

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky

Návrh části IS pro PS Hroch Pardubice

Zuzana Pávová

Bakalářská práce
2012

Univerzita Pardubice
Fakulta ekonomicko-správní
Akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Zuzana Pávová**
Osobní číslo: **E09120**
Studijní program: **B6209 Systémové inženýrství a informatika**
Studijní obor: **Informatika ve veřejné správě**
Název tématu: **Návrh části IS pro PS Hroch Pardubice**
Zadávací katedra: **Ústav systémového inženýrství a informatiky**

Zásady pro vypracování:

- 1) Zkoumání současného stavu, potřeb a požadavků PS Hroch
- 2) Možnosti zpracování agendy, návrh řešení vybraných funkcionalit
- 3) Implementace a zhodnocení tohoto návrhu

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] ŠIMONOVÁ, Stanislava. Databázové systémy I . Vyd. 1. Pardubice : Univerzita Pardubice, 2007. 106 s. ISBN 978-80-7194-988-6.
- [2] HUB, Miloslav. Technologie internetu - PHP 5. Vyd. 1. Pardubice : Univerzita Pardubice, 2009. 88 s. ISBN 978-80-7395-163-4.
- [3] Pohybové Studio Hroch [online]. 2010 [cit. 2011-06-29]. POHYBOVÉ STUDIO HROCH. Dostupné z WWW: <<http://pshroch.ddmstolbova.com/>>.

Vedoucí bakalářské práce:

Ing. Milan Tomeš

Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce: **3. října 2011**

Termín odevzdání bakalářské práce: **30. dubna 2012**

doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.

doc. Ing. Jiří Křupka, Ph.D.

vedoucí ústavu

V Pardubicích dne 3. října 2011

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byla jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako Školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 30. 4. 2012

Zuzana Pávová

PODĚKOVÁNÍ:

Tímto bych ráda poděkovala svému vedoucímu práce Ing. Milanu Tomešovi za jeho odbornou pomoc, cenné rady a poskytnuté materiály, které mi pomohly při zpracování bakalářské práce, a za přístup ke školnímu serveru cipisek.upce.cz. Dále bych ráda poděkovala DDM Beta Pardubice, za poskytnuté materiály a informace.

ANOTACE

Práce se zabývá návrhem a následnou implementací části informačního systému pro PS Hroch Pardubice. Nejdůležitější částí návrhu je vytvoření elektronického formuláře pro přihlašování dětí do kurzů, a návrh aplikace pro práci s údaji přihlášených dětí. Systém je navržen především tak, aby usnadnil práci zaměstnancům s přepisováním údajů přihlášených dětí do databáze.

KLÍČOVÁ SLOVA

Relační databáze, E-R diagram, Use-Case diagram, MySQL, PHP

TITLE

The design of the part of the information system for PS Hroch Pardubice

ANNOTATION

The work deals with the design and subsequent implementation of an information system for PS Hroch Pardubice. The most important parts of the proposal is to create electronic forms for registration of children into courses, and design of application for work with those children's data. The system is primarily designed to facilitate the work of employees with rewriting children's data to the database.

KEYWORDS

Relation database, E-R diagram, Use-Case diagram, MySQL, PHP

OBSAH

SEZNAM PŘÍLOH	- 9 -
ÚVOD	10
1 SOUČASNÝ STAV	11
1.1 INFORMAČNÍ SYSTÉM DDM	12
1.2 PŘIHLAŠOVÁNÍ DO KURZŮ	13
1.2.1 Povinné údaje	13
1.3 VÝKAZY	13
1.4 STANOVENÍ POŽADAVKŮ NA IS	14
1.4.1 Funkční požadavky	14
1.4.2 Nefunkční požadavky	15
2 UML	16
2.1 NAVRHOVÁNÍ PŘÍPADŮ UŽITÍ	16
2.1.1 Scénář případu užití	16
2.1.2 Případy užití	16
2.1.3 Hranice systému	17
2.1.4 Aktéři	17
2.2 VZTAHY MEZI PŘÍPADY UŽITÍ	17
2.2.1 Relace <<include>>	17
2.2.2 Relace <<extend>>	18
2.3 DIAGRAM PŘÍPADU UŽITÍ	18
3 POUŽITÉ TECHNOLOGIE	19
3.1 JAZYK HTML	19
3.1.1 Tvorba tabulek v HTML	19
3.1.2 Tvorba formulářů v HTML	20
3.1.3 Prvky formuláře	20
3.2 PHP	21
3.3 ARCHITEKTURA KLIENT- SERVER	21
3.4 MYSQL	22
3.4.1 Funkce MySQL	22
3.4.2 phpMyAdmin	23
3.4.3 Jazyk SQL	24
4 NÁVRH SYSTÉMU	25
4.1 NÁVRH APLIKACE A PŘIHLÁŠENÍ DĚTÍ	25
4.2 NÁVRH DATABÁZE	26
4.2.1 Konceptuální úroveň	26
4.2.2 Technologická úroveň	27
4.3 USE CASE	29
4.3.1 Scénář k případům užití	30
5 TVORBA APLIKACE A ELEKTRONICKÉHO FORMULÁŘE	34
5.1 TVORBA ELEKTRONICKÉHO FORMULÁŘE	34
5.1.1 Kontrola odesílaných údajů	36
5.2 VYTVOŘENÍ DATABÁZE	37
5.2.1 Databázové tabulky	38
5.2.2 Odeslání údajů do databáze	40
5.3 TVORBA ADMINISTRAČNÍHO ROZHRAŇÍ	40
5.3.1 Přihlášení k aplikaci	41
5.3.1 Přidání dítěte do databáze	41
5.3.2 Přidání zaměstnance	41
5.3.3 Změna přístupových údajů	42
5.3.4 Vymazání /Editace údajů dětí v databázi	43
5.3.5 Vytvoření seznamu na soutěž	43
5.3.6 Vytvoření statistik	44
5.3.7 Odhlášení z aplikace	44

6	TESTOVÁNÍ FUNKČNOSTI	45
6.1	TESTOVACÍ ÚČASTNÍCI KURZŮ	45
6.2	SEZNAM ÚČASTNÍKŮ	46
6.3	SEZNAM NA SOUTĚŽ	47
6.4	ZHODNOCENÍ	48
6.4.1	<i>Návrhy do budoucna</i>	48
	ZÁVĚR	49
	POUŽITÁ LITERATURA	50

SEZNAM TABULEK

Tab. 1 Přehled prvků formuláře.....	20
Tab. 2 Seznam entit	26
Tab. 3 Seznam relací	27
Tab. 4 Scénář pro případ užití Vyplnit přihlášku	30
Tab. 5 Scénář pro případ užití Vyhledat záznam.....	31
Tab. 6 Scénář pro případ užití Upravit záznam.....	31
Tab. 7 Scénář pro případ užití Smazat záznam	32
Tab. 8 Scénář pro případ užití Zobrazit seznam.....	32
Tab. 9 Scénář pro případ užití Tisk výstupů.....	33
Tab. 10 Scénář pro případ užití Vypsát statistiky.....	33
Tab. 11 Tabulka 2012_deti.....	38
Tab. 12 Tabulka zamestnanci	39
Tab. 13 Tabulka kurz.....	39
Tab. 14 Tabulka pohlavi.....	39
Tab. 15 Tabulka pojistovna	39
Tab. 16 Tabulka post	40
Tab. 17 Tabulka psc.....	40
Tab. 18 Testovací účastníci kurzů	45

SEZNAM OBRÁZKŮ

Obr. 1 Počet dětí v posledních letech	11
Obr. 2 Topologie sítě v DDM.....	12
Obr. 3 Případ užití	16
Obr. 4 Aktér.....	17
Obr. 5 Příklad diagramu případu užití	18
Obr. 6 Vytvoření tabulky.....	19
Obr. 7 Průběh komunikace mezi klientem a serverem.....	21
Obr. 8 Aplikace phpMyAdmin.....	23
Obr. 9 Návrh aplikace a přihlašování dětí	25
Obr. 10 ERD návrhu databáze.....	26
Obr. 11 RMD databáze.....	28
Obr. 12 Návrh systému pomocí diagramu případu užití	29
Obr. 13 Formulář.....	34
Obr. 14 Ruční zadání tabulky v PHP MyAdmin.....	37
Obr. 15 Administrační rozhraní.....	40
Obr. 16 Přihlašovací formulář	41
Obr. 17 Formulář pro přidání zaměstnance	42
Obr. 18 Změna přihlašovacích údajů.....	42
Obr. 19 Seznam účastníků kurzu 3	46
Obr. 20 Seznam účastníků kurzu 11	46
Obr. 21 Seznam kurzu 3 na soutěž	47
Obr. 22 Seznam kurzu 11 na soutěž	47

SEZNAM PŘÍLOH

Příloha A Papírová přihláška

Příloha B Výkaz odučených hodin v měsíci

Příloha C Formulář s vypsánými chybovými hláškami

SEZNAM ZKRATEK

PS Pohybové Studio

IS Informační systém (Information System)

DDM Dům dětí a mládeže

UML Unifikovaný modelovací jazyk (Unified Modeling Language)

HTML HyperText Markup Language

PHP Hypertextový preprocesor (Hypertext Preprocessor)

MS Microsoft

IT Information Technology

URL Uniform Resource Locator

LAN Local Area Network

XHTML eXtensible HyperText Markup Language

DDL Data definition Language

DML Data Manipulation Language

ERD Entity Relationship Diagram

RMD Relační model dat

NF Normální forma

RČ Rodné číslo

PSČ Poštovní směrovací číslo

ČP Číslo popisné

ÚVOD

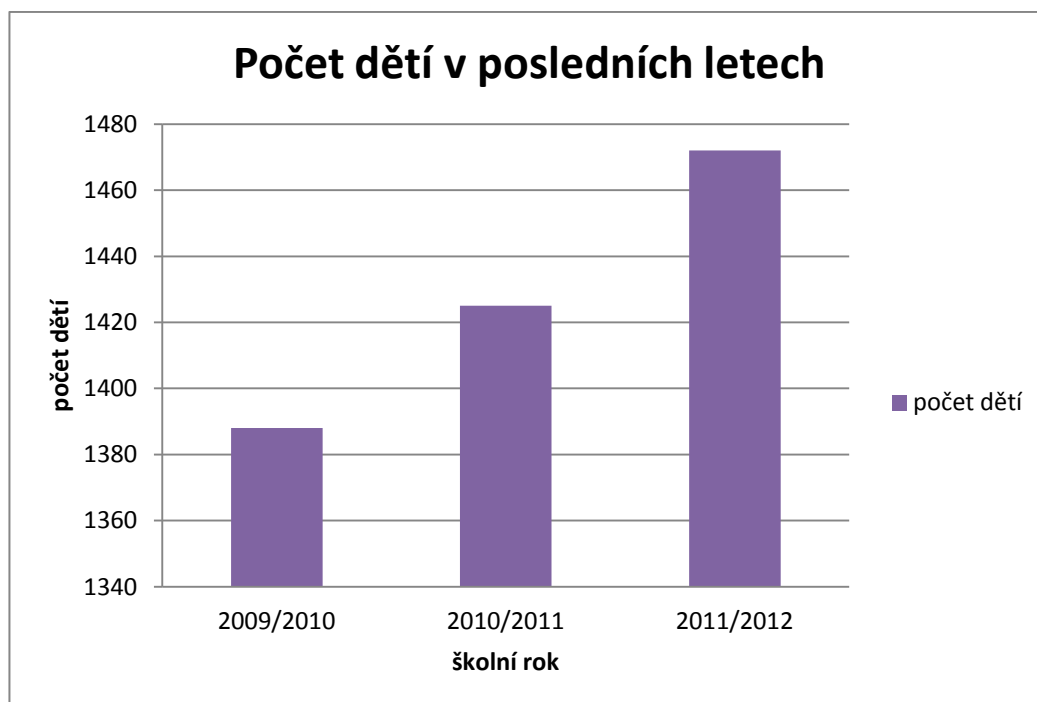
Vzhledem k stále stoupajícímu počtu účastníků PS Hroch a nutné dokumentaci jejich údajů přichází problém s přehledným úložištěm těchto dat, přepisováním těchto dat do počítače atd. Zastaralost IS přináší problémy v komunikaci mezi zaměstnanci, mezi zaměstnanci a rodiči dětí a také hlavně v organizaci věcí a dat. Mezi největší problémy v DDM patří přihlašování dětí do jednotlivých kurzů. Proto tato práce se bude zaměřovat především řešením tohoto problému. Po dohodě s vedením DDM a s vedoucím bakalářské práce bude práce zaměřena na vytvoření aplikace, která bude otestována studenty Univerzity Pardubice, a která tedy bude prozatím provozována na školním serveru Univerzity Pardubice, cipisek.upce.cz, aby mohla být použita v následujícím školním roce 2012/2013 ke svým účelům již v DDM. Aplikace bude určena pouze pro potřeby PS Hroch. Pokud se osvědčí přihlašování dětí na nový školní rok 2012/2013, bude tato aplikace rozšířena pro celý DDM.

Celá práce je rozdělena do 6 kapitol. Úvodní kapitola je zaměřena na seznámení se současným stavem PS Hroch. Následující kapitola teoreticky seznamuje s metodikou UML, pomocí které bude samotný návrh aplikace vytvořen. Třetí kapitola zaznamenává použité technologie- architekturu klient-server a jazyky HTML, PHP a SQL. Tyto jazyky jsou klíčové pro celou práci, neboť pomocí nich je vytvořena celá praktická část této bakalářské práce. Ve čtvrté kapitole bude vytvořen a popsán návrh systému. Pátá kapitola se zabývá samotnou tvorbou aplikace a elektronických přihlášek. Poslední kapitola testuje funkčnost navržené aplikace.

