

**Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií**



Webový portál pro podporu předmětu PRIPO

Bakalářská práce

Autor: Ondřej Vondra
Studijní obor: Aplikovaná informatika

Vedoucí práce: Mgr. Josef Horálek, Ph.D.

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 15.8 2016

Ondřej Vondra

Poděkování:

Děkuji vedoucímu bakalářské práce Mgr. Josefu Horálkovi, Ph.D. za metodické vedení práce a cenné rady. Rovněž děkuji své rodině a přátelům za podporu během studia. Dále děkuji garantovi předmětu „Principy počítačů“ prof. RNDr. Petru Mikuleckému, Ph.D. za poskytnuté materiály. Závěrem děkuji Mgr. Petru Čechákovi za podněty v rámci stylistické a gramatické části práce.

Anotace

V současné době jsou výukové systémy využívány stále častěji. K jejich tvorbě lze využít širokou škálu technologií, které, i když na první pohled se zdají stejné, mají při bližším pohledu značné rozdíly. Tyto systémy jsou hojně využívány i na Fakultě informatiky a managementu na Univerzitě Hradec Králové. Jeden takový výukový systém a jeho tvorba, konkrétně pro předmět „Principy počítačů“, je popsán v následující práci. Součástí tohoto systému jsou také animace vysvětlující problematiku látku, jež jsou zde rovněž představeny.

Klíčová slova:

Výuková aplikace, ASP.NET, Principy počítačů, Animované výukové materiály

Annotation

Title:

Web portal to support the subject PRIPO

Learning systems are currently used more and more often. For their creation there are a lot of tools that can be used, although at the first glance seem to be the same. The closer look can show the major differences. These systems are widely used even on the Faculty of informatics and management on University of Hradec Králové. One of these educational systems and its creation, specifically for the subject „Principy počítačů“ („Computer Principles“), is shown in this thesis. Part of this system are animations that explains problematical materials. These animations are shown there too.

Keywords:

Educational application, ASP.NET, Computer principles, Animated educational materials

Obsah

Seznam zkratk	1
Seznam obrázků	3
Seznam tabulek	3
Seznam grafů	3
1 Úvod	4
2 Analýza předmětu „Principy počítačů“	5
3 Výběr prostředí pro tvorbu animací	7
3.1 Flash	7
3.2 GIF	7
3.2.1 GIMP	8
3.3 Microsoft Silverlight	9
3.4 Porovnání	9
4 Tvorba animací pro vybrané partie předmětu	12
4.1 Převody soustav	12
5 Výběr technologie pro webovou prezentaci	14
5.1 Technologie pro tvorbu aplikace	14
5.1.1 PHP	14
5.1.2 ASP.NET	14
5.1.3 Perl	15
5.1.4 JSP	16
5.1.5 Python	16
5.1.6 Porovnání technologií	17
5.2 Technologie pro uložení dat	19
5.2.1 Microsoft SQL Server	19
5.2.2 SQLite	20
5.2.3 MySQL	21
5.2.4 PostgreSQL	22
5.2.5 Porovnání technologií	23
6 Návrh struktury webové prezentace	26
6.1 Analýza	26
6.1.1 Doménový model	27
6.1.2 Datový model	28
6.2 ASP.NET MVC	29

6.2.1	Výhody ASP.NET MVC	30
6.2.2	Využití v projektu.....	31
6.3	Přístup k datům	31
6.3.1	Nhibernate 4.0	32
6.3.2	Entity Framework 7	32
6.3.3	Výběr ORM Frameworku.....	33
6.4	Uživatelské role	34
6.5	Popis aplikace	35
7	Přístupy k vytvoření cvičných testů.....	37
8	Výběr hostingu	38
8.1	ASPone s.r.o.	38
8.2	ASP2.cz	38
8.3	Porovnání poskytovatelů.....	39
9	Návrh optimální integrace do LMS Blackboard.....	41
9.1	LMS Blackboard	41
10	Závěr.....	42
11	Použitá literatura.....	45
11.1	Seznam literatury	45
11.2	Seznam pramenů.....	46
	Příloha A – Podklad pro zadání bakalářské práce	49

Seznam zkratek

ACID – Atomicity, Consistency, Isolation, Durability

ADO.NET – ActiveX Data Object.NET

AJAX – Asynchronous JavaScript and XML

ANSI – American National Standards Institute

API – Application Programming Interface

ASP.NET – Active Server Pages.NET

CLR – Common Language Runtime

CRUD – Create, Read, Update, Delete

DBMS – Database Management System

GAP – GIMP animation package

GIF – Graphics Interchange Format

GIMP – GNU Image Manipulation program

GNU – GNU's Not Unix

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

IIS – Internet Information Services

ISAPI – Internet Server Application Programming Interface

JSP – Java Server Pages

LINQ – Language Integrated Query

LMS – Learning Management System

MSQL – Mini SQL

MVC – Model, View, Controller

MVCC – Multi-Version Concurrency control

OOP – Object-oriented programming

ORM – Object-relational mapping

PBKDF2 – Password-Based Key Derivation Function 2

PHP – PHP: HyperText Preprocessor

RDBMS – Relational database management system

RIA – Rich Internet Application

SQL – Structured Query Language

TDD – Test-driven development

UI – User Interface

VB.NET – Visual Basic.NET

WebCT – Web Course Tools

Windows DNA – Windows Distributed InterNet Applications Architecture

WPF – Windows Presentation Foundation

Seznam obrázků

Obrázek 1 – Převod z dvojkové do desítkové soustavy	12
Obrázek 2 – Převod z desítkové do dvojkové soustavy	12
Obrázek 3 – Převod z šestnáctkové do desítkové soustavy	12
Obrázek 4 – Převod z desítkové do šestnáctkové soustavy	12
Obrázek 5 – Převod z osmičkové do desítkové soustavy	13
Obrázek 6 – Převod z desítkové do osmičkové soustavy	13
Obrázek 7 – Převod z dvojkové do šestnáctkové soustavy	13
Obrázek 8 – Převod z šestnáctkové do osmičkové soustavy 1. fáze	13
Obrázek 9 – Převod z šestnáctkové do osmičkové soustavy 2. fáze	13
Obrázek 10 – Doménový model	27
Obrázek 11 – Datový model	28
Obrázek 12 – MVC model	29
Obrázek 13 – Využití MVC architektury v projektu	31
Obrázek 14 – Rozdíly mezi zobrazením menu pro jednotlivé uživatelské role	35
Obrázek 15 – Vzhled webové aplikace (Nepřihlášený uživatel)	35

Seznam tabulek

Tabulka 1 – Podpora technologií jednotlivými nástroji	10
Tabulka 2 – Porovnání technologií pro tvorbu animací	11
Tabulka 3 – Počet hostingů pro dané technologie	18
Tabulka 4 – Porovnání technologií pro tvorbu aplikace	19
Tabulka 5 – Limity databází PostgreSQL	23
Tabulka 6 – Počet hostingů podporujících databázové technologie	24
Tabulka 7 – Porovnání technologií pro uložení dat	25
Tabulka 8 – Hodnotící systém	37
Tabulka 9 – Porovnání poskytovatelů webhostingu	40

Seznam grafů

Graf 1 – Celkové známky studentů od roku 1995/96 do roku 2015/16	5
Graf 2 – Průměrné výsledky studentů od roku 1995/96 do roku 2015/16	6

1 Úvod

Využití výukových systémů je v současné době na vzestupu, a to nejenom na vysokých školách. Možností tvorby takových systému je velké množství. Mezi nejčastější technologie patří ASP.NET, Python, PHP, Perl a JSP, přičemž někdy jsou rozdíly mezi těmito technologiemi drobné a jindy značné.

Právě zmíněný nárůst výše uvedených technologií a rozdíly v možnostech, které nabízejí jednotlivé technologie v souvislosti s cílem zapojení těchto systémů na Fakultě informatiky a managementu na Univerzitě Hradec Králové, byly prvním důvodem pro volbu tohoto tématu. Předmět „Principy počítačů“ byl zvolen, protože zakončení tohoto předmětu spolu se zápočtovými testy je stále prováděno písemnými testy.

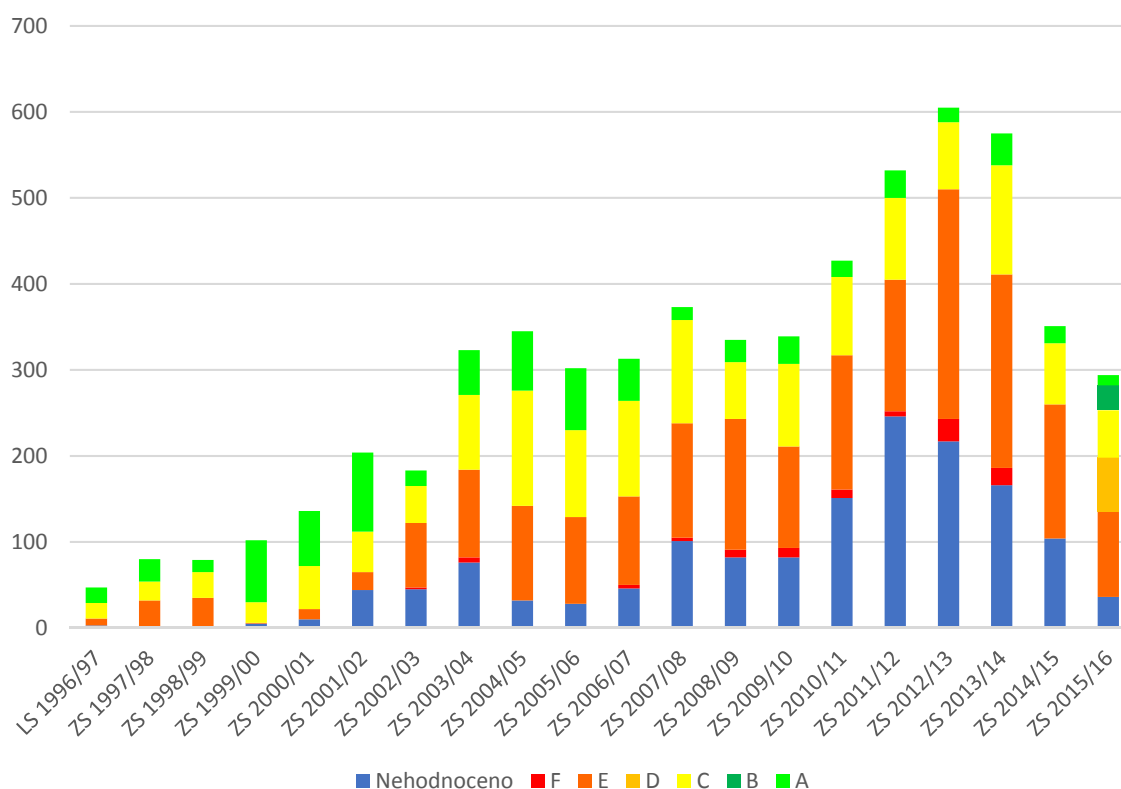
Hlavním cílem této práce je tedy vytvořit výukový systém, který pomůže studentům lépe pochopit problematiku látku předmětu „Principy počítačů“ a vhodně je připraví k úspěšnému absolvování předmětu. Nedílnou součástí tvorby tohoto systému je optimální návrh samotného systému. Zároveň jsou v této práci představeny technologie pro vývoj podobných aplikací, stejně jako možnosti jejich hostování. Jako možnost hostování se nabízí cepsos.cz, což je jeden z univerzitních výukových systémů. Dalším cílem je vypracování analýzy předmětu „Principy počítačů“ na základě výsledků z předchozích let. Jako další cíl lze uvést vytvoření naučných animací, které spolu s výukovými texty a cvičnými testy studentům napomohou při studiu. Posledním z cílů je návrh optimální integrace do LMS Blackboard. Tato práce by tedy měla přispět k lepším výsledkům studentů.

2 Analýza předmětu „Principy počítačů“

Před samotnou tvorbou, ať už animací či aplikace, bylo nutné provést analýzu, zda je vůbec nutné tuto aplikaci vytvářet. Analýza bohužel nemohla být uskutečněna na základě výsledných testů, ale byla provedena z celkových výsledků všech studentů. Informace použity pro tuto analýzu byly získány z informačního portálu Stag.uhk.cz.

Dle těchto výsledků byl vytvořen graf 1, ze kterého lze pozorovat všechny výsledky studentů od akademického roku 1995/96 do nyní. Z grafu je vidět postupné zhoršení prospěchu studentů. Jsou zde však zaneseny pouze celkové informace o prospěchu, a proto byl vytvořen graf 2 na základě průměrových hodnot.

Graf 1 – Celkové známky studentů od roku 1995/96 do roku 2015/16



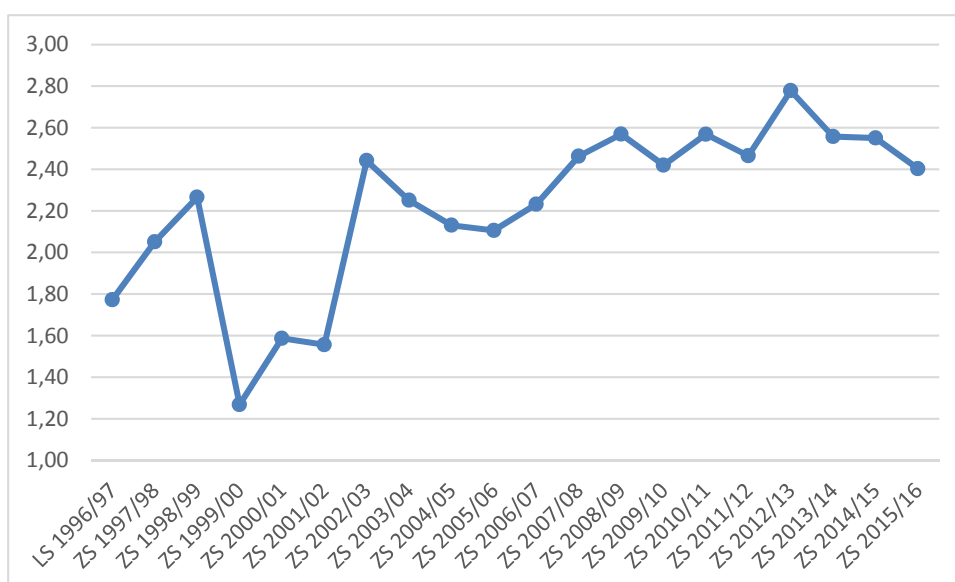
Zdroj: Upraveno dle STAG UHK, nedatováno

Tento graf zobrazuje průměrné hodnoty všech studentů včetně těch, kteří neuspěli. Lze zde zřetelněji vidět hlavní problematika studentů, a tou je nedostatečná příprava k závěrečné zkoušce. Za poslední dva roky je patrné mírné zlepšení. Toto zlepšení je ale velice nepatrné, jelikož průměr za poslední rok (2015/16) byl 2,40. Tento výsledek je velice znepokojivý, protože se jedná o předmět zahrnující základní dovednosti a znalosti, bez kterých se student vysoké školy v oboru informatika a technika neobejde.

