zákazníka. Na výber je niekoľko šablón a takisto je možné zvoliť farbu prostredia. Pre pokročilých užívateľov tu je možnosť samostatného vytvorenia celej šablóny. Na otestovanie študentov je na výber z niekoľkých druhov otázok. K dispozícii sú napríklad výber z možností, áno/nie otázky, vyplnenie prázdneho miesta, spojenie správnych možností a iné.

Na umožnenie interakcie medzi študentmi Mindflash poskytuje niekoľko prostriedkov. Samozrejmosťou sú verejné a privátne chaty a fórum. Okrem toho systém poskytuje adresáre, kam môžu študenti nahrávať súbory, ku ktorým majú potom prístup ostatní študenti kurzu. Ďalšou praktickou službou je možnosť jednoduchého zasielania emailov od učiteľa študentom daného kurzu.

Najväčšou slabinou systému Mindflash je obmedzený výber jazyka. K dispozícii je angličtina, španielčina a portugalčina. Vo vývoji je francúzština a taliančina.

4 Prehľad kurzu Algoritmy

Algoritmy sú všeobecné, deterministické (v každom kroku algoritmu je jednoznačne stanovené, čo bude nasledovať), konečné a resultatívne (vedú k správne výsledku). Algoritmy spolu s dátovými štruktúrami tvoria programy.

4.1 Abstraktné dátové štruktúry

Dátový typ je definovaný množinou hodnôt, ktoré môže objekt tohto typu nadobúdať a množinou operácií nad týmito objektmi. Abstraktný dátový typ potláča hľadisko ako sú dáta zobrazené a ako sa nad nimi prevádzajú operácie a zdôrazňuje čo dáta reprezentujú a čo s nimi operácie prevádzajú [5]. Abstraktný dátový typ je teda odvodený z dátového typu ponechaním jeho najvýznamnejších vlastností a vynechaním ostatných vlastností, týkajúcich sa najmä implementácie v pamäti a realizácie operácií konkrétnymi prostriedkami.

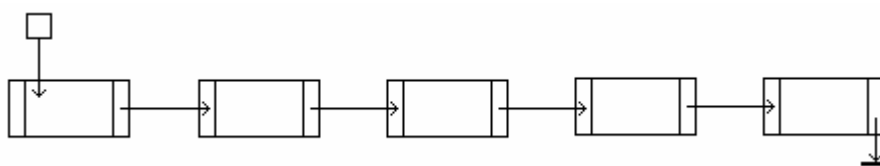
Použitie abstraktných dátových štruktúr ma význam hlavne v algoritmoch, ktoré sa týmto stávajú omnoho jednoduchšie, prehľadnejšie a najmä efektívnejšie. Ďalšou výhodou je, že sa zvýši spoľahlivosť programu tým, že je umožnený prístup a manipulácia s abstraktnými dátovými štruktúrami len pomocou stanovených operácií. Tým sa zabráni priamemu prístupu k všetkým objektom abstraktných dátových štruktúr.

Medzi najvýznamnejšie abstraktné dátové štruktúry patria zoznam, zásobník, fronta a strom.

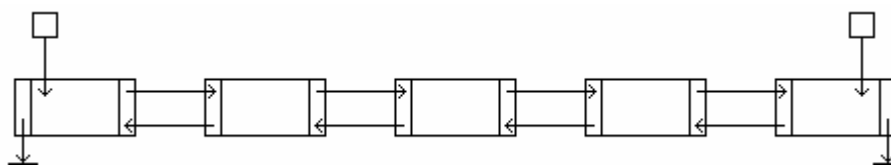
4.1.1 Zoznam

Zoznam je lineárna, homogénna, dynamická dátová štruktúra. Lineárnosť zoznamu je daná tým, že každý prvok má jednoznačne definovaného predchodcu a následníka, výnimkou sú iba prvý a posledný prvok zoznamu. Prvkom zoznamu môže byť ľubovoľný dátový typ.

Aby bolo možné nadefinovať operácie nad zoznamom, je potrebné stanoviť špeciálnu vlastnosť zoznamu, aktivitu. Hovoríme, že zoznam je aktívny, ak je jeden z prvkov zoznamu aktívny, a zoznam nie je aktívny, ak ani jeden z prvkov zoznamu nie je aktívny. Existujú dva typy lineárnych zoznamov. Sú nimi jednosmerný zoznam a dvojsmerný zoznam.



Obr. 4.1: Jednosmerný zoznam



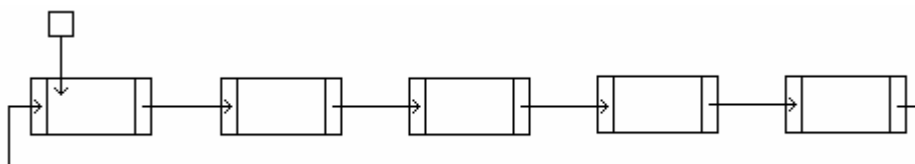
Obr. 4.2: Dvojsmerný zoznam

Operácie nad abstraktnou dátovou štruktúrou jednosmerný zoznam sú definované nasledovne:

- ListInit – vytvorí prázdny zoznam prvkov daného typu
- Active – vracia boolovskú hodnotu, vráti true v prípade, že je zoznam aktívny alebo false v prípade, že zoznam nie je aktívny
- First – nastaví prvý prvok zoznamu na aktívny, ak je zoznam prázdny, operácia je bez účinku
- Succ – aktívnym sa stane následník aktívneho prvku, ak je aktívny posledný prvok zoznamu, aktivita sa stratí, ak je zoznam prázdny, operácia je bez účinku
- InsertFirst – vloží nový prvok na začiatok zoznamu
- PostInsert – vloží nový prvok za aktívny prvok
- Copy – vráti hodnotu aktívneho prvku, ak zoznam nie je aktívny, nastáva chyba
- CopyFirst – vráti hodnotu prvého prvku, ak je zoznam prázdny, nastáva chyba
- DeleteFirst – zruší prvý prvok zoznamu, ak je zoznam prázdny, operácia je bez účinku
- PostDelete – zruší prvok za aktívnym prvkom, ak zoznam nie je aktívny, alebo za aktívnym prvkom sa nenachádza ďalší prvok, je operácia bez účinku
- Actualize – hodnota aktívneho prvku je prepísaná novou hodnotou, ak zoznam nie je aktívny, je operácia bez účinku

Dvojsmerný zoznam má všetky operácie jednosmerného zoznamu a k nim inverzné operácie Last, Pred, InsertLast, PreInsert, CopyLast, DeleteLast, PreDelete.

Kruhový zoznam sa vytvorí z lineárneho zoznamu tak, že následníkom posledného prvku sa stane prvý prvok. Podobne ako lineárny zoznam aj kruhový zoznam môže byť jednosmerný alebo dvojsmerný. Kruhový zoznam nemá ani začiatok ani koniec, z praktického hľadiska však je stanovený jeden prvok ako prvý a na neho sa budú vzťahovať operácie pracujúce so začiatkom a koncom zoznamu.



Obr. 4.3: Jednosmerný kruhový zoznam

Medzi najčastejšie používané vyššie operácie nad zoznamami patria zistenie počtu prvkov zoznamu, zlúčenie zoznamov, rozdelenie zoznamu, vytvorenie kópie zoznamu, vyhľadanie prvku s určitou hodnotou a zoradenie prvkov podľa hodnoty.

4.1.2 Zásobník

Zásobník je lineárna, homogénna, dynamická dátová štruktúra. Neprázdny zásobník obsahuje jeden prvok označený ako vrchol zásobníku. Po vložení ďalšieho prvku do zásobníka sa stane nový prvok vrcholom zásobníka a predchádzajúci vrchol sa stane jeho následníkom. Pri vyberaní prvkov zo zásobníka sa vyberá vždy prvok, ktorý bol vložený ako posledný a jeho následník sa stane novým vrcholom zásobníka. Prvky sa teda zo zásobníka vyberajú v obrátenom poradí ako boli do zásobníka vkladané. Takýto spôsob práce sa nazýva LIFO z anglického Last In, First Out.

Operácie nad abstraktnou dátovou štruktúrou zásobník sú definované nasledovne:

- StackInit – vytvorí prázdny zásobník
- Push – vloží prvok na vrchol zásobníka
- Pop – odstráni prvok na vrchole zásobníka
- Top – získa hodnotu prvku na vrchole zásobníka
- SEmpty – vracia boolovksú hodnotu, vráti true v prípade, že je zásobník prázdny alebo false v prípade, že zásobník nie je prázdny

Zásobník nachádza svoje využitie v algoritmoch, kde je potrebné implementovať reverziu poradia, pri vytváraní rekurzívnych podprogramov a algoritmov s návratom. Je tiež vhodný na prevod infixovej notácie na prefixovú.

4.1.3 Fronta

Fronta je lineárna, homogénna, dynamická dátová štruktúra. Obsahuje dva konce, na jednom sa prvky do fronty pridávajú (tzv. koniec fronty) a na druhom sa prvky z fronty odoberajú (tzv. začiatok fronty). Poradie vyberania prvkov je opačné ako u zásobníku. Prvky sa z fronty vyberajú v takom poradí, a akom do nej boli vložené. Tento spôsob práce sa nazýva FIFO z anglického First In, First Out.

Operácie nad abstraktnou dátovou štruktúrou fronta sú definované nasledovne:

- QueueInit – vytvorí prázdnu frontu
- QueUp – vloží prvok na koniec fonty
- Remove – odstráni prvok na začiatku fronty
- Front – získa hodnotu prvku na začiatku fronty
- QEmpty – vracia boolovksú hodnotu, vráti true v prípade, že je fronta prázdna alebo false v prípade, že fronta nie je prázdna

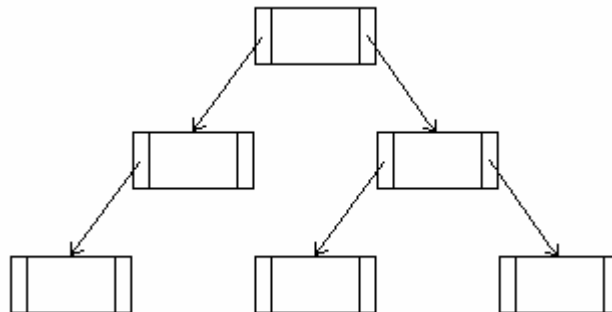
Fronta nachádza uplatnenie najmä v úlohách z oblasti hromadnej obsluhy.

Variantom klasickej fronty je obojstranne ukončená fronta, ktorá umožňuje vkladanie aj vyberanie prvkov na oboch koncoch fronty. Operácie tohto variantu sú podobné ako u klasickej fronty a sú doplnené o inverzné operácie vkladania, odstránenia a získania hodnoty.

4.1.4 Strom

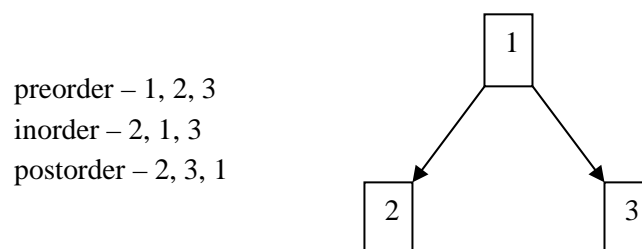
Strom je nelineárna štruktúra a je to acyklický súvislý graf. Koreňový strom obsahuje jeden špeciálny uzol, ktorý sa nazýva koreň. Koreň je taký uzol, že z každého uzlu stromu vedie len jedna cesta do koreňa. Z každého uzlu stromu vedie jedna hrana smerom ku koreňu do uzlu, ktorý sa nazýva otcovský a ľubovoľný počet hrán k uzlu, ktoré sa nazývajú synovské. Uzly stromu, ktoré nemajú synovské uzly, sa nazývajú koncové uzly alebo listy. Všetky ostatné uzly sa nazývajú vnútorné uzly. Podstrom koreňového stromu je strom, ktorý získame, ak budeme niektorý vnútorný uzol považovať za koreň stromu. Výška prázdneho stromu je 0, výška stromu obsahujúceho len koreň je 1 a výška iného stromu je rovná počtu hrán od koreňa stromu k najvzdialenejšiemu uzlu zvýšenému o 1.

Špeciálnym prípadom koreňového stromu je binárny strom. Je to strom, ktorého každý uzol má najviac dva synovské uzly. Binárny strom je váhovo vyvážený, ak pre všetky jeho uzly platí, že počty uzlov ich ľavého podstromu a pravého podstromu sa rovnajú, alebo sa líšia o 1. Binárny strom je výškovo vyvážený, ak pre všetky jeho uzly platí, že výška ľavého podstromu sa rovná výške pravého podstromu, alebo sa líšia o 1 [5].



Obr. 4.4: Binárny strom

Dôležitou operáciou nad stromom je priechod stromom, ktorý transformuje stromovú štruktúru na lineárnu štruktúru. Významnými priechodmi sú preorder, inorder a postorder, ktoré sú nadefinované nasledovne:



Obr. 4.5: Priechody stromom

4.2 Vyhľadávacie algoritmy

Vyhľadávanie je založené na jednoduchom princípe hľadania zhody (úplnej alebo čiastočnej) v prvku označenom ako kľúč. Účelom vyhľadávania je obvykle získanie dát uložených v položke zviazanej s kľúčom.