Cílem této práce tedy bude navrhnout systém pro elektronické přihlašování dětí do kurzů PS Hroch pomocí formuláře, a usnadnění tak práce zaměstnancům s přepisováním jejich údajů do databáze. Tento systém bude testován na školním serveru cipisek.upce.cz. Dále bude navrženo administrační rozhraní, kde budou moci oprávnění zaměstnanci sledovat, upravovat a mazat údaje dětí, tisknout seznamy dětí na soutěže a přidávat děti do kurzů. Také zde bude fungovat automatická statistika, která bude vykazovat počty dětí a zaměstnanců, které se musí odesílat na statistický úřad.

1 SOUČASNÝ STAV

Pohybové studio Hroch je taneční skupina, která je v Pardubicích známa už od r 1986 [12]. Zázemí má v Domě dětí a mládeže Beta Pardubice (dále jen DDM), což je školské zařízení Statutárního města Pardubice [4]. V současné době je v DDM 1472 zapsaných dětí. Nárůst počtu dětí v posledních letech je zobrazen v grafu na Obr. 1. V PS Hroch tancuje kolem 800 členů ve věku od 2 do 25 let a stále se rozrůstá. PS Hroch má 12 soutěžních skupin, rozdělených do kurzů podle věku a stylu tance. Každou skupinu vede jeden lektor, který je zaměstnancem DDM nebo pracuje na dohodu o provedení práce či dohodu o pracovní činnosti, tedy je externím zaměstnancem. DDM navštěvují děti nejen kvůli výuce tance, ale učí se zde například i keramika, estetická výchova, hra na kytaru, kulečnick, tanec pro začátečníky, tancování rodičů s dětmi apod., proto je DDM rozděleno do 3 oddělení: oddělení pohybové výchovy, oddělení estetické výchovy a oddělení sportovní a společenské výchovy. O každé oddělení se stará vždy jeden či dva vedoucí pracovníci. Celý Dům dětí a mládeže spravuje ředitelka DDM. Tato bakalářská práce bude zaměřena pouze na oddělení pohybové výchovy, pod které spadá taneční skupina PS Hroch Pardubice.



Obr. 1 Počet dětí v posledních letech

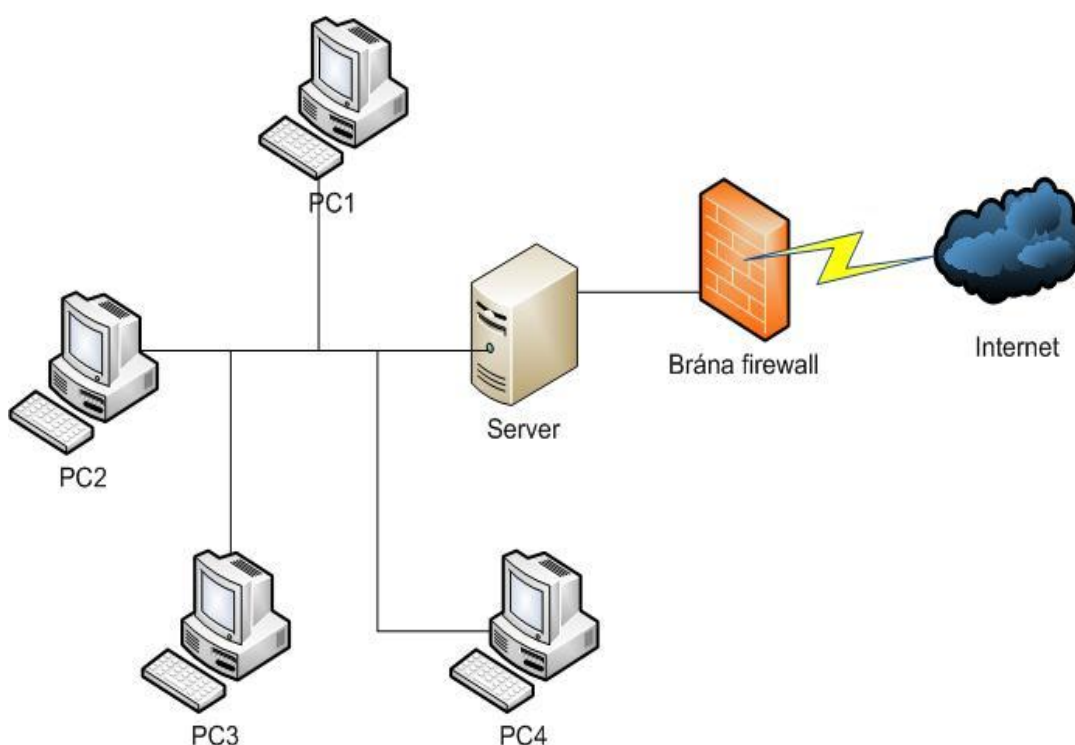
Zdroj: Vlastní zpracování, z podkladů vnitřní dokumentace DDM

1.1 Informační systém DDM

V současné době je informační systém DDM velmi zastaralý. Podle §2, 364 vyhláška ze dne 6. září 2005 o vedení dokumentace škol a školských zařízení a školní matriky a o předávání údajů z dokumentace škol a školských zařízení a ze školní matriky, musí školy a školská zařízení vést školní matriku v elektronické i listinné formě. DDM jako elektronickou matriku využívá Microsoft Office Excel, kam údaje dětí přepisují zaměstnanci z podaných papírových přihlášek. Papírové přihlášky dětí se uchovávají ve speciální místnosti v krabicích popsané školním rokem.

Jelikož zaměstnanci DDM nemají IT vzdělání, ani neprošli žádným kurzem s IT tematikou, práce s „balíčkem“ MS Office jim vyhovuje, neboť na vše, co potřebují provádět s údaji dětí, jim MS Office vystačí. Bohužel ale netuší, kolik práce by jim usnadnila jednoduchá aplikace, kde by měli všechny funkce, které využívají, pohromadě.

V DDM se využívá severu, který propojuje lokální počítačovou síť, takže data mohou jednotliví pracovníci sdílet. V každé kanceláři je k dispozici jeden počítač s operačním systémem Windows XP, dohromady tedy 4 počítače a 1 server. Topologie síť v DDM je zobrazena na Obr. 2



Obr. 2 Topologie síť v DDM

Zdroj: Vlastní zpracování

1.2 Přihlašování do kurzů

System přihlašování dětí je velmi náročný, zejména pro zaměstnance DDM, neboť děti jsou přihlašovány pomocí papírových přihlášek.

1.2.1 Povinné údaje

Na přihláškách do kurzů musí být uvedeno:

- jméno a příjmení dítěte
- národnost
- rodné číslo
- pohlaví
- adresa trvalého bydliště
- jméno a příjmení zákonného zástupce a jeho telefonní nebo jiný kontakt
- podpis zákonného zástupce

Papírová přihláška, která je v současnosti využívána, a která obsahuje všechny tyto povinné údaje, je zobrazena v Příloze A.

Tyto údaje musí zaměstnanci DDM přepisovat do elektronické matriky (MS Excel) kvůli následnému odeslání těchto dat na statistický úřad, kde jsou tyto údaje dále zpracovány. Tento proces je velmi časově i finančně náročný. Sice největší „nátlak“ přihlášek je pouze v červnu, září a říjnu, přesto přepsat do počítače takové množství údajů, zabere v tomto období zaměstnancům zhruba 25% času jejich pracovní doby

1.3 Výkazy

Pro vykazování počtu dětí, docházky a co bylo na dané hodině odučeno, slouží třídnice. Do třídnic se mimo jiné zapisuje i cíl výuky na začátku školního roku a na konci školního roku je zapsáno, zda tento cíl byl naplněn.

Na konci každého měsíce, lektori jednotlivých kurzů, vyplňují výkazy, kam zapíší počet odučených hodin v jednotlivých kurzech, počet hodin na soutěžích (pokud se nějakých zúčastnili) a počet hodin víkendových tréninků. Ukázka výkazu je zobrazena v příloze B.

1.4 Stanovení požadavků na IS

V této kapitole budou popsány funkční a nefunkční požadavky na navrhovanou aplikaci.

1.4.1 Funkční požadavky

Mezi nejdůležitější požadavky na navrhovanou aplikaci patří:

- aplikaci musí zvládnout obsloužit každý zaměstnanec DDM bez IT vzdělání
- obsahování pouze základních a nutných funkcí:
 - schopnost ukládání údajů dětí a zaměstnanců do databáze
 - možnost prohlížení výstupů (statistik)
 - možnost editace údajů
 - možnost smazání údajů
 - registrace do aplikace
 - kontrola vyplněného formuláře před odesláním do databáze
 - vytvoření seznamu dětí na soutěže
 - tisk seznamů

Jelikož je DDM školským zařízením města Pardubice, musí každý rok vykazovat různé počty a statistiky. Tyto statistiky jsou také jedním z požadavků na funkce navrhované aplikace. Do statistik, jak už bylo zmíněno, se uvádí údaje o dětech, které jsou zapsány v předchozí kapitole, ale také musí vykazovat následující počty:

- celkový počet dětí
- počet dětí z Pardubic
- počet chlapců
- počet dívek
- počet zaměstnanců
- počet externích zaměstnanců

Dále musí být vykazovány soutěže, olympiády a další různé akce, které se ale netýkají PS Hroch, tudíž nejsou prozatím v této aplikaci zahrnuty.

1.4.2 Nefunkční požadavky

Na základě funkčních požadavků na IS a na základě zjišťování současného stavu DDM a jeho technického vybavení byly stanoveny tyto nefunkční požadavky na IS:

- síťová architektura klient-server
- podpora běhu na operačních systémech MS Windows
- snadno ovladatelné a přehledné prostředí aplikace

2 UML

V této kapitole bude popsáno co je to vlastně UML, jež bude použito pro návrh části informačního systému PS Hroch.

UML (Unified modeling language) je unifikovaný modelovací jazyk, který podle [8] umožňuje modelovat jednoduché i složité aplikace pomocí stejné formální syntaxe. Tím umožňuje výsledky práce sdílet s ostatními návrháři. Vybrané modely jsou pochopitelné i pro zadavatele aplikace a umožní kvalitní vyjasnění požadavků uživatelů na vytvářený systém.

2.1 Navrhování případů užití

Tato kapitola je velmi důležitá, neboť pouze to, co popisuje soubor případů užití, se bude programovat. Jinými slovy navržený systém nebude obsahovat žádnou jinou funkčnost, než jakou popisují případy užití.

2.1.1 Scénář případu užití

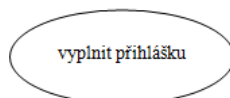
Sice stále nebylo definováno, co vlastně případ užití znamená, ale pomocí scénáře případu užití bude tento pojem vysvětlen v následující kapitole.

Scénář je sekvence kroků popisujících interakce mezi aktérem a systémem.[8] Scénáře se zapisují pomocí tabulky, kde je přehledně zobrazen dialog uživatele se systémem. Může být základní, a také alternativní (doplňující), který představuje postup při zjištění různých chyb nebo mimořádných stavů.[15]

2.1.2 Případy užití

Případ užití (Use Case) je sada scénářů, které spojuje dohromady společný cíl, zachycují přesně funkčnost, která bude budoucím informačním systémem pokryta, a vymezují tak jednoznačně rozsah práce. Každý případ užití popisuje jeden ze způsobů použití systému, popisuje tedy jednu jeho požadovanou funkčnost.[8]

Pro znázornění případu užití, jak ukazuje Obr. 3, se používá ikona ve tvaru elipsy, jejíž návěští popisuje odpovídající službu.



Obr. 3 Případ užití

Zdroj: vlastní zpracování

2.1.3 Hranice systému

Hranice systému představuje vymezení hranice mezi systémem a okolím systému. Je základním vymezením oblasti, která spadá do kompetence systému. Vše, co je mimo hranici systému, představuje okolí. V okolí systému mohou vznikat rozmanité události. Některé z těchto událostí představují stimuly, na které musí systém reagovat, jiné nemusí se systémem vůbec souviset. Pro zobrazení hranice systému se používá 2D-symbol ve tvaru obdélníku značeného návěstím a zahrnujícím evidované případy užití.[15]

2.1.4 Aktéři

Aktérem se nazývá role, ve které vystupuje uživatel v rámci své komunikace se systémem. Aktéři spouštějí případy užití. V systému může jeden aktér spouštět více případů užití a obráceně, jeden případ užití může být prováděn více aktéry. Pro zobrazení aktérů zavádí UML symbol postavičky, jak je vidět na Obr. 4. Aktéři nemusí být vždy jen lidé, může jimi být například nějaký externí systém. [8]



Obr. 4 Aktér

Zdroj: Vlastní zpracování

2.2 Vztahy mezi případy užití

V následujících kapitolách budou popsány nejdůležitější vztahy mezi případy užití. Celá kapitola 2.2 čerpá informace z [8]

2.2.1 Relace<<include>>

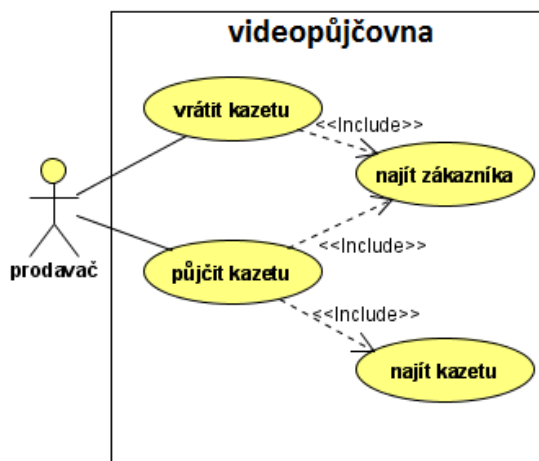
Relace se objevuje tam, kde existuje podobná nebo stejná část sekvence scénáře opakující se ve více případech užití. V takovém případě není vhodné udržovat více kopií shodných částí scénářů, ale doporučuje se vyčlenit samostatný případ užití obsahující opakující se část scénáře. Jinými slovy jedná se o vyčlenění společného chování ze scénářů základních případů užití. Důležité je, že základní případ užití není bez rozšiřujícího případu užití kompletní.