Z výsledků je tedy jasně vidět, že i přes vyšší počet studentů zde stále zůstává velice podobný a znepokojivý průměr, který je třeba zlepšit (STAG UHK, nedatováno).

Na základě těchto dat je tedy zřejmé, že je nutno zpracovat všechny kapitoly a, kde bude potřeba, vytvořit animované příklady. Rozpracování pouze některých kapitol tedy nepřipadá v úvahu.

Graf 2 – Průměrné výsledky studentů od roku 1995/96 do roku 2015/16



Zdroj: Vlastní tvorba dle STAG UHK, nedatováno

3 Výběr prostředí pro tvorbu animací

3.1 Flash

Flash je dynamický a robustní 2D navrhovací a vývojářský nástroj, který je využíván designéry a vývojáři po celém světě, pro vytváření prezentací, her a dalšího obsahu, vyžadujícího uživatelskou interakci. Flashové projekty zahrnují jednoduché animace, video komplexní prezentace a mnoho dalšího.

Flash umožňuje tvorbu mediově bohatých aplikací pomocí integrace různých mediálních elementů, jako jsou obrázky, zvuk, videa a speciální efekty. Flash je většinou využíván k tvorbě obsahu doručovaného přes internet, kvůli malé velikosti souborů, která je dosažena využíváním vektorové grafiky. Vektorová grafika pojme značně méně paměti než bitmapová grafika, protože je reprezentována matematickými výrazy místo velkými daty. Bitmapová grafika je větší, protože každý jednotlivý pixel obrazu vyžaduje samostatnou část dat pro reprezentaci.

Existují dva způsoby vkládání grafiky do Flashových projektů. První variantou je vytvoření vlastní grafiky přímo ve vývojovém prostředí Flashe a druhou je importování externích obrázků do Flashového dokumentu. Pomocí naimportované grafiky a nástroje Timeline lze vytvořit Flashové filmy.

Flashové soubory jsou ukládány s příponou .fla, která je příponou zdrojového souboru. Když je Flashový dokument připraven k exportování na webové stránky, je exportován jako .swf komprimovaný soubor. Ke spuštění .swf souboru musí klient disponovat Flash playerem.

Flashové vývojové prostředí se skládá ze čtyř částí. Tyto části jsou „stage“, která obsahuje všechny grafické, textové a jiné elementy, „timeline“, která organizuje elementy dle časové osy, „tools“ a „panels“, které poskytují prostředky pro tvorbu a úpravu Flashových dokumentů (Lakshmi Prayaga a Hamsa Suri, 2008, s. 1-2).

3.2 GIF

Animované obrázky GIF jsou speciální formou souborového formátu GIF užívaného na webových stránkách. Principy a vlastnosti týkající se dané GIF animace představují, a to jak obecně, tak i v rovině specifické pro konkrétní web, množství bodů této animace.

Animované GIFy jsou v podstatě forma image-flipping animací, kde jeden obraz překrývá druhý v určitém pořadí. Rychlost střídání obrázku je atributem souborového

formátu a je nezávislá na jiných médiích. To znamená, že GIFové animace lze použít pro prezentaci s pomalou rychlostí změny obrázků či pro plynulý pohyb s vysokou rychlostí změny obrázků. Dalším atributem na úrovni souboru je loop. Tento atribut může být nastaven na kontinuální tak, že se animace opakuje stále dokola, a pokud je animace vytvořena bez přechodu, může se zdát, že nemá ani začátek ani konec. Atribut loop lze také nastavit na pevná čísla, takže například hodnota 1 způsobí, že animace proběhne jen jednou a zastaví se na svém posledním snímku.

GIFové animace jsou předrenderované rastrové obrázky, které mají barevnou hloubku až 256 barev. Kompresní algoritmus použitý u souboru GIFového formátu je zvláště užitečný pro ploché obrázky, ale není moc vhodný pro fotografie s vysokým tónováním. Všechny obrázky, které tvoří animaci, jsou uloženy v rámci obrazové banky a jsou postupně zobrazovány. Velikost GIFového obrázku závisí na čtyřech faktorech. Tři z těchto faktorů jsou velikost obrazu (X, Y a velikost pixelu), počet snímků a barevná hloubka. Čtvrtý faktor je závislý na kompresním poměru (Alan Peacock, 2002, s. 112-113).

3.2.1 GIMP

GIMP (GNU Image Manipulation Program) je jedním z předních open-source programů pro úpravu obrazu.

Jedná se o počítačový program pro vytváření a úpravu digitálních obrázků. Zejména je navržen pro editaci digitálních fotografií a webové grafiky. Lze ho též použít pro tvorbu animovaných obrázků GIF.

GIMP je zcela open-source software. Je vytvořen a aktualizován komunitou a šířen všemi uživateli. Vznikl roku 1995 jako projekt Spencera Kimballa a Petera Mattis. Během jednoho roku se rozrostl do populárního open-source editoru obrázků, díky příspěvkům od uživatelů z celého světa.

Dnes GIMP používá bezpočet lidí na celém světě. Funguje na většině verzí systému Windows, Macintosh, Linux a UNIX. (Akkana Peck, 2008, s. 1-2).

Existuje mnoho zásuvným modulů (plug-in), kterými je možné základní verzi programu GIMP rozšířit. Mezi tyto rozšíření patří například GAP, což je rozšíření, které napomáhá při tvorbě animací.

3.3 Microsoft Silverlight

Jedná se o Framework pro vytváření prohlížečově hostovaných RIA (Rich Interactive Application), které běží na různých operačních systémech. Silverlight pracuje díky plug-inu v prohlížeči. Při prohlížení webových stránek obsahujících Silverlight obsah tento plug-in spustí kód obsažený na stránce. Důležitou součástí je skutečnost, že plug-in Silverlight poskytuje bohatší prostředí než tradiční směs HTML a JavaScriptu, která pohání běžné webové stránky. Lze vytvářet Silverlight stránky, které přehrávají video nebo mají hardwarově akcelerovanou 3D grafiku používající vektorové animace (Matthew MacDonald, 2012, s. XXX).

Silverlight 1 původně obsahoval jen 2D kreslení a mediální podporu přehrávání. Vývojáři byli nuceni programovat v JavaScriptu, protože tato verze nezahrnovala podporu .NET Framework. Verze 2 přinesla velké změny. Dodala podporu .NET Framework a uživatelské prostředí založené na WPF¹. Silverlight 5 udržuje stejný model vývoje, jako byl nastaven u verze 2 (Matthew MacDonald, 2012, s. XXXV).

Silverlight aplikace je možné vytvářet ručně či pomocí profesionálních nástrojů pro vývoj. Tyto aplikace je možno tvořit buď v grafickém editoru Microsoft Expression Blend nebo ve vývojářském prostředí Visual Studio, které obsahuje nástroje pro kódování, testování a debugování. Protože oba nástroje slouží k vývoji Silverlight aplikace, je možné je oba kombinovat. Microsoft vyvinul nástroj Visual Studio LightSwitch kombinující tyto dva programy. V tomto případě je kladen důraz na používání šablon a generování kódu, díky čemuž je možno rychle vytvářet sofistikované aplikace (Matthew MacDonald, 2012, s. 1-2).

3.4 Porovnání

Pro tvorbu animací byly vybrány technologie Flash, Gif a Silverlight. Porovnávání těchto technologií bylo prováděno na základě daných kritérií. Každé kritérium bylo bodově ohodnoceno od jedné do pěti. Technologie s nejvíce body byla zvolena pro použití. Všechny informace o technologiích byly získány z oficiálních stránek hlavních poskytovatelů. Pro Flash bylo vybráno vývojové prostředí Animate CC a informace byly získávány z oficiální stránky Adobe (Adobe, 2016a,b), pro Gif bylo zvoleno vývojové prostředí GIMP a bylo čerpáno z jejich oficiálních stránek (GIMP, 2001-2016) a pro Microsoft Silverlight bylo

¹ **WPF** – Poskytuje vývojářům jednoduchý programovací model pro vývoj moderních aplikací (Microsoft, 2016a).

zvoleno rozšíření Visual Studio LightSwich a údaje pocházejí z oficiálních stránek Microsoft (Microsoft Silverlight, nedatováno).

Jako první kritérium byla zvolena dostupnost ve smyslu ceny vývojového prostředí. Vývojové prostředí Animate CC je dostupné za cenu 647 Kč², čímž se řadí pod nejdražší vývojová prostředí. Díky tomu, že GIMP je free open-source vývojový nástroj, je zdarma. Visual Studio LightSwich je rozšíření klasického Visual Studia, a to je pro studenty rovněž zdarma. Proto jsou tyto dva nástroje nejvhodnější z hlediska ceny.

Dalším hodnotícím kritériem je použitelnost hotových animací na webových stránkách, v prezentacích a v LSM Blackboard. Podpora jednotlivých technologií je zobrazena v tabulce 1. Kromě Microsoft Silverlight, protože je sice možno takové animace implementovat do powerpointových prezentací, ale pouze jako webovou stránku, obě technologie získávají hodnocení pět.

Tabulka 1 – Podpora technologií jednotlivými nástroji

Technologie	Webové stránky	Powerpointové prezentace	LMS Blackboard
Flash	Ano	Ano	Ano
Gif	Ano	Ano	Ano
Silverlight	Ano	Přes HTML stránku	Ano

Zdroj: Vlastní tvorba na základě informací z oficiálních zdrojů

Poslední hodnotící kritérium je vlastní zkušenost s danými technologiemi. Jelikož zkušenosti autora ať už s Flashovými animacemi, či s animacemi v Microsoft Silverlight jsou zanedbatelné, hodnocení těchto nástrojů je jedna. Naopak zkušenosti s GIFovými animacemi v prostředí GIMP jsou daleko větší a díky tomu získává hodnocení pět.

Na základě hodnotících kritérií byly pro tvorbu zvoleny GIFové animace vytvářené v prostředí GIMP. Výsledky a bodová hodnocení jsou zobrazeny v tabulce 2.

² Všechny ceny jsou stanoveny ke dni 18. 7. 2016 podle měnových kurzů stanovených dle České národní banky.

Tabulka 2 – Porovnání technologií pro tvorbu animací

Technologie	Dostupnost	Použitelnost	Vlastní zkušenost	Výsledek
Flash	1	5	1	7
Gif (GIMP)	5	5	5	15
Microsoft Silverlight	5	3	1	9

Zdroj: Vlastní tvorba na základě oficiálních zdrojů

4 Tvorba animací pro vybrané partie předmětu

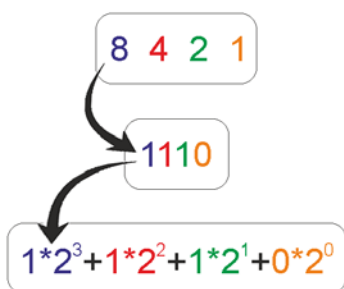
Jak bylo zjištěno v kapitole 3.4 („Porovnání“), nejvhodnější technologií pro tvorbu animací v tomto případě jsou gifové animace v grafickém editoru GIMP. Animace jsou vytvořeny jako frame-by-frame animace. Většina animací obsahuje okolo 150 až 200 framů. Tyto animace byly vytvářeny za použití postupného skládání vrstev. Konkrétně se jedná o animace vysvětlující problematiku číselných soustav.

4.1 Převody soustav

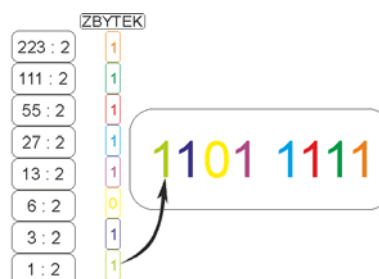
První animace vysvětluje převod z dvojkové do desítkové soustavy. Je v ní ukázán jednoduchý způsob převodu za použití mocnin dvojky. Na obrázku 1 je vidět výsledný frame této animace.

Další představuje opačný převod z desítkové do dvojkové soustavy. Tato animace je „framově“ náročnější, jelikož je pro převod použit způsob postupného dělení číslem dvě. Část animace je znázorněna na obrázku 2.

Obrázek 1 – Převod z dvojkové do desítkové soustavy Obrázek 2 – Převod z desítkové do dvojkové soustavy



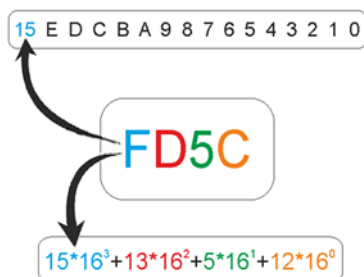
Zdroj: Vlastní tvorba



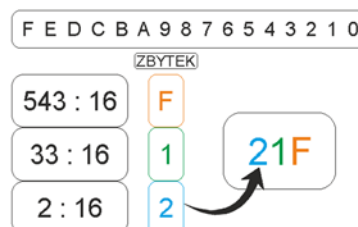
Zdroj: Vlastní tvorba

Animace pro převod z šestnáctkové do desítkové soustavy je zobrazena na obrázku 3. Opačně z desítkové do šestnáctkové je animace ukázána na obrázku 4.

Obrázek 3 – Převod z šestnáctkové do desítkové soustavy Obrázek 4 – Převod z desítkové do šestnáctkové soustavy



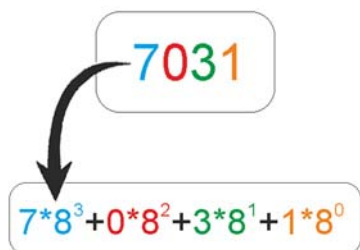
Zdroj: Vlastní tvorba



Zdroj: Vlastní tvorba

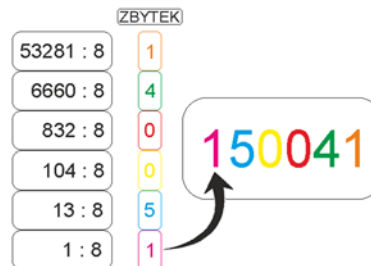
Převody mezi osmičkovou a desítkovou soustavou lze vidět v dalších animacích. Jedná se o převody z osmičkové soustavy do desítkové a opačně. Ukázky z těchto animací jsou zobrazeny na obrázcích 5 a 6.