Dáta môžu byť uložené v rôznych typoch dátových štruktúr (tabuľka, zoznam, strom atď.) a vyhľadávanie v jednotlivých dátových štruktúrach je možné implementovať rôznymi algoritmami. Metódy vyhľadávania sa klasifikujú podľa vlastností ako napr. rýchlosť a pamäťová náročnosť.

Hlavným typom dát súvisiacim s vyhľadávaním je tabuľka. Existujú dva základné druhy implementácie vyhľadávacej tabuľky:

- dátovou štruktúrou so sekvenčným prístupom – vyhľadávanie prebieha prechádzaním dátovej štruktúry od začiatku prvok po prvok
- dátovou štruktúrou s priamym prístupom – vyhľadávanie sa uskutočňuje priamym prístupom do pamäti.

Vo všeobecnosti neexistuje univerzálna metóda, ktorá by mala za každých podmienok najlepšie vlastnosti. To ako bude metóda efektívna závisí na faktoroch ako počet položiek v dátovej štruktúre alebo či prevláda úspešné vyhľadávanie nad neúspešným. Preto je dôležité dôkladne zvážiť výber metódy.

4.2.1 Sekvenčné vyhľadávanie

Pole môže mať dva druhy implementácie: statickú a dynamickú. Statickou formou implementácie je pole, ktoré má vopred danú pevnú dĺžku. Tento druh poľa je vhodný, ak vopred poznáme aspoň približný počet dátových položiek, ktoré budú v poli uložené.

Sekvenčné vyhľadávanie môže prebiehať buď v zoradenom, alebo nezoradenom poli. U nezoradeného poľa nie sú položky v poli usporiadané podľa relácie \leq resp. \geq . Vkladanie do nezoradeného poľa je jednoduchá operácia, kedy sa nový prvok vždy pripojí na koniec poľa. Vyhľadávanie je založené na sekvenčnom prehľadávaní poľa a porovnávaní hodnoty hľadanej položky s kľúčom. Vyhľadávanie sa končí neúspešne ak prideme na koniec poľa.

U zoradeného poľa sú prvky v poli zoradené podľa relácie usporiadania od najmenšieho po najväčší. Na rozdiel od vyhľadávania v nezoradenom poli je operácia vkladania nového prvku zložitejšia. Najprv je potrebné nájsť miesto kam bude prvok vložený a potom treba posunúť časť poľa, ktorá obsahuje prvky väčšie ako nový prvok, o jednu pozíciu doprava. Toto nám urýchli operáciu vyhľadávania, ktoré sa končí neúspešne ak narazíme na kľúč, ktorý je väčší ako hľadaný kľúč.

Z charakteru prehľadávaní poľa vyplýva, že najrýchlejšie budú nájdené položky, ktoré sa nachádzajú na začiatku a naopak najpomalšie bude vyhľadanie položiek z konca poľa.

Dynamickou voľbou pre implementáciu poľa je zreťazený zoznam. Princíp algoritmov sa pre všetky metódy zachováva. Hlavná výhoda oproti statickému poľu spočíva v tom, že nemusíme vopred stanoviť maximálny počet položiek. Zjednoduší sa aj operácia vkladania novej položky do zoradeného zoznamu. Vloženie spočíva v presmerovaní niekoľkých ukazateľov a teda nám odpadá presúvanie zvyšku zoznamu o jednu pozíciu doprava. Na druhú stranu má zreťazený zoznam nevýhodu v tom, že potrebujeme ďalší priestor v pamäti pre ukazatele, ktoré prepájajú prvky zoznamu.

4.2.2 Nesekvenčné vyhľadávanie v zoradenom poli

Ak sú prvky vyhľadávacej tabuľky zoradené podľa veľkosti kľúča položiek, je možné použiť algoritmy, ktoré sú efektívnejšie ako sekvenčné vyhľadávanie. Tieto algoritmy sa podobajú numerickým metódam pre hľadanie koreňa funkcie jednej premennej, ak poznáme interval, v ktorom je práve jeden koreň. Obzvlášť je vidieť podobnosť s metódou polenia intervalu [5].

Najčastejším zo spôsobov nesekvenčného vyhľadávania je binárne vyhľadávanie. Vyhľadávaný kľúč sa porovná s kľúčom položky nachádzajúcej sa v polovici zoradeného poľa. Ak sú kľúče zhodné, vyhľadávanie končí úspešne. Ak je vyhľadávaný kľúč menší ako kľúč položky, pokračuje sa vo vyhľadávaní v ľavej polovici poľa. Ak je vyhľadávaný kľúč väčší ako kľúč položky, pokračuje sa vo vyhľadávaní v pravej polovici poľa. Vyhľadávanie končí neúspešne v prípade, že prehľadávaná časť poľa neobsahuje žiadny prvok.

Dijkstrov variant binárneho vyhľadávania predpokladá, že pole môže obsahovať viac položiek s rovnakým kľúčom. V prípade úspešného vyhľadania algoritmus nájde najľavejšiu položku z položiek s rovnakým kľúčom.

Keďže multiplikatívne operácie, a teda aj polenie intervalu, sú časovo veľmi náročné, stráca sa pri ich častom opakovaní efektívnosť algoritmu. Toto sa snaží riešiť uniformné binárne vyhľadávanie. Je založené na princípe určenia hraníc intervalu odchýlkou od stredu. Pre danú vyhľadávaciu tabuľku sa vopred spočíta tabuľka odchýlok, ktorá sa následne využíva v algoritme.

4.2.3 Binárne vyhľadávacie stromy

Binárny vyhľadávací strom (BVS) je usporiadaný binárny strom, pre ktorého každý uzol platí, že jeho ľavý podstrom je buď prázdny, alebo pozostáva z uzlov, ktorých hodnoty kľúčov sú menšie než hodnota kľúča daného uzlu a podobne jeho prvý podstrom je buď prázdny, alebo pozostáva z uzlov, ktorých hodnoty kľúčov sú väčšie než hodnota kľúča daného uzlu [5].

Algoritmy vkladania a vyhľadávania BVS sú veľmi efektívne a majú nízku náročnosť na implementáciu. Pre tieto vlastnosti sú BVS používané v mnohých aplikáciách a sú hodnotené ako jedna z najvýznamnejších metód. Priechodom BVS typu inorder získame usporiadanú postupnosť kľúčov.

Pre BVS sú typické rekurzívne algoritmy vkladania a vyhľadávania. Obidve operácie majú aj svoje nerekurzívne varianty, ale ich použitie nie je príliš časté.

Efektívnosť algoritmov pracujúcich s BVS výrazne závisí na tvare stromu. Ideálny prípad je ak je strom vyvážený. Vtedy sa počet krokov od koreňa k uzlu rovná $\log N$. Pravdepodobnosť, že sa také niečo stane v reálnej situácii je však takmer nulová. Pre náhodné kľúče sú ale veľmi vzácne aj silne nevyvážené stromy. Preto sú BVS v priemere vyvážené dostatočne na to, aby boli efektívnejšie ako mnohé iné metódy.

4.2.4 Tabuľky s rozptýlenými položkami

Prístup do poľa prvkov sa uskutočňuje pomocou mapovacej funkcie. Nájdenie vhodnej mapovacej funkcie je najťažšou a najdôležitejšou úlohou. Nevhodná mapovacia funkcia výrazne zneefektívni túto metódu a v tom najhoršom prípade ju môže degradovať na sekvenčné pole. Hlavné požiadavky na mapovaciu funkciu sú rýchlosť a čo najmenší počet kolízií. Kolízia je jav, kedy sa dva rôzne kľúče namapujú na to isté miesto. Kľúče, ktoré sa namapujú na to isté miesto sa nazývajú synonymá.

Mapovacie funkcie často využívajú operáciu zvyšku po celočíselnom delení. Úspešnosť takejto funkcie výrazne závisí na zvolenej veľkosti tabuľky. Jedna veľkosť tabuľky môže zabezpečiť rovnomerné rozptýlenie položiek, zatiaľ čo iná veľkosť môže spôsobiť príliš veľký počet kolízií.

Tabuľka s rozptýlenými položkami (TRP) sa skladá z mapovacieho priestoru, reprezentovaného poľom a zo zoznamov synonym, ktoré sa začínajú v položkách mapovacieho priestoru.

Vyhľadávanie v TRP sa skladá z dvoch krokov. V prvom kroku sa pomocou rozptyľovacej funkcie prevedie hodnota kľúča na index poľa, v ktorom sa začína lineárny zoznam synonym. V druhom kroku sa sekvenčným priechodom týmto zoznamom hľadá položka s daným kľúčom. Vyhľadávanie v TRP má teda indexsekvenčný charakter.

Podľa druhu zoznamu synonym existujú TRP:

- s explicitným zoradením synonym – každý prvok obsahuje adresu následníka
- s implicitným zoradením synonym – adresa následníka je funkciou adresy predchodcu.

4.3 Radiace algoritmy

Radenie je usporiadanie položiek podľa relácie lineárneho usporiadania na kľúči. Radiace algoritmy, sú charakterizované niekoľkými vlastnosťami, ktoré určujú na aký druh úloh sa algoritmy hodia. Sekvenčnosť radenia vyjadruje, že algoritmus pristupuje k položkám v takom poradí, v akom sú usporiadané v dátovej štruktúre. Opakom sekvenčnosti je priamy prístup k položkám štruktúry. Prirodzenosť radenia vyjadruje, že doba potrebná na zoradenie už zoradenej množiny údajov je menšia ako doba potrebná na zoradenie náhodne usporiadanej množiny a tá je menšia ako doba

potrebná na usporiadanie opačne zoradenej množiny údajov. Stabilita radenia znamená, že algoritmus zachováva vzájomné poradie položiek s rovnakou hodnotou kľúča. Táto vlastnosť je potrebná najmä pri radení podľa viacerých kľúčov.

4.3.1 Radenie na princípe výberu

Radenie na princípe výberu (select sort) je najjednoduchší spôsob radenia. Pole je rozdelené na dve časti, prvá časť poľa od indexu 1 až po index k je zoradená a druhá časť poľa od indexu $k+1$ až po n nie je zoradená. Algoritmus v každom cykle nájde položku s najmenšou hodnotou v časti poľa od indexu $k+1$ po n a premiestni ju na pozíciu s indexom $k+1$. Teraz je už zoradená časť poľa od indexu 1 po $k+1$. V prvom kroku algoritmu je zoradená časť vstupného poľa prázdna. Symetricky je algoritmus aplikovateľný pre zoradenie od najväčšieho prvku po najmenší, kedy je namiesto položky s najmenšou hodnotou hľadaná položka s najväčšou hodnotou. Táto metóda je nestabilná, lebo vymenený prvok sa môže dostať za prvok so zhodnou hodnotou.

Iným variantom radenia na princípe výberu je bublinový výber (bubble sort). Princíp tejto metódy je zhodný s predchádzajúcou metódou, odlišný je spôsob vyhľadávania minima, respektíve maxima. Nezoradená časť poľa sa prechádza sprava doľava a postupne sa porovnávajú každé dve susedné položky. Ak nie sú položky zoradené podľa kľúča, vymenia si pozície. Takto sa položka s najmenšou hodnotou dostane na najľavejšiu pozíciu doteraz nezoradenej časti poľa. Radenie končí, ak počas priechodu nezoradeného poľa nedôjde k výmene položiek. Bublinový výber je metóda stabilná a prirodzená. Je to najmenej efektívna metóda, ale je najrýchlejšia pre zoradené pole.

4.3.2 Radenie na princípe vkladania

Pole položiek je rozdelené na dve časti, v ľavej časti sú prvky zoradené a v pravej časti sú prvky v ľubovoľnom usporiadaní. Zoberieme najľavejší prvok z pravej časti poľa a hľadáme index k v ľavej časti poľa taký, pre ktorý bude platiť, že všetky hodnoty položiek s indexom menším ako k budú menšie ako aktuálne vkladaná hodnota a všetky hodnoty položiek s indexom väčším alebo rovným k budú väčšie ako aktuálne vkladaná hodnota. Následne časť poľa od indexu k po koniec usporiadanej časti posunieme o jedno miesto doprava a vložíme aktuálne vkladajúcu položku na miesto s indexom k . V prvom kroku algoritmu, je usporiadaná časť poľa tvorená jedným prvkom. Cyklus radenia končí zaradením poslednej položky.

Metóda bublinového vkladania (bubble insert sort) kombinuje vyhľadanie indexu k s posunom poľa. Princíp spočíva v postupnom porovnávaní vkladajúceho prvku s prvkami zoradenej časti poľa sprava doľava a posunutí každého väčšieho prvku o jednu pozíciu doprava. Metóda je stabilná a chová sa prirodzene.

Metóda vkladania s binárnym vyhľadávaním (binary insert sort) používa na vyhľadanie pozície pre vkladajúci prvok algoritmus binárneho vyhľadávania. Binárne vyhľadávanie musí v prípade

zhodnosti kľúčov nájst' miesto za najpravejším zo zhodných kľúčov. Metóda je stabilná a chová sa prirodzene.

4.3.3 Radenie na princípe rozdelenia

Radenie na princípe rozdelenia (quick sort) pracuje na princípe rozdel' a panuj. Rozdelí pole položiek na dve časti, pričom v ľavej časti sa budú nachádzať položky, ktorých hodnota je menšia ako hodnota zvolenej položky a v pravej časti sa budú nachádzať položky, ktorých hodnota je väčšia ako hodnota zvolenej položky. Takto sa bude s rozdeľovaním pokračovať až, kým nebudú rozdelené polia obsahovať len dve položky. Za hodnotu, ktorá rozdeľuje pole na dve časti je najlepšie zvoliť medián daných položiek. Medián je taká hodnota, pre ktorú platí, že polovica hodnôt v poli je menšia ako medián a polovica hodnôt v poli je väčšia ako medián. Pretože stanovenie mediánu je náročné, plní jeho úlohu hodnota ľubovoľného prvku poľa, najčastejšie hodnota prvku zo stredu poľa. Je dokázané, že táto hodnota splní úlohu podobne ako medián.