2.2.2 Relace <<extend>>

Tímto typem relace přidává rozšiřující případ užití nové, doplňkové chování do základního případu užití. Podstatným rozdílem oproti relaci typu <<include>> však je, že základní případ užití je sám o sobě zcela soběstačný. Deklaruje pouze tzv. body rozšíření, které ovšem nejsou součástí scénáře, ale pouze ukazují místo ve scénáři, kde může být funkčnost případně rozšířena. Rozšiřující případ užití tedy přidává chování k základnímu případu užití právě v tomto bodě rozšíření. Jinými slovy, vztah <<extend>> modeluje volitelné části případů užití. Rozšiřující případ užití ví, jak se přidat do základního scénáře, naopak základnímu scénáři je jedno, kdo ho rozšiřuje.

2.3 Diagram případu užití

Jazyk UML se skládá z mnoha grafických prvků, které se dají vzájemně kombinovat do podoby diagramů. Protože UML je jazyk, prvky se kombinují podle pevně daných pravidel.[13] V diagramu případu užití jsou vyjádřeny hranice systému rámečkem, uvnitř rámečku jsou případy užití, které vytvářejí chování systému, a vně jsou znázorněni aktéři. Ukázka diagramu případu užití je na následujícím Obr. 5. Vazba <<extend>> by mohla případně rozšířit případ užití vrátit kazetu případem užití zaplatit poplatek za pozdní vrácení.



Obr. 5 Příklad diagramu případu užití

Zdroj: [16]

3 POUŽITÉ TECHNOLOGIE

Celá tvorba aplikace je zpracována pomocí skriptovacího jazyka PHP a databázového systému MySQL. K vytvoření jednotlivých stránek aplikace bude využit skriptovací jazyk HTML. V následujících kapitolách je tato problematika vysvětlena.

3.1 Jazyk HTML

HTML (HyperText Markup Language) je hypertextový značkovací jazyk, sloužící k vytvoření HTML stránky. K vytvoření HTML stránky není potřeba speciálního programu, stačí textové editory, jako např. NotePad, Poznámkový blok ve Windows, atd., nicméně existují i HTML editory, které výrazně usnadňují práci, např. PSPad, ve kterém bude tento návrh systému zpracován.

Pomocí HTML značek (tagů) se dají vytvářet webové stránky. Tagů existuje velmi mnoho a zapisují se do špičatých závorek. Pro návrh aplikace pro PS Hroch postačí pouze popsat tagy pro tvorbu tabulek a formulářů.

3.1.1 Tvorba tabulek v HTML

Tabulky obecně slouží k přehlednému zobrazení dat. Pro účely této bakalářské práce pouze postačí, jak vytvořit základní strukturu tabulky. Celá tato kapitola čerpá informace z [5]

K vytvoření jednoduché tabulky je třeba znát pouze tři základní párové značky: <table>, <tr>, <td>, popř. <th>.

Z párového tagu <table> prohlížeč pozná, že půjde o tabulku, ale neví, co tabulka bude obsahovat, a proto slouží právě značky <tr> a <td>. Značka <tr> vytvoří nový řádek a <td> vytvoří novou buňku. Značka <th> definuje hlavičku tabulky, což je zvýrazněný horní řádek, který popisuje, co je v kterém sloupci (<th> není povinné). Zdrojový kód tabulky, která má hlavičku, dva řádky a v každém řádku dvě buňky je zobrazen na Obr. 6

```
<table>
  <tr>
    <th>hlavička1 </th>
    <th>hlavička2 </th>
  </tr>
  <tr> <!--začátek řádku-->
    <td>sloupec 1 řádek 1</td> <!--nová buňka-->
    <td>sloupec 2 řádek 1</td> <!--nová buňka-->
  </tr> <!--ukončení řádku řádku-->
  <tr>
    <td>sloupec 1 řádek 2</td>
    <td>sloupec 2 řádek 2</td>
  </tr>
</table>
```

Obr. 6 Vytvoření tabulky

Zdroj:[5]

3.1.2 Tvorba formulářů v HTML

Webový formulář je část kódu HTML uzavřená do značek `<form>` a `</form>`. Prvek `form`, může obsahovat atribut `action`, prostřednictvím kterého se určuje, kam budou zadané údaje odeslány. Atribut `action` obsahuje název souboru se skriptem v jazyce PHP, jenž údaje zapsané do formuláře zpracuje.

Atribut `method` určuje způsob odeslání údajů zadaných do webového formuláře. Tento atribut může být nastaven na hodnotu `get` nebo `post`. Metodou `get` jsou údaje posílány jako součást adresy URL, zatímco metodou `post` budou údaje posílány odděleně. Nebudou tedy viditelné v adrese URL. Obecně se doporučuje právě tato metoda předání údajů. Kromě „ukrytí“ posílaných údajů nabízí totiž možnost odeslání většího množství dat.

Prvek `form` je koncipován jako rodičovský prvek, jenž zapouzdřuje všechny prvky, které vytvářejí konečnou podobu webového formuláře.

3.1.3 Prvky formuláře

V následující tabulce (Tab. 1), jsou uvedeny nejobvyklejší prvky formuláře, které budou dále použity při návrhu elektronické přihlášky do kurzů PS Hroch Pardubice.

Tab. 1 Přehled prvků formuláře

HTML	ukázka	popis
<code><input type="text" name="jmeno"></code>	<input type="text" value="obsah"/>	Normální vstupní pole
<code><input type="password" name="heslo" ></code>	<input type="password" value="*****"/>	Vstupní pole pro heslo. Uživatel vidí pouze náhradu textu (hvězdičky)
<code><input type="checkbox" name="skrt"></code>	<input checked="" type="checkbox"/>	Zaškrťovací políčko (nezaškrtnuté se neodesílá)
<code><input type="radio" name="puntik" ></code> Horní <code><input type="radio" name="puntik" checked="" ></code> Dolní	<input type="radio"/> Horní <input checked="" type="radio"/> Dolní	Přepínač. Vždy je aktivní pouze jedna volba se stejným name.
<code><input type="submit" value="odeslat"></code>	<input type="submit" value="odeslat"/>	Odesílací tlačítko
<code><input type="reset" value="vymazat"></code>	<input type="reset" value="vymazat"/>	Tlačítko obnovující původní hodnoty
<code><select name="roletka"></code> <code><option value="3">3</option></code>	<input type="text" value="3"/> ▼	Z roletkového menu je možno vybrat 1 hodnotu

Zdroj: Vlastní zpracování

3.2 PHP

Problematika vytvoření formuláře pro zadání údajů na stránce HTML je vyřešena, proto další kapitola je věnována skriptu PHP, jenž zadané údaje zpracuje. PHP (Hypertext Preprocessor) je výkonným skriptovacím jazykem pro programování na straně serveru. [6]

Nejdůležitějšími pilíři dle [9] jsou:

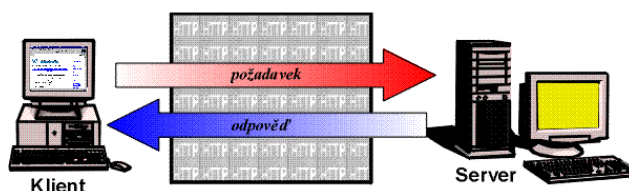
- kód jazyka PHP je uzavřen do párových značek `<?php a ?>`
- každý příkaz v jazyce PHP je ukončen středníkem
- pro výstup textu se používají příkazy `print()` a `echo()`

3.3 Architektura klient- server

Architektura klient-server je v podstatě založena na lokální síti (LAN), personálních počítačích a databázovém serveru. Na personálních počítačích běží program podporující např. vstup dat, formulaci dotazu atd. Dotaz se dále předává pomocí jazyka SQL na databázový server, který jej vykoná a vrátí výsledky zpět na personální počítač. Databázový server je tedy nejvíce zatíženým prvkem systému a musí být tvořen dostatečně výkonným počítačem. Celá komunikace probíhá tímto způsobem:

- uživatel zadává dotaz (buď přímo v SQL, nebo musí být do tohoto jazyka přeložen)
- dotaz je odeslán na databázový server
- databázový server vykoná dotaz
- výsledek dotazu je poslán zpět na vysílací počítač, kde je zobrazen

Kromě jazyka SQL, který představuje standardní dotazovací jazyk, existují ještě další standardy pro navazování komunikace mezi aplikacemi ještě před vlastním zahájením komunikace v SQL.[2] Na následujícím obrázku (Obr. 7) je zobrazen průběh komunikace mezi klientem a serverem.



Obr. 7 Průběh komunikace mezi klientem a serverem

Zdroj:[1]

3.4 MySQL

Databázi existuje hodně druhů, například MS Acces, PostgreSQL, MySQL atd. Jako databáze se dá dokonce využít například tabulka MS Excel.[7]

MySQL je nejpoužívanějším databázovým systémem. Je známý pro svou rychlost, robustnost a malé režijní náklady na připojení. To je důležité pro obsluhu webových stránek, které musí být obslouženy tak rychle, jak je to jen možné. [3]

3.4.1 Funkce MySQL

V této kapitole jsou popsány nejpoužívanější funkce, které lze v souvislosti s MySQL a PHP používat, a které budou dále použity při tvorbě aplikace. Tato kapitola čerpá informace z [10]

mysql_connect() vytvoří spojení s MySQL serverem. Nejsou-li zadány nepovinné parametry, použijí se následující hodnoty: server = 'localhost ', uziv_jmeno = jméno uživatele, pod kterým běží právě spuštěný skript a heslo = bez hesla.

mysql_close() uzavře spojení s MySQL serverem, které je asociováno s určitým identifikátorem spojení. Pokud spojení není zadáno, uzavře poslední otevřené spojení.

mysql_select_db() nastaví aktuální databázi na serveru a asociuje ji s uvedeným identifikátorem spojení.

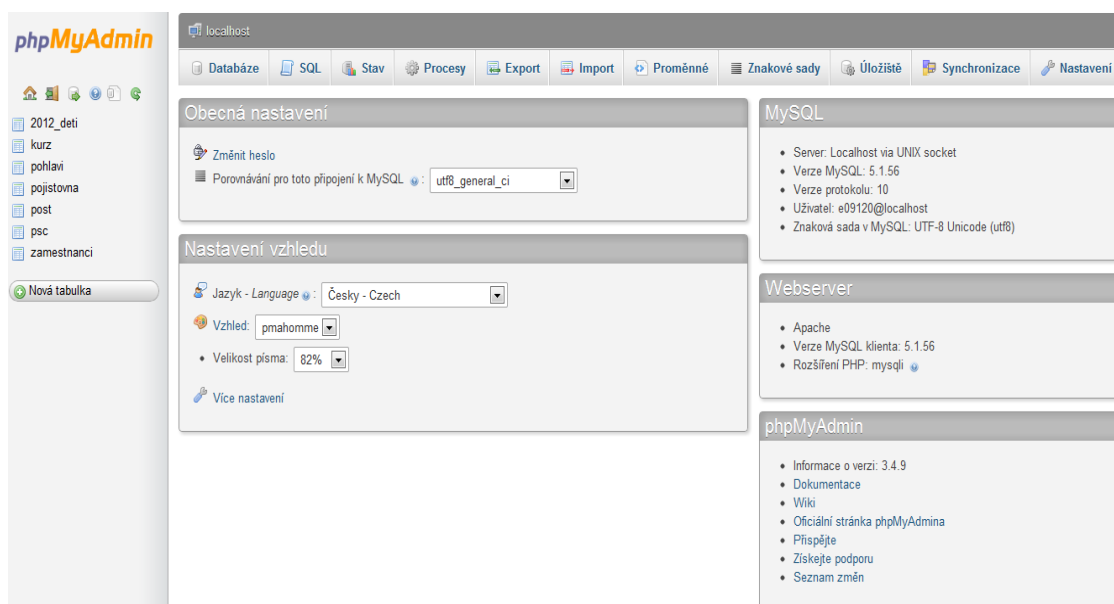
mysql_fetch_row() načte jeden záznam výsledku do pole s číselnými klíči. Každá hodnota sloupce je samostatná hodnota pole. Klíče jsou číslovány od 0.

mysql_fetch_array() je rozšířenou verzí mysql_fetch_row(). Navíc jsou zde data uložena v poli nejen pod číselnými klíči, ale také pod asociativními textovými klíči a jmenujícími se podle názvu sloupce SQL tabulky.

mysql_num_rows() vrací počet záznamů ve výsledku dotazu. Tento příkaz je použitelný pouze pro dotaz typu SELECT.