Obrázek 5 – Převod z osmičkové do desítkové soustavy



Zdroj: Vlastní tvorba

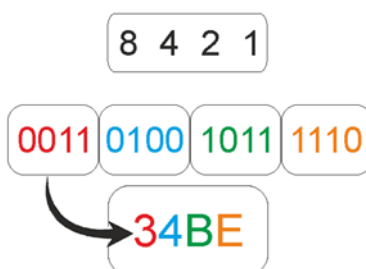
Obrázek 6 – Převod z desítkové do osmičkové soustavy



Zdroj: Vlastní tvorba

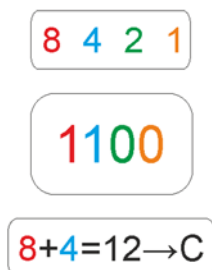
Mezi poslední zpracované animace pro převody soustav patří převod z dvojkové do šestnáctkové soustavy, který je ukázán na obrázku 7, a převod z šestnáctkové do osmičkové soustavy, jenž je rozdělen do dvou fází. V první fázi je zobrazen převod z šestnáctkové do dvojkové soustavy. Tento převod je znázorněn na obrázku 8. Druhá fáze, tedy převod z dvojkové do osmičkové soustavy je zobrazen na obrázku 9.

Obrázek 7 – Převod z dvojkové do šestnáctkové soustavy



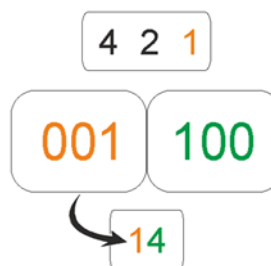
Zdroj: Vlastní tvorba

Obrázek 8 – Převod z šestnáctkové do osmičkové soustavy 1. fáze



Zdroj: Vlastní tvorba

Obrázek 9 – Převod z šestnáctkové do osmičkové soustavy 2. fáze



Zdroj: Vlastní tvorba

5 Výběr technologie pro webovou prezentaci

5.1 Technologie pro tvorbu aplikace

5.1.1 PHP

V roce 1994 byl Rasmusem Lerdorfem vytvořen skriptovací jazyk PHP pro sledování návštěvníků jeho online resumé. Původně zkratka znamenala „Personal Home Page“. Postupem času spolu s rostoucím využíváním a možnostmi využití, začal se měnit i význam názvu, stával se z něj „PHP: Hypertext Preprocessor“.

Podle oficiální stránky php.net je PHP široce používaný mnohoúčelový, skriptovací jazyk, který je obzvláště vhodný pro vývoj webových aplikací a může být vložen do HTML kódu. Vložením do HTML kódu je myšlena skutečnost, že lze do standardní HTML stránky vložit PHP kód, kdekoli je potřeba. Vložený kód zajistí dynamický obsah.

PHP byl na rozdíl od komplikovaných jazyků vytvořen pro tvorbu webových skriptů, nikoli pro vytváření stand-alone aplikací, i když nyní s trochou úsilí to možné je. PHP skripty lze spouštět jen jako reakce na určité události.

Jedná se o multiplatformní³ technologii zpracovávající veškeré výpočty a dotazy na straně serveru. K tomuto zpracování musí být dostupný aplikační webový server, jako například Apache nebo Microsoft IIS⁴. Je nutné, aby všechny PHP skripty byly přístupné přes URL. Skripty psané v PHP běžící na jednom serveru budou s žádnou nebo drobnou úpravou fungovat i na jiném serveru (Larry Ullman, 2012, s. X - XI).

5.1.2 ASP.NET

Historii ASP.NET je možné sledovat zpátky do roku 1995, kdy si Microsoft uvědomil, že ve vývoji poněkud zaostává, protože průmysl se směřuje směrem k internetu. Do té chvíle Microsoft vyvíjel pouze své patentované technologie, nástroje a síťové protokoly pro Microsoft Network. V této době byly vývojářské možnosti jak na straně serverového tak desktopového programování omezené.

³ **Multiplatformní** – funguje na většině operačních systémů včetně Windows, Unix a Macintosh (Larry Ullman, 2012, s. XI).

⁴ **IIS** – Visual Basic aplikace, která z webového serveru reaguje na požadavky prohlížeče (Microsoft, 2016b).

Na počátku roku 1996 měl Microsoft první zlepšení na poli internetového vývoje a to za pomoci ISAPI (Internet Server Application Programming Interface) technologie jako součásti IIS (Internet Information Server) (Mesbah Ahmed a kolektiv, 2002, s. 2-3). Téhož roku Microsoft vydal veřejnou beta verzi IIS 3. Mezi hlavní přínosy této verze patří vývojové prostředí zvané ASP (Active Server Pages).

Microsoft udělal opravdu dobře, když nenechal zestárnout ASP a přenesl ho do jednadvacátého století jako ASP.NET pomocí VB.NET, z něhož se stala plnohodnotná objektivě orientovaná vývojová platforma pro tvorbu webových aplikací.

ASP.NET je klíčovou součástí širší Microsoft .NET iniciativy pro vývoj aplikací. .NET je jak aplikační architektura pro nahrazení Windows DNA modelu tak soubor nástrojů, služeb, aplikací a serverů na bázi .NET Framework a CLR (Common Language Runtime). Spíše než inkrementálním upgradem je ASP.NET kompletním přepsáním využívajícím všech pokročilých funkcí .NET. ASP.NET má výhodu v tom, že může nabídnout .NET podporu pro 20 a více .NET jazyků a mnoho softwarových knihoven.

Aplikace psané v ASP.NET jsou rychlé, efektivní, lehce spravovatelné, flexibilní, ale především mají snadno pochopitelný kód (Mesbah Ahmed a kolektiv, 2002, s. 10-12).

5.1.3 Perl

Perl je populární open-source programovací jazyk, který byl vytvořen Larrym Wallem. První verze Perl byla vytvořena v roce 1987. Od té doby byl rozšířen a vylepšen velkou komunitou programátorů (Jules J. Berman, 2011, s. 2).

Perl je programovací jazyk na vysoké úrovni. Jedná se o interpretovaný jazyk, což znamená, že programy se stávají z prostého kódu a není třeba žádný krok kompilace při spouštění programu. Perl je navržen pro snadné použití. Při tvorbě aplikace je potřeba mnohem méně času než u ostatních programovacích jazyků. Perl není PERL. Nejedná se o zkratku, jde o název programovacího jazyka, zatímco perl je název překladače potřebného pro spuštění programu.

Jednou z hlavních výhod Perlu je rychlost psaní kódu, jeho jednoduchost, dostupnost a flexibilita. Ta je ale také jednou z jeho hlavních nevýhod, protože v některých situacích může být mnohem pomalejší než ostatní programovací jazyky. Někomu se může zdát jako další nevýhoda i skutečnost, že funguje hlavně v příkazovém řádku spíše než s grafickým prostředím (Simon Andrews, 2007-2014, s. 4).

5.1.4 JSP

JavaServer Pages je Java technologie umožňující vývojářům rychle vytvářet dynamické webové aplikace. I když existují ostatní technologie, pouze JSP poskytuje výkonný, přenosný a snadno rozšiřitelný Framework, který podporuje vývoj webových aplikací.

Zatímco JSP technologie může být použita k vytvoření jednoduché webové stránky, její skutečná síla spočívá ve schopnosti poskytovat základ pro tvorbu webových aplikací. JSP stránky mohou být použity k vybudování on-line webové stránky, veřejného fóra, nebo jiného typu interaktivního webu. Zatímco jiné technologie mohou také hrát roli při vytváření těchto typů webových stránek, JSP je silná technologie, která podporuje tvorbu dynamických webových stránek i s malými či žádnými předchozími zkušenostmi. Funkce jako Expression Language⁵ a JSP standardizovaná tagová knihovna dovoluje tvorbu malých JSP dokumentů s velkým počtem funkcí. Specifikace JSP také umožňuje vývojářům vytvářet vlastní akce, díky které může jediný řádek v JSP stránce poskytovat Behind-the-Scenes⁶ funkcionalitu.

JSP webové aplikace závisí na HTTP protokolu⁷ a na Java Servlet. JSP aplikace mají jednoduchou adresářovou strukturu, která seskupuje konfigurační informace a prostředky pro jednodušší identifikaci a zpracování. Webové aplikace vyžadují tzv. popisovač nasazení (Deployment descriptor), který v případě, že se jedná o konkrétní webové aplikace, poskytuje informace o konfiguraci do webového aplikačního serveru (Robert J. Brunner, 2003, s. 1-2).

5.1.5 Python

Python je nejspíš nejjednodušší a nejpřehlednější programovací jazyk v širokém užití. Jeho kód je snadno čitelný i programovatelný, ale hlavně je stručný. Python je velmi expresivní jazyk, což znamená, že je možné psát mnohem méně řádků než by bylo zapotřebí pro ekvivalentní aplikace v jiných programovacích jazycích.

⁵ **JSP Expression Language** – umožňuje přístup k aplikačním datům. Tento přístup se provádí výhradně přes rámcové proměnné a atributy (Robert J. Brunner, 2003, s. 72).

⁶ **Behind-the-Scenes** – zpracování probíhající na serveru.

⁷ **HTTP protokol** – poskytuje klient-server komunikaci přes internet (Robert J. Brunner, 2003, s. 2).

Jedná se o multiplatformní jazyk, ale je možné vytvořit programy závislé na platformě. Toto je jen zřídka nutné, protože téměř všechny Python knihovny jsou standardními knihovnami na většině platform.

Jednou z největších předností Python právě skutečnost, že obsahuje velmi kompletní standardní knihovnu, která umožňuje pomocí jednoho nebo pár řádků vytvořit funkcionality jako stahování souborů z internetu, rozbalování komprimovaných archivních souborů a podobně. Kromě standardních knihoven jsou k dispozici tisíce knihoven třetích stran, které poskytují silnější a sofistikovanější funkce než standardní knihovny.

Python lze použít k procedurálnímu, objektovému a v menší míře i funkcionálnímu programování, i když Python je převážně objektově orientovaný jazyk (Mark Summerfield, 2010, s. 1).

5.1.6 Porovnání technologií

Největší zřetel byl brán na výběr technologie pro tvorbu webové aplikace. Při špatné volbě této technologie by při vývoji mohly nastat nechtěné problémy. Porovnání technologií pro vývoj webové aplikace je provedeno na základě důležitých kritérií, ke kterým bylo při rozhodování přihlíženo. Technologie jsou ohodnoceny bodově, od jedné do pěti, přičemž technologie s nejvíce body je doporučena k použití. Veškeré informace byly získány z oficiálních stránek poskytovatelů.

Prvním hodnotícím kritériem jsou požadované vlastnosti. Pod tyto vlastnosti spadá fakt, zda je technologie zpracovávána na straně serveru či klienta. Většina vybraných technologií je zpracovávána výhradně na serveru. To znamená, že zajišťují plynulý běh aplikace. Perl a Python jsou sice zpracovávány na serveru, ale je možné je využívat i pro zpracování na klientské straně komunikace. V tomto ohledu jsou tedy technologie srovnatelné. Další požadovanou vlastností bylo programovací paradigma. Jelikož má autor zkušenosti především s objektově orientovanými jazyky, byl brán zřetel především na to, zda jsou jazyky právě objektově orientované. Z tohoto pohledu jsou vidět větší změny. ASP.NET a JSP jsou vytvořeny právě pro objektový svět, zatímco PHP a Python sice v novějších verzích nabízejí podporu objektového programování, ale ve své podstatě jsou to jen skriptovací jazyky. Perl také nabízí podporu objektového programování, ale v tomto případě se jedná a interpretovaný jazyk. Pod hodnotícím kritériem „požadované vlastnosti“ tedy ASP.NET a JSP jsou hodnoceny pěti body a ostatní technologie třemi.

Druhé hodnotící kritérium jsou možnosti hostování výsledné aplikace. Jelikož zahraniční hostingy byly z důvodu vyšší odezvy vyloučeny, informace pro toto kritérium byly získávány ze stránky hostingy.cz. V tabulce 3 jsou uvedeny počty poskytovatelů webového hostingu pro dané technologie. Tyto počty zahrnují i placené hostingy, a to z důvodu větší rozmanitosti vzorku. Z tabulky 3 je vidět, že pro PHP je největší počet nabízených hostingů, nejspíš z důvodu jeho oblíbenosti. Získává proto pět bodů. Se stejným počtem hostingů, tedy s 25, je Perl a Python. Oba jsou tedy hodnoceni čtyřmi body. Jako další je ASP.NET s hodnocením třemi body a jako poslední JSP se dvěma body. Jedná se ale pouze o placené hostingy a při pohledu na varianty zdarma zůstávají pouze hostingy podporující technologie PHP a ASP.NET (Hostingy.cz, 2003-2016).

Tabulka 3 – Počet hostingů pro dané technologie

Technologie	Počet hostingů
PHP	135
ASP.NET	19
Perl	25
JSP	7
Python	25

Zdroj: Vlastní tvorba dle Hostingy.cz, 2003-2016

Předposledním kritériem je množství podporovaných technologií. Toto kritérium je jedno z důležitějších zejména kvůli podpoře plánovaného využití technologií. První důležitou technologií jsou asynchronní operace, díky kterýmž je aplikace mnohem rychlejší. Jelikož se jedná o moderní technologie, všechny mají podporu asynchronních operací, například ve formě přídatných frameworků. Jedním z těchto frameworků je AJAX. Další velice užitečnou technologií je ORM Framework, díky níž není třeba vytvářet SQL kódy. Každá technologie má svůj vývojáři oblíbený framework. Mezi nejznámější patří Entity Framework nebo NHibernate, které využívá především ASP.NET. NHibernate vychází z jeho předchůdce Hibernate, jenž je široce používaným JSP. Pro PHP byl vytvořen Doctrine, který vychází z NHibernate. Stejně tak byl pro Python vytvořen SQLAlchemy a pro Perl například Rose::DB::Object. Všechny technologie jsou tedy v této hodnotící kategorii na stejné úrovni.