Rýchlosť quick sortu je daná najmä algoritmom rozdelenia poľa. Algoritmus prechádza pole zľava doprava až kým narazí na hodnotu väčšiu alebo rovnú deliacej hodnote, následne prechádza pole sprava doľava až kým narazí na hodnotu menšiu alebo rovnú deliacej hodnote a nájdené položky zamení. Takto postupuje, až kým sa indexy postupujúce proti sebe neprekrížia.

Quick sort patrí medzi najrýchlejšie metódy radenia polí. Je to metóda nestabilná a pracuje neprirodzene.

4.3.4 Radenie na princípe zlučovania

Predstaviteľom tohto typu radenia je metóda merge sort. K poľu nezoradených prvkov sa pripojí rovnako veľké prázdne pole. Algoritmus postupne prechádza nezoradené pole z oboch strán a hľadá neklesajúce postupnosti. Následne tieto postupnosti spojí do jednej neklesajúcej postupnosti, ktorú vloží zľava do pridaného poľa. Pokračuje ďalej v hľadaní neklesajúcich postupností v pôvodnom a poli, a neklesajúcu postupnosť, ktorá vznikne ich spojením, tento krát vloží do pripojeného poľa sprava. Takto postupuje, až pokiaľ sa indexy prehľadávajúce pôvodné pole nestretnú, respektíve pokiaľ sa pripojené pole nenaplní. Potom si jednotlivé polia vymenia svoju úlohu, pridané pole bude zdrojom postupností a tie sa budú vkladať do pôvodného poľa. Algoritmus radenia pokračuje, až pokiaľ niektoré pole nebude obsahovať iba jednu neklesajúcu postupnosť.

Nevýhodou algoritmu je, že na svoju realizáciu potrebuje dvojnásobok miest, než koľko obsahuje položiek. Metóda merge sort je nestabilná a nechová sa prirodzene.

5 Návrh E-learningového systému

Vytvorenie dobrého E-learningového systému si vyžaduje dôkladnú analýzu požiadaviek na daný systém a následne podrobný návrh každej časti systému. Systém bude obsahovať administrátorské rozhranie pre správu celého systému, kurzov a užívateľov, rozhranie supervisora, pre správu študentov a študentské rozhranie umožňujúce vzdelávanie a testovanie vedomostí. Následne pre vytvorený systém pripravím a spracujem kurz Algoritmy. Aby bol kurz prístupný študentom aj zo zahraničia, ktorí nemusia ovládať český jazyk, bude celý spracovaný v anglickom jazyku. Kurz bude obsahovať učivo z oblasti abstraktných dátových štruktúr, vyhľadávacích a radiacich algoritmov. Takisto pripravím databázu otázok z oblasti algoritmov, ktorá preskúša nadobudnuté vedomosti študentov. Prvým krokom v návrhu každého E-learningového systému je výber formátu učenia.

5.1 Výber formátu

Navrhovaný systém má byť určený ako doplnkový prostriedok ku klasickej výuke kurzu Algoritmy a najmä od toho závisí výber formátu vzdelávania systému. Vzhľadom k tomu, že systém nemá študenta priamo hodnotiť, nútiť ho prechádzať celú látku kurzov a robiť všetky testy, je vhodné zvoliť neformálnu formu vzdelávania. Tým sa vzdelávanie študentov stane dobrovoľné, nebudú ničím nútení sa mu venovať a nebudú podstupovať stres a obavy z prípadného neúspechu.

Aby bolo vzdelávanie študentom dostupné kedykoľvek bez obmedzenia je potrebné zvoliť asynchrónnu formu vzdelávania. To umožní študentom prístup k informáciám vtedy keď to potrebujú a v podobe aká sa im v danej situácii najviac hodí. Nebudú závislí na pracovných hodinách učiteľa.

Systém, by mal podporovať rôzne druhy vzdelávania a bude iba na študentovi, ktorý z druhov si vyberie. Mal by študentovi umožniť nechať sa viesť preberanou látkou, usmerňovať ho tak, aby prichádzal k informáciám v logickom poradí. Mal by mu poskytovať prehľady preberanej látky a zhrnutia. Po každej prebranej časti by mu mal poskytnúť možnosť overenia nadobudnutých vedomostí vo forme testu a v prípade, že má v prebranej látke nedostatky, mal by mu poskytnúť možnosť priameho sprístupnenia látky, ktorej sa nedostatky týkajú.

Na druhú stranu by systém mal umožniť študentom, ktorí nemajú záujem o komplexnú výuku, možnosť rýchlej navigácie v informáciách týkajúcich sa daného kurzu, ktorá ich v čo najkratšom čase odkáže priamo na miesto, kde nájdu potrebné informácie. To zamedzí zbytočnému zdržiavaniu študentov a zvýši efektivitu ich štúdia.

5.2 Konceptuálny návrh

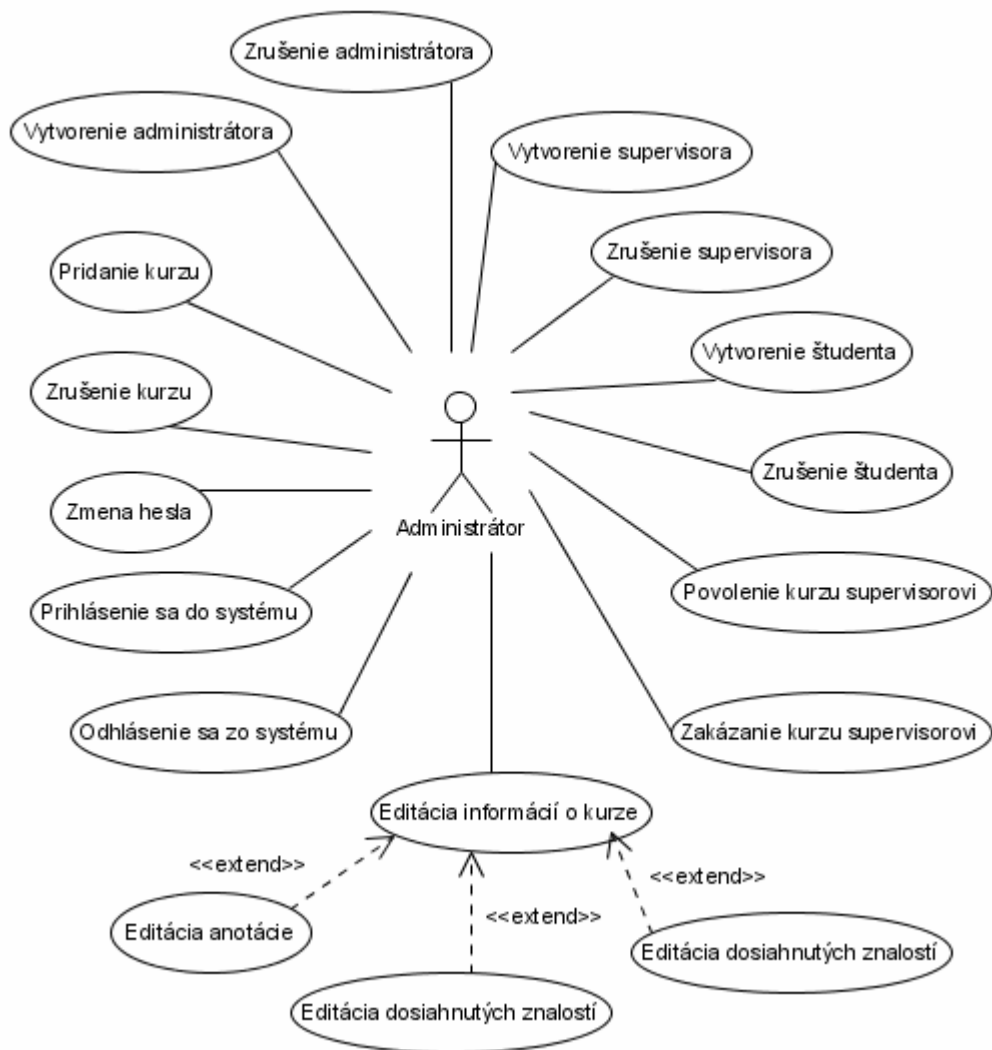
Ďalším krokom vedúcim k lepšiemu pochopeniu činnosti systému je vytvorenie konceptuálneho návrhu. Tento návrh popisuje štruktúru systému, role jednotlivých druhov užívateľov a ako sa bude systém správať. Súčasťou konceptuálneho návrhu je tiež návrh kurzu a činností spojených so štúdiom.

5.2.1 Návrh systému

Celý systém bude rozdelený na tri základné časti, každá z nich reprezentuje jeden druh užívateľa a to administrátora, supervisora alebo študenta. Prístup do každej sekcie bude zabezpečený pomocou prihlasovacieho mena a hesla. Celý systém bude vytvorený v anglickom jazyku, čím sa stane praktickým pre vzdelávanie zahraničných študentov, ale zároveň bude podporovať aj vzdelávanie ostatných študentov v anglickom jazyku a anglickej terminológii, ktoré sú neoddeliteľnou súčasťou oboru informatiky.

5.2.1.1 Administrátor

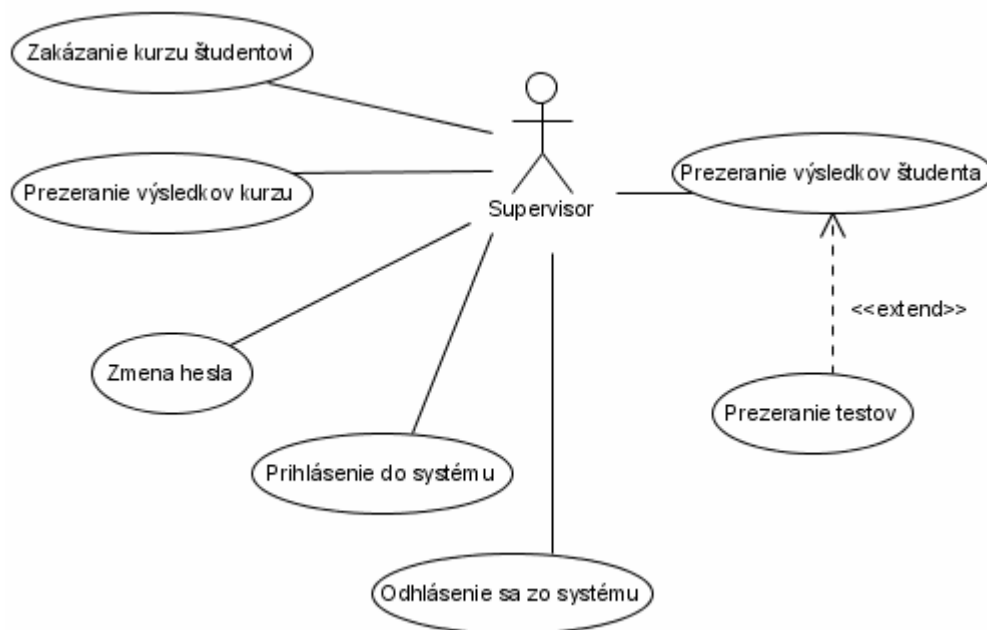
Úlohou administrátora je správa systému, užívateľov a kurzov. Má možnosť pridávania a mazania užívateľov a kurzov. Takisto má možnosť editovať informácie o kurzoch, priradovať kurzom supervisorov a editovať osobné údaje o užívateľoch. Plný popis funkcie administrátora je vidieť na nasledujúcom diagrame prípadov použitia.



