

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **System pro správu pedagogických úvazků**

Plzeň, 2015

Štěpánka Duffková

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 5. května 2014

Štěpánka Duffková

# Abstract

## **Pedagogical Obligation Management System**

This bachelor thesis describes the realization of a pedagogical obligation management system at the Department of Computer Science and Engineering at the University of West Bohemia in Pilsen. It explains the major terms related to databases and provides an overview of used technology, including the explanations for their selection. This thesis also describes the realization of the database based on the IS/STAG data and creation of system with a user-friendly environment that provides access to the database. Requirements for this application and database are based on the requirements of the contracting authority. Solution created in this bachelor thesis simplifies the process of managing obligations.

# Abstrakt

Tato práce popisuje realizaci systému, jenž umožňuje správu pedagogických úvazků na Katedře informatiky a výpočetní techniky Západočeské univerzity v Plzni. Objasňuje zásadní pojmy týkající se databází a poskytuje přehled využitých technologií, včetně zdůvodnění jejich výběru. Dále popisuje realizaci databáze založené na datech získaných z IS/STAG a tvorbu systému s uživatelsky přívětivým prostředím, který zprostředkovává přístup k databázi. Požadavky na funkcionalitu aplikace i databáze vycházejí z požadavků zadavatele. Řešení uživateli usnadňuje proces správy úvazků.

# Poděkování

Tímto bych ráda poděkovala paní Ing. Janě Krutišové za odborné vedení, věnovaný čas a poskytnutí cenných rad při vedení práce.

Dále bych ráda poděkovala své rodině a přátelům za podporu poskytovanou během celého studia.

# Obsah

<b>1</b>	<b>ÚVOD</b>	<b>1</b>
<b>2</b>	<b>VYMEZENÍ POJMŮ PRÁCE</b>	<b>2</b>
<b>3</b>	<b>ANALÝZA POŽADAVKŮ</b>	<b>10</b>
	3.1 Problematika .....	10
	3.2 Funkční požadavky .....	10
	3.3 Identifikace uživatelů.....	11
<b>4</b>	<b>POUŽÍVANÉ TECHNOLOGIE</b>	<b>12</b>
	4.1 Systém řízení báze dat .....	12
	4.2 Aplikační rozhraní .....	12
	4.2.1 Podpůrné technologie .....	13
	4.3 Webový server .....	14
	4.4 Instalace .....	15
	4.5 Vývojové prostředí.....	15
	4.6 Ostatní využité technologie.....	16
<b>5</b>	<b>TVORBA DATABÁZE</b>	<b>17</b>
	5.1 Datový model.....	17
	5.1.1 OSOBA.....	18
	5.1.2 PŘEDMĚT.....	19
	5.1.3 ROZVRHOVÁ AKCE .....	21
	5.1.4 PRACOVISTĚ.....	23
	5.1.5 UŽIVATEL.....	24
	5.2 Vztahy mezi entitami .....	25
	5.3 Omezení pravomocí uživatelů .....	26
	5.4 Webové služby nad IS/STAG .....	26

<b>6</b>	<b>TVORBA APLIKACE</b>	<b>27</b>
6.1	Návrh systému .....	27
6.2	Vývojový diagram .....	27
6.3	Připojení k databázi.....	28
6.4	Komunikace s databází .....	28
6.5	Přihlášení k aplikaci a odhlášení.....	29
6.6	Mapa aplikace .....	29
6.7	Funkce skriptů.....	30
6.8	Systémové požadavky.....	32
<b>7</b>	<b>TESTOVÁNÍ APLIKACE</b>	<b>33</b>
7.1	Druhy testování.....	33
7.2	Výsledky testování.....	34
<b>8</b>	<b>ZÁVĚR</b>	<b>36</b>
	<b>PŘEHLED ZKRATEK</b>	<b>37</b>
	<b>SEZNAM OBRÁZKŮ</b>	<b>39</b>
	<b>SEZNAM TABULEK</b>	<b>40</b>
	<b>LITERATURA</b>	<b>41</b>
	<b>PŘÍLOHY</b>	<b>44</b>

# 1 Úvod

Cílem této bakalářské práce je vývoj systému, jenž bude sloužit jako prostředek správy úvazků pedagogů působících na Katedře informatiky a výpočetní techniky Západočeské univerzity v Plzni. Systém uživateli umožní kontrolu úvazků plynoucích z pracovní smlouvy zaměstnanců a zároveň nabídne možnost úprav rozvrhových akcí souvisejících se závazky. Správu zprostředkuje přehledné uživatelské prostředí přizpůsobené potřebám zadavatele.

Bakalářská práce je rozdělena do osmi kapitol. První část je věnována odborné terminologii. Definuje pojmy týkající se především databází, které jsou využívány v celém tomto projektu, a tím je jejich znalost nanejvýš vhodná. Následující kapitola se zaměřuje na požadavky zadavatele. Obsahuje popis problematiky související s tímto tématem. Dále shrnuje požadavky a cíle, jichž je třeba v rámci vývoje dosáhnout. Kapitola následující se zaměřuje na technologie, které jsou pro vývoj využity. Charakterizuje vybraný software a poskytuje informace, na jejichž základě je technologie vybrána. Tvorba a popis databáze