Posledním důležitým kritériem jsou vlastní zkušenosti autora. Autor měl možnost vyzkoušet si některé technologie v rámci výuky formou projektů. Některé technologie jako Perl a Python pro autora nebyly ničím lákavé, a proto se zkušenosti s těmito jazyky dají považovat za nulové. Jak již bylo zmíněno, některé technologie byly nastíněny v rámci výuky, konkrétně to jsou JSP a PHP. Zkušenosti se tedy nedají považovat za relevantní, ačto jsou přece jenom větší než u Perlu nebo u Pythonu a získávají tedy dva body. Největší

zkušenosti autora, ať už ve formě školního projektu či vlastní pracovní zkušenosti, jsou zaručeně s technologií ASP.NET. Proto tato technologie získává pět bodů.

Po důkladném zvážení a nastudování potřebných informací byla vybrána technologie ASP.NET. Výsledky hodnocení na základě zadaných kritérií jsou zaneseny do tabulky 4 (Peter Cowburn, 2016; Microsoft, 2016c; Perl, 2002-2016; Oracle, nedatováno; Python, 2001-2016).

Tabulka 4 – Porovnání technologií pro tvorbu aplikace

Technologie	Požadované vlastnosti	Možnosti hostování	Množství podporovaných technologií	Vlastní zkušenost	Výsledek
PHP	3	5	5	2	15
ASP.NET	5	3	5	5	18
Perl	3	4	5	0	12
JSP	5	2	5	2	14
Python	3	4	5	0	12

Zdroj: Vlastní tvorba za použití oficiálních zdrojů

5.2 Technologie pro uložení dat

5.2.1 Microsoft SQL Server

Jedná se o Databázový management systém (DBMS), který zajišťuje ukládání, vyhledávání, správu a manipulaci s daty. SQL Server je na trhu již velmi dlouho a celosvětově je používán miliony uživatelů. Důvody, proč se jedná o jeden z nejpoužívanějších databázových systémů, jsou následující:

- **Výkon:** SQL Server je velmi rychlý.
- **Důvěryhodnost:** Je používán některými z nejdůležitějších a nejprestižnějších organizací a stránek, z nichž všichni do databáze svěřují svá citlivá data.
- **Integrace:** Jelikož se jedná o produkt společnosti Microsoft, je přímo spojen s některými jejich nabídkami a produkty.
- **Jednoduchost:** Jedná se o jeden z nejjednodušších DBMS z hlediska instalace a správy.

Jednou z nevýhod je fakt, že se nejedná o multiplatformní DBMS, nýbrž o čistě Windows záležitost. Jde také o komerční produkt, a proto pro ty, co hledají free open-source variantu SQL Server, není nejlepší volbou. Byl často kritizován pro nepodporování high-end podnikových funkcí jako clustering a tolerance chyb, tato kritika byla ale adresována především na verzi SQL Server 2005.

Databáze, jako jsou SQL Server, Oracle a MySQL, jsou založeny na architektuře klient-server. Aplikace na této architektuře jsou rozděleny do dvou odlišných částí. „Server portion“ je část software, který je zodpovědný za veškeré přístupy a manipulaci s daty.

Jediný, kdo pracuje s daty, je serverový software. Všechny žádosti o poskytnutí, doplňování a mazání dat jsou zpracovávány serverovým softwarem. Tyto požadavky či změny pocházejí z počítačů, na nichž běží klientský software. Klient je software, se kterým uživatel přímo pracuje (Ben Forta, 2007, s. 36-38).

5.2.2 SQLite

SQLite je procesní knihovnou, která implementuje soběstačný, bezserverový, transakční SQL databázový stroj. Kód pro SQLite je umístěn na veřejné doméně a je tedy zdarma k použití pro jakýkoli účel (Soukromý/Komerční). Jedná se o nejrozšířenější databázi na světě s nespočtem aplikací, včetně několika velkých projektů.

SQLite má integrovaný SQL databázový stroj. Na rozdíl od většiny ostatních SQL databází nemá samostatný serverový proces. SQLite zapisuje přímo do běžných souborů na disku. Kompletní SQL databáze s velkým množstvím tabulek, triggerů a pohledů je obsažena přímo na disku v jediném souboru. Souborový formát databáze je multiplatformní – databáze může být libovolně kopírována mezi 32-bit a 64-bit systémy. Díky těmto vlastnostem je SQLite populární.

Jedná se o kompaktní knihovnu se všemi funkcemi, přičemž velikost této knihovny může být menší než 500 KB, v závislosti na nastavení cílové platformy a optimalizace. Pokud jsou volitelné vlastnosti, které zabírají mnoho místa, vynechány, velikost SQLite knihovny může být snížena i pod 300 KB. SQLite lze také spustit v minimálním režimu zásobníku, což je populární volba databázového stroje u zařízení s omezenou pamětí jako jsou mobilní telefony, PDA a MP3 přehrávače. Jedná se ale o kompromis mezi využíváním paměti a rychlostí. Obecně platí, že čím rychleji SQLite běží, tím více paměti zabírá. Nicméně výkon je i při nedostatku paměti velmi dobrý.

SQLite je velmi pečlivě testován před vydáním nové verze, díky čemuž je velice spolehlivý. Většina ze zdrojového kódu je vyvinuta především pro testování a ověřování. Transakce jsou ACID⁸ i při přerušení zaviněného systémovou chybou či výpadkem proudu. Vše je zajištěno pomocí ověřených automatizovaných testů díky speciálním testovým svazkům, které simulují systémové selhání. Samozřejmě i při všech těchto testech stále existuje jisté riziko chyb, ale na rozdíl od podobných projektů SQLite poskytuje log s chybami, které pak lze snadno opravit.

SQLite je vyvíjeno mezinárodním týmem vývojářů, kteří pracují na plný úvazek pro SQLite. Vývojáři i nadále rozšiřují možnosti SQLite a stále zvyšují jeho spolehlivost a výkonnost, při zachování zpětné kompatibility s publikovaným rozhraním, SQL syntaxí a databázovými souborovými formáty. Zdrojový kód je zcela zdarma a je k dispozici komukoli (SQLite, nedatováno).

5.2.3 MySQL

Jedná se o klient-server relační databázový management systém původem ze Skandinávie. MySQL obsahuje SQL server, klientské programy pro přístup k serveru, nástroje pro správu a programovací rozhraní pro psaní vlastních programů.

Kořeny MySQL jsou v roce 1979, s databázovým nástrojem UNIREG, který stvořil Michael Widenius pro švédskou společnost TcX. V roce 1994 začala společnost TcX hledat RDBMS s SQL rozhraním pro použití při vývoji webových aplikací. Bylo testováno několik komerčních serverů, ale žádný nebyl dostatečně rychlý pro jejich potřeby. V důsledku toho začal Michael Widenius vyvíjet nový server. Programovací rozhraní bylo navrženo tak, aby bylo podobné tomu, které používá msql, pro které bylo k dispozici mnoho volně dostupných nástrojů, které by tak mohly být použity i pro MySQL.

V roce 1995, David Axmark z Detron HB začal tlačit na TcX, aby uvolnili MySQL. MySQL 3.11.1 bylo vypuštěno do světa roku 1996 ve formě binárních distribucí pro Linux a Solaris. V současné době MySQL funguje na mnoha platformách a je k dispozici v binární i zdrojové formě. Společnost MySQL AB byla založena pro poskytování distribucí MySQL s otevřeným zdrojovým kódem i s komerční licencí. Dále nabízí technickou podporu,

⁸ **ACID** – Atomicity, Consistency, Isolation, Durability - Atomicita, Konzistence, Izolace, Trvanlivost

monitorovací služby a školení. Roku 2008 Sun Microsystems převzala MySQL AB a nadále poskytuje open source licence MySQL (Paul DuBois, 2009, s. 1-2).

Mezi hlavní nástroje, které MySQL obsahuje, patří:

- **SQL Server.** Jedná se o motor, který pohání MySQL a poskytuje přístup k databázím.
- **Klientské a pomocné programy.** Mezi které patří klientský program, který umožňuje přímo zadávat dotazy a zobrazovat výsledky. K dispozici jsou také některé administrativní a pomocné programy, které pomáhají při provozu webových stránek. Některé umožní sledovat a kontrolovat server, jiné zase import dat, provádění záloh, kontrolu tabulek kvůli problémům a další.
- **Klientská knihovna pro vytváření programů.** Jelikož je tato knihovna psána v jazyce C, lze psát klientské programy pouze v jazyce C. Je ale možné propojit knihovnu s jinými jazykovými procesory, jako jsou Perl, PHP nebo Ruby.

Kromě softwaru dodávaného s MySQL existuje mnoho talentovaných vývojářů, kteří poskytují své výtvořky zdarma. Díky nim je možné získat mnoho nástrojů třetích stran a použití MySQL je snadnější (Paul DuBois, 2009, s. 5).

5.2.4 PostgreSQL

PostgreSQL je výkonný open-source objektově-relační databázový systém. Má více než 15 let aktivního vývoje a osvědčené architektury, díky čemuž získal značnou pověst pro svou spolehlivost. Funguje na všech hlavních operačních systémech včetně Linux, Macintosh a Windows. Je plně ACID kompatibilní, má plnou podporu cizích klíčů, joinů, pohledů, triggerů a uložených procedur. Zahrnuje většinu datových typů z SQL 2008. Podporuje také ukládání velkých objektů včetně obrázků, audio souborů či videa. Má nativní programovací rozhraní pro většinu programovacích jazyků.

Může se pochlubit sofistikovanými funkcemi jako MVCC (Multi-Version Concurrency Control), obnova podle bodu v čase nebo asynchronní replikace. Podporuje mezinárodní znakové sady a multibyte kódování znaků Unicode. Je široce škálovatelný a dokáže v nezměrném množství řídit velké množství souběžně se dotazujících uživatelů. Některé z aktivních systémů PostgreSQL spravují více než 4 TB dat. V tabulce 5 lze vidět omezení databází PostgreSQL.

Tabulka 5 – Limity databázi PostgreSQL

Limit	Hodnota
Maximální velikost databáze	Neomezeno
Maximální velikost tabulky	32 TB
Maximální velikost řádku	1,6 TB
Maximální velikost pole	1 GB
Maximum řádků v tabulce	Neomezeno
Maximum sloupců v tabulce	250 -1600 (záleží na datovém typu sloupce)
Maximum indexů v tabulce	Neomezeno

Zdroj: PostgreSQL, 1996-2016

PostgreSQL se pyšní dodržováním standardů. Jeho implementace SQL je v souladu s ANSI-SQL:2008 standardem. Má plnou podporu subqueries, včetně subselect při použití FROM klauzule. Při plně relačním systémovém katalogu, který sám o sobě podporuje vícero schémat na databázi, je tento katalog přístupný prostřednictvím informačního schématu, které je definováno v normě SQL

Funkce integrity dat zahrnují (složené) primární klíče, cizí klíče s omezením a kaskádovými změnami.

Má také celou řadu rozšíření s pokročilými funkcemi. Mezi tyto funkce například spadá automatická inkrementace hodnot sloupce pomocí sekvencí a limit umožňující návrat dílčích kroků. PostgreSQL podporuje složené, částečné, unikátní a funkcionální indexy (PostgreSQL, 1996-2016).

5.2.5 Porovnání technologií

Porovnávání technologií pro uložení dat je provedeno na základě zvolených hodnotících kritérií, která jsou uvedena v tabulce 7. Klasifikace je vypočtena na základě bodového hodnocení od jedné do pěti. Technologie s nejvyšším počtem bodů je vybrána k použití. Veškeré informace jsou získávány z oficiálních zdrojů pro jednotlivé technologie.

Prvním z hodnotících kritérií jsou možnosti nasazení. Toto kritérium bylo posuzováno na základě počtu poskytovatelů webového hostingu podporujícího tuto databázovou technologii. Protože pro webovou aplikaci byly vybrány jen hostingy v České

republike, tak i zde tomu bylo stejně. Informace byly stejně jako u kapitoly 5.1.6 („Porovnání technologií“) získány z webové stránky hostingy.cz. Jelikož pro tvorbu byla vybrána

technologie ASP.NET, byl zadán filtr jen na hostingy podporující tuto technologii. Z tabulky 6 je opět zřejmé, že PHP je nejoblíbenější technologií pro tvorbu webových aplikací, jelikož MySQL je její nejpoužívanější databázovou technologií.

Tabulka 6 – Počet hostingů podporujících databázové technologie

Díky nejvyššímu počtu hostingů podporující MySQL získává tato technologie pět bodů. Hned za ní je Microsoft SQL Server, který se taktéž může pochlubit velkou podporou hostingovými poskytovateli. Ten tedy získává čtyři body. Podporu SQLite a PostgreSQL nabízí nejméně poskytovatelů, a z toho důvodu nezískávají velký počet bodů.

Technologie	Počet hostingů
Microsoft SQL Server	14
SQLite	3
MySQL	17
PostgreSQL	6

Zdroj: Vlastní tvorba dle Hostingy.cz, 2003-2016

Dalším kritériem je podpora technologie webové prezentace. Jedná se o velmi důležité hledisko, protože kdyby byla zvolena špatná technologie pro databázi, mohly by nastat problémy s komunikací mezi aplikací a databází. Jelikož ASP.NET je moderní technologie, je možné výslednou aplikaci propojit s jakoukoli z vybraných databází. I přes tento fakt je možné mezi nimi najít drobné výhody a nevýhody. Hlavní výhodou Microsoft SQL Serveru je fakt, že stejně jako technologie vybraná pro tvorbu aplikace (ASP.NET) je vytvářena a spravována firmou Microsoft a tudíž je zde velká podpora a kompatibilita. Na rozdíl od MySQL nebo PostgreSQL se sice nejedná o open-source variantu, ale existují licence, které jsou pro tuto technologii dostupné zdarma. Jelikož zde nenajdeme velké rozdíly v kvalitě, jsou všechny technologie z hlediska podpory technologie pro webovou prezentaci hodnoceny pěti body.

Posledním kritériem jsou vlastní zkušenosti autora. Stejně jako u technologie pro tvorbu webové aplikace i zde lze zkušenosti autora rozdělit na dva druhy, a to školní a pracovní. Při školních projektech bylo autorovi umožněno vyzkoušet si MySQL v kombinaci s PHP jakožto malý projekt. Hodnocení této zkušenosti je tudíž tři. Stejně tomu je i u Microsoft SQL Serveru, který autor mohl vyzkoušet v rámci jiného většího projektu, nicméně s touto technologií autor vytvářel i mimoškolní projekty a tudíž zkušenosti s ní jsou největší mezi testovanými. Z tohoto důvodu získává hodnocení pět. Autor pracoval i s SQLite, ale tato zkušenost se nedá považovat za relevantní a z tohoto důvodu tato

technologie získává hodnocení jedna. Jedinou technologií, se kterou autor nepracoval je PostgreSQL, a tudíž je hodnocena nulou.