Obr. 5.1: Diagram prípadov použitia pre administrátora

5.2.1.2 Supervisor

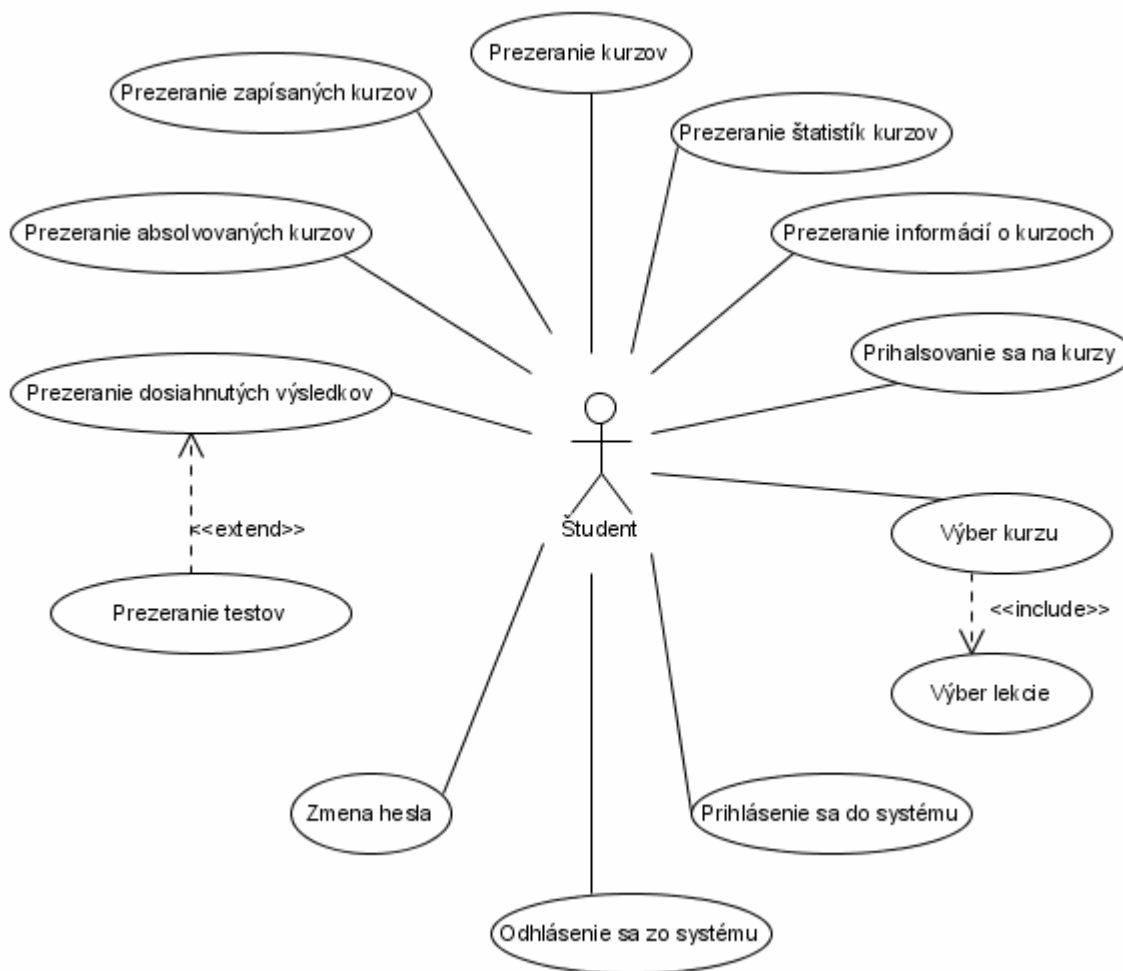
Supervisor bude mať v systéme hlavne pozorovaciu úlohu. Supervisorovi bude priradený zoznam kurzov, na ktoré má dohliadať. Má možnosť prehliadania výsledkov testov všetkých študentov daných kurzov a takisto môže študentov z kurzu vyhodiť.



Obr. 5.2: Diagram prípadov použitia pre supervisory

5.2.1.3 Študent

Navrhovaný systém je orientovaný hlavne na študenta a preto má študent aj najviac možností v rámci systému. Má prístup k popisu všetkých kurzov, k štatistikám ich výsledkov, môže sa registrovať na kurzy a študovať ich. Počas celého štúdia kurzu a aj po dokončení kurzu má prístup k všetkým svojim výsledkom a môže neobmedzene študovať daný kurz a robiť testy. Podrobný popis funkcií študenta vidíme na nasledujúcom diagrame.



Obr. 5.3: Diagram prípadov použitia pre študenta

Po tom, čo bol popísaný návrh štruktúry E-learningového systému bude v ďalšej časti popísaný návrh samotného kurzu Algoritmov.

5.2.2 Návrh kurzu

Podobne ako systém, tak aj kurz Algoritmov bude celý v angličtine. Celý kurz bude rozdelený na štyri lekcie a to:

1. Úvod do algoritmov (Introduction to algorithms)
2. Abstraktné dátové štruktúry (Abstract data structures)
3. Vyhľadávacie algoritmy (Searching algorithms)
4. Radiace algoritmy (Sorting algorithms)

Každá lekcia sa skladá z preberanej látky a z testu overujúceho stupeň pochopenia látky. Navyše bude k dispozícii záverečný test zo všetkých lekcii.

Každá lekcia je ďalej rozdelená na menšie sekcie, z ktorých každá sa dá chápať ako samostatný celok v rámci vzdelávania. Súčasťou všetkých lekcii bude úvodná strana, ktorá bude obsahovať anotáciu danej lekcie a zoznam tém (sekcii), ktorým sa bude lekcia venovať. Záver každej lekcie bude

tvoriť zhrnutie znovu zdôrazňujúce to najdôležitejšie z práve prebranej lekcie a opätovné vymenovanie všetkých tém danej lekcie.

Aby bola výuka pomocou navrhovaného systému kvalitná, budú textové informácie o predmete Algoritmy dopĺňané vysvetľujúcimi obrázkami a najmä zdrojovým kódom v jazyku C, ktorý bude implementovať jednotlivé dátové štruktúry, operácie alebo algoritmy. Preto je potrebnou prerekvizitou pre štúdium kurzu aspoň základná znalosť jazyka C.

Testy budú pozostávať z rôznych druhov otázok, tak aby pokryli čo najväčšiu škálu preberanej látky. Náročnosť otázok bude prispôbená tomu, aby vynikajúce výsledky dosahovali len tí najlepší študenti, ale zároveň, aby bodové zisky slabších študentov nepôsobili demotivujúco, ale naopak, aby ich povzbudzovali do ďalšieho štúdia.

5.3 Návrh databázy

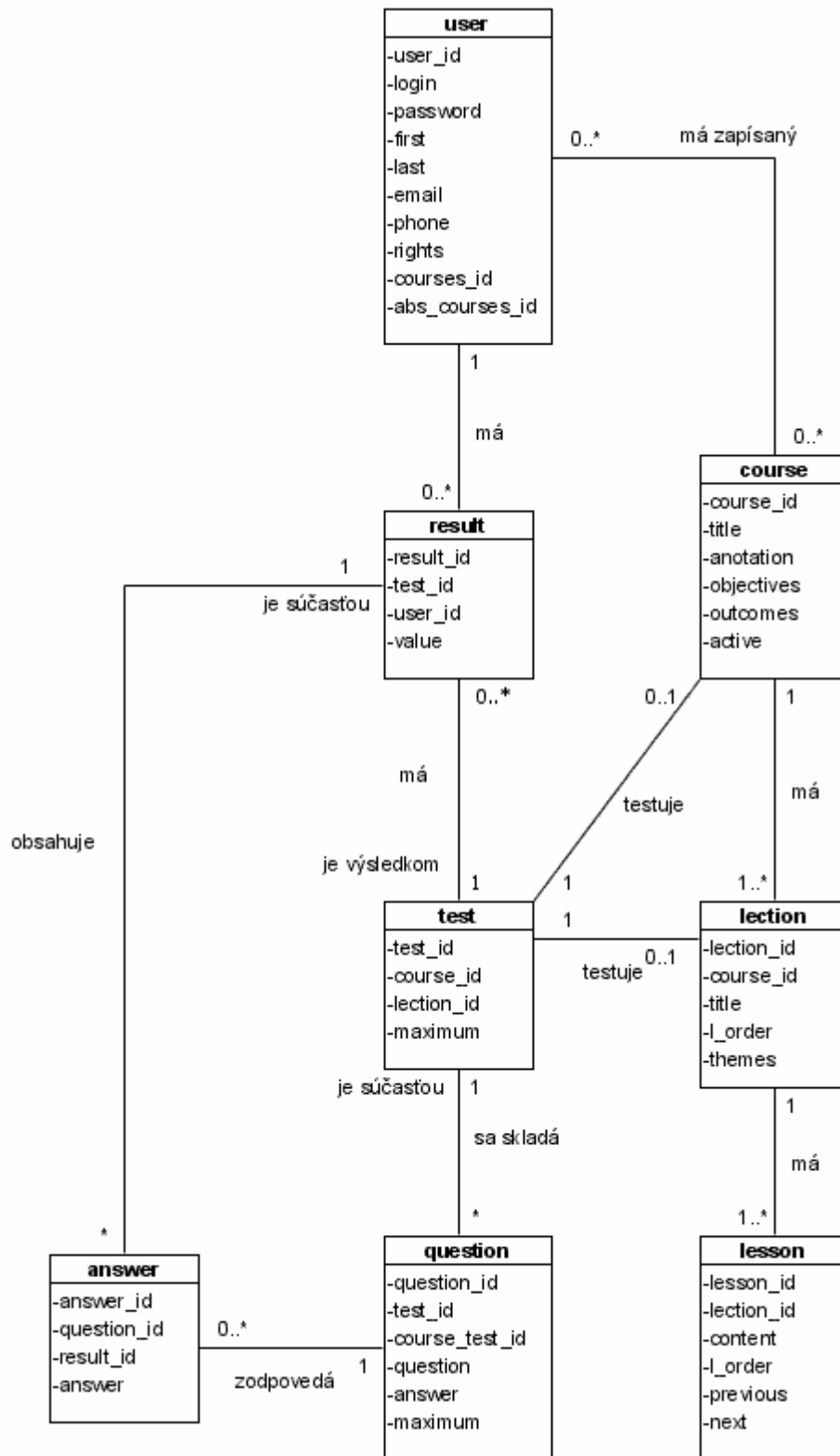
Na základe analýzy požiadaviek na systém a konceptuálneho návrhu systému som vytvoril návrh databázy pre E-learningový systém. Pre to aby systém správne fungoval a implementoval potrebné funkcie bude databáza pozostávať z ôsmich tabuliek:

- user
- course
- lection
- lesson
- test
- question
- answer
- result

V nasledujúcej časti budú vysvetlené vzťahy medzi jednotlivými tabuľkami pomocou ER diagramu.

5.3.1 ER diagram

Entitno-relačný diagram je základným diagramom pri návrhu dátových modelov určených pre relačné databázové systémy. ER diagram je konceptuálny diagram, ktorý berie reálny svet ako entity s danými medzientitnými vzťahmi. ER diagram slúži na vizuálnu prezentáciu dátových objektov.



Obr. 5.4: ER diagram E-learningového systému

Výhody ER diagramu sú:

- dobre spolupracuje s relačným modelom. Jeho konštrukcia je ľahko transformovateľná do relačných tabuliek.

- je ľahko pochopiteľný, preto je možné ho používať aj na prezentáciu modelu koncovým užívateľom.
- diagram môže byť použitý pre návrh implementácie dátového modelu v danom databázovom systéme.

ER diagram zobrazuje dáta v pokojnom stave, nezobrazuje operácie, ktoré sa budú nad dátami v databáze vykonávať. Výsledný ER diagram E-learningového systému, ktorý vznikol na základe návrhu systému je zobrazený na obrázku 5.4.

5.3.2 Popis tabuliek

V tabuľkách sú primárne kľúče označené pomocou **PK**, cudzie kľúče pomocou **FK** a položky, ktoré musia byť v rámci databázy unikátne pomocou **U**.

5.3.2.1 user

Tabuľka user slúži na ukladanie všetkých užívateľov systému, ich prihlasovacích a osobných údajov ako aj ich kurzov.

user			
stĺpec		typ	popis
user_id	PK	int	Primárny kľúč, jednoznačná identifikácia užívateľa
login	U	varchar	Prihlasovacie meno užívateľa
password		varchar	Heslo užívateľa
first		varchar	Krstné meno užívateľa
last		varchar	Priezvisko užívateľa
email		varchar	Užívateľov email
phone		varchar	Užívateľov telefón, nepovinná položka
rights		int	Práva užívateľa, určujú či je užívateľ administrátor (1), supervisor (2) alebo študent (3)
courses_id		varchar	Zoznam identifikačných čísel zapísaných kurzov oddelených čiarkami (týka sa supervisory a študenta)
abs_courses_id		varchar	Zoznam identifikačných čísel absolvovaných kurzov oddelených čiarkami (týka sa študenta)

Tabuľka 5.1: Popis stĺpcov tabuľky user

5.3.2.2 course

Tabuľka course obsahuje všetky kurzy, aj aktívne, aj neaktívne, ktoré sú v systéme. Tabuľka obsahuje položky title, anotation, objectives a outcomes, v ktorých sú uložené všetky všeobecné informácie o kurzoch. Tieto informácie budú dostupné všetkým užívateľom k nahliadnutiu

5.3.2.3 lection

Tabuľka lection obsahuje všetky informácie o lekciách.

lection			
stĺpec		typ	popis
lection_id	PK	int	Primárny kľúč, jednoznačná identifikácia lekcie
course_id	FK	int	Cudzí kľúč, určuje, ku ktorému kurzu lekcia patrí
title		varchar	Názov lekcie
l_order		int	Lokálne poradie lekcie v rámci kurzu
themes		text	Obsahuje vymenované témy a číslo určujúce lokálne poradie prvej obrazovky týkajúcej sa danej témy. Jednotlivé témy sú oddelené čiarkou (téma=l_order, ...). Služi na vytvorenie rýchlej navigácie v kurzoch.