3.4.2 phpMyAdmin

Nástroj phpMyAdmin je populární nástroj pro správu databáze MySQL. Jedná se o svobodný software. [11] Tento software je napsaný za pomoci skriptovacího jazyka PHP, obsahuje klientský kód XHTML, CSS a Javascript. Protože poskytuje kompletní webové rozhraní pro administraci databází MySQL, je nejvhodnější a nejrozšířenější aplikací. Jelikož se informační technologie stále rozvíjí, rostou rychle i nároky na spravování databází. Z tohoto důvodu jsou vyvíjeny nové standardy MySQL a tak s nimi musí i tato aplikace držet krok. Nejčastější operace, které se v aplikaci phpMyAdmin uskutečňují, jsou: vytvoření a odstranění databáze, a vytvoření, přejmenování, kopírování a odstranění databázové tabulky. Náhled aplikace phpMyAdmin je zobrazena na následujícím obrázku (Obr. 8). V levém menu jsou zobrazeny veškeré tabulky uložené v databázi, v horním menu různé operace, které lze v tomto prostředí provádět a uprostřed obecná nastavení a informace o MySQL, Webserveru a phpMyAdmin.



Obr. 8 Aplikace phpMyAdmin

Zdroj: Vlastní zpracování

3.4.3 Jazyk SQL

SQL (Structured Query Language) je neprocedurální programovací jazyk, který slouží hlavně k vytváření dotazů na data. Je vhodný podle [14], protože je:

- **velmi rozšířen**- většina podnikových informačních systémů dnes ukládá svoje data do relačních databází, kde se s nimi pracuje právě jazykem SQL. Hojně se tento jazyk používá i při tvorbě webových stránek a webových i jiných aplikací.
- **lepší práce s daty**- SQL databáze jsou snadno dostupné i pro osobní použití. Práce s daty v databázi je mnohem sofistikovanější, než v tabulkovém editoru.

Jazyk SQL lze rozdělit podle [9] na dvě základní podmnožiny:

- **jazyk DDL**(Data definition Language) – syntaxe pro tvorby tabulek
- **jazyk DML** (Data Manipulation Language)- syntaxe manipulace s daty

Příkazy z podmnožiny **DDL** umožňují definici datových struktur a tvorbu objektů, jako jsou tabulky, sestavy, indexy, apod. Umožňují také měnit jejich strukturu nebo je odstraňovat. Do této skupiny patří například příkazy:

- **CREATE**- umožní vytvořit novou databázi, nebo novou tabulku.
- **ALTER TABLE**- umožní upravit tabulku.
- **DROP**- umožní zrušit databázi, tabulku, nebo indexy.

Příkazy z podmnožiny **DML** umožní uživateli upravovat databázi, případně vybrat potřebná data v aplikaci. Zahrnuje například tyto příkazy:

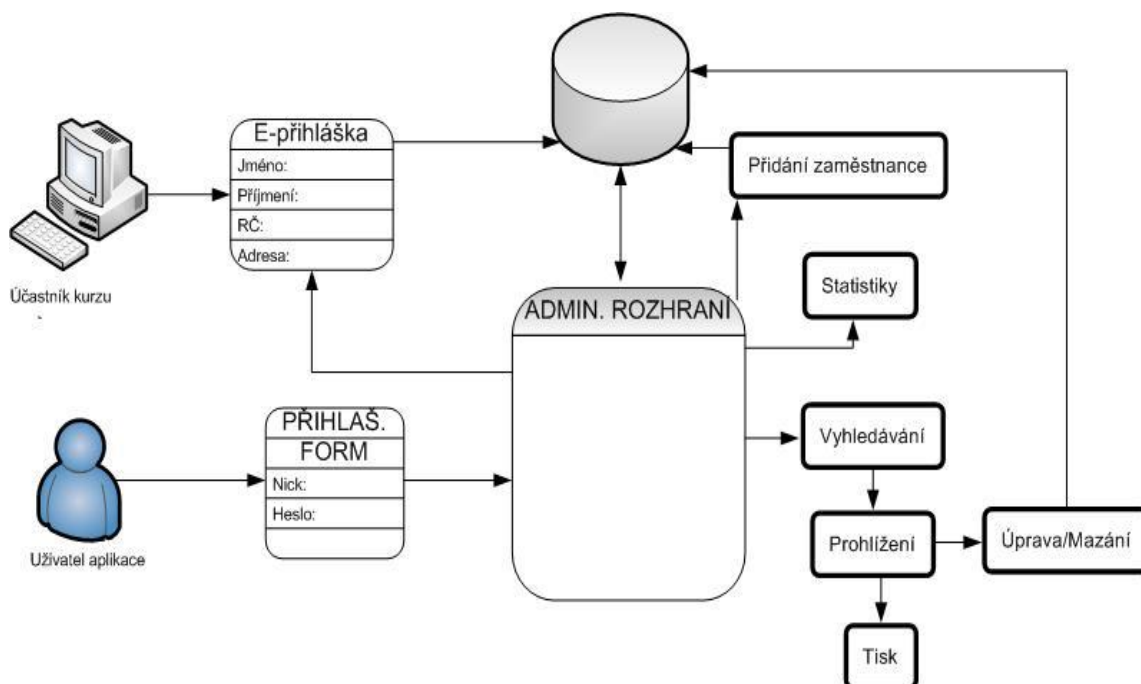
- **SELECT**- je nejvýznamnějším příkazem, pomocí kterého je realizován výběr požadovaných dat z databáze.
- **INSERT**- pomocí tohoto příkazu je možné vložit jeden nebo několik řádku do tabulky. Při vkládání se kontrolují integritní omezení definované pro danou tabulku.
- **DELETE**- umožní smazat data z tabulky, nebo podmnožinu řádků tabulky, na základě definované podmínky výběru.
- **UPDATE**- umožní změnit hodnoty uložené v databázi.

4 NÁVRH SYSTÉMU

Tato kapitola bude zaměřena na návrh aplikace a celého systému v rámci IS.

4.1 Návrh aplikace a přihlášení dětí

Na základě požadavků byla vytvořena základní struktura aplikace, přihlašování do ní pomocí přihlašovacího formuláře a přihlašování dětí do kurzů pomocí E-přihlášky. Tato struktura je zobrazena na Obr. 9.



Obr. 9 Návrh aplikace a přihlašování dětí

Zdroj: Vlastní zpracování

Jak je znázorněno na Obr. 9, pro přihlášení uživatele do aplikace slouží přihlašovací formulář. V Administračním rozhraní si uživatel může vybrat, zda bude vyhledávat děti podle jména a příjmení, nebo podle kurzu, zda potřebuje přidat dítě do databáze, nebo zda potřebuje zobrazit statistiky za uplynulý rok. Položky elektronické přihlášky odpovídají položkám, které byly původně vyplňovány do papírové přihlášky. Údaje z této přihlášky jsou odeslány do databáze. Přihláška může být vyplněna i uživateli aplikace. Každý uživatel aplikace může data přidávat, upravovat i mazat. Každý také může měnit své přihlašovací jméno (nick) a heslo.

4.2 Návrh databáze

Tato kapitola bude popisovat jednotlivé úrovně návrhu databázového modelu. Při návrhu databáze bude použita relační databáze, bude zde zobrazen E-R diagram a relační model dat.

Databáze je navržena tak, aby v sobě uchovávala veškeré údaje odeslané z elektronické přihlášky a údaje, které jsou do databáze vloženy či upraveny uživateli aplikace.

Jednotlivé tabulky a sloupce tedy odpovídají prvkům elektronické přihlášky a údajům o zaměstnancích.

4.2.1 Konceptuální úroveň

Na konceptuální úrovni návrhu databázového modelu je vytvořen seznam entit. Entita je rozlišitelný a identifikovatelný objekt reality. Tedy je to objekt reálného světa, který je schopen nezávislé existence a je jednoznačně odlišitelný od ostatních objektů. [15]

Z návrhu vyllynuly 3 entity, popsané v Tab. 2. Jejich vzájemné vztahy jsou zobrazeny v ER diagramu (ERD) na Obr. 10.

Tab. 2 Seznam entit

Název entity	Klíčový atribut	Atributy
Dítě	ID_d	Jméno, RČ, adresa, pohlaví, příslušnost, pojišťovna, škola, třída, poznámka
Kurz	Kurz	Název kurzu, čas konání, cena
Zaměstnanec	ID_z	Jméno, přihlašovací údaje, post

Zdroj: Vlastní zpracování

Pomocí vztahů mezi entitami jsou entity znázorněny v ERD. ERD je znázorněn na následujícím Obr. 10.



Obr. 10 ERD návrhu databáze

Zdroj: Vlastní zpracování

Vztahy, které jsou v ERD, představují:

- **Zapsáno:** Entita Dítě musí být zapsáno pouze v 1 Kurzu. V Kurzu musí být zapsáno více (1-N) dětí.
- **Učí:** V každém Kurzu musí učit 1 Zaměstnanec, ale Zaměstnanec může učit ve více (1-N) Kurzech

4.2.2 Technologická úroveň

Na této úrovni je vytvořen model systému, zohledňující technologickou koncepci řešení, tj. ve strukturovaném pojetí jde o koncepci organizace dat a technologickou koncepci jejich zpracování. Technologický model není zatížen implementačními specifiky řešení. Technologický návrh určuje, jak je obsah systému v dané technologii realizován.[15]

Na této úrovni se převádí konceptuální model (ERD) do databázové úrovně modelování pomocí tzv. transformace. Transformací vznikají první návrhy relací, které jsou zobrazené v Tab. 3.

Tab. 3 Seznam relací

Název relace	Primární klíč	Cizí klíč	Atributy
Dítě	ID_d	Kurz	Jméno, RČ, adresa, pohlaví, příslušnost, pojišťovna, škola, třída, poznámka
Kurz	Kurz	ID_z	Název kurzu, čas konání, cena
Zaměstnanec	ID_z		Jméno, přihlašovací údaje, post

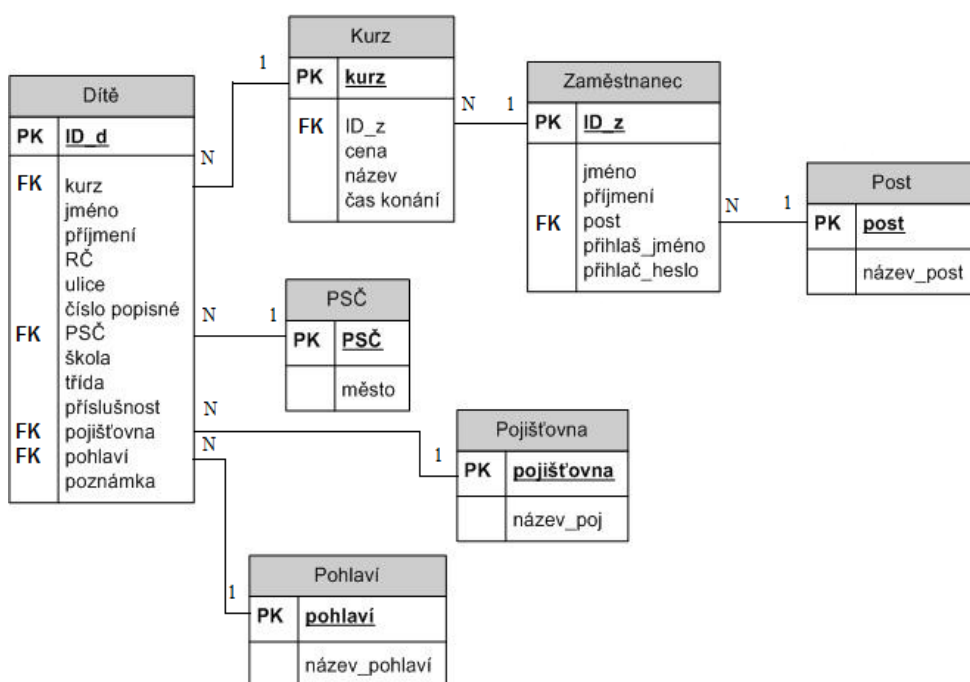
Zdroj: Vlastní zpracování

Pomocí relací lze vytvořit relační model dat (RMD). Nejdříve je ale nutné odstranit anomálie v datovém modelu. Důsledkem normalizace dat je postupná dekompozice datového modelu rozdělením atributů do většího počtu relací, které již nevykazují dané nedostatky.

Postupně je množina všech relací převáděna do tzv. vyšších normálních forem. Pro tento návrh postačí pouze 3 základní normální formy [15]:

- 1NF (první normální forma): 1NF splňuje relace, která neobsahuje vícehodnotové atributy.
- 2NF: splňuje relace, je-li v 1NF a každý neklíčový atribut je plně funkčně závislý na celém primárním klíči relace.
- 3NF: splňuje relace, jestliže je v 2NF a každý neklíčový atribut je netranzitivně závislý na primárním klíči.

Výsledný relační model dat, po transformaci i normalizaci, je zobrazen na Obr. 11. Na tomto obrázku jsou zobrazeny veškeré relace se svými primárními i cizími klíči a jednotlivými atributy. Je zde také vidět jak jsou relace mezi sebou propojeny pomocí vztahů, díky primárním a cizím klíčům.