Jelikož, podle výsledků z tabulky 7, Microsoft SQL Server dosáhl nejvíce bodů, tedy čtrnácti, byl vybrán k použití v projektu. (Microsoft, 2016d; SQLite, nedatováno; MySQL, 2016; PostgreSQL, 1996-2016).

Tabulka 7 – Porovnání technologií pro uložení dat

Technologie	Možnosti nasazení	Podpora technologie webové prezentace	Vlastní zkušenost	Výsledek
Microsoft SQL Server	4	5	5	14
SQLite	2	5	1	8
MySQL	5	5	3	13
PostgreSQL	3	5	0	8

Zdroj: Vlastní tvorba za použití oficiálních zdrojů

6 Návrh struktury webové prezentace

6.1 Analýza

Před vytvářením jakékoli aplikace je velice nápomocné provést analýzu a návrh výsledného vzhledu a funkčnosti aplikace. Tento návrh je prováděn pomocí dvou modelů. Jedná se o doménový model, který je vidět na obrázku 10 a z něho následně přepracovaný datový model zobrazený na obrázku 11.

Toto přepracování se provádí pomocí převedení doménového modelu, který je reprezentován třídami, do datového modelu, jenž reprezentují tabulky. Doménový model zobrazuje pouze použité třídy s atributy a jejich vazby s danou kardinalitou. V datovém modelu jsou k atributům přidány datové typy, jsou vytvořeny primární a cizí klíče a jako poslední je provedena normalizace.

Normalizace je velice užitečná technika, jak zajistit plynulý běh databáze. Jde tedy o omezení redundance dat a o celkové zjednodušení složitosti zápisu. Normalizace se rozděluje do několika forem. Tyto formy jsou vysvětleny níže v textu na příkladech, jak byly použity v projektu.

První normální forma:

Podle této normální formy je databáze upravena tak, že všechny tabulky obsahují pouze atomické proměnné. Například tedy správné odpovědi v tabulce otázek byly odděleny do samostatné tabulky, aby bylo možné ukládat více odpovědí pro jedinou otázku. Na stejném principu byly odděleny odpovědi na testové otázky.

Druhá normální forma:

Tato normální forma definuje především závislost neklíčových atributů na celém primárním klíči a nejen na jeho částech. Databáze splňuje tuto formu, pouze pokud splňuje předchozí. Není nutno se s ní v tomto případě zabývat a je možné rovnou říci, že databáze splňuje i tuto normální formu. Je tomu tak, protože všechny neklíčové atributy jsou jasně závislé na primárním klíči a není zde použit žádný vícehodnotový primární klíč.

Třetí normální forma:

Jelikož databáze splňuje jak první, tak druhou normální formu, je možné, aby splňovala i třetí. Princip této formy spočívá v tom, že žádný z neklíčových atributů není tranzitivně závislý na primárním klíči. To znamená, že žádný atribut není závislý na jiném

atributu, který je závislý na primárním klíči. Lze tedy s jistotou říci, že celá databáze splňuje i tuto normální formu.

Tyto tři normalizační formy jsou nejčastěji využívány díky své jednoduchosti a přínosu ke kvalitě tvořené databáze. Další normalizační formy jsou u projektu v tomto rozsahu zbytečné. Jak bylo ukázáno, celá databáze splňuje všechny tři normalizační formy a je díky nim připravena pro poměrně snadné pochopení a správu.

6.1.1 Doménový model

Na modelu zobrazeném obrázkem 10 je zobrazen náčrt základních entit systému a vztahů mezi nimi. Většina atributů vypovídá svými názvy sama za sebe. Některé ovšem tak jednoduché nejsou. Ty, jenž nemusí být tak zřejmé, jsou podrobněji vysvětleny v následujícím textu.

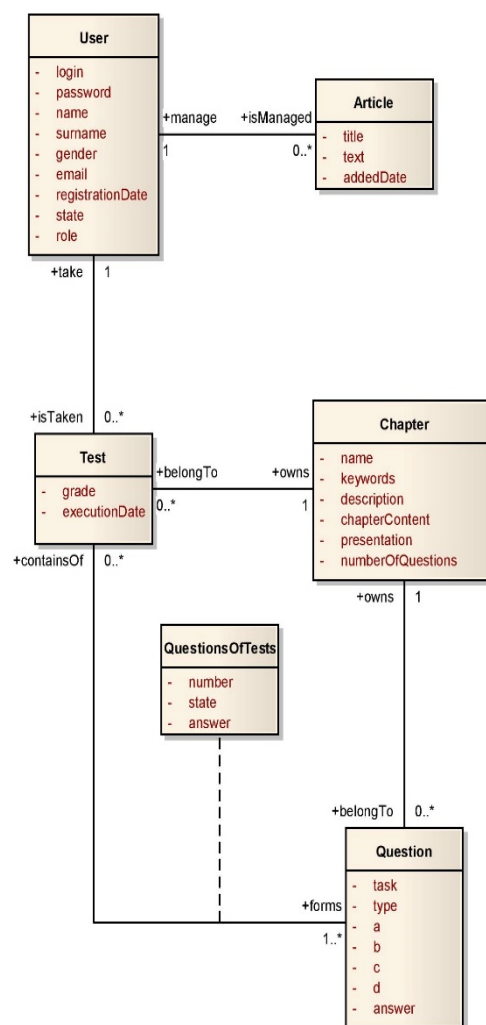
Atribut „state“ (stav) udává ve třídě „User“ stav, v jakém se nachází uživatelský účet. Tyto stavy jsou buďto „Aktivní“, „Registrovaný“, či „Deaktivovaný“. Deaktivované účty jsou po roce automaticky smazány. Stejně tak je tomu i u registrovaných účtů, které nejsou do třiceti dnů potvrzeny.

Stejný atribut je i ve třídě „QuestionsOfTests“. Zde se jedná o podobný případ a vypovídá o stavu otázky, zda je ohodnocena, či nikoli.

Dalším atributem je „numberOfQuestions“ v tabulce „Chapters“. V tomto případě jde o počet otázek, které se budou generovat do testu.

Posledním atributem, který potřebuje vysvětlení je atribut „type“, jenž připadá do třídy „Question“ a udává typ otázky. Celkem jsou zde rozlišeny dva typy otázek. Podrobněji je toto rozdělení vysvětleno v kapitole 7 („Přístupy k vytvoření cvičných testů“).

Obrázek 10 – Doménový model

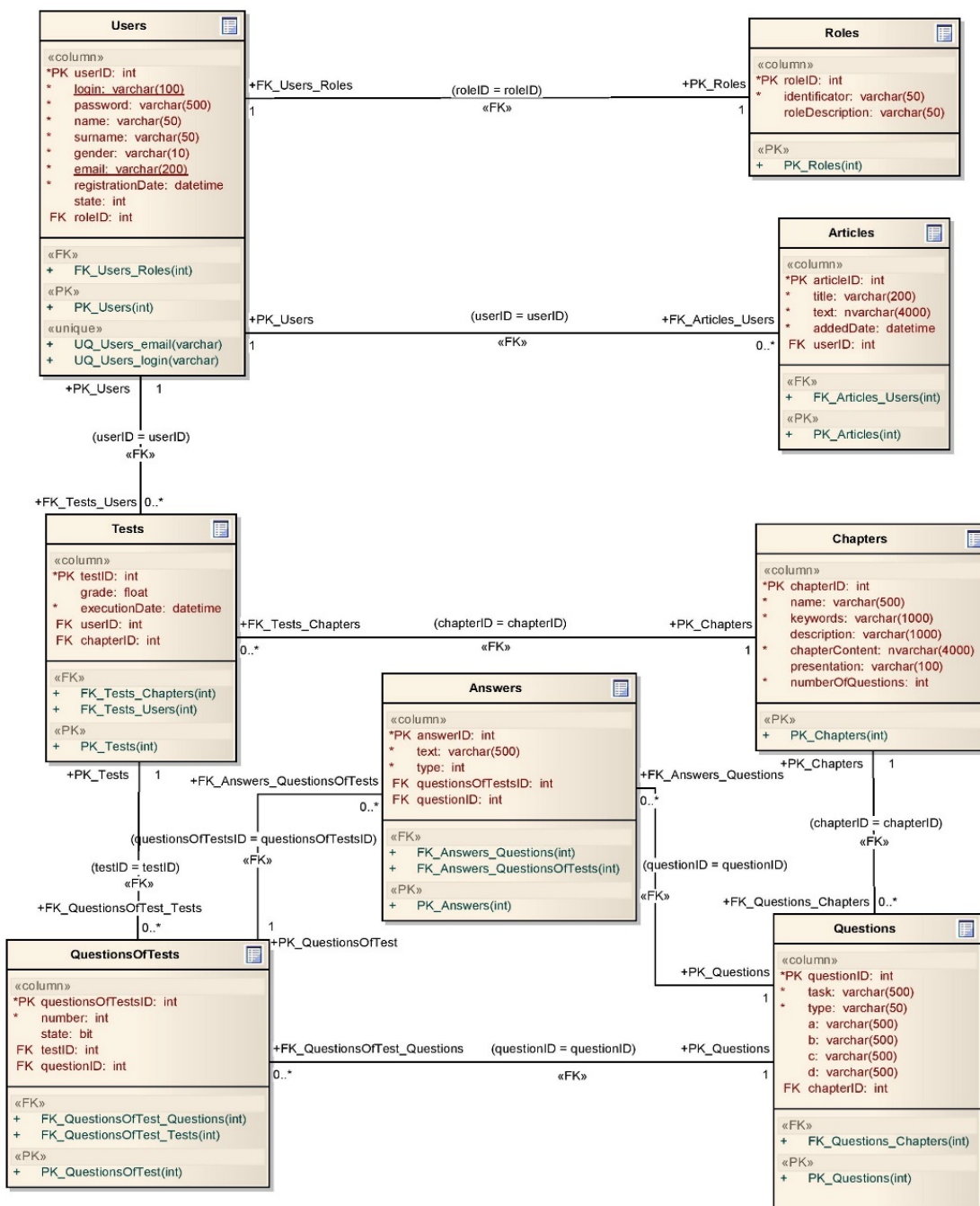


Zdroj: Vlastní tvorba

6.1.2 Datový model

Jak již bylo zmíněno, v modelu ukázaném na obrázku 11 je zobrazen návrh databáze. Tento model přímo vychází z doménového modelu. Jelikož pro vytvoření obou modelů byl použit vývojářský nástroj Enterprise Architect a výsledné vygenerování skriptu pro vytvoření databáze, byla tedy jednoduchá záležitost.

Obrázek 11 – Datový model



Zdroj: Vlastní tvorba

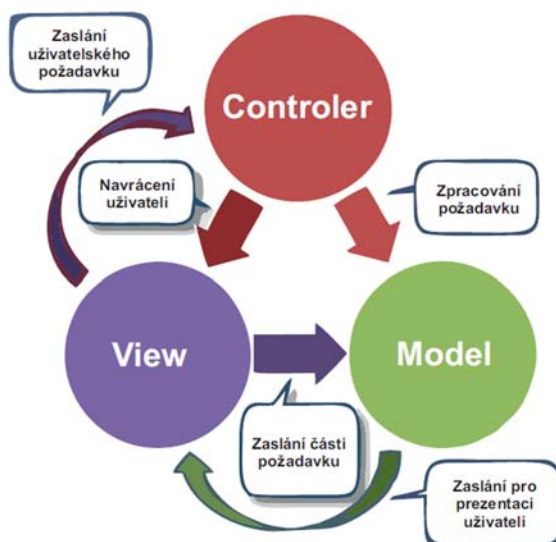
6.2 ASP.NET MVC

ASP.NET MVC plně podporuje Framework pro vytváření webových aplikací, které používají architekturu Model-View-Controller. Stejně jako u ASP.NET Web Forms, tak ASP.NET MVC je na vrcholku ASP.NET Frameworku. To znamená, že je možné použít v těchto aplikacích stejné API pro zabezpečení, stavový management, ukládání do paměti a další, jako u ASP.NET aplikací.

Ve světě ASP.NET MVC bylo mnoho vylepšení zahrnuto v samotném Frameworku. Hlavním cílem této architektury je izolovat business logiku od uživatelského rozhraní. Především je zaměřena na udržovatelnost, vylepšenou testovatelnost a čistou strukturu aplikace.

Každá aplikace ASP.NET MVC má tři základní části: model (Model), pohled (View) a kontrolery (Controller). Stručně řečeno, model se skládá ze všech tříd, které zpracovávají data a business logiku. Zpracování dat pomocí modelových tříd je iniciováno ze strany kontrolerů, které mají na starosti požadavky uživatelů. Jakmile je zpracování dat kompletní, kontroler vytvoří odpověď uživateli odesláním výsledku do pohledu, který vyprodukuje HTML stránku. Ta je následně vykreslena do prohlížeče.

Obrázek 12 – MVC model



Zdroj: Upraveno dle Jose Rolando Guay Paz, 2013, s. 7, Figure 1-2.

Obrázek 12 znázorňuje jednoduchou implementaci architektury MVC. Rovné šipky označují přímé sdružení, zatímco zakřivené šipky identifikují nepřímé asociace.

Model v architektuře MVC reprezentuje část aplikace, která implementuje datovou logiku domény. Provoz modelu většinou pochází ze tříd, které reprezentují objekty v datovém úložišti jako je databáze. Jsou to tedy například použité třídy v „DataAccess“.

Pohledy jsou viditelné prvky v aplikaci. Jsou to části, které obvykle uživatelé zobrazují části dat z modelu. Stránka pohledu obvykle obdrží objekt s modelem z kontroleru a nezáleží mu na tom, jak byla data získávána. Pohledová stránka obsahuje HTML kód (někdy i související UI kód), díky kterému ví jak prezentovat data z modelu přes prohlížeč uživateli.

Kontrolery jsou třídy, které shromažďují požadavky uživatelů, pracují s modelem a v poslední fázi vybírají příslušný pohled k renderování správného UI (Jose Rolando Guay Paz, 2013, s. 7).