Tabuľka 5.2: Popis stĺpcov tabuľky lection

5.3.2.4 lesson

Tabuľka lesson služi na ukladanie študijnej látky kurzov a lekcí, sú v nej uložené všetky študijné informácie.

lesson			
stĺpec		typ	popis
lesson_id	PK	int	Primárny kľúč, jednoznačná identifikácia obrazovky
lection_id	FK	int	Cudzí kľúč, určuje, ku ktorej lekcii obrazovka patrí
content		text	Obsah obrazovky - študijné materiály vo forme pripravenej na zobrazenie, formátované pomocou html kódu
l_order		int	Číslo slúžiace na určenie náväznosti obrazoviek
previous		int	Číslo náväznosti predchádzajúcej obrazovky, pre úvodnú obrazovku lekcie rovné 0
next		int	Číslo náväznosti nasledujúcej obrazovky, pre poslednú obrazovku lekcie rovné 0

Tabuľka 5.3: Popis stĺpcov tabuľky lesson

5.3.2.5 test

Tabuľka test obsahuje všetky testy a ich priradenie ku kurzu respektíve k lekcii.

test			
stĺpec		typ	popis
test_id	PK	int	Primárny kľúč, jednoznačná identifikácia testu
course_id	FK	int	Cudzí kľúč, určuje, ku ktorému kurzu test patrí, ak test patrí k lekcii, je hodnota rovná 0
lection_id	FK	int	Cudzí kľúč, určuje, ku ktorej lekcii test patrí, ak test patrí ku kurzu, je hodnota rovná 0
maximum		int	Maximum bodov, ktoré je možno v teste dosiahnuť

Tabuľka 5.4: Popis stĺpcov tabuľky test

5.3.2.6 question

Tabuľka question obsahuje otázky a odpovede ku všetkým testom v databáze.

question			
stĺpec		typ	popis
question_id	PK	int	Primárny kľúč, jednoznačná identifikácia otázky
test_id	FK	int	Cudzí kľúč, určuje, ku ktorému testu otázka patrí
course_test_id	FK	int	Cudzí kľúč, určuje, ku ktorému konečnému testu lekcie otázka patrí
question		text	Znenie otázky, spolu s možnými odpoveďami, formátované pomocou html kódu
answer		text	Číslo správnych odpovedí oddelené čiarkami
maximum		int	Maximum možných bodov za otázku

Tabuľka 5.5: Popis stĺpcov tabuľky question

5.3.2.7 answer

Tabuľka answer obsahuje všetky odpovede užívateľov na otázky z testov. Položky question_id a result_id sú cudzími kľúčmi, ktoré viažu odpoveď k otázke respektíve výsledku. Poslednou položkou tabuľky je answer, ktorá obsahuje čísla odpovedí oddelené čiarkou.

5.3.2.8 result

Tabuľka result obsahuje všetky výsledky študentov z jednotlivých testov. Položky test_id a user_id sú cudzími kľúčmi, ktoré viažu výsledok k testu a užívateľovi. Tabuľka obsahuje aj položku value obsahujúcu počet získaných bodov študenta.

V kapitole bol podrobne popísaný návrh databázy, pomocou ER diagramu boli popísané relácie medzi tabuľkami a následne bol popísaný formát a význam dát všetkých položiek tabuliek.

6 Implementácia

Na základe poznatkov získaných v predchádzajúcich etapách vývoja som implementoval E-learningový systém pre výuku Algoritmov. V tejto časti popisujem najdôležitejšie časti procesu implementácie systému.

6.1 Použité programovacie technológie

V nasledujúcej kapitole stručne popíšem niektoré programovacie technológie, ktoré som použil pri tvorbe diplomovej práce.

6.1.1 PHP

PHP je rozšírený univerzálny skriptovací jazyk, ktorý je obzvlášť vhodný pre vývoj dynamických webových aplikácií. Vytvorené bolo v roku 1994, je to OpenSource produkt, čo znamená že je voľne šíriteľný s voľne dostupnými zdrojovými kódmi, ktoré môžeme v prípade potreby upravovať, a distribuovať ďalej.

Samotný kód PHP sa umiestňuje do HTML stránky, medzi tagy `<? a ?>` značiace začiatok a koniec časti kódu písaného v jazyku PHP. Pre funkčnosť kódu je potrebné zmeniť koncovku súboru `html` na koncovku `php`. PHP kód sa prevedie vždy, keď má byť daná stránka zobrazená. Kód sa interpretuje na webovom servery a generuje výstup viditeľný pre užívateľa už nie v PHP, ale napríklad formou HTML. Cieľom PHP teda bolo uľahčiť vývoj jednoduchých webových aplikácií, u ktorých sa nevyplatí nasadiť sofistikované a drahé aplikačné prostredie [13].

6.1.1.1 Smarty

Smarty je šablónovací systém vytvorený použitím skriptovacieho jazyka PHP. Jeho cieľom je oddeliť aplikačnú logiku od obsahu prezentácie. Oddelenie PHP kódu od HTML kódu robí programy prehľadnejšími a lepšie modifikovateľnými. Zmena algoritmov v PHP si nevyžaduje priamu zmenu v HTML kóde a naopak zmena zobrazovaných šablón si nevyžaduje zásah do PHP [15].

Šablónovací stroj taktiež umožňuje použitie riadiacich štruktúr, cyklov, vstavaných funkcií pre prácu s reťazcami, časom a pod. Použitie funkcií v prezentačnej vrstve nie je obmedzené a to vďaka tzv. pluginom. Niekoľko ich je naprogramovaných v základnej distribúcii, ostatné je možné doprogramovať.

Smarty poskytuje možnosť cacheovania všetkých častí webovej stránky. Programátor si vyberá, či bude šablónová funkcia registrovaná ako cacheovateľná alebo ako ncacheovateľná.

Smarty spoločne s ďalšími funkciami umožňuje efektívne objektové programovanie v PHP, ktoré je prehľadnejšie ako klasické programovanie v PHP. Umožňuje jednoduchú modifikáciu kódu a to iba tých častí, ktorých sa zmena priamo týka.

6.1.2 MySQL

MySQL je rýchly, robustný, relačný databázový systém, umožňuje efektívne ukladať, hľadať a získavať dáta. Server MySQL je viac užívateľský a viacvláknový server, teda umožňuje, aby bolo k databáze pripojených zároveň viac užívateľov. Ako dotazovací jazyk k databáze používa jazyk SQL (Structured Query Language), ktorý je celosvetovo považovaný, za štandardný dotazovací jazyk pre databáze.

Hlavné prednosti MySQL sú jeho vysoká výkonnosť, rýchlosť, spoľahlivosť a prispôsobivosť. Je to najčastejšie používaný voľne šíriteľný databázový systém, ktorý sa v praxi využíva najmä v prepojení s PHP.

Samotné MySQL neponúka užívateľovi žiadne uspokojujúce rozhranie, až na príkazový riadok, kde sa dajú zadávať SQL dotazy, preto bol vytvorený voľne dostupný balík phpMyAdmin, čo je program písaný v jazyku PHP, ktorý sa automaticky spúšťa na servery, a zahŕňa užívateľské rozhranie pre prehľadnejšie a ľahšie spravovanie miestnych databáz.

6.1.2.1 Spolupráca PHP s MySQL

PHP a MySQL sa často využívajú spoločne pri realizovaní zložitejších internetových aplikácií, ktoré zahŕňajú mimo iného aj spoluprácu s databázou. PHP má v sebe podporu príkazov SQL, umožňuje pripojiť sa pomocou kódu napísaného v PHP na daný databázový server, vybrať potrebnú a požadovanú databázu, a do PHP kódu vložený SQL príkaz previesť. Dáta získané SQL príkazom sa potom pomocou PHP na strane serveru použijú napr. pri zobrazovaní na výstup vo forme HTML tabuľky.

6.1.3 JavaScript

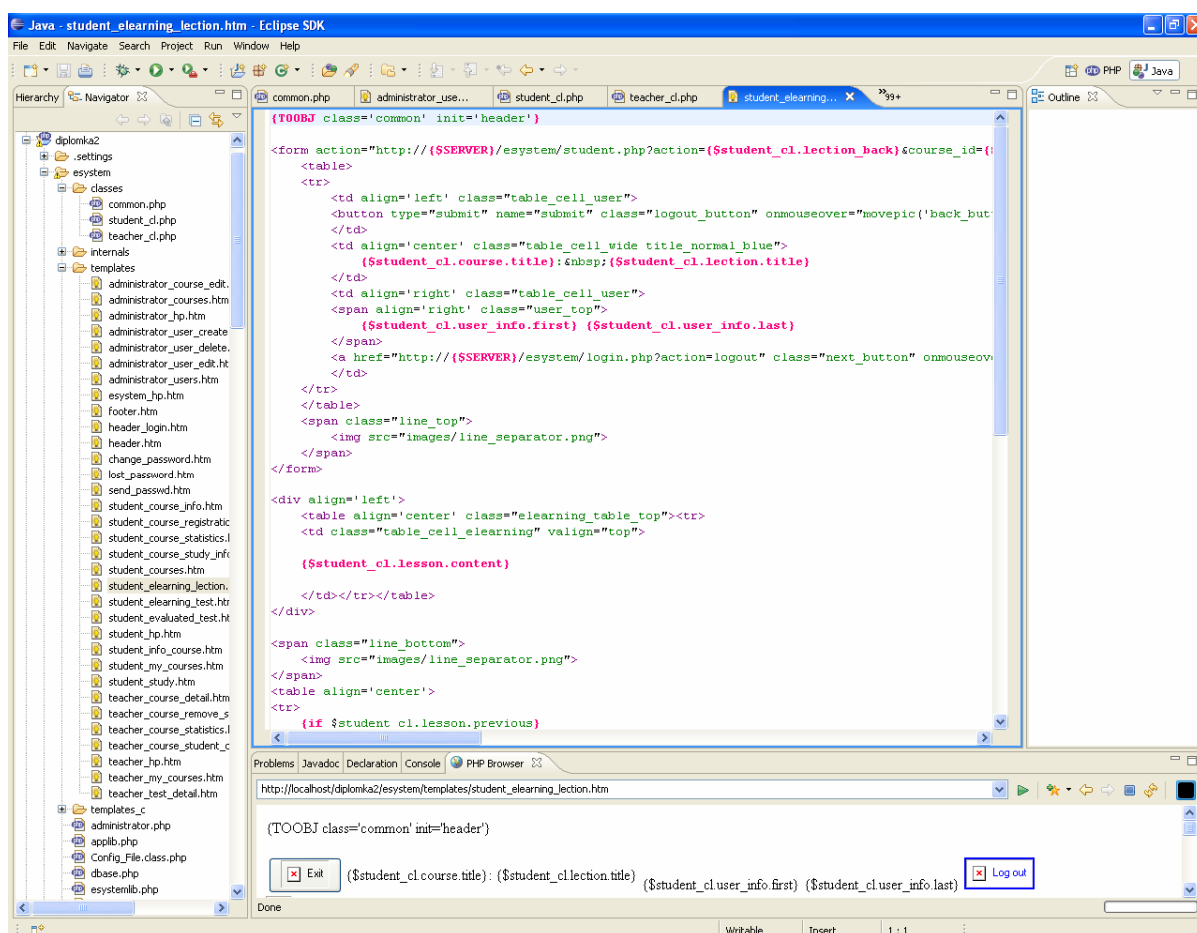
JavaScript je interpretovaný programovací jazyk so základnou objektovo orientovanou koncepciou. Klientská verzia tohto jazyka je súčasťou väčšiny všeobecne rozšírených prehliadačov ako sú napr. Internet Explorer alebo Mozilla Firefox.

Jadro jazyka (t.j. časť univerzálna, nezávislá od prehliadačov) je syntakticky veľmi podobná C/C++ alebo Jave. Podobnosť však končí na úrovni syntaxe. JavaScript je jazykom, ktorý má potlačenú typovú kontrolu. Ďalej ide o interpretovaný jazyk, ktorý preberá veľa myšlienok z jazyku Perl. Príklady prevzatých pojmov sú regulárne výrazy alebo práca s poľami [14].

6.1.4 Eclipse

Pre vývoj aplikácie bolo potrebné zvoliť vhodné vývojové prostredie. Muselo spĺňať niekoľko základných kritérií. V prvom rade muselo byť voľne dostupné, bez akýchkoľvek obmedzení. Prostredie muselo byť schopné rozoznávať jazyky HTML a PHP. Malo by poskytovať užívateľsky príjemné a prehľadné ovládanie, s funkciami uľahčujúcimi programovanie.

Všetky tieto podmienky spĺňa prostredie Eclipse. Eclipse je pôvodne vytvorený pre vývoj Java aplikácií a HTML stránok, ale po pridaní pluginov pre podporu PHP sa z neho stáva veľmi vhodný nástroj pre vývoj dynamických webových stránok. Prostredie obsahuje prehľadné menu všetkých súborov projektu, ktoré umožňuje rýchlu navigáciu. Po otvorení súboru poskytne pre rýchlu navigáciu v zdrojovom kóde zoznam všetkých funkcií a vlastností. V dolnej časti obsahuje okno priamo zobrazujúce vzhľad webovej stránky aktuálne otvoreného súboru. Pomocné okná sú usporiadané tak, že aj napriek ich veľkému počtu zostáva pomerne veľký priestor pre písanie kódu. Zdrojový kód je automaticky štruktúrovaný a farebne odlíšený.



Obr. 6.1: Vývojové prostredie Eclipse

6.2 Implementácia systému

E-learningový systém bol implementovaný tak, aby bol plne funkčný pod internetovými prehliadačmi Internet Explorer 6, 7, Mozilla Firefox a Opera 8. Aby bol systém prispôsobený užívateľom s priemerným technickým vybavením, je systém optimalizovaný pre rozlíšenie 1280 x 1024 pixelov. Plne funkčný systém je dostupný na URL: http://www.stud.fit.vutbr.cz/~xhroza00/diplomova_praca. Pre testovanie sú vytvorené užívateľské účty administrátor: administrator/administrator, supervisor: supervisor/supervisor a študenti student1/student1, student2/student2.

Po spustení hlavnej stránky najprv prebehne inkudovanie niekoľkých súborov. Ako prvý sa nainkluduje Session.php, ktorého jedinou úlohou je spustenie session. Ako ďalšie sa inkudujú súbory global_vars.php, dbase.php, applib.php, esystemlib.php a MainSystem.php.