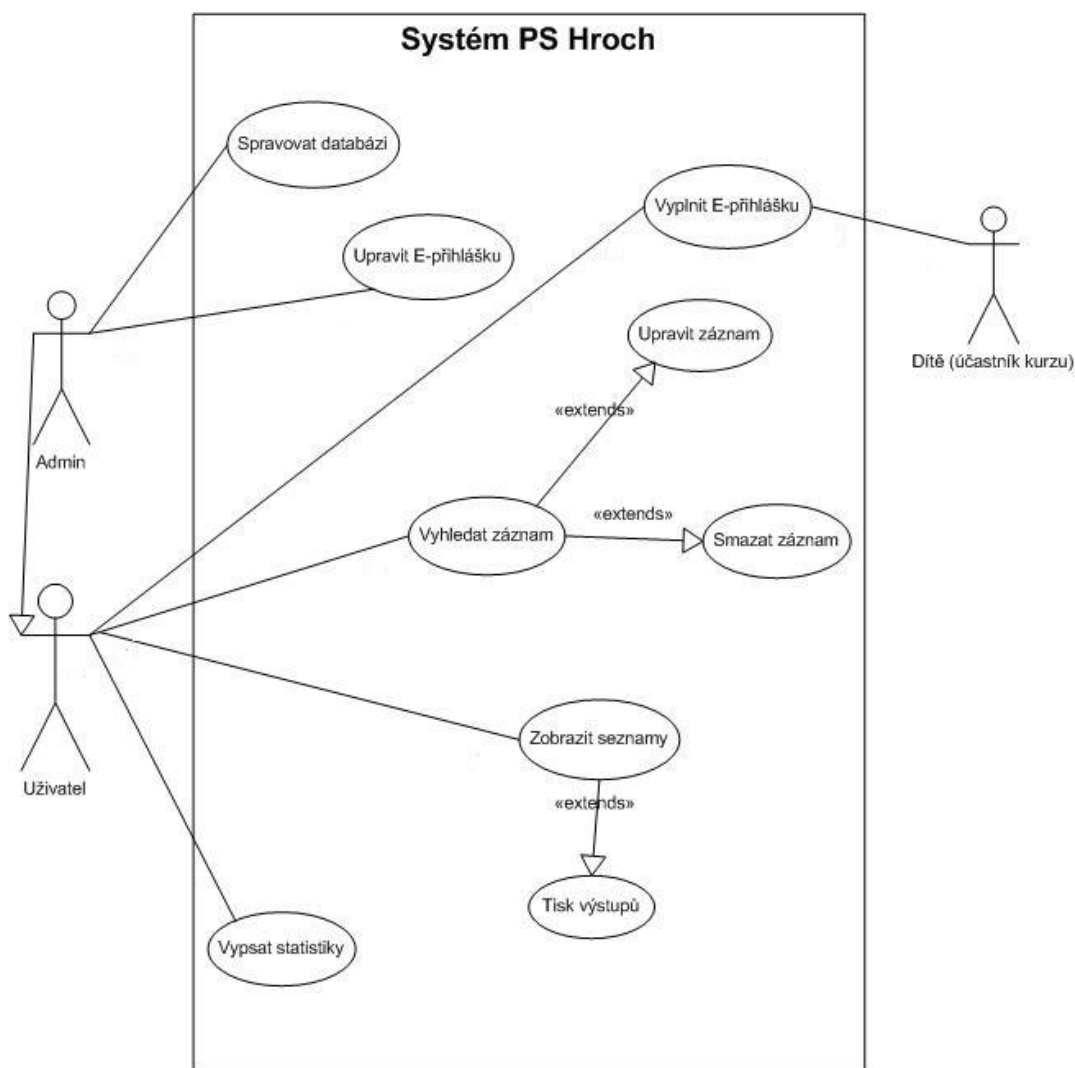
Obr. 11 RMD databáze

Zdroj: Vlastní zpracování

4.3 Use Case

Návrh systému bude také modelován pomocí diagramu případů užití, neboť případy užití zachycují přesně funkčnost, která bude budoucím informačním systémem pokryta a vymezují tak jednoznačně rozsah práce.[15]

Nejprve musí být stanoveni aktéři. Těmi budou uživatelé (děti), ti co budou systém využívat pro vkládání osobních údajů. Poté zaměstnanci PS Hroch (uživatelé aplikace), kteří budou systém využívat k úpravám a mazání údajů z databáze, zobrazování statistik a k zobrazování a tisku seznamů na soutěže. V důležité roli bude také admin, který má možnost všech operací, které má uživatel a navíc může editovat přihlašovací formuláře a spravovat databázi. Takto navržený systém zobrazuje Obr. 12. Relace extend rozšiřuje případ užití vyhledat záznam tak, že poté co je záznam vyhledán, bude uživatel moci záznam smazat nebo upravit. Také rozšiřuje případ užití zobrazit seznamy o případ užití tisk výstupů.



Obr. 12 Návrh systému pomocí diagramu případu užití

Zdroj: Vlastní zpracování

4.3.1 Scénář k případům užití

Tato kapitola je věnována popisům jednotlivých případů užití prostřednictvím jejich scénářů. Vždy je vypsán hlavní scénář, kde je popsána komunikace uživatele se systémem. Ve většině případů je vypsán i alternativní scénář, kde je popsáno, co se stane, pokud selže scénář hlavní. U všech případů užití je možnost alternativního scénáře selhání připojení k databázi, proto tento scénář již nebude vypisován.

Případ užití **Vyplnit přihlášku** je zaznamenán v Tab. 4. Zabývá se zaevidováním údajů člena do databáze. Účastník kurzu se přihlásí na stránky PS Hroch, a po kliknutí na odkaz k přihlášení na nový školní rok mu systém zobrazí formulář, který musí být vyplněn. Účastník vyplní povinné údaje (viz kapitola 1.2.1) označené hvězdičkou a odešle. Po odeslání provede systém kontrolu zadaných údajů a v případě, že je vše vyplněno správně, se zobrazí tabulka s odeslanými údaji pro kontrolu a vytvoří se nový záznam do databáze členů. V případě chybného vyplnění údajů začíná alternativní scénář, kdy je účastník informován, že nevyplnil všechny povinné údaje, nebo je vyplnil chybně, a je vrácen zpět na formulář k doplnění údajů. Účastníka může do databáze zapsat i admin či zaměstnanec.

Tab. 4 Scénář pro případ užití Vyplnit přihlášku

Případ užití	Vyplnit E-přihlášku		
Popis	Systém zaeviduje všechny údaje nového člena		
Aktér	Dítě (popř. zástupce dítěte), uživatel, admin		
Vstup	Aktér je na stránkách, kde se nachází přihlašovací formulář		
Výstup	Uložení údajů do databáze		
	Krok	U/S	Činnost
Hlavní scénář	1	U	Spustí případ kliknutím na odkaz přihláška na rok 2012/2013
	2	S	Zobrazí formulář s údaji k vyplnění
	3	U	Vyplní údaje a odešle
	4	S	Ověří údaje uživatele
	5	S	Vytvoří nového člena a uloží zadané údaje do databáze
Alternativní scénář	5	S	Upozorní uživatele, že nevyplnil všechny povinné údaje nebo je vyplnil chybně a pokračuje krokem 2

Zdroj: Vlastní zpracování

Případ užití **Vyhledat záznam**, zobrazený v Tab. 5 slouží pouze zaměstnancům popř. adminovi. Vyhledávání záznamů je uskutečňováno buďto pomocí vyhledávání podle jména a příjmení, nebo podle kurzů. Po vyhledání záznamu lze provádět operace jako upravit údaje nebo vymazat člena z databáze. **Úprava údajů a smazání záznamů** je popsáno ve scénáři v Tab. 6 a Tab. 7.

Tab. 5 Scénář pro případ užití Vyhledat záznam

Případ užití	Vyhledat záznam		
Popis	Systém zobrazí seznam všech záznamů odpovídajících zadání		
Aktér	Zaměstnanec, Admin		
Vstup	Administrační rozhraní		
Výstup	Zobrazení seznamu odpovídajících členů		
	Krok	U/S	Činnost
Hlavní scénář	1	U	Napiše hledané jméno a příjmení
	2	S	Zobrazí seznam přihlášených členů podle zadání
Alternativní scénář1	1	U	Napiše pouze jméno
	2	S	Zobrazí seznam přihlášených členů se zadaným jménem ale s různými příjmeními
Alternativní scénář2	1	U	Napiše pouze příjmení
	2	S	Zobrazí seznam přihlášených členů se zadaným příjmením ale s různými jmény
Alternativní scénář3	1	U	Vyhledá seznam členů pomocí zadání čísla kurzu
	2	S	Zobrazí seznam přihlášených členů daného kurzu

Zdroj: Vlastní zpracování

Tab. 6 Scénář pro případ užití Upravit záznam

Případ užití	Upravit záznam		
Popis	Systém zaeviduje upravené údaje		
Aktér	Zaměstnanec, Admin		
Vstup	Zaevidovaný člen v databázi, vyhledán záznam tohoto člena		
Výstup	Upravené údaje člena		
	Krok	U/S	Činnost
Hlavní scénář	1	U	Stiskne tlačítko upravit
	2	S	Otevře formulář s vyplněnými přihlašovacími údaji
	3	U	Upraví údaje a odešle
	4	S	Ověří údaje uživatele
	5	S	Zaeviduje nové informace a uloží je do databáze
Alternativní scénář	5	S	Upozorní uživatele, že nevyplnil všechny povinné údaje nebo je vyplnil chybně, pokračuje krokem 3

Zdroj: Vlastní zpracování

Tab. 7 Scénář pro případ užití Smazat záznam

Případ užití	Smazat záznam		
Popis	Systém smaže člena z databáze		
Aktér	Zaměstnanec, Admin		
Vstup	Zaevidovaný člen v databázi, vyhledán záznam tohoto člena		
Výstup	Odstraněný člen z databáze		
	Krok	U/S	Činnost
Hlavní scénář	1	U	Stiskne tlačítko smazat
	2	S	Vypíše na obrazovku dialogové okno, zda chce člena opravdu smazat
	3	U	Stiskne OK
	4	S	Odstraní člena z databáze
Alternativní scénář	3	U	Stiskne ne
	7	S	Vrátí uživatele zpět na seznam členů

Zdroj: Vlastní zpracování

Případ užití **Zobrazit seznam** využívají zaměstnanci (popř. admin). Seznamy účastníků jsou potřeba zobrazovat a tisknout kvůli akcím, jako jsou soutěže nebo vystoupení, kde je potřeba znát datum narození a věk účastníků, pro zařazení do správné věkové kategorie. V Tab. 8 je zobrazen scénář k tomuto případu užití.

Tab. 8 Scénář pro případ užití Zobrazit seznam

Případ užití	Zobrazit seznam		
Popis	Systém zobrazí seznam dětí podle zadaného kurzu nebo seznam pracovníků podle zadání		
Aktér	Zaměstnanec, Admin		
Vstup	Zaevidované děti/zaměstnanci v databázi		
Výstup	Zobrazen seznam dětí/ zaměstnanců		
	Krok	U/S	Činnost
Hlavní scénář	1	U	Stiskne tlačítko seznam na soutěž
	2	S	Dá na výběr uživateli číslo kurzu
	3	U	Vybere číslo kurzu
	4	S	Zobrazí seznam (jméno, příjmení, datum narození, věk)
Alternativní scénář	1	U	Stiskne tlačítko seznam pracovníků
	2	S	Zobrazí seznam zaměstnanců a externích zaměstnanců

Zdroj: Vlastní zpracování

Scénář pro případ užití **Tisk výstupů** je zobrazen v Tab. 9. Tisknout budou moci zaměstnanci a samozřejmě i admin, a to seznam dětí na soutěže.

Tab. 9 Scénář pro případ užití Tisk výstupů

Případ užití	Tisk výstupů		
Popis	Uživatel vytiskne potřebné údaje		
Aktér	Zaměstnanec		
Vstup	Zaevidované děti v databázi, zobrazen seznam dětí		
Výstup	Vytisknuté potřebné údaje		
	Krok	U/S	Činnost
Alternativní scénář 1	1	U	Zadá příkaz pro tisk
	2	S	Vytiskne seznam
	2	S	Nahlásí, že není spuštěna tiskárna
	3	U	Spustí tiskárnu a zadá příkaz pro tisk
Alternativní scénář 2	4	S	Vytiskne seznam
	2	S	Nahlásí, že došel papír v tiskárně
	3	U	Vloží papíry do tiskárny a zadá příkaz pro tisk
	4	S	Vytiskne seznam

Zdroj: Vlastní zpracování

Případ užití **Vypsat statistiky** je pro vedoucí DDM velmi prospěšný, neboť pouhým kliknutím zjistí stavy počtů dětí, zaměstnanců atd. (viz. kapitola 1.4.1), které musí odevzdávat na statistický úřad. Scénář pro případ užití Vypsat statistiky je zobrazen v Tab. 10

Tab. 10 Scénář pro případ užití Vypsat statistiky

Případ užití	Vypsat statistiky		
Popis	Vypsání statistik počtů dětí, zaměstnanců		
Aktér	Zaměstnanec, Admin		
Vstup	Uložené údaje dětí v databázi, ze kterých se budou vytvářet statistiky		
Výstup	Výpis statistik		
	Krok	U/S	Činnost
Hlavní scénář	1	U	Stiskne tlačítko statistiky počtů
	2	S	Nabídne školní rok
	3	U	Vybere, za který školní rok chce statistiky vypsat
	4	S	Vypíše zadané počty

Zdroj: Vlastní zpracování

Admin (programátor) bude moci, mimo jiné výše popsané, upravovat či mazat přihlašovací formuláře. Zároveň se také bude starat o databázi, kam jsou jednotlivé údaje odesílány. Pro tyto případy užití není třeba zobrazovat scénář.

5 TVORBA APLIKACE A ELEKTRONICKÉHO FORMULÁŘE

Celá aplikace je tvořena pomocí značkovacího jazyka HTML, skriptovacího jazyka PHP, kaskádových stylů a komunikuje s databázovým systémem MySQL. Pro zápis programu byl použit uživatelsky velmi jednoduchý a přehledný software PSPad. Ošetření vstupních polí formuláře bylo vytvořeno pomocí skriptovacího jazyka JavaScript. Nutností pro provoz této aplikace je zajištění webového serveru s podporou PHP, databázový systém MySQL a webový prohlížeč. Tato aplikace je optimalizována pro prohlížeč Mozilla Firefox, který v současné době zaměstnanci využívají. Jelikož aplikace bude testována na školním serveru cipisek.upce.cz, všechny tyto nutné požadavky jsou splněny.