6.2.1 Výhody ASP.NET MVC

Ve srovnání s ASP.NET Web Forms⁹ má ASP.NET MVC výhodu v tom, že zahrnuje nejen všechny základní funkce ASP.NET, ale i funkce z architektury MVC. Některé z těchto výhod jsou následující:

- MVC architektura sama usnadňuje správu složitosti tím, že jasně rozděluje funkčnost aplikace do tří hlavních částí: Model, Pohled, Kontroler.
- Webové aplikace založené na této architektuře nepoužívají formuláře založené na serveru. Díky tomu je MVC Framework ideální pro vývojáře, kteří chtějí plnou kontrolu nad chováním aplikace. Ve Web Form stránce lze užít <form> pouze jednou, což je docela zásadní omezení. V MVC žádné takové omezení není.
- ASP.NET MVC poskytuje lepší podporu pro vývoj řízený testy TDD (Test-driven development).
- Funguje velmi dobře při vývoji webové aplikace, která je vyvíjena více vývojáři a webovými designéry, kteří potřebují vysoký stupeň kontroly nad HTML kódem (Jose Rolando Guay Paz, 2013, s. 8).

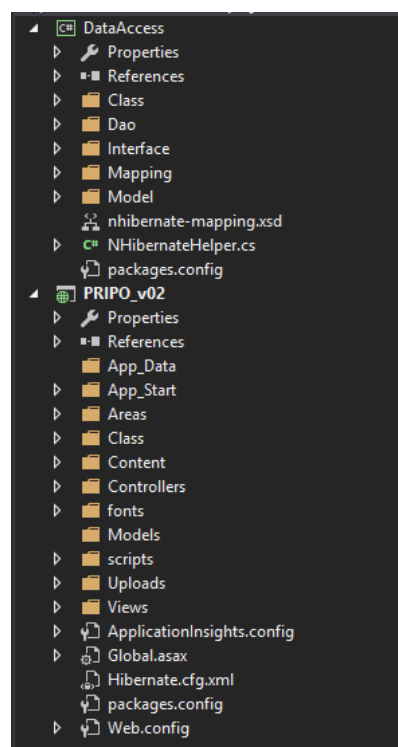
⁹ **ASP.NET Web Forms** – Framework pro vytváření webových aplikací, ve kterých stránky využívají syntaxi Razor (Jose Rolando Guay Paz, 2013, s. 6).

6.2.2 Využití v projektu

Jelikož jsou v dnešní době ASP.NET aplikace psány především s architekturou MVC, ať už z důvodu bezpečnosti, či jen pro přehlednost nebo rychlost aplikace, je MVC balíček nainstalován hned při založení projektu. Jak už bylo zmíněno v kapitole 5.1.6 („Porovnání technologií“), autorovy zkušenosti jsou především s technologií ASP.NET konkrétně tedy s aplikacemi založenými na ASP.NET MVC.

Na obrázku 13 lze vidět datovou strukturu aplikace. Ta je rozdělena do dvou projektů. Je zde nadefinována knihovna „DataAccess“, která je samostatně kompilována. Díky této knihovně je umožněn přístup k databázi pomocí ORM Frameworku NHibernate. Je zde obsažen i model aplikace, který, jak bylo řečeno, implementuje datovou logiku aplikace.

Obrázek 13 – Využití MVC architektury v projektu



Zdroj: Vlastní tvorba

6.3 Přístup k datům

Pro přístup k datům byl vybrán standard ORM, což je technika umožňující mapovat objekty do relační databáze a naopak. Termín „objekt“ většinou znamená instanci třídy. Nástroje, které využívají ORM techniky, se také označují jako ORM. Jedná se o takzvané objektově relační mappery. ORM zpracovává všechny CRUD operace a vývojář tak nemusí napsat jediný řádek SQL kódu. Nicméně hlavní výhodou této techniky je propojení objektově orientovaného světa (aplikace) a relačního světa (databáze). Rozdíly mezi relačním světem a objektovým světem přináší obtíže hlavně z pohledu impedance. Program v OOP musí při připojení k relační databázi mít stejnou impedanci, aby nedocházelo k pomalému či nekorektnímu přenosu dat. Musí být koherentní. Toto zajišťuje mimo jiné ORM (Suhás Chatekar, 2015, s. 1-2).

Zvolený mapper pro porovnání se nazývá NHibernate. Je založen na svém předchůdci Hibernate, který je používán především na platformě Java. Byl obohacen o funkce z .NET jako například LINQ queries. NHibernate je open source projekt, který prosperuje výhradně z příspěvků komunity. Stejně jako u jiných open source projektů je zde velká transparentnost kódu (Suhás Chatekar, 2015, s. 3-4).

Jako další z mapperů byl pro porovnání vybrán Entity Framework. Jedná se o sadu technologií v ADO.NET, které podporují vývoj datově orientovaných softwarových aplikací. Protože Entity Framework je komponentou .NET Frameworku, aplikace využívající Entity Framework mohou být spouštěny na jakémkoli počítači s podporou .NET Frameworku (Microsoft, 2016e).

6.3.1 NHibernate 4.0

Veřejně dostupná se verze NHibernate 4.0.3.400 stala 17. srpna 2014. Tato verze je poměrně důležitá z mnoha úhlu pohledu. Jedná se o hlavní vydání téměř po třech a půl letech, a to znamená, že přináší značné změny a opravy chyb (Suhas Chatekar, 2015, s. 5). V projektu je sice využívána verze 4.0.4, ale oproti verzi 4.0.3 nepřináší, kromě oprav chyb při práci s C# překladačem "Roslyn" v Microsoft Visual Studio 2015, žádné změny. Tudíž je zajímavější popsat předešlé změny.

- NHibernate 4.0 je vytvořen pro .NET 4.0. Uživatelé, ale mohou i nadále využívat starší verze NHibernate v projektech postavených na .NET 4.0. Nicméně tato verze přidává vývojářům možnost nové funkce .NET 4.0. Hlavní funkcí této verze je nezávislost na knihovně *lesi.Collections*, která u dřívějších verzí podporovala *lesi.Collections.Generic.ISet<T>*, protože .NET 4.0 má nativní podporu pro sety ve formě *ISet<T>* a implementace jako je *HashSet<T>*. Dříve, když starší verze .NET neměly podporu pro sety, *lesi.Collections* knihovny poskytovaly vospělou funkcionalitu pro sety. NHibernate závisel na této knihovně, protože využívá sety pro znázornění kolekcí.
- Díky „mapování kódem“ není třeba psát a zpravovat těžkopádné mapovací XML soubory. Nicméně uživatelé nebyli s touto funkcí spokojeni, dokud verze NHibernate 4.0 neopravila několik hlavních problémů a dodala celkovému *mapování kódem* facelift (Suhas Chatekar, 2015, s. 5).

6.3.2 Entity Framework 7

Entity Framework 7 je výkonnější a obsahuje značné změny oproti jeho předchůdci Entity Frameworku 6.x. V této verzi jsou ponechány vývojářské funkce jako v předešlé, mezi tyto funkce se řadí DbContext, DbSet a další. Nová verze je odlehčena o přebytečný kód, proto je daleko rychlejší a spolupracuje s cloudem pomocí ASP.NET vNext na Windows zařízeních stejně jako je tomu u klasických .NET scénářů.

Mezi hlavní vlastnosti Entity Frameworku 7 se řadí:

- Entity Framework spolupracoval především s relačními databázemi. Nyní však přináší velkou podporu „Nerelačních“ databází jako je například *Azure table storage*.
- Mezi další zajímavé vlastnosti patří možnost pracovat s více klíči nad jednou tabulkou. Kromě primárního klíče tak lze pracovat i s alternativními klíči.
- Pro zachycení veškeré interakce s databází přichází Entity Framework 7 se zabudovanou protokolovací funkcí. Tato funkce je postavena na `Microsoft.Framework.Logging` a implementovanou `Microsoft ILoggingFactory`, která podporuje všechny .NET platformy.
- Jako další zajímavou funkci je nutno zmínit „Stínové vlastnosti“. Jedná se o vlastnosti, které neexistují v entity třídách, ale jsou považovány za jejich součásti. Hodnota a stav těchto vlastností jsou udržovány pouze ve sledování změn. Tyto vlastnosti se mohou účastnit všech databázových operací, ale neměly by být dostupné přes entity třídy zbytku aplikace. (Bhavik Patel, 2015).

6.3.3 Výběr ORM Frameworku

- Entity Framework je těsně spojen s databází. Při vytváření aplikace s jeho použitím je stále nutné pracovat s databázovými výrazy jako například s cizími klíči, omezeními a tabulkovými vztahy. Tato nevýhoda je u projektu menšího rozsahu nepatrná.
- + Říká se, že NHibernate je daleko před Entity Frameworkem z hlediska funkcí. NHibernate má mnoho funkcí, které Entity Framework nemá. Mezi tyto funkce patří vyrovnávací paměť druhé úrovně, konkurenční strategie, bohaté možnosti mapování atd.
- Pravděpodobně jediná funkce, kterou nepodporuje NHibernate jsou asynchronní databázové operace. Entity Framework tyto operace podporuje (Vladimir Khorikov, nedatováno).
- + Dlouholetá zkušenost s NHibernate

Při rozhodování zda použít Entity Framework či NHibernate byl brán zřetel hlavně na zkušenosti s danými prostředími. Jelikož oba frameworky jsou prakticky na stejné úrovni, bylo rozhodnuto zejména na základě vlastních zkušeností s danými frameworky. Byl tedy vybrán framework NHibernate.

6.4 Uživatelské role

V aplikaci je použit role management pro přístupy na základě rolí. Celkem je možné aplikaci prohlížet čtyřmi způsoby. Základním způsobem je zobrazení pro nepřihlášené uživatele. Takto jsou znázorněny veškeré kapitoly a informace o výuce. Na hlavní stránce se uživateli zobrazí novinky a změny ve výuce předmětu „Principy počítačů“. Bez přihlášení uživatel nemůže spouštět cvičné testy.

Po registraci je uživateli vytvořen uživatelský účet s uživatelskou rolí „student“. Po jeho aktivaci, vyučujícím, či administrátorem, je možné se pod tímto účtem přihlásit a studentovi je zobrazeno více možností než nepřihlášenému uživateli. Mezi tyto volby patří zobrazení dalších tlačítek v menu, nebo podstupování cvičných testů. Kromě položky „kapitoly“ jsou přidány položky „testy“, ve kterých jsou zachyceny všechny testy vygenerované pro přihlášeného uživatele a položka se jménem přihlášeného uživatele, která odkazuje na detailní zobrazení informací ohledně daného účtu. V detailním zobrazení jsou uživateli zpřístupněny možnosti pro editaci účtu.

Další uživatelskou rolí je „vyučující“. V tomto zobrazení jsou do menu přidány další položky, jako jsou „články“, které umožňují správu novinek a změn ve výuce. Položka „kapitoly“ zobrazuje jiný pohled, než pro nepřihlášeného uživatele či studenta. Vyučujícímu je zobrazen pohled na správu kapitol. Je možné tedy přidávat a editovat kapitoly. Další přidanou položkou jsou „otázky“. Tato položka umožňuje správu otázek, ze kterých jsou následně generovány cvičné testy pro studenty. Stejně jako u studentského pohledu i učitel si může zobrazit testy. Na rozdíl od studenta, ale vidí testy vygenerované všemi studenty. Další způsob jak zobrazit testy je přes položku „studenti“, ve které si lze zobrazit výpis studentů s jejich podrobnými informacemi, testy a dosaženým průměrem. Při zobrazení studentů je možno aktivovat deaktivované, či nepotvrzené uživatelské účty.

Poslední uživatelskou rolí a zároveň rolí s největšími právy je role „administrátor“. Tento pohled je ve své podstatě úplně stejný jako pohled přes účet s uživatelskou rolí „vyučující“. Jediným rozdílem je zobrazení uživatelů. Zatímco učitel vidí pouze výpis studentů s jejich podrobnými informacemi a bez možnosti jejich úpravy, administrátorovi je umožněno zobrazit si všechny uživatele s možností přidání nového uživatele a s editací uživatelských rolí.

Hlavní rozdíly mezi rolemi jsou v zobrazeném menu. Tyto difference jsou vidět na obrázku 14.

Obrázek 14 – Rozdíly mezi zobrazením menu pro jednotlivé uživatelské role

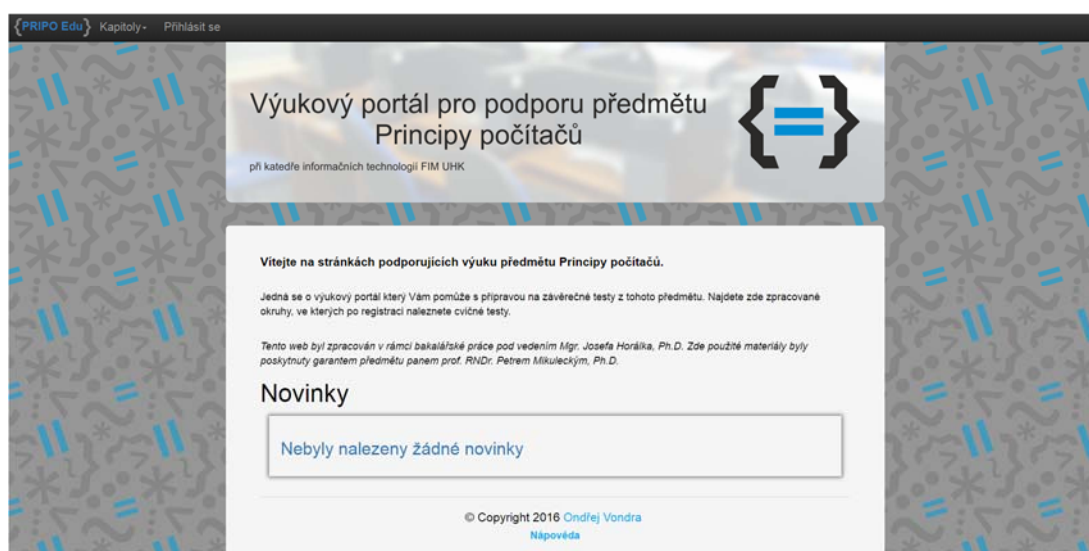


Zdroj: Vlastní tvorba

6.5 Popis aplikace

V aplikaci je použit vzhled z cepsos.cz, kvůli budoucímu přesunu aplikace pod tuto doménu. Aplikace je založena na tomto layoutu, je tedy použito pozadí a hlavička stránky. Veškeré stylování je provedeno pomocí Bootstrap Frameworku. Jeho využití značně usnadňuje tvorbu responzivního moderního webu. Umožňuje totiž využití značného množství předdefinovaných elementů. Jak již bylo zmíněno, celý web je tvořen za použití responzivního designu. To umožňuje práci s aplikací jak na klasickém desktopovém počítači, tak i na většině mobilních zařízení. Na obrázku 15 je vidět pohled na aplikaci po prvním vstupu.