Súbor global_vars.php nadefinuje premenné SERVER a WORKDIR, obsahujúce URL serveru a cestu k pracovnému adresáru, v ktorom sa nachádzajú všetky súbory systému. Tieto premenné sú potom v systéme využívané pre jednoduchší a prehľadnejší zápis cieľov odkazov a formulárov. Súbor dbase.php je trieda implementujúca základné operácie s databázou. Konštruktor triedy sa pokúsi pripojiť k databáze na základe vopred nastavených prihlasovacích parametrov. Ak je prihlásenie úspešné, pokračuje ďalej v činnosti, ak sa pripojenie nepodarí, ukončí činnosť s chybovým oznámením.

Súbory applib.php a esystemlib.php obsahujú funkcie implementujúce všetky operácie prebiehajúce v systéme. Súbor applib.php obsahuje funkcie, ktoré nepoužívajú priamy prístup k databáze. Ide prevažne o funkcie uľahčujúce programovanie, alebo pomocné funkcie. Obsahuje funkciu print_a, ktorá je základným prvkom ladenia programu. Funkcia prehľadne zobrazí v tabuľke obsah akéhokoľvek druhu premennej. Ďalšími významnými funkciami knižnice sú funkcie pre spracovanie výsledkov databáze. Tie predspracujú získané dáta tak, aby s nimi bolo možné ďalej jednoduchšie pracovať.

Súbor esystemlib.php na druhú stranu obsahuje funkcie priamo súvisiace so systémom. Väčšina funkcií súboru má za úlohu získavanie alebo vkladanie dát do databáze. Medzi významné a náročnejšie funkcie patria hlavne funkcie spracovávajúce a vyhodnocujúce testy.

Posledný inkudovaný súbor MainSystem.php najprv nainkluduje súbor OBJ.php, ktorý slúži ako rozhranie pre ostatné triedy, implementujúce prácu so šablónami a ich zobrazovanie. Následne sú nainkludované súbory týkajúce sa šablónovacieho systému smarty. Tento systém umožní oddelenie prezenčnej vrstvy od aplikačnej, umožní opakované použitie niektorých častí na viac rôznych účelov a sprehľadní zdrojový kód. Súbor obsahuje funkciu DOSmarty, ktorá na základe prednastavených parametrov vytvorí inštanciu jedného z objektov reprezentujúcich jednotlivé typy užívateľov, zavolá jednu z metód daného objektu a určí šablónu, ktorá sa má zobrazit'.

Systém obsahuje tri základné triedy, každú pre jeden typ užívateľa. Triedy sú uložené v adresári classes. Funkcie spoločné pre všetkých užívateľov sú spojené v jednej triede spolu

s funkciami administrátora. Všetky tri triedy sú vydedené z triedy OBJ popísanej vyššie. Triedy implementujú funkcie, ktoré sa volajú pred zobrazením šablón, tie sa nazývajú smarty funkcie, a funkcie, ktoré sú volané zo šablón, aby do nich doplnili zdrojový kód, ktorý je spoločný pre viacero šablón, tie sa nazývajú rendrovacie funkcie. Smarty funkcie majú za úlohu získať z databáze dáta potrebné pre danú stránku a pripraviť ich pre zobrazenie. Tieto dáta sú následne pomocou smartyho predané do šablóny a tam sa pomocou smarty tagov zobrazujú.

Vďaka oddeleniu prezenčnej vrstvy od aplikačnej vrstvy, šablóny obsahujú iba HTML kód a jednoduché smarty tagy. Smarty umožňuje pomocou rendrovacích funkcií kdekoľvek v kóde šablóny začleniť inú šablónu. Toto v systéme využívam najmä na pridanie hlavičky a pätičky HTML stránky, ktoré sú takmer u všetkých šablón spoločné. Takéto pridanie inej šablóny zaberá iba jeden riadok zdrojového kódu a zobrazovaná šablóna teda bude obsahovať iba kód, ktorý sa jej priamo týka. Smarty tiež umožňuje kdekoľvek v šablóne sprístupniť dáta, ktoré boli k šablóne priradené. Stránky teda zostávajú dynamické bez obmedzení. Ďalšou výhodou je možnosť použitia ríadiacich konštrukcií a cyklov v šablónach, syntax je veľmi podobná syntaxi PHP, preto nie je problém osvojiť si prácu s nimi. Smartyho je takisto možné rozšíriť o ďalšie funkcie, ktoré budú po zaregistrovaní voľne prístupné v šablónach. Týmto spôsobom je možné dynamicky vytvárať napríklad comboboxy a checkboxy, ktoré sa v šablóne jednoducho vytvoria priradením premennej, v ktorej sú už predspracované dáta pre zobrazenie v danom formáte.

Celé riadenie systému, v ktorom sa rozhoduje, ktorá šablóna a objekt budú vybrané je rozdelené do štyroch súborov. Súbor login.php je určený pre časti systému spoločné pre všetkých užívateľov a ostatné tri súbory, administrator.php, teacher.php a student.php, sú určené vždy pre jeden typ užívateľa. Riadenie je realizované pomocou switchu, ktorý na základe parametru action z premennej GET vyberie správnu akciu. V každej akcii sa naplnia položky globálnej premennej global_komponents. Položka action obsahuje názov smarty funkcie zavolanej pred zobrazením šablóny, ktorej meno je uložené v položke template. Posledná nevyhnutná položka premennej je položka object, ktorá obsahuje názov objektu, v ktorom sa nachádza daná funkcie. Po nastavení parametrov je zavolaná funkcia DOSTsmarty, ktorá spustí potrebný kód.

6.2.1 Systém

Hlavná stránka systému obsahuje formulár pre prihlásenie do systému (Obr. 6.1). Ak užívateľ do formuláru zadá nesprávne prihlasovacie údaje, systém ho na to upozorní zvýrazneným chybovým oznámením. Pre prípad, že užívateľ heslo zabudol, obsahuje formulár aj odkaz na stránku, ktorá mu na základe zadaného loginu vygeneruje nové heslo a automaticky ho zašle na email, ktorý má uložený v databáze systému.

The image shows a login form for an 'E-Learning' system. In the top left corner, there is a small square icon with a blue background, featuring a black graduation cap (mortarboard) and a stack of books. To the right of this icon, the text 'E - Learning' is written in a large, blue, stylized font. Below the title, a red error message reads 'Log in was not successful, please try again'. Underneath the error message, there are two input fields: the first is labeled 'Login:' and the second is labeled 'Password:'. At the bottom left of the form, there is a blue underlined link that says 'Forgot your password?'. At the bottom right, there is a blue button with the text '<OK>'. The entire form is enclosed in a blue rectangular border.

Obr. 6.2: Formulár pre prihlásenie do systému

Po úspešnom prihlásení systém na základe údajov uložených v databáze zistí, či užívateľ je administrátor, supervisor alebo študent. Do session uloží základné údaje o užívateľovi, ako sú jeho identifikačné číslo a jeho práva. Tieto údaje slúžia na zabezpečenie systému proti neoprávnenému používaniu. Umožňujú užívateľovi prístup iba do sekcií, ktoré sú pre neho určené a zamietnu mu prístup do sekcií určených pre iný typ užívateľa.

Každý typ užívateľa má hneď na svojej hlavnej stránke možnosť zmeny svojho hesla. Zmena prebieha tradičným spôsobom, kedy je zadané staré heslo, nové heslo a potvrdenie nového hesla. Zmena hesla prebehne iba v prípade, že staré heslo je zhodné s heslom uloženým v databáze a nové a potvrdené heslo sú rovnaké. Systém z bezpečnostných dôvodov neumožňuje zadať heslo, ktoré je kratšie ako päť znakov.

Všetci užívatelia majú neustále, nezávisle na zanorení v systéme, možnosť odhlásenia sa. Po odhlásení sa zo session zmažú všetky údaje o užívateľovi a zobrazí sa hlavná stránka systému s formulárom pre prihlásenie.

6.2.2 Administrátor

Administrátor je užívateľ s najväčšími právomocami v systéme. Má na starosti správu všetkých kurzov a užívateľov.

Administrátor má možnosť pridávať kurzy do systému, odoberať kurzy zo systému a editovať kurzy. Pridávanie aj odoberanie spočíva v nastavení aktivity kurzu na 1 resp. 0. Môže teda iba aktivovať kurz, ktorý je už vytvorený v databáze. Samotné vytvorenie kurzu, lekcii, učiva a testových otázok musí vykonať programátor a následne ho celý vložiť do databáze. Takisto odoberanie kurzu nezmaže žiadne informácie o kurze z databáze.

Administrátor má možnosť priamo cez webové rozhranie editovať anotáciu kurzu, ciele kurzu a získané znalosti po ukončení kurzu. Takisto má možnosť pridať resp. odobrať kurz supervisorom. Systém pre každý kurz vygeneruje zoznam supervisorov, ktorý daný kurz nemajú priradený, tých ponúkne na pridanie, a zoznam supervisorov, ktorým daný kurz je priradený, tých ponúkne na odobratie.

Správa užívateľov spočíva vo vytvorení, zmazaní a editovaní užívateľov všetkých druhov. Pri vytváraní užívateľa systém požaduje zadanie prihlasovacieho mena, hesla, mena, priezviska a emailu. Práve na tento email bude užívateľovi zaslané nové heslo v prípade, že si o neho zažiada na úvodnej stránke. Poslednou povinnou položkou pri vytváraní užívateľa je typ užívateľa. Systém administrátorovi ponúkne na výber administrátora, supervisora a študenta.

Pri mazaní užívateľa sú pre lepšiu prehľadnosť administrátorovi ponúknuté, tri skupiny užívateľov, každá pre jeden typ. Administrátor má možnosť zmazať ktoréhokolvek užívateľa, okrem seba samého. Po zmazaní užívateľa sú všetky informácie o ňom zmazané z databázy.

Pri editácii užívateľa má administrátor možnosť zmeniť všetky užívateľove údaje, ako aj typ užívateľa.

6.2.3 Supervisor

Supervisor má v systéme prevažne pasívnu úlohu. Má pridelené predmety, ktoré spravuje zo študijného hľadiska. Má k dispozícii štatistiky svojich predmetov (Obr. 6.2), ktoré obsahujú výpis všetkých lekcíí, a teda aj testov, týkajúcich sa daného predmetu. Ku každému testu je uvedený priemerný počet dosiahnutých bodov a maximálny možný počet bodov za test. Aby si užívateľ mohol vytvoriť lepší obraz o dôveryhodnosti štatistík, je pri každom teste uvedený aj počet študentov, ktorí daný test absolvovali. Štatistika tiež obsahuje súhrnné údaje pre predmet ako celok.

Course statistics: Algorithms

Lection title	Average Points	Maximum	Students
Introduction to algorithms	9.3	10	3
Abstract data structures	16	18	3
Searching algorithms	12	16	4
Sorting algorithms	13.7	16	3
Final test	36.7	40	3
Together	87.7	100	3

Obr. 6.3: Štatistiky predmetu Algoritmy

Supervisor má prístup k detailným študijným výsledkom každého študenta. Systém mu poskytuje zoznam študentov daného kurzu, a po stlačení tlačidla sa mu zobrazí detail zvoleného študenta. Detail obsahuje získaný počet bodov zo všetkých testov, ktoré študent absolvoval a celkový počet bodov získaný v kurze. Po kliknutí na lekcii sa supervisorovi zobrazí opravený test študenta so zvýraznenými zaškrtnutými odpoveďami a správnymi odpoveďami.

Poslednou funkciou supervisora je možnosť vylúčenia študenta z kurzu. Vylúčenie môže byť prevedené napríklad na základe porušenia študijných pravidiel, alebo po vylúčení študenta zo školy.

6.2.4 Študent

Časť systému určená pre študentov je z funkčného hľadiska najrozsiahlejšia. Hlavná stránka dáva študentovi na výber zo šiestich volieb, ktorými sú okrem zmeny hesla Dostupné kurzy, Absolvované kurzy, Aktuálne zapísané kurzy, Registrácia kurzov a samotné Štúdium.

Dostupné kurzy pozostávajú zo zoznamu kurzov, ktoré majú v databáze nastavenú položku active na 1. Po kliknutí na kurz sa študentovi zobrazia informácie o kurze: zoznam supervisorov, ktorí na kurz dohliadajú, anotácia, ciele kurzu, získané znalosti, zoznam lekcii kurzu a odkaz na štatistiky kurzu. Štatistiky majú rovnakú podobu ako štatistiky prístupné supervisorovi.

Absolvované kurzy sú kurzy, ktorými študent prešiel v minulosti. Tento odkaz študentovi umožňuje stály prístup k svojim študijným výsledkom, aj po dokončení kurzu. Prehľad študijných výsledkov pozostáva zo zoznamu všetkých lekcii a testov a bodového ohodnotenia každého testu. Prehľad takisto obsahuje odkazy na všetky študentove testy, kde si kedykoľvek môže pozrieť aké chyby v teste spravil.

Aktuálne zapísané kurzy sú obdobou absolvovaných kurzov s tým rozdielom, že študent ešte neabsolvoval všetky testy daného kurzu. Študent si môže prezerat' výsledky všetkých častí kurzu, ktoré absolvoval.