5.1 Tvorba elektronického formuláře

Cílem této bakalářské práce je především vytvoření elektronického formuláře pro přihlašování dětí do různých kurzů PS Hroch a usnadnění tak práce zaměstnanců s přepisováním jejich údajů do elektronické podoby. K vytvoření formuláře postačí znalost tvorby HTML stránky, zejména tvorba tabulek a formulářových prvků. To je popsáno v kapitole 3.1. Formulář, který se zobrazí uživateli, je zobrazen na Obr. 13.

PŘIHLÁŠKA			
školní rok 2011/2012			
Hroch č.:	3		
* Jméno:	<input type="text"/>	* Příjmení:	<input type="text"/>
* Pohlaví:	Dívka	* Zdrav. pojist'ovna:	111
		* Stát. přísl.:	<input checked="" type="radio"/> ČR <input type="radio"/> ostatní
* Rodné číslo (bez lomítka):	<input type="text"/>	ÚČASTNÍK zájmového vzdělávání:	<input checked="" type="radio"/> Školák <input type="radio"/> Předškolák
Škola:	<input type="text"/>	Třída:	<input type="text"/>
* Ulice:	<input type="text"/>	* Číslo popisné:	<input type="text"/>
* PŠČ:	<input type="text"/>	* Město:	<input type="text"/>
** Matka jméno:	<input type="text"/>	Příjmení:	<input type="text"/>
		Č.mobilu:	<input type="text"/>
** Otec jméno:	<input type="text"/>	Příjmení:	<input type="text"/>
		Č.mobilu:	<input type="text"/>
* ... povinné údaje			
** ...údaje alespoň jednoho z rodičů musí být vyplněny			
Sdělení pro rodiče: Úplata za kroužek je splatná na zahajovací schůzce nebo převodem na účet č.: 234032128/0300, VS-rodné číslo dítěte. Zpráva pro příjemce-jméno a kurz dítěte Nepřítomnost dětí na kroužku prosím omluvte vedoucí kroužku. Informace o osobních datech účastníka kroužku budou použity pouze pro potřeby DDM BETA, budou řádně uschovány, aby nedošlo k jejich zneužití. Účastníci ZU se budou řídit provozním řádem DDM BETA Pardubice, Štoltbova 2665.			
Souhlasíme s pořizováním a uveřejňováním fotografií a nahrávek za účelem propagace DDM BETA.			
<input type="button" value="odeslat"/>			

Obr. 13 Formulář

Zdroj: Vlastní zpracování

První řádek formuláře obsahuje roletkové menu, ze kterého uživatel vybere číslo kurzu. Vyplnění toho záznamu je povinné, proto je přednastaveno číslo kurzu 3.

Druhý řádek formuláře slouží pro zapsání jména a příjmení do textového pole. Tyto položky jsou povinné, bez jejich vyplnění formulář nepůjde odeslat. Kontrola údajů bude popsána v kapitole 5.1.1.

Třetí řádek formuláře obsahuje výběr pohlaví a zdravotní pojišťovny z roletkového menu a výběr státní příslušnosti pomocí přepínače. Při výběru pohlaví se do databáze odesílá hodnota 1, pokud uživatel zvolí dívku nebo hodnota 2 pro chlapce, tento systém funguje v současné době v matrice DDM, proto toto značení je zachováno. Státní příslušnost je před-vyplněna na hodnotu ČR, díky vlastnosti checked, neboť většina účastníků těchto kurzů je české národnosti a tudíž nemusí být prováděna kontrola vyplněného pole.

Na čtvrtém řádku je k vyplnění rodné číslo, a zda je účastník školní či předškolní. Rodné číslo je ošetřeno podmínkou psaní pouze čísel, ostatní nečíselné znaky nemohou být použity, díky události onkeypress. Ta stanovuje, že pokud uživatel stiskne klávesu jinou než číselnou (každý znak má svůj kód, číselné znaky jsou v intervalu <48;57>), tak se tento znak nezobrazí. Je také nastavena maximální délka na 10 znaků- ostatní znaky již nebudou zapsány. Toto pole musí být vždy vyplněno, takže zde bude probíhat kontrola, zda je vyplněno minimálně 9 číslic(RČ mají 10 čísel až od 1.1.1954). Dělitelnost rodného čísla jedenácti není v podmínce, neboť rodná čísla s datem narození před rokem 1954 tuto vlastnost nemají, a vzhledem k budoucímu přihlašování nejen dětí ale i dospělých a seniorů do kurzů DDM je toto omezení tedy vyloučeno. Výběr účastníka je zapsán pomocí přepínače a je přednastaven na hodnotu školák pomocí vlastnosti checked. Jedna z těchto 2 hodnot musí být vždy vybrána, proto není třeba další kontroly.

Pátý řádek obsahuje údaje o škole, její název a třídu, kterou účastník navštěvuje. Tyto položky nejsou povinné, neboť ne všichni účastníci do školy chodí.

Na šestém řádku jsou k vyplnění povinné údaje ulice a číslo popisné. Číslo popisné je obdobně jako rodné číslo ošetřeno podmínkou psaní pouze číslic.

Sedmý řádek je tvořen totožně jako řádek šestý. Na tomto řádku jsou položky poštovní směrovací číslo a město. PSČ má opět podmínku psaní pouze číslic. Položka město je obyčejná textová hodnota, stejně jako položka ulice.

Na osmém a devátém řádku jsou k vyplnění údaje o rodičích, jejich jméno, příjmení a telefonní kontakt, vše obyčejné textové vstupy, telefonní kontakty jsou omezeny podmínkou psaní 9 číslic.

Na posledním řádku formuláře je tlačítko na odeslání. Syntaxe odesílacího tlačítka je uvedena v kapitole: 3.1.3.

Všechny řádky jsou uzavřené do párového tagu `<table>` a celá tabulka do párového tagu `<form>`. Formulář se odešle po kontrole formulářových prvků díky události `onSubmit`, která kontroluje vybrané prvky formuláře a je nadefinovaná v tagu `<form>`. V tagu `<form>` je kromě této události také parametr `action` obsahující stránku, na kterou se správně vyplněný formulář odešle a parametr `method` s nastavením „post“, díky kterému nebudou při zpracování data vidět v URL adrese. Poznámka koncepční

5.1.1 Kontrola odesílaných údajů

Samotná kontrola formulářových prvků je prováděna pomocí skriptovacího jazyka JavaScript. Tato kontrola probíhá díky funkci, která je volána při odesílání formuláře. Probíhá pouze u formulářových prvků označených hvězdičkou, které nemají před-vyplněnou volbu. V této funkci jsou nadeklarovány jednotlivé proměnné, do kterých je vložena hodnota odesílaného prvku, vyhledaná pomocí `id` prvku. Dále se vytvoří podmínky. Pro jméno, příjmení, ulici, číslo popisné a město je vytvořena podmínka: „pokud se délka znaků rovná 0“, je podmínka vyhodnocena jako pravdivá, vrátí hodnotu `TRUE`, a je zobrazena hláška, která je vyhledána pomocí `id`. Pokud pravdivá není, vrátí hodnotu `FALSE` a pokračuje v dalších podmínkách. Pro rodné číslo je vytvořena podmínka: „pokud je délka rodného čísla menší než 9“, je vrácena hodnota `TRUE` a zobrazena hláška. Pro PSČ je podmínka stejná jako u rodného čísla pouze s rozdílem, pokud se nerovná délce 5 znaků.

V zobrazeném formuláři, je hláška umístěna vždy vpravo od formulářového prvku. Pomocí kaskádových stylů je nastavena na neviditelnou a při zjištění špatného vyplnění při kontrole je zobrazena červenou barvou. V příloze C je zobrazen formulář s vypsánými hláškami. Pokud je formulář bezchybně vyplněný a odeslaný, zobrazí se uživateli stránka s vypsánými odeslanými údaji, které slouží ke kontrole. V případě zjištění chyby až po odeslání, při kontrole, bude na stránce uveden email, na který může být odesláno hlášení o špatném vyplnění formuláře a poslat opravené údaje. Na této stránce by měla být i kolonka pro napsání emailové adresy uživatele, a odeslání údajů dítěte na tento email. Jelikož se musí udržovat ve školských zařízeních i papírová kartotéka, a údaje nezletilých dětí musí být podepsány zákonným zástupcem dítěte, bude muset zákonný zástupce dítěte vyplněné údaje

vytisknout a podepsat, poté donést podepsanou přihlášku do DDM k založení do evidence. Pokud účastník nebude moci přihlášku vytisknout, bude vytištěna v DDM na začátku školního roku a předána zákonnému zástupci dítěte k podpisu. Z důvodu využívání školního serveru, jako testovacího serveru, nejsou některé funkce podporovány, proto prozatím funkce na odesílání emailu není vytvořena.

Údaje z formuláře jsou odeslány nejen účastníkovi ke kontrole, ale musí být odeslány také do databáze, proto v dalších kapitolách bude popsáno, jak byla databáze vytvořena a jak budou údaje v databázi využívány.

5.2 Vytvoření databáze

Ke zpracování údajů z formulářů a pro práci s těmito daty je využíván databázový systém MySQL, neboť tato aplikace musí být otestována a testování probíhá na školním serveru, který databázi MySQL využívá. V této práci je použita verze MySQL: 5.1.56

Pro vytvoření jednotlivých tabulek v databázi bylo využito nástroje PHP MyAdmin. Návrh tabulek může být uskutečněn buď ručním zadáním, nebo pomocí příkazu SQL. Ukázka ručního zadání je na Obr. 14.

The screenshot shows the PHP MyAdmin interface. On the left, a sidebar contains a list of databases: 2012_deti, kurz, pohlavi, pojistovna, post, psc, and zamestnanci. A 'Nová tabulka' (New table) button is highlighted with a blue border. The main area displays the 'Struktura' (Structure) tab for the 'zamestnanci' table. The table name 'zamestnanci' is entered in the 'Jméno tabulky:' field. The table structure is defined as follows:

Pole	id_z	jmeno
Typ	INT	VARCHAR
Délka/Množina	3	20
Výchozí	Žádná	Žádná
Porovnávání		utf8_czech_ci
Vlastnosti		
Nulový	<input type="checkbox"/>	<input type="checkbox"/>
Index	PRIMARY	---
AUTO_INCREMENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Komentáře		
MIME typ		
Transformace při prohlížení		
Parametry transformace		

Obr. 14 Ruční zadání tabulky v PHP MyAdmin

Zdroj: Vlastní zpracování

V modrém obdélníku je tlačítko pro vytvoření nové tabulky. V zeleném obdélníku je kolonka pro název tabulky. Definice jednotlivých polí (sloupců) tabulky je označeno v červeném obdelníku. Zde je ukázka pro zadanání tabulky zamestnanci se sloupci id_z a jmeno (nejsou zde uvedeny všechny prvky, neboť budou jednotlivé tabulky včetně svých prvků popsány v kapitole 5.2.1).

V celé databázi je prozatím 7 tabulek, tabulka 2012_deti, tabulka zamestnanci, tabulka kurz, tabulka pohlavi, tabulka pojistovna, tabulka kurz a tabulka psc. Při pozdější implementaci do DDM by jistě přibyly další tabulky, jako databáze dětí minulých let, akce, které se konají, atd. Pro účely bakalářské práce postačí pro testování systému pouze tyto výše uvedené tabulky.

5.2.1 Databázové tabulky

V této kapitole jsou ukázány a popsány jednotlivé tabulky v databázi se svými prvky (sloupci).

Tab. 11 Tabulka 2012_deti

Sloupec	Datový typ	Max délka	Popis
id_d	INT	10	Automaticky přiřazené ID, primární klíč, ne nulový
kurz	VARCHAR	2	Číslo kurzu, je indexem- propojení na tabulku Kurz, ne nulový
jmeno	VARCHAR	20	Jméno účastníka, ne nulový
prijmeni	VARCHAR	20	Příjmení účastníka, ne nulový
phlavi	INT	2	Pro dívku vypíše 1, pro chlapce 2, je indexem- propojení na tabulku Pohlavi, ne nulový
prislusnost	VARCHAR	10	Může nabývat hodnoty „ČR“ nebo „jiná“, ne nulový
ulice	VARCHAR	50	Název ulice účastníka, ne nulový
cp	VARCHAR	10	Číslo popisné domu, ne nulový
psc	INT	5	Poštovní směrovací číslo, je index- propojení na tabulku PSC,ne nulový
rc	VARCHAR	11	Rodné číslo účastníka, ne nulový
ucastnik	VARCHAR	20	Vypíše, zda je účastník školní či předškolní, ne nulový
skola	VARCHAR	30	Název školy v následujícím školním roce hodnota Null
trida	VARCHAR	2	Třída, do které bude účastník chodit v následujícím šk. roce, hodnota Null
pojistovna	INT	3	Číslo zdravotní pojišťovny, je index- propojení na tabulku Pojistovna, ne nulový
poznamka	VARCHAR	300	V tomto poli jsou zaznamenány údaje o rodičích- jméno, příjmení a telefonní číslo, ne nulový

Zdroj: Vlastní zpracování

Tab. 12 Tabulka zamestnanci

Sloupec	Datový typ	Max délka	Popis
id_z	INT	3	Automaticky přiřazené ID, primární klíč, ne nulový
jmeno	VARCHAR	20	Jméno uživatele, ne nulový
prijmeni	VARCHAR	20	Příjmení uživatele, ne nulový
post	VARCHAR	1	Nabývá hodnoty „e“ nebo „z“, je indexem – propojení na tabulku Post, ne nulový
nick	VARCHAR	20	Nick uživatele, slouží pro přihlášení k aplikaci, může být nulový
heslo	VARCHAR	20	Heslo uživatele, slouží pro přihlášení k aplikaci, může být nulový