Obrázek 15 – Vzhled webové aplikace (Nepřihlášený uživatel)



Zdroj: Vlastní tvorba

Pro zabezpečení přenosu všech zadaných údajů je použita metoda POST. Před přenosem zadaného hesla, ať už při registraci, či při přihlašování, je toto heslo hashováno pomocí PBKDF2 (Password-Based Key Derivation Function 2). Jedná se o standard, jenž

přidává k zadanému heslu „salt“, což je posloupnost bitů známá jako „cryptographic salt“. Heslo s přidaným „salt“ je přeneseno a uloženo v databázi.

Pro stránkování a vyhledávání jsou použity asynchronní operace pomocí AJAXu. Jak již bylo zmíněno, díky těmto operacím je zajištěn rychlejší běh aplikace, neboť se neaktualizuje celá stránka, ale jen její obsah.

7 Přístupy k vytvoření cvičných testů

Testové otázky jsou přidávány vyučujícím. Každá otázka připadá k jedné vytvořené kapitole předmětu „Principy počítačů“. Aplikace je schopna rozpoznat dva druhy testových otázek. Konkrétně se jedná o otázky zaškrťovací s více správnými možnostmi a doplňovací, kdy student musí zadat správnou odpověď, převážně ve formě jednoho slova. K výpisu všech otázek mají přístup pouze vyučující a administrátor, jak bylo zmíněno v kapitole 6.4 („Uživatelské role“). Student se k otázkám dostane pouze formou testů.

Testovací systém je v aplikaci vytvořen na základě fungujícího školního modelu. Jedná se tedy o hodnocení podle dosaženého procentuálního výsledku. Podrobněji je hodnotící systém zobrazen v tabulce 8. Pro každou kapitolu je náhodně vygenerovaný cvičný test podle zadaného počtu otázek. Testy jsou náhodně generovány z databáze otázek. Pro každou kapitolu jsou vybrány otázky pouze pro tuto kapitolu a při generování je zajištěna unikátnost otázek v testu. Aby student prokázal znalosti dané kapitoly, musí dokončit vygenerovaný test pro tuto kapitolu nejméně pětkrát a s průměrem 2 a lepším. Pokud student zodpoví libovolnou otázku špatně, ve výsledném zobrazení neuvidí správnou odpověď testů, nýbrž je vyzván k pečlivějšímu prostudování dané kapitoly. Po dokončení je kapitola označena jako dokončená a studentovi se znepřístupní testy pro tuto kapitolu. Může ovšem stále prohlížet své již vyplněné testy. Po dokončení všech kapitol je studentovi zpřístupněn finální celkový test ze všech kapitol, po jehož splnění je řádně připraven splnit zkoušku z tohoto předmětu. Pro tento test není stanoveno žádné omezení a student jej může spustit i po úspěšném splnění vícekrát.

Tabulka 8 – Hodnotící systém

Výsledek	Známka
100 – 90 %	1
89 - 80 %	1.5
79 - 70 %	2
69 - 60 %	2.5
59 - 50 %	3
49 – méně %	4

Zdroj: Vlastní tvorba

Zpětná vazba pro vyučujícího je zřízena formou zobrazení výsledků studenta. Vyučující si může zobrazit všechny testy a jejich výsledky nebo pouze testy pro daného studenta. Vyučující vidí studentovy odpovědi a zároveň i odpovědi správné. Toto zobrazení platí pouze pro otázky typu zaškrťovací a doplňovací.

Samotné otázky byly poskytnuty garantem předmětu „Principy počítačů“, panem prof. RNDr. Petrem Mikuleckým, Ph.D.

8 Výběr hostingu

Při tvorbě webové aplikace bylo nutné zvážit možnosti hostování aplikace. Jelikož je aplikace vytvořena především pro studenty Fakulty informatiky a managementu na Univerzitě Hradec Králové, byl vybrán hosting pod doménou cepsos.cz. Bohužel v době vytváření této práce hosting cepsos.cz nepodporuje zvolené technologie a je tedy nutné zvolit jiného poskytovatele webového hostingu s podporou použitých technologií. Mezi vybrané poskytovatele patří ASPone s.r.o. a ASP2.cz. Při výběru hostingu byl brán zřetel především na cenu. Z tohoto důvodu byly zvoleny jen tyto dva webhostingy. Ostatní nabízely pouze free trial verze s krátkou zkušební dobou. Dále byli, z důvodu odezvy, vybráni jen poskytovatelé z České republiky. Při přístupu na zahraniční servery by nemusel být zaručen plynulý chod aplikace.

8.1 ASPone s.r.o.

Jedná se o dynamicky se rozvíjející kreativní firmu nabízející webový hosting jak v České republice, tak v zahraničí. Firma se počátkem roku 2007 jako první v České republice objevila na trhu s tehdy novou technologií Windows Server 2008, IIS 7, ASP.NET Framework 3.5 a s dalšími moderními technologiemi. Společnost však má s technologií ASP.NET zkušenosti již od prvního vydání Microsoft .NET Framework 1.0 Beta2. Nyní jsou domovem více než tisíce webových projektů, ať už od drobných podnikatelů, jednotlivců či velkých firem. Díky zkušenostem a podpoře patří mezi největší poskytovatele webhostingu s podporou technologie ASP.NET v České republice.

Společnost se v roce 2011 stala členem CZ-NIC a dostala akreditaci, díky čemuž má přímý přístup k centrálnímu registru českých domén. Specializuje se především na společnost Microsoft®, což zajišťuje nejaktuálnější verze všech poskytovaných moderních technologií. Mezi tyto technologie patří ASP.NET, MVC, IIS, MSSQL a další. Kromě webhostingu firma nabízí i spolehlivý serverhosting ve formě dedikovaných či virtuálních serverů. Nyní se firma specializuje na tvorbu, provoz a správu vlastních CLOUD řešení podle toho co od nich zákazník požaduje. Samozřejmostí u těchto řešení jsou nejvyšší technologie Supermicro, DELL, Hewlett-Packard a Cisco (ASPone, 2007-2016).

8.2 ASP2.cz

Jedná se o hosting spadající pod projekt czechia.com provozovaný společností ZONER software, a.s. V roce 1996 společnost založila oddělení internetových služeb a spustila projekt czechia.com. Jedná se o první společnost vlastníci .com doménu (CZECHIA,

nedatováno). První zmínky o tomto projektu jsou z roku 2005, kdy společnost Zoner poprvé spustila svůj připravovaný projekt s názvem asp2.cz . První verze byla spuštěna na operačním systému Windows Server 2003 SP1, podporovala IIS 6.0, .NET Framework 2.0 a databázi MS Server 2005. Téhož roku přišlo mnoho dalších změn a vylepšení již fungujícího systému. Za zmínku stojí přidání utility statistiky pro konkrétní domény, kde jsou zobrazeny návštěvnosti a celková datová náročnost konkrétní webové aplikace.

Jedním z dalších důležitých roků se stal rok 2007, kdy přišla podpora AJAX a ASP.NET technologie. Postupem tohoto roku se začala zavádět nová verze .NET Frameworku, konkrétně verze .NET Framework 3.5, která byla nasazena v půlce roku jako beta2 verze a koncem téhož roku jako ostrá verze.

V roce 2008 přišel jeden z větších upgrade systému, v podobě migrace na operační systém Windows server 2008. S tímto upgradem přišlo i pár změn v podobě nasazení nové SQL databáze SQL Server 2008 a přechod na IIS7. Téhož roku byla spuštěna testovací verze ASP.NET MVC ve verzi beta.

Nyní je projekt ASP2.cz provozován na operačním systému Windows Server 2012. Jsou použity nejnovější technologie jak pro aplikace, tak pro databáze. Aktuální nasazená databáze je MS SQL Server 2016, ta spolu s .NET Frameworkem 4.6 a IIS 8.5 tvoří kvalitní základ pro výborný hosting webových aplikací na platformě ASP (ASP2.cz, 2005-2013).

8.3 Porovnání poskytovatelů

Jak již bylo zmíněno, poskytovatelé webového hostingu byli vybráni především na základě ceny a odezvy. Z tohoto důvodu byla tato dvě kritéria vynechána z výsledného porovnání. Všechny informace pro porovnání byly čerpány z oficiálních stránek vybraných poskytovatelů. Veškerá kritéria jsou bodově hodnocena od jedné do pěti a poskytovatel s nejvíce body je zvolen jako vyhovující.

Prvním porovnávaným kritériem je velikost poskytovaného úložiště jak pro aplikaci, tak pro data. Velikosti úložiště jsou v tabulce 9 znázorněny pod jedním průměrovaným výsledkem. Zatímco ASP2.cz poskytuje pouhých 25 MB diskového prostoru pro aplikaci a potřebné soubory, ASPone nabízí 40 MB diskového prostoru. Z pohledu prostoru pro uložení databáze ASP2.cz nabízí sice v součtu stejný diskový prostor, tedy 30 MB, ale tento prostor je na rozdíl od ASPone rozdělen do dvou základních částí každé databáze tedy do logového souboru (10 MB) a do datového souboru (20 MB) pro uložení samotných dat.

Z obou pohledů, ať už z pohledu velikosti diskového prostoru, či možné velikosti databáze je ASPone ve velké převaze oproti ASP2.cz.

Jako další hodnotící kritérium byla zvolena aktuálnost technologií. ASP2.cz nabízí opravdu moderní technologie na to, že se jedná o free hosting. Provoz hostingu je nasazen na Windows Serveru 2012. Ten je oproti porovnávanému konkurenčnímu free hostingu, který běží na Windows Serveru 2008 R2, mnohem modernější. Podpora .NET Frameworku je prakticky stejná u obou poskytovatelů. Při pohledu na databázové technologie jsou ale vidět drobné rozdíly. ASPone v této verzi hostingu nabízí pouze MS SQL Server 2012, zatímco ASP2.cz může nabídnout podporu nejaktuálnější verze MS SQL Server 2016. Jelikož je podpora požadovaných technologií na vyrovnané úrovni s menší převahou ASP2.cz, bodové ohodnocení je tomu přizpůsobeno.

Posledním důležitým kritériem je zkušenost autora s klientským prostředím obou poskytovatelů. Zkušenosti autora se vztahují především k hostingu od ASPone z důvodu hostování předešlých projektů. Při pohledu na klientské prostředí je v převaze co do přehlednosti ASPone. U ASP2.cz je prostředí poněkud zmatečné. Z tohoto důvodu víc bodů, tedy 5, dostává ASPone. ASP2.cz z důvodu malé zkušenosti získává pouze 3 body.

Z tabulky 9 je zřetelné, že oba poskytovatelé nabízí kvalitní hosting a hlavními důvody zvolení ASPone jako vyhovujícího poskytovatele jsou velikost úložiště a vlastní zkušenosti. Výsledná zaregistrovaná doména je tedy pripo.aspone.cz (ASPone, 2007-2016; ASP2.cz, 2005-2013).

Tabulka 9 – Porovnání poskytovatelů webhostingu

Poskytovatel	Velikost úložiště	Aktualita technologií	Vlastní zkušenost	Výsledek
ASPone s.r.o.	5	4	5	14
ASP2.cz	3	5	3	11

Zdroj: Vlastní tvorba za použití oficiálních zdrojů

9 Návrh optimální integrace do LMS Blackboard

Jelikož je webová aplikace vytvořena na podobné bázi jako LMS Blackboard, její integrace do systému není úplně nutná. Díky testovacímu systému je aplikace samostatnou výukovou pomůckou, která pomůže studentům při studiu. Její samotná integrace do LMS Blackboard by tedy mohla být provedena pouze jako odkaz na výukový materiál spadající pod univerzitní studijní materiály. Stejnou formou pak může být aplikace integrována ať už do přednášek či jiných učitelských výukových materiálů nabízených studentům.

9.1 LMS Blackboard

Koncem roku 2010 se Blackboard, Inc. prosadili jako vůdci mezi LMS (Learning Management Systém) v K-12 a vyšších výukových prostředích. Růst systému zajistilo jeho sloučení s CourseInfo v roce 1999. Akviziční strategie Blackboard má za následek dalekosáhlé produktové řady, které zahrnují výukové systémy LMS.

Jednou z výhod tohoto systému je bezproblémová integrace stupňovaných úkolů a komunikace s hodnotícím centrem. Když instruktor vytvoří známkovanou úlohu na obsahové stránce nebo na diskusní tabuli, tyto úlohy jsou generovány automaticky hodnotícím centrem.

První LMS byl vytvořen v roce 1996 WebCT a jednalo se o největší konkurenci pro Blackboard. Byl znám pro svou možnost být upraven vyučujícími. Jedná se také o komplexní program s velkým množstvím kontrolních panelů pro vytváření výukového obsahu. Jednoduše charakterizovat tyto dva LMS lze takto: Blackboard je přívětivý a jednoduše použitelný, zatímco WebCT je výrazně sofistikovanější.

Kromě základní verze systému existují i přídavné moduly jako je větší Blackboard obsahový systém, komunitní systém a výsledkový systém. Když jsou k Blackboardu přidány moduly obsahového a komunitního systému, přidávají institučně a výukově založený obsah (Yefim Kats, 2010, s. 8-9).

10 Závěr

V této bakalářské práci je popsán postup při tvorbě výukové webové aplikace pro podporu předmětu „Principy počítačů“ vyučovaným na Fakultě informatiky a managementu na Univerzitě Hradec Králové. Práce je rozdělena do tří hlavních částí.

Prvním krokem byla analýza problémových partií. Ta byla provedena na základě informací o všech výsledcích studentů získaných z informačního systému STAG. Z těchto výsledků byl následně vypočten celkový průměr, čímž bylo zjištěno, že výsledky jsou opravdu potřeba zlepšit. Většina studentů se nejspíše spokojí se známkou E, než by se pokusila látku lépe pochopit.

Další částí práce jsou již zmíněné animace. Nejprve bylo nutné vybrat prostředí pro tvorbu těchto animací a poté přejít k samotné tvorbě. Všechny animace byly vytvořeny technologií Gif za pomoci grafického editoru GIMP. Tento postup byl zvolen na základě porovnání provedeném v kapitole 3.4 („Porovnání“). Konkrétní případy animací spolu s jejich popisem jsou zobrazeny v kapitole 4 („Tvorba animací pro vybrané partie předmětu“).