Vo voľbe registrácia kurzu systém ponúkne študentovi na registráciu všetky kurzy, ktoré ešte neabsolvoval alebo nemá zapísané. Hneď po registrácii je kurz študentovi plne dostupný k štúdiu. Nemusí čakať na žiadne potvrdenie registrácie zo strany supervisora.

Posledná voľba, Štúdium, je hlavnou časťou celého implementovaného systému. Systém študentovi ponúkne všetky aktuálne zapísané kurzy a všetky absolvované kurzy. Po zvolení kurzu má študent na výber všetky lekcii daného kurzu, test z každej lekcii a záverečný test kurzu skúšajúci prebranú látku celého kurzu.

6.2.4.1 Lekcia

Obrazovka lekcii je rozdelená na tri základné časti (Obr. 6.3). Horná časť obsahuje v ľavom rohu tlačidlo na opustenie lekcii. V strede sa nachádza názov kurzu a názov lekcii a v ľavej časti sa nachádza meno a priezvisko študenta spolu s tlačidlom pre odhlásenie užívateľa.

Stredná časť je vyhradená pre učivo danej lekcie, kde sa zobrazuje text formátovaný v html načítaný z databáze z tabuľky lesson. Každá stránka učiva je v hornej časti uvedená názvom preberanej témy a pod ním sa nachádza samotné učivo týkajúce sa témy.

The screenshot shows a web page for 'Algorithms: Searching algorithms' by Martin Uhlir. The main content is titled 'BST with sentinel'. It includes a diagram of a binary search tree with a sentinel node and two C code snippets. The diagram shows a tree with root 30, left child 20, right child 40, and further children 12, 34, and 45. A sentinel node is shown at the bottom, with arrows pointing to it from nodes 12, 34, and 45. The code snippets are as follows:

```

//variable item is global
void TInit(link Root){
    item = (link) malloc(sizeof(*item));
    Root = item;
}

//variable item is global
void SearchTree(link Root, TKey k){
    item->Key = K;
    while(Root->Key != K){
        if(Root->Key > K)
            Root = Root->left;
        else
            Root = Root->right;
    }
    return (Root == item);
}

```

At the bottom of the page, there is a navigation bar with links: Overview ~ Introduction ~ Unordered array ~ Ordered array ~ Binary searching ~ BST ~ AVL tree ~ Hashing table ~ Conclusion.

Obr. 6.4: Príklad stránky lekcie

Spodná časť je vyhradená na posúvanie sa po stránkach lekcie. Pre posun o jednu stránku sú k dispozícii dve tlačidlá v ľavom a pravom rohu. Pre rýchlejšiu navigáciu v lekcii, slúžiacu najmä pre študentov hľadajúcich len konkrétnu informáciu sú k dispozícii odkazy na najdôležitejšie témy lekcie, ako aj odkaz na začiatok lekcie a na zhrnutie. Tieto odkazy sa nachádzajú v strede spodnej časti.

Každá lekcia obsahuje úvodnú stranu pozostávajúcu z názvu lekcie, cieľov štúdia a zoznamu najdôležitejších tém lekcie. Bezprostredne za úvodnou stranou nasleduje úvod do problematiky danej lekcie obsahujúci najmä teoretické vysvetlenie pojmov vyskytujúcich sa v lekcii.

Ďalšie strany lekcie sú už priamo venované problematike, ktorej sa lekcia má týkať. Počet strán je závislý na rozsahu preberanej látky. Látka je rozdelená na strany tak, aby sa na každej strane vždy vyskytovalo učivo len z jednej témy a zároveň, aby sa študent nemusel posúvať po strane dolu.

Záver každej lekcie tvorí strana zhŕňajúca najdôležitejšie informácie z práve prebranej lekcie nasledovaná stranou s opätovným vymenovaním najdôležitejších tém lekcie.

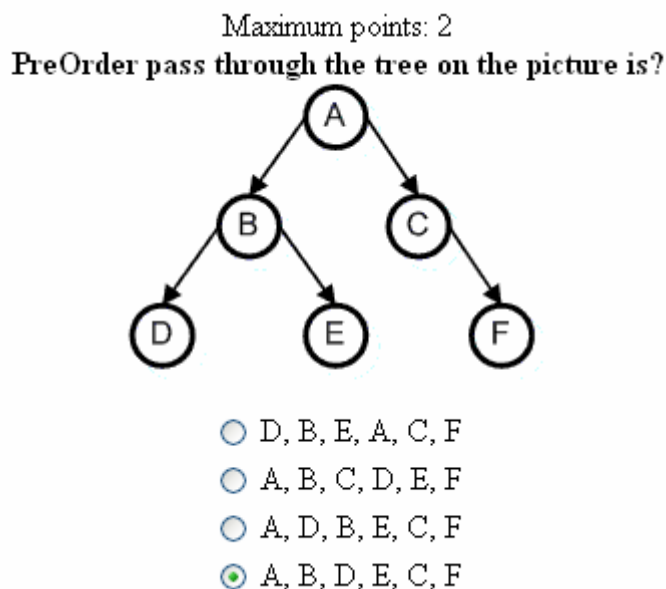
Obsah učiva sa skladá z textu vysvetľujúceho preberanú látku a z obrázkov, ktoré slúžia pre lepšie pochopenie preberanej látky. Keďže vytvorený kurz sa venuje problematike algoritmov, súčasťou výuky sú aj funkcie a algoritmy zapísané v zdrojovom kóde. Pre zápis zdrojového kódu som si vybral jazyk C, ktorý je v súčasnosti najrozšírenejším neobjektovým jazykom.

6.2.4.2 Test

Študent má na overenie si dosiahnutých vedomostí k dispozícii test ku každej lekcii a navyše jeden test z celej lekcie. Študent môže opakovať testy koľkokrát bude chcieť, aj po absolvovaní celého kurzu. Prvý výsledok z každého testu bude uložený do databázy a bude braný ako oficiálny výsledok študenta. Výsledok tohto testu bude započítaný do štatistík a bude študentovi kedykoľvek k dispozícii k nahliadnutiu.

Keďže testy majú slúžiť na overenie si nadobudnutých vedomostí, nie sú nijako časovo obmedzené. Je iba na študentovi ako rýchlo a dôkladne ich bude riešiť.

Na úvod testu sú stručne vysvetlené pravidlá testu. Test sa skladá z dvoch typov otázok. Prvým typom je voľba jednej správnej odpovedi zo štyroch možných. Ak je zaškrtnutá správna odpoveď, študent dostane za otázku plný počet bodov, ak je zaškrtnutá nesprávna odpoveď, alebo nie je zaškrtnutá žiadna odpoveď, študent dostane za otázku nula bodov. Tento druh otázok je implementovaný pomocou Radio buttonov.



Obr. 6.5: Otázka s voľbou jednej správnej odpovedi

Druhým typom otázok je voľba niekoľkých správnych odpovedí zo štyroch možných, tzv. multiple choice. Správna je vždy minimálne jedna odpoveď a maximálne štyri odpovede. Ak sú

zaškrtnuté všetky správne odpovede a ani jedna nesprávna odpoveď, študent dostane za otázku plný počet bodov. Ak študent spraví jednu chybu, teda buď nezaškrtnie jednu správnu odpoveď, alebo zaškrtnie jednu nesprávnu odpoveď, dostane za otázku polovičný počet bodov. V prípade ak spraví dve alebo viac chýb, dostane za otázku nula bodov. Celkový bodový zisk z testu sa získa súčtom získaných bodov za jednotlivé otázky.

Maximum points: 4

Which of following affirmations about hashing table with implicit linking are correct?

- items could not be deleted
- each item contains address of its successor
- the step of implicit link can be defined by another function
- the best size for the table is a prime number

Obr. 6.6: Otázka s voľbou n správnych odpovedí

Každá lekcia obsahuje databázu otázok týkajúcu sa učiva prebraného v tejto lekcii. Podľa počtu bodov za test sa z databázy náhodne vyberie príslušný počet otázok. Testy sa teda stále obmieňajú a študent pri opakovanom spustení testu nedostane tie isté otázky. Tým sa testovanie stáva atraktívnym aj po absolvovaní testu. Študent si takto môže zdokonaľovať svoje vedomosti až dokiaľ nebude schopný akokoľvek zostavený test zodpovedať na plný počet bodov.

Testy obsahujú dvojbodové a štvorbodové otázky. Počet štvorbodových otázok je automaticky zvolený tak, aby počet možných získaných bodov za ne nepresiahol jednu štvrtinu. Ostatné body potrebné k doplneniu maxima bodov za test sú doplnené dvojbodovými otázkami. Počet možných bodov za otázku je vždy uvedený pred znením otázky.

Podobne ako obsah učiva lekcii aj otázky sú troch druhov. Najčastejšie sú klasické textové otázky, ktoré sa priamo pýtajú na prebranú látku. Druhým druhom sú otázky spojené s obrázkom, kedy študent musí nájsť správnu odpoveď na otázku preštudovaním priloženého obrázku. Posledným druhom otázok sú otázky obsahujúce funkciu alebo algoritmus zapísaný v zdrojovom kóde. V tomto prípade má študent doplniť chýbajúci príkaz, alebo podmienku tak, aby algoritmus fungoval správne. Iným variantom je určenie riadku, v ktorom sa nachádza logická chyba, ktorá znefunkčňuje algoritmus.

Maximum points: 2

Choose the the right statement to fill the missing line.

```
void SiftDown(TArr A, int left, int right){
    int i = left, j = 2 * left, Temp = A[i];
    boolean Cont = (j <= right);

    while(Cont){
        if(j < right)
            if(A[j] < A[j+1])
                j++;
        if(Temp >= A[j])
            Cont = false;
        else{
            A[i] = A[j];
            i = j;
            j = 2 * i;
            missing line
        }
    }
    A[i] = Temp;
}
```

- Cont = true;
- swap(A[i], A[j]);
- Cont = (j <= right);
- no line is missing

Obr. 6.7: Otázka so zdrojovým kódom v jazyku C

Po tom čo študent vyplní celý test, klikne na tlačidlo Ukončiť test a test bude ihneď vyhodnotený. Študentovi sa zobrazí výsledok testu spolu s otázkami, zaškrtnutými odpoveďami a správnymi odpoveďami. Správne odpovede budú pre prehľadnosť zvýraznené modrou farbou. Každá otázka tiež obsahuje počet získaných bodov za jej zodpovedanie. Ak bola niektorá otázka zodpovedaná nesprávne, bude sa pod otázkou nachádzať odkaz na učivo, v ktorom bola téma otázky preberaná.

Points gained 0/2

Which of following affirmations about Bubble insert sort are correct?

- Method is stable
- Method has linearithmic complexity
- Method behaves naturally
- Method works in situ

[Go to lesson](#)

Obr. 6.8: Opravená otázka s odkazom na učivo

Ak bol daný test absolvovaný po prvý krát, jeho výsledok sa uloží do databáze a bude študentovi kedykoľvek k dispozícii na prehliadnutie, spolu s otázkami a odpoveďami.

Ak bol práve absolvovaný test posledným neabsolvovaným testom daného kurzu, kurz sa automaticky preradí medzi absolvované kurzy. Študent môže aj naďalej bez obmedzenia prechádzať lekcie alebo riešiť testy.

7 Záver

7.1 Zhodnotenie výsledkov

V práci bola popísaná problematika E-learningu a možnosti jeho využitia vo vzdelávaní súčasnej spoločnosti. Boli popísané základné druhy E-learningu a výhody a nevýhody každého z nich. Podrobne bola popísaná tvorba systému od úplného začiatku až po návrh testovania a údržby systému po jeho vypustení. Takisto boli popísané základné vlastnosti existujúcich kvalitných systémov pre tvorbu E-learningových kurzov, ich možnosti, ktoré ich robia tak úspešnými ako aj niektoré ich nedostatky. Tiež bol poskytnutý stručný náhľad na problematiku algoritmov, ktorá je vhodná pre vytvorenie E-learningového systému.

Bol implementovaný E-learningový systém, spĺňajúci všetky základné požiadavky kladené na takýto systém. Systém umožňuje prehľadnú správu kurzov a užívateľov. Umožňuje sprístupnenie, deaktivovanie kurzov a editáciu informácií o kurzoch. Ďalej umožňuje vytváranie, mazanie a editovanie užívateľov. Systém tiež poskytuje správu študentov a ich výsledkov v podobe rozhrania supervisora. Študentské rozhranie umožňuje prehľadnú orientáciu v prístupných kurzoch a v informáciách o nich. Súčasťou systému sú tiež štatistiky všetkých kurzov. Systém poskytuje užívateľsky príjemné rozhranie pre štúdium kurzov, ktoré študenta systematicky naviguje celým učivom, ale v prípade potreby mu umožňuje rýchly priamy prístup k vybraným informáciám. Znalosti získané štúdiom systém overuje pomocou implementovaných testov.

Bol navrhnutý a vytvorený kurz pre výuku algoritmov v angličtine. Kurz pozostáva zo štyroch lekcí: Úvod do algoritmov, Abstraktné dátové štruktúry, Vyhľadávacie algoritmy a Radiace algoritmy. Ku kurzu bola vytvorená rozsiahla databáza otázok, ktorá preveruje dosiahnuté znalosti študentov.

7.2 Ďalšie možnosti vývoja

E-learningový systém je navrhnutý a implementovaný pre správu a štúdium ľubovoľného počtu kurzov. Preto je vhodným rozšírením systému vytvorenie a následné pridanie ďalších kurzov. Keďže kurz Algoritmy obsahuje a testuje učivo v podobe textu, obrázkov a zdrojového kódu, je systém pripravený pre pridanie prakticky akéhokoľvek druhu kurzu, nezávisle na obsahu jeho učiva.