Zdroj: Vlastní zpracování**Tab. 13 Tabulka kurz**

Sloupec	Datový typ	Max délka	Popis
kurz	INT	20	Primární klíč, ne nulový
nazev	VARCHAR	20	Název kurzu, ne nulový
cena	VARCHAR	4	Cena kurzu, může být hodnota null
cas	VARCHAR	10	Čas konání kurzu, může být hodnota null
id_z	INT	5	ID zaměstnance, který kurz vyučuje, ne nulový

Zdroj: Vlastní zpracování**Tab. 14 Tabulka pohlavi**

Sloupec	Datový typ	Max délka	Popis
pohlavi	INT	2	Primární klíč, ne nulový, (pouze hodnota 1 nebo 2)
nazev	VARCHAR	5	Název pohlaví

Zdroj: Vlastní zpracování**Tab. 15 Tabulka pojistovna**

Sloupec	Datový typ	Max délka	Popis
pojistovna	INT	3	Primární klíč, ne nulový
nazev	VARCHAR	20	Název pojišťovny, ne nulový

Zdroj: Vlastní zpracování

Tab. 16 Tabulka post

Sloupec	Datový typ	Max délka	Popis
post	VARCHAR	1	Primární klíč, ne nulový, nabývá hodnoty „e“ (externista) nebo „z“ (zaměstnanec)
nazev	VARCHAR	20	Název postu, ne nulový

Zdroj: Vlastní zpracování

Tab. 17 Tabulka psc

Sloupec	Datový typ	Max délka	Popis
psc	INT	5	Primární klíč, ne nulový
mesto	VARCHAR	20	Název města, ne nulový

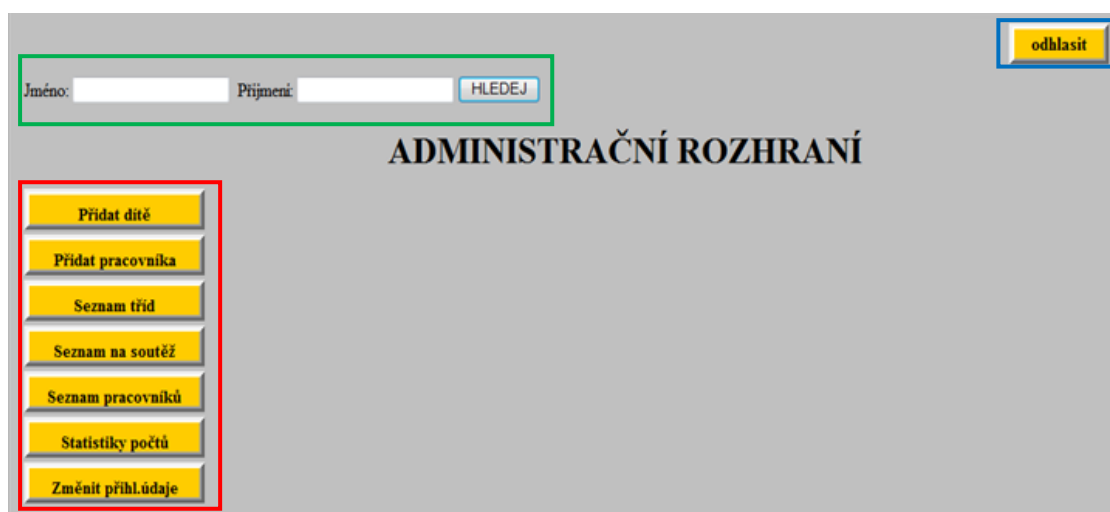
Zdroj: Vlastní zpracování

5.2.2 Odeslání údajů do databáze

Pro odeslání jakýchkoli údajů do databáze musí být s databází nejprve navázáno spojení pomocí funkce MySQL_Connect. Pokud již existují tabulky v databázi s požadovanými sloupci a je navázáno spojení s databází, je možné odeslané údaje uložit pomocí SQL dotazu a příkazu MySQL_Query, který provede dotaz, do zvolené tabulky.

5.3 Tvorba administračního rozhraní

Jelikož k databázi má přístup pouze programátor (admin), musí být vytvořeno rozhraní pro zaměstnance, kde budou moci provádět různé operace s daty. Na Obr. 15 je zobrazeno administrační rozhraní tak, jak se zobrazí uživateli po přihlášení do této aplikace.

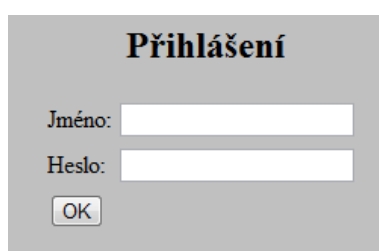
**Obr. 15** Administrační rozhraní

Zdroj: Vlastní zpracování

V červeném obdélníku jsou zobrazeny tlačítka pro vykonání jednotlivých úkonů, které budou popsány v dalších kapitolách. V zeleném obdélníku jsou kolonky pro vyhledávání dětí v databázi, vyhledávání může být buďto podle jména, nebo podle příjmení, nebo podle jména a příjmení. V modrém obdélníku je tlačítko na odhlášení se z aplikace.

5.3.1 Přihlášení k aplikaci

Pro přihlášení do aplikace slouží formulář, který je zobrazen na Obr. 16, a který vyžaduje přihlašovací jméno (nick) a heslo. Uživatelské jméno i heslo dostane zaměstnanec přiděleno od programátora, nebo od zaměstnance, který nového zaměstnance přidělil do databáze. To si bude moci zaměstnanec v administračním rozhraní změnit.



The image shows a login form with a grey background. At the top center, the title "Přihlášení" is displayed in bold black text. Below the title, there are two input fields. The first is labeled "Jméno:" and the second is labeled "Heslo:". Both labels are in a dark grey font. Below the "Heslo:" field, there is a small button with the text "OK" inside it.

Obr. 16 Přihlašovací formulář

Zdroj: Vlastní zpracování

Po vyplnění je přihlašovací formulář odeslán a vykoná se podmínka kontrolující odeslané přihlašovací jméno a heslo s přihlašovacím jménem a heslem, uloženým v databázi. Pokud se tyto údaje shodují, vykoná se příkaz na vybrání uživatelského id podle přihlašovacího jména a hesla a uživatel tak „vstoupí“ do aplikace pod svým id. Pokud se uživatelské jméno a heslo neshodují s uživatelským jménem a heslem v databázi, vypíše se hláška o špatném zadání kombinace přihlašovacího jména a hesla.

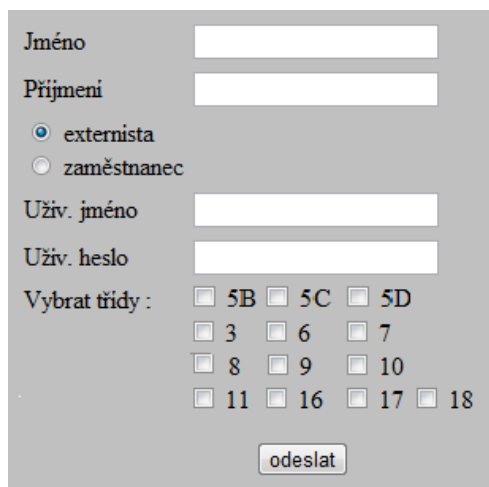
5.3.1 Přidání dítěte do databáze

Pokud bude třeba přidat dítě, bude v tomto administračním rozhraní tlačítko na přidání dítěte. Formulářová přihláška je totožná s přihláškou, kterou budou vyplňovat rodiče, takže zde pracují veškeré kontroly a před-vyplněné položky stejně jako v kapitole 5.1.1.

5.3.2 Přidání zaměstnance

V DDM pracují buďto zaměstnanci, nebo externí zaměstnanci, kteří pracují pouze na dohodu o pracovní činnosti nebo na dohodu o vykonání práce. Proto je nutné tyto pracovníky rozlišovat. Pro přidání pracovníka do databáze je tedy nutné vyplnit jeho jméno, příjmení, uživatelské jméno pro přihlášení do aplikace a heslo, kurzy (třídy), ve kterých vyučuje a zdali

je zaměstnancem či externím zaměstnancem. Uživatelské jméno a heslo nemusí být vyplněno, neboť ne všichni zaměstnanci musí mít přístup k této aplikaci- to bude na rozhodnutí ředitelky DDM. Na Obr. 17 je zobrazen formulář na přidání zaměstnance do databáze.

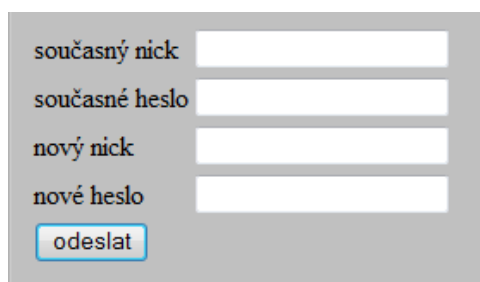


Obr. 17 Formulář pro přidání zaměstnance

Zdroj: Vlastní zpracování

5.3.3 Změna přístupových údajů

Pokud si chce zaměstnanec DDM změnit své přístupové údaje, musí nejprve vyplnit své stávající přihlašovací jméno a heslo. Pomocí podmínky, která testuje, zda se sobě rovná současné přihlašovací jméno a heslo s přihlašovacím jménem a heslem uloženým v databázi, se ověří, zda toto přihlašovací jméno s heslem existuje. Pokud ano, do databáze se přepíše nové přihlašovací jméno a heslo. Formulář na změnu přihlašovacích údajů je zobrazen na Obr. 18.



Obr. 18 Změna přihlašovacích údajů

Zdroj: Vlastní zpracování

5.3.4 Vymazání /Editace údajů dětí v databázi

K vymazání nebo úpravě údajů některého dítěte je potřeba nejprve dítě vyhledat, buď pomocí vyhledávače jména a příjmení (viz zelený obdélník na Obr. 15), nebo zobrazením seznamu třídy (v levém menu v administračním rozhraní). V seznamu třídy se zobrazí všechny děti, které daný kurz (třídu) navštěvují s veškerými svými údaji.

Vpravo od posledního záznamu jsou dvě tlačítka, 1. tlačítko je upravit a 2. tlačítko je smazat. Po stisknutí tlačítka smazat, vyskočí dialogové okno s hláškou: „Opravdu chcete smazat?“, pokud stisknutí tlačítka byl pouze omyl, stisknutím storno, je zaměstnanec vrácen zpět. Pokud opravdu chce dítě vymazat z databáze a stiskne tlačítko OK, zobrazí se stránka s vypsanou hláškou, že data byla vymazána a veškeré údaje jsou z databáze vymazány.

Pokud chce uživatel upravit údaje dítěte, stiskne tlačítko upravit, zobrazí se uživateli předvyplněný formulář s již zadanými údaji, aby nemusel vše vypisovat znovu. Uživatel přepíše údaje, které chce upravit a stiskne tlačítko odeslat. I zde probíhá kontrola správně vyplněných údajů (viz. 5.1.1) a údaje do databáze neuloží, dokud nejsou vyplněny správně.

5.3.5 Vytvoření seznamu na soutěž

Na taneční soutěže, je vždy nutností trenéra, mít s sebou seznam zúčastněných dětí. Na tomto seznamu musí být jméno, příjmení, datum narození a věk. Jelikož věk dětí na přihlášce k zadání není, musí být vytvořena funkce na vypočítání věku z data narození. Datum narození se snadno zjistí z rodného čísla. Pomocí funkce substr jsou do proměnných rok, měsíc a den vloženy vždy dvě číslice z rodného čísla, která tomuto odpovídají, a do proměnné letos je vloženo poslední dvojčíslí ze současného roku. Poté je testováno, zda je v proměnné měsíc číslo větší nebo rovno 51. Pokud ano, (jedná se o dívku) je provedeno odečtení hodnoty 50 od tohoto čísla, neboť v rodném čísle dívky, se k měsíci narození připočítává hodnota 50. Rok narození je vytvořen díky podmínce, která testuje, zda je číslo v proměnné letos větší či rovna číslu v proměnné rok. Pokud je tato podmínka vyhodnocena kladně, uloží se do proměnné, století, číslo 20, v opačném případě se do této proměnné uloží hodnota 19. Celé datum narození je uloženo do proměnné DatNar.

Pokud je známo datum narození, je možné vytvořit funkci na výpočet věku, která odčítá datum narození od data současného dne. Po vydělení číslem 1000 a zjištění celého čísla z této hodnoty vyjde přesný věk, který je vrácen při volání této funkce.

5.3.6 Vytvoření statistik

Jedním z požadavků na aplikaci bylo vytvoření statistik, které jsou potřeba vykazovat každý rok, a které musí být odesílány k dalšímu zpracování na statistický úřad (viz. kapitola 1.4.1). Jelikož jsou formuláře na přihlášení prozatím navrhovány pouze pro kurzy PS Hroch, v těchto statistikách budou tedy data, která se týkají pouze pohybového oddělení DDM. Po zavedení aplikace do DDM bude také na výběr, ze kterého roku chce uživatel vypsat statistiky, jelikož je v této práci pracováno prozatím pouze s jedním školním rokem, není třeba se tímto nastavením zabývat.

Pro vytvoření počtů je vždy využito dotazu na výběr požadavku a poté pomocí příkazu `MySQL_Num_Rows` zjištěn počet řádků. Tím je zjištěn příslušný počet, na který je dotazováno.