Jednou z nejdůležitějších částí před samotnou tvorbou je výběr technologií, ve kterých bude výsledná aplikace vytvořena. Tento výběr se dá rozčlenit na výběr technologie pro tvorbu webové aplikace a tvorbu databáze spojené s touto aplikací.

Volba technologií pro tvorbu webové aplikace byla provedena vybráním nejznámějších a nejpoužívanějších technologií v tomto oboru. Po výběru byly tyto technologie krátce představeny a porovnány na základě zadaných kritérií, které si autor sám zvolil. Porovnání bylo vyhodnoceno, stejně tak jako v případě animací, na základě bodového hodnocení a technologie s nejvíce body byla doporučena k použití. Touto technologií byl ASP.NET od společnosti Microsoft, který byl nakonec použit pro tvorbu výukového systému.

Výběr databázové technologie byl proveden podobně jako výběr technologie pro tvorbu webové aplikace. Byly tedy vybrány nejpoužívanější databázové technologie, s přihlédnutím na předpokládaný rozsah projektu, a ty byly opět jako technologie pro aplikaci stručně představeny. Po tomto představení již bylo možné vypracovat porovnání. Toto porovnání bylo stejně důležité jako srovnání v kapitole 5.1.6 („Porovnání technologií“), jelikož výsledná databáze musí spolupracovat s aplikací a být schopná pojmout rozsah projektu. Kritéria v hodnocení byla tedy zvolena velmi pečlivě. Technologie

s nejvíce body byla opět doporučena k využití coby nejvhodnější. Vybranou technologií pro tvorbu databáze byl Microsoft SQL Server, především díky skutečnosti, že tato technologie pro tvorbu webové aplikace je ASP.NET, a tudíž spolu jdou velmi snadno použít, jelikož jsou obě od společnosti Microsoft.

Samotná tvorba webové aplikace byla doprovázena dalšími pomocnými úkony, jako je například výběr způsobu přístupu k datům, jenž byl vyhodnocen jako důležitá část, a proto byl krátce představen. Zvolený způsob je přístup pomocí ORM Frameworku, který umožnil autorovi oprostít se od psaní SQL příkazů. Jelikož se jako moderní způsob tvorby webových aplikací ukázalo použití architektury MVC, tak i ta byla krátce představena se svými výhodami a bylo ukázáno, jak byla použita v projektu. Hlavní částí návrhu aplikace bylo vytvoření analytického modelu, který byl později rozpracován do fyzického modelu databáze, z něž byla databáze vygenerována. Jednou z posledních částí návrhu aplikace byl testovací systém, jenž je podrobněji popsán v kapitole 7 („Přístupy k vytvoření cvičných testů“). Po připravení všech pomocných úkonů nastal čas pro samotnou tvorbu. Celková tvorba a výsledná aplikace je popsána v kapitole 6 („Návrh struktury webové prezentace“).

Poslední zásadní část práce je věnována výběru hostingu pro chod aplikace. Vzhledem k tomu, že aplikace byla sice vytvořena tak, aby jí bylo možné hostovat pod doménou cepsos.cz., bohužel v době vytvoření této práce tento hosting nebylo možné použít, protože nepodporoval vybranou technologii webové prezentace. Z tohoto důvodu bylo tedy třeba vybrat jiného poskytovatele webového hostingu. Jelikož byl brán zřetel v první řadě na cenu a odezvu, byli vybráni poskytovatelé z České republiky nabízející hosting podporující zvolené technologie zdarma. Ti byli krátce představeni a následně porovnání v kapitole 8 („Výběr hostingu“). Z tohoto porovnání byl zvolen poskytovatel aspone.cz. Výsledná adresa portálu je tedy pripo.aspone.cz.

Po dokončení a spuštění aplikace na vybraném hostingu bylo třeba zpracovat kapitoly, které jsou obsaženy na webu společně s vytvořenými animacemi. Tyto kapitoly byly vypracovány na základě poskytnutých přednášek od pana prof. RNDr. Petra Mikuleckého, Ph.D. Tyto přednášky byly upraveny a doplněny o již zmíněné animace. Kromě kapitol bylo nutné vytvořit i testovací otázky. Ty byly taktéž poskytnuty panem prof. RNDr. Petrem Mikuleckým, Ph.D. a byly tedy pouze upraveny a poté vloženy do webové aplikace.

Poslední částí byl návrh optimální integrace do LMS Blackboard. Jelikož vytvořená aplikace je soběstačný výukový prvek, byla tato integrace doporučena pouze v podobě hypertextového odkazu na stránku aplikace.

Cílem této práce bylo vytvořit výukovou aplikaci, která pomůže studentům Fakulty informatiky a managementu na Univerzitě Hradec Králové se studiem předmětu „Principy počítačů“, který jim v předchozích letech dělal problémy. Tento cíl práce byl splněn, avšak zda a v jakém rozsahu bude vytvořená aplikace splňovat svůj účel, se ukáže až časem.

11 Použitá literatura

11.1 Seznam literatury

AHMED, M. A KOLEKTIV 2002: ASP.NET. Web Developer's Guide. Rockland: Syngress Publishing Inc. ISBN 1-928994-51-2.

BERMAN, J. J. 2011: Perl: the Programming Language. Burlington: Jones & Bartlett Learning. ISBN 978-1-449-67389-5.

BRUNNER, R. J. 2003: Practical Guide to JavaServer Pages. San Francisco: Morgan Kaufmann Publishers. ISBN 1-55860-836-2.

DUBOIS, P. 2009: MySQL. Fourth Edition. Boston: Pearson Education, Inc. ISBN 978-0-672-32938-8.

FORTA, B. 2007: Sams Teach Yourself Microsoft SQL Server T-SQL in 10 Minutes. Boston: Pearson Education, Inc. ISBN 978-0-768-69000-2.

CHATEKAR, S. 2015: Learning NHibernate 4. Birmingham: Packt Publishing Ltd. ISBN 978-1-78439-356-4.

KATS, Y. 2010: Learning Management System Technologies and Software Solutions for Online Teaching: Tools and Applications. Hershey: IGI Global. ISBN 978-1-61520-853-1.

MACDONALD, M. 2012: Pro Silverlight 5 in C#. New York: Apress. ISBN 978-1-4302-3479-1.

PAZ, J. R. G. 2013: Beginning ASP.NET MVC 4. New York: Apress. ISBN 978-1-4302-5752-3.

PEACOCK, A. 2002: Animation for multimedia and new media. In: M. Kuperberg, Guide to Computer Animation: for tv, games, multimedia and web. 2002, Woburn: Focal Press, 91-121. ISBN 0-240-51671-0.

PECK, A. 2008: Beginning GIMP: From Novice to Professional. Second Edition. New York: Apress. ISBN 978-1-4302-1070-2.

PRAYAGA, L. a SURI, H. 2008: Begening Game Programming with Flash. Boston: Thomson Course Technology PTR. ISBN 978-1-59863-398-6.

SUMMERFIELD, M. 2010: Programming in Python 3. A complete introduction to the Python language. Second Edition. Boston: Pearson Education, Inc. ISBN 978-0-321-68056-3.

ULLMAN, L. 2012: PHP and MySQL for Dynamic Web Sites. Forth Edition. Berkeley: Peachpit Press. ISBN 978-0-321-78407-0.

11.2 Seznam pramenů

ABOUT GIMP (online). GIMP, 2001-2016 (citace srpen, 2., 2016). Přístup z internetu: URL: <http://www.gimp.org/about>

ABOUT MYSQL (online). MySQL, 2016 (citace srpen, 2., 2016). Přístup z internetu: URL: <https://www.mysql.com/about>

ABOUT (online). PostgreSQL, 1996-2016 (citace srpen, 2., 2016). Přístup z internetu: URL: <https://www.postgresql.org/about>

ABOUT (online). Python, 2001-2016 (citace srpen, 2., 2016). Přístup z internetu: URL: <https://www.python.org/about>

ABOUT SILVERLIGHT (online). Microsoft Silverlight, nedatováno (citace srpen, 2., 2016). Přístup z internetu: URL: <https://www.microsoft.com/silverlight/what-is-silverlight>

ABOUT SQLITE (online). SQLite, nedatováno (citace srpen, 2., 2016). Přístup z internetu: URL: <http://sqlite.org/about.html>

ADOBE FLASH PLAYER (online). Adobe, 2016a (citace srpen, 2., 2016). Přístup z internetu: URL: <http://www.adobe.com/software/flash/about>

ADOBE FLASH PLAYER DISTRIBUTION (online). Adobe, 2016b (citace srpen, 2., 2016). Přístup z internetu: URL: <http://www.adobe.com/cz/products/players/flash-player-distribution.html>

ANDREWS, S. 2007-14: Learning To Program With Perl (online). Babraham Bioinformatics, 2016 (citace srpen, 2., 2016). Přístup z internetu: URL: <http://www.bioinformatics.babraham.ac.uk/training/Perl%20Introduction.pdf>

COWBURN, P. ed. 2016: PHP Manual (online). PHP, 2001-2016 (citace srpen, 2., 2016). Přístup z internetu: URL: <http://php.net/manual/en/index.php>

ENTITY FRAMEWORK OVERVIEW (online). Microsoft, 2016e (citace srpen, 2., 2016). Přístup z internetu: URL: <https://msdn.microsoft.com/en-us/library/bb399567.aspx>

CHANGELOG (online). ASP2.cz, 2005-2013 (citace srpen, 2., 2016). Přístup z internetu: URL: <https://www.asp2.cz/Archiv.aspx?Type=1>

INTRODUCTION TO WPF IN VISUAL STUDIO 2015 (online). Microsoft, 2016a (citace srpen, 2., 2016). Přístup z internetu: URL: [https://msdn.microsoft.com/en-us/library/aa970268\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/aa970268(v=vs.110).aspx)

JAVASERVER PAGES TECHNOLOGY (online). Oracle, nedatováno (citace srpen, 2., 2016). Přístup z internetu: URL: <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>

KHORI KOV, V. nedatováno: Entity Framework 6 (7) vs NHibernate 4: DDD perspective (online). Enterprise Craftsmanship, nedatováno (citace srpen, 2., 2016). Přístup z internetu: URL: <http://enterprisecraftsmanship.com/2014/11/29/entity-framework-6-7-vs-nhibernate-4-ddd-perspective/>

KURZY VYHLAŠOVANÉ ČNB (online). Česká národní banka, 2016 (citace červenec, 18., 2016). Přístup z internetu: URL: <http://www.cnb.cz/cs/index.html>

LEARN (online). Learn Perl, 2002-2016 (citace srpen, 2., 2016). Přístup z internetu: URL: <https://www.perl.org/learn.html>

O WEBHOSTINGU CZECHIA.COM (online). CZECHIA, nedatováno (citace srpen, 2., 2016). Přístup z internetu: URL: <https://www.czechia.com/clanek/czechia>

PATEL, B. 2015: 10 Entity Framework 7 Features That You Must Know (online). C# Corner, 2016 (citace srpen, 2., 2016). Přístup z internetu: URL: <http://www.c-sharpcorner.com/UploadFile/55d96a/10-entity-framework-7-features-that-you-must-know>

PORTÁL IS/STAG (online). STAG UHK, nedatováno (citace srpen, 2., 2016). Přístup z internetu: URL:

https://stag.uhk.cz/portal/studium/prohlizeni.html?pc_pagenavigationalstate=H4sIAAAAAA AAAAGNgYGBkYDMytDQxNhdmZADxOIpLEktSvVMrwTwRXUsjI2NjcyMDYzML UxNzc1MDcyOgDAMAgbcppjoAAAA*#prohlizeniSearchResult

PROFIL SPOLEČNOSTI (online). ASPone, 2007-2016 (citace srpen, 2., 2016). Přístup z internetu: URL: <http://www.aspone.cz/cz/O-nas/Profil-spolecnosti>

SQL SERVER 2016 PŘEHLED (online). Microsoft, 2016d (citace srpen, 2., 2016). Přístup z internetu: URL: <https://www.microsoft.com/cs-cz/server-cloud/products/sql-server>

VISUAL BASIC CONCEPTS (online). Microsoft, 2016b (citace srpen, 2., 2016). Přístup z internetu: URL: [https://msdn.microsoft.com/en-us/library/aa733738\(v=vs.60\).aspx](https://msdn.microsoft.com/en-us/library/aa733738(v=vs.60).aspx)

WEB APPLICATIONS (ASP.NET) (online). Microsoft, 2016c (citace srpen, 2., 2016). Přístup z internetu: URL: <https://msdn.microsoft.com/en-us/library/aa286485.aspx>

WEBHOSTING (online). Hostingy.cz, 2003-2016 (citace srpen, 2., 2016). Přístup z internetu: URL: <http://www.hostingy.cz/webhosting.html>

Příloha A – Podklad pro zadání bakalářské práce

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2014/2015

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Vondra Ondřej	Skála 41, Boháňka - Skála	I1301386

TÉMA ČESKY:

Webového portálu pro podporu předmětu PRIPO

TÉMA ANGLICKY:

Web portal to support the subject PRIPO

VEDOUcí PRÁCE:

Mgr. Josef Horálek, Ph.D. - KIT

ZÁSADY PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit webový portál pro podporu výuky předmětu Principy počítačů (PRIPO) na FIM UHK. V teoretické práci autor provede analýzu problematických partií předmětu PRIPO, na základě vyhodnocení výsledků závěrečných testů. Tyto partie zpracuje s využitím aktuálních technologií pro tvorbu webu a vhodných animací a tyto technologie představí. V praktické části pak autor vytvoří webový portál pro podporu předmětu PRIPO a navrhne jeho optimální integraci do LMS systému Blackboard.

Osnova práce:

Úvod

Představení stěžejních oblastí předmětu PRIPO

Analýza problematických oblastí

Návrh webového portálu

Představení realizace webového portálu


Závěr

SEZNAM DOPORUČENÉ LITERATURY:

HENNESSY, John L a David A PATTERSON. Computer architecture: a quantitative approach. 5th ed. Amsterdam: Elsevier, c2011, 1 sv. (různé stránkování). ISBN 978-0-12-383872-8.

SCHOCKEN, Noam Nisan and Shimon a David A PATTERSON. The elements of computing systems: building a modern computer from first principles. 1st MIT Press pbk. ed. Cambridge, Mass: MIT, 2008, 1 sv. (různé stránkování). ISBN 978-026-2640-688.

Podpis studenta:



Datum:

29.9.2015

Podpis vedoucího práce:



Datum:

29.9.2015