Rozšírenia sa však ponúkajú aj pre samotný systém. Úloha supervisora je v súčasnom systéme až na jednu funkciu pasívna. Rozšírením jeho možností je možné z neho spraviť učiteľa. Pre učiteľa by napríklad mohla byť vytvorená možnosť opravovať testy študentov. Testy by tak nemuseli pozostávať len z otázok, ktoré si vyžadujú zaškrtnutie správnych odpovedí. Mohli by sa pridať otázky so slovnými odpoveďami, ktoré by boli zobrazené učiteľovi a on by ich hodnotil. Učiteľovi by sa pri

každéj otázke zobrazila aj správna odpoveď, čím by sa zamedzilo zbytočných chybám v opravovaní. Tento variant má však nevýhodu pre študentov, ktorý by takto nemali hneď po dokončení testu k dispozícii jeho výsledok, ale museli by čakať až pokiaľ učiteľ test opraví.

Ďalším možným rozšírením pôsobnosti učiteľa je možnosť študentov klásť dotazy k látke kurzov, ktoré by im učiteľ následne zodpovedal. Toto by bolo vhodné realizovať napríklad pomocou newsgroup alebo fóra moderovaného učiteľom. Prístup k dotazom a odpoveďami by mali všetci študenti daného kurzu. Na dotazy študentov by mal mať možnosť odpovedať nielen učiteľ, ale aj ostatní študenti. Fórum by nebolo určené len na kladenie a zodpovedanie dotazov, ale bolo by aj miestom, kde by sa študenti mohli podeliť so svojimi postrehmi, prípadne problémami z praxe týkajúcimi sa tematiky kurzu.

K zvyšovaniu kvality systému by výrazne prospel systém hodnotenia kurzov študentmi. Po dokončení kurzu by mal každý študent možnosť vyjadriť sa ku kvalite, náročnosti a aktuálnosti kurzu napríklad vo forme vyplnenia dotazníku. Tvorcovia kurzu by na základe štatistík získaných z dotazníkov určili slabé miesta kurzu a pokúsili by sa ich vylepšiť.

Pre rýchlejšie a pohodlnejšie vytváranie kurzov je možné do systému pridať rolu vývojára. Vývojár by mal mať možnosť priamo v systéme vytvárať a editovať učivo lekcie, ktoré by následne bolo vložené do databázy. Takisto by mal mať možnosť pomocou rozhrania pridávať testové otázky.

Systém je v súčasnosti zameraný na doplnkové vzdelávanie študentov a nemá ich hodnotiť a nútiť vzdelávať sa, ale slúžiť ako prostriedok pre lepšie pochopenie učiva. Naskytuje sa tu teda možnosť zmeny podstaty systému na systém, ktorý by sa stal súčasťou regulárnej, hodnotenej výuky. To by si vyžadovalo výrazný zásah do bezpečnosti systému, najmä čo sa týka časti určenej pre študentov. Zmeniť by sa musel aj systém testovania študentov. Testy by boli sprístupnené študentom len v stanovený termín a nie počas celého štúdia. Študenti by stratili možnosť neobmedzeného testovania, lebo by prišlo k úniku testových otázok. Aj napriek tomu by museli byť testové otázky pravidelne menené, aby sa zachovala náročnosť testov a spravodlivosť hodnotenia. Študenti by mali na riešenie testov presne vymedzený čas a po jeho uplynutí, by sa testy uzavreli v stave, v ako by sa práve nachádzali. Takisto by bolo potrebné zamedziť študentom prístup na internet a iné zdroje informácií, ktoré by mohli použiť pri riešení testov.

Literatúra

- [1] Barešová A.: e-LEARNING ve vzdělávání dospělých, VOX 2003, ISBN 80-86324-27-3
- [2] Carliner S.: Designing E-Learning, ASTD 2002, ISBN 1-56286-332-0
- [3] Horton W.: Evaluating E-Learning, ASTD 2004, ISBN 1-56286-300-2
- [4] Sedgewick R.: Algoritmy v C, SoftPress 2003, ISBN 8086497569
- [5] Honzík, J., Hruška, T., Máčel, M.: Vybrané kapitoly z programovacích technik, Ed.stř.VUT Brno, 1991
- [6] Moodle – a Free, Open Source Course Management System for Online Learning. Stránka je dostupná online na URL: <http://moodle.org> (máj 2006)
- [7] Moodle CZ. Stránka je dostupná online na URL: <http://moodle.cz> (máj 2006)
- [8] Chlebec J., Majorová M.: Moodle SK Web pre podporu používateľov LMS MOODLE na Slovensku. Stránka je dostupná online na URL: <http://www.moodle.sk> (máj 2006)
- [9] Mindflash Tech – Online e-Learning System. Stránka je dostupná online na URL: <http://www.mindflash.com> (máj 2006)
- [10] e-Learning centre. Stránka je dostupná online na URL: <http://www.e-learningcentre.co.uk> (máj 2006)
- [11] E-Learning magazine. Stránka je dostupná online na URL: www.elearningmag.com (máj 2006)
- [12] Fakulta informačních technologií VUT v Brně: Algoritmy. Dokument je dostupný online na URL: <http://www.fit.vutbr.cz/study/course-1.php?id=59> (máj 2006)
- [13] Hruška T., Burget R.: Internetové aplikace IV, 2007. Dokument je dostupný online na URL: <https://www.fit.vutbr.cz/study/courses/WAP/private/opory/OporaWAP4ProgramovaniServeru.pdf>
- [14] Hruška T.: Internetové aplikace VI, 2007. Dokument je dostupný online na URL: <https://www.fit.vutbr.cz/study/courses/WAP/private/opory/OporaWAP6ProgramovaniKlienta.pdf>
- [15] Smarty TEMPLATE ENGINE. Stránka je dostupná online na URL: <http://smarty.php.net> (máj 2006)
- [16] PHP. Stránka je dostupná online na URL: <http://www.php.net> (máj 2006)
- [17] Janovský D.: Jak psát web. Stránka je dostupná online na URL: <http://www.jakpsatweb.cz> (máj 2006)


Zoznam príloh

Príloha 1. Ukážka E-learningového systému Moodle

Príloha 2. CD

Príloha 1 Ukážka E-learningového systému

Moodle



flash.help

- Úvod
- Diskusní fórum
- Pokročilé vyhledávání
- FAQ (Časté dotazy)
- Dotazy & připomínky
- Download
- Odkazy sem

Flash

- Seznámení
- Základní pravidla
- Nástroje
- Barvy a atributy
- Knihovna
- Snímky, pohyb
- Pohyb po křivce, maska**
- Vložení do HTML
- Novinky v MX2004

ActionScript

- Řízení animace
- Proměnné
- Vložení zvuku
- Podmínky a smyčky
- Funkce
- Atributy instancí
- UI komponenty (MX)
- Objekty - úvod
- Objekty "CORE"
- Objekty "MOVIE"

Příklady

- Flash GuestBook 2.0
- Fotogalerie
- Flash GuestBook
- SmartClip
- Vlastní kurzor a duplikace MovieClipu
- Ovládání autička
- Textový scrollbar
- MovieClip scrollbar

[hledat](#) ↕ AT

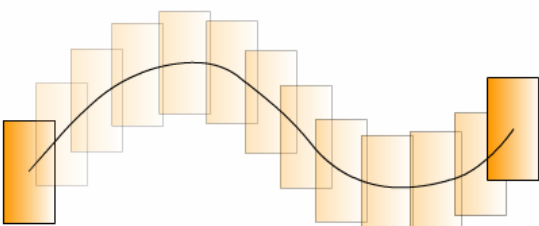
Pohyb po křivce, maska

Toto jsou dva speciální případy Motion Tweenu, kdy je kromě klasické vrstvy použita ještě jedna pomocná.

Pohyb po křivce

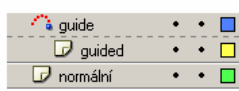
Pokud vytvoříte Motion Tween, bude vždy rozdíl polohy mezi počátečním a koncovým klíčovým snímkem vyplněn pohybem po přímce.

Lze to však udělat i jinak. Pohyb může kopírovat libovolnou trasu. K tomu nám pomůže tzv. vodící vrstva (Guide)



Postup:

- Vytvoříte klasickou vrstvu
- Klikněte pravým tl. na vrstvu a zvolte "**Properties**" (lze použít i dvojklik). Poté zvolte Type - **Guide**
- Poté vytvoříte další vrstvu, která **MUSÍ** být umístěna pod vodící vrstvou a stejně, jako v předchozím případě otevřete okno "**Properties**", ale zvolte Type - **Guided**
- Pokud jste všechno udělali správně, mělo by to vypadat takto:



- Do vodící vrstvy (guide) nakreslete čáru
- Ve vrstvě níže (guided) vytvoříte klasický Motion Tween, ale v počátečním a koncovém klíčovém snímku uchopíte instance za středy a umístíte je na konce křivky (měly by se samy uchopit)
- Případně můžete nastavit na panelu "Properties":

Obr. P1.1: Příklad kurzu vytvořeného v systému Moodle

Improving Sendmail Security by Turning it Off

<http://www.deer-run.com/~hal/sysadmin/sendmail.html>

Hal Pomeranz, Deer Run Associates

Sendmail is, for better or worse, the de facto standard Mail Transfer Agent (MTA) for Unix systems. While many tomes could be written about the pros and cons of replacing Sendmail with an alternate MTA-- such as Postfix, Qmail, or Exim-- the reality is that most Unix shops have a huge installed base of machines running Sendmail daemons as part of their default install. However, the surprising news is that the vast majority of these systems do not need to be running a Sendmail daemon at all.

This is likely to become a critical security issue for sites in the near future. The Sendmail buffer overflow exploit announced in March is almost certainly going to be programmed into an automated worm within the next six months. Such a worm could do for Unix systems what Code Red did to the Windows world-- simply because there are so many potentially vulnerable Unix systems on the network today. Shutting off the Sendmail daemon on 99.9% of the systems in your environment would greatly reduce the potential impact of such a worm.

The Role of the Sendmail Daemon

When discussing security issues with Unix System Administrators, many of them are confused about the need for a running Sendmail daemon on their systems. This confusion is understandable since, for at least the last two decades, the commercial Unix vendors have been shipping their operating systems with active Sendmail daemons in the default install. Most administrators simply assume that this daemon is necessary for the users and automated processes running on the system to be able to emit email from the machine.

The reality, however, is that the Sendmail daemon on a machine is only responsible for two things:

1. Listening on port 25/tcp for *incoming* messages from outside of the machine
2. Flushing the local queue of unsent messages on a periodic basis

Having a Sendmail daemon listening on 25/tcp is what opens the system up for remote exploits like the recently announced buffer overflow issue. But note that the system only needs to be listening on 25/tcp if it is actively expecting to receive email messages from other systems. However, the only systems on the network that receive external email are the machines that are acting as mail servers and mail relays. A given site will typically only have a small handful of these machines (many of which are Windows machines running Exchange these days anyway). The other 99.9% of the systems on the network are email "clients"-- that is they may emit email messages from time to time, but never expect to receive a message from another machine. On these machines it is probably best to simply disable the Sendmail daemon altogether in order to completely eliminate the potential for remote exploits.

But what about outgoing email generated from the local system? It turns out that Unix email clients send outgoing email by invoking the Sendmail binary directly off the disk. This new Sendmail process simply attempts to deliver the message to its next hop directly, rather than communicating with the local Sendmail daemon running on the system. So while it is necessary to have a proper Sendmail configuration file on each local system, it is not necessary to have a Sendmail daemon running in order for users and processes on the system to send email out of the machine. Actually, the release of Sendmail v8.12 has complicated this simple view of the world somewhat, as we will discuss later in this article.

Remember, however, that the local Sendmail daemon is also responsible for managing the local queue of unsent messages. For example, when a user creates an email message, the local Sendmail process invoked by the user's mail client may not be able to reach the next hop mail server immediately. This message is then queued in the system's local mail queue. If the local Sendmail daemon on the system is shut off, then there is nothing to process the local mail queue, and this unsent message will never be delivered! So when shutting off the local Sendmail daemon to avoid remote exploits, it is important to remember to arrange for some mechanism to flush the local mail queue on a regular basis.

So let's tackle all of these issues. First we'll look at how to manage the local mail queue, and then discuss appropriate Sendmail configurations for "client" type machines-- both for older versions of Sendmail and the new v8.12 releases.

Obr. P1.2: Příklad kurzu vytvořeného v systému Moodle

Ovládáte vztažná zájmena? - Pokus 1

1 Ten,(kdož) jsem čekal, nepřišel.
Body: --/1
Odpovědět:

2 Maminka, pod(jejíž) rukama vykvetlo tolik růží, byla pobledlá.
Body: --/1
Odpovědět:

3 Pohlédl na své ruce, (jež) se živil celý život.
Body: --/1
Odpovědět:

4 Prý je nejsilnější.(což) se samozřejmě nedá věřit.
Body: --/1
Odpovědět:

5 Vítáme vás, (kdož) se podařilo najít cestu k nám.
Body: --/1
Odpovědět:

6 Obviněný se dopustil podvodu, (což) se dopustil trestného činu.
Body: --/1
Odpovědět:

7 Ten člověk, (jenž) znám, zemřel.
Body: --/1
Odpovědět:

Obr. P1.3: Příklad testu vytvořeného v systému Moodle