5.3.7 Odhlášení z aplikace

Posledním nutným úkonem, je odhlášení přihlášeného uživatele. To je velice jednoduché, neboť stačí zrušit `session` proměnnou, která vznikla při přihlášení do aplikace.

6 TESTOVÁNÍ FUNKČNOSTI

Pro testování funkčnosti bylo využito 6 studentů Univerzity Pardubice, neboť pouze ti mají přístup ke školnímu serveru cipisek.upce.cz a mohou tak odeslat elektronickou přihlášku. Z důvodů ochrany osobních údajů nebudou zveřejněné údaje studentů pravdivé. Tato fáze testování slouží ke zjištění, zda přihlašování a aplikace fungují správně. Funkčnost se testovala na seznamu dětí, zda se vypisují veškeré informace, a na seznamu dětí na soutěž, kde je třeba vypsát i věk a datum narození. Ostatní funkce pracují na stejném nebo podobném principu, tudíž nebudou popisovány a zobrazovány.

6.1 Testovací účastníci kurzů

V následující tabulce (Tab. 18) jsou vypsáni studenti a jejich údaje o jejich PC, použitím internetovém prohlížeči, rozlišení monitoru a používaném operačním systému, kteří budou testovat funkčnost elektronických přihlášek.

Tab. 18 Testovací účastníci kurzů

Jméno	Prohlížeč	Rozlišení	OS	PC/notebook
Zuzana Pávová	Google Chrome 18.0	1366x768	Windows 7	Ntb: ASUS P50IJ
Markéta Drtinová	Mozilla Firefox 11.0	1200x800	Windows 7	Ntb: Toshiba Satellite L300
Pavla Vaníčková	Internet Explorer 9	1280x800	Windows 7	Ntb: Acer Extensa 5430
Ondřej Pelinka	Opera 11.52	1366x768	Windows XP	Ntb: HP Compaq 615
Martin Šanda	Safari 5.1.4	1280x768	Windows 7	Ntb: Fujitsu Siemens Computers ESPRIMO Mobile V5535
David Konstantinov	Mozilla Firefox 9.0	1280x 800	Windows XP	Ntb: ASUS F3JP

Zdroj: Vlastní zpracování

6.2 Seznam účastníků

Bylo vybráno 6 testovacích studentů (viz předchozí kapitola). Muži se registrovali do kurzu číslo 3, ženy do kurzu číslo 11 (náhodně vybráno). Na následujícím obrázku (Obr. 19) je zobrazen seznam kurzu č. 3 a na Obr. 20 seznam kurzu č. 11.

seznam třídy Hroch č.: - odeslat

Seznam třídy H3

Kurz	Jméno	Příjmení	Pohlaví	Poj.	Prisl.	RČ.	Účastník	Škola	Třída	Bydliště	PSČ	Město	Poznámka	UPRAVIT	SMAZAT
3	David	Konstantin	2	111	ČR	8801183665	školaček	Univerzita Pardubice		Na Dražce 1668	53003	Pardubice	Petr Konstantin 774994816	upravit	smazat
3	Ondřej	Peřínka	2	111	ČR	8907314845	školaček	Univerzita Pardubice		Sruby 5	65644	Sruby	Ivana Peřínková 777878582 Aleš Peřínka 608745569	upravit	smazat
3	Martin	Šanda	2	217	ČR	8808190895	školaček	Univerzita Pardubice		Novomlýnská 529	28401	Katná Hora	Iva Šandová 603258761 Boris Šanda	upravit	smazat

Obr. 19 Seznam účastníků kurzu 3

Zdroj: Vlastní zpracování

seznam třídy Hroch č.: - odeslat

Seznam třídy H11

Kurz	Jméno	Příjmení	Pohlaví	Poj.	Prisl.	RČ.	Účastník	Škola	Třída	Bydliště	PSČ	Město	Poznámka	UPRAVIT	SMAZAT
11	Markéta	Drtinová	1	211	ČR	8955072997	školačka	Univerzita Pardubice		Nerudova 1407	43401	Most	Paola Drtinová 723355606 Ondřej Drtina 607502033	upravit	smazat
11	Zuzana	Pávková	1	111	ČR	8962232829	školačka	Univerzita Pardubice		Erno Kolář 957	53012	Pardubice	Jaroslava Pávková 728684111 Pavel Pávek 728684068	upravit	smazat
11	Paola	Vaničková	1	111	ČR	8955073192	školačka	Univerzita Pardubice		Vauška IV 413	58266	Krucemburk	Marie Vaničková 728125118 Bohoslav Vaniček 728247315	upravit	smazat

Obr. 20 Seznam účastníků kurzu 11

Zdroj: Vlastní zpracování

6.3 Seznam na soutěž

Zobrazení seznamu na soutěž, kde je potřeba jméno, datum narození a věk, je zobrazeno na následujících obrázcích (Obr. 21 a Obr. 22).

seznam třídy Hroch č.: -

Seznam PS Hroch Pardubice

Choreografie:

Jméno:	Příjmení:	Datum narození:	Věk:
David	Konstantinov	18. 01. 1988	24
Ondřej	Pelínka	31. 07. 1989	22
Martin	Šanda	19. 08. 1988	23

Obr. 21 Seznam kurzu 3 na soutěž

Zdroj: Vlastní zpracování

seznam třídy Hroch č.: -

Seznam PS Hroch Pardubice

Choreografie:

Jméno:	Příjmení:	Datum narození:	Věk:
Markéta	Drtinová	07. 5. 1989	22
Zuzana	Pávová	23. 12. 1989	22
Pavla	Vaničková	07. 5. 1989	22

Obr. 22 Seznam kurzu 11 na soutěž

Zdroj: Vlastní zpracování

6.4 Zhodnocení

Testování probíhalo od 1. dubna do 8. dubna, kdy již byla aplikace i elektronické přihlášky připravené. Vzhledem k výběru 6 studentů (viz kapitola 6.1) byly do databáze zaznamenány údaje pouze těchto 6 lidí. Během 1. -3. dubna se jednotliví studenti přihlásili do těchto kurzů a ve zbylých 5 dnech měli čas na testování funkcí aplikace.

„Testovací účastníci kurzů“ neměli žádné připomínky k elektronickým přihláškám. Údaje k vyplnění byly srozumitelné a žádný z účastníků neměl problémy s vyplněním přihlašovacího formuláře. Jelikož mezi účastníky jsem zvolila i sebe, testovala jsem i nevyplnění povinných údajů. Zjistila jsem, že vypisování chybových hlášek a psaní pouze číslic do vybraných kolonek funguje správně.

Jediné, co nebylo zcela přesné, bylo nastavení zobrazení v jednotlivých prohlížečích. Po odeslání formuláře se zobrazuje stránka s vyplněnými údaji ke kontrole, hláška, o úspěšně odeslaných datech a tlačítko (odkaz) pro vyplnění další přihlášky. V prohlížeči Google Chrome toto tlačítko překrývalo část z vypsání hlášky. Tato chyba byla opravena a již funguje tato stránka bez chyby.

6.4.1 Návrhy do budoucna

Aplikace by měla být rozšířena pro celý DDM, tudíž by přibyly další kurzy. Dále by měla být rozšířena o předchozí roky, kam by měli být zpětně dopsány účastníci kurzů minulých let, aby se mohla vyvíjet statistika počtů, a zjišťovat tak, zda počet dětí přibývá či ubývá. Dobrým řešením by také bylo zjistit seznam škol v Pardubickém kraji, a tento seznam dát účastníkům na výběr při vypisování názvu školy v elektronické přihlášce (pro ostatní školy vložit textové pole), neboť každý účastník může v současné době školu napsat v jiném tvaru. Z těchto záznamů by se daly vytvářet statistiky, ze které školy přichází nejvíce účastníků, a z které naopak málo, a tudíž do těchto škol dávat například letáky s nabízenými kurzy.

ZÁVĚR

Úkolem této bakalářské práce bylo navrhnout systém pro elektronické přihlašování dětí PS Hroch pomocí elektronického formuláře, a usnadnění tak práce zaměstnancům s přepisováním jejich údajů do databáze. Dále bylo úkolem, na základě informací získaných v DDM, navrhnout a vytvořit aplikaci jako součást informačního systému, která by usnadnila práci zaměstnancům.

Na základě požadavků na IS byla navržena a vytvořena aplikace. Tato aplikace byla testována na školním serveru cipisek.upce.cz, kde byly testovány vybrané funkcionality. Aplikace byla zpřístupněna i „testovacím účastníkům kurzů“, aby prověřili funkčnost aplikace. Vzhledem k bezproblémovému provozu, je tento návrh i aplikace předložena do DDM, kde v červnu 2012 bude probíhat přihlašování účastníků PS Hroch Pardubice.

Na základě těchto faktů, lze tedy konstatovat, že cíle této práce byly splněny.

POUŽITÁ LITERATURA

- [1] Aplikace na Webu: 5. Základy protokolu HTTP. *Aplikace na Webu: 5. Základy protokolu HTTP* [online]. 1999 [cit. 2012-04-14]. Dostupné z: <http://www.kosek.cz/clanky/iweb/05.html>
- [2] Databázové systémy - typy databázových systémů, vývojové trendy a hlavní historické milníky. – Státnice na FM TUL. *Databázové systémy* [online]. 28. 1. 2011 [cit. 2012-04-14]. Dostupné z: http://statnice.obrys.cz/index.php?title=Hlavn%C3%AD_strana
- [3] DELISLE, Marc. *PhpMyAdmin: efektivní správa MySQL*. Vyd. 1. Brno: Zoner Press, 2004, 264 s. ISBN 80-868-1509-9
- [4] *Dům dětí a mládeže BETA Pardubice* [online]. 1999 [cit. 2012-04-14]. Dostupné z: <http://ddmstolbova.com/>
- [5] *HTML Tutoriál 6. - Tabulky - Webtvorba* [online]. 2008 [cit. 2011-11-25]. Vytvoření nejjednodušší tabulky. Dostupné z: <http://webtvorba.howto.cz/html-tutorial-6-tabulky/>
- [6] HUB, Miloslav. *Technologie internetu - PHP 5: distanční opora*. Vyd. 1. Pardubice: Univerzita Pardubice, 2009, 88 s. ISBN 978-80-7395-163-4
- [7] JANOVSKEÝ, Dušan. *Možnosti PHP- formuláře, databáze, soubory, mail* [online]. 2011, 2011 [cit. 2011-11-25]. Možnosti PHP. Dostupné z: <http://www.jakpsatweb.cz/php/moznosti-php.html>
- [8] KANISOVÁ, Hana; MÜLLER, Miroslav. *UMLsrozumitelně*. 2. aktualizované vydání. Brno : Computer Press, a.s., 2006. 176 s. ISBN 80-251-1083-4
- [9] LACKO, Luboslav. *PHP 5 a MySQL 5 : Hotová řešení*. Vyd. první. Brno : Computer Press, a.s., 2007. 320 s. ISBN 978-80-251-1695-1
- [10] *Manuál PHP* [online]. 2005 [cit. 2012-04-14]. Dostupné z: <http://jonatan.spse.pilsedu.cz/doc/php-man/index.html>
- [11] *PhpMyAdmin.cz* [online]. [cit. 2012-04-14]. Dostupné z: <http://phpmyadmin.cz/>
- [12] *Pohybové Studio Hroch* [online]. 2010 [cit. 2011-11-25]. O nás. Dostupné z: <http://pshroch.ddmstolbova.com/>
- [13] SCHMULER, Joseph. *Myslíme v jazyku UML : Knihovna programátora*. První vydání. Praha : Grada Publishing, spol. s r.o., 2001. 360 s. ISBN 80-247-0029-8

- [14] SQL Výuka. *SQL Výuka* [online]. 2009 [cit. 2012-04-14]. Dostupné z: <http://sql-vyuka.cz/d/node/18>
- [15] ŠIMONOVÁ, Stanislava. *Databázové systémy I*. Vyd. 1. Pardubice: Univerzita Pardubice, 2007. 106 s. ISBN 978-80-7194-988-6
- [16] *UML: use case diagram - diagram případů užití* [online]. 2005, 2006 [cit. 2011-11-26]. Use case diagram - diagram případů užití. Dostupné z: <http://mpavus.wz.cz/uml/uml-b-use-case-3-2-1.php>

Příloha C: Formulář s vypsanými chybovými hláškami

PŘIHLÁŠKA

školní rok 2011/2012

Hroch č.:	3 ▾						
* Jméno:	<input type="text"/>	povinná položka	* Příjmení:	<input type="text"/>	povinná položka		
* Pohlaví:	Dívka ▾		* Zdrav. pojišť. ovna:	111 ▾		* Stát. přísl.:	<input checked="" type="radio"/> ČR <input type="radio"/> ostatní
* Rodné číslo (bez lomítka):	<input type="text"/>	obsahuje 10 číslic!	ÚČASTNÍK zájmového vzdělávání:	<input checked="" type="radio"/> Školák <input type="radio"/> Předškolák			
Škola:	<input type="text"/>		Třída:	<input type="text"/>			
* Ulice:	<input type="text"/>	povinná položka	* Číslo popisné:	<input type="text"/>	povinná položka(číslice)		
* PSČ:	<input type="text"/>	obsahuje 5 číslic!	* Město:	<input type="text"/>	povinná položka		
** Matka jméno:	<input type="text"/>	Příjmení:	<input type="text"/>	Č.mobilu:	<input type="text"/>		
** Otec jméno:	<input type="text"/>	Příjmení:	<input type="text"/>	Č.mobilu:	<input type="text"/>		
* ... povinné údaje							
** ... údaje alespoň jednoho z rodičů musí být vyplněny							
<input type="button" value="odeslat"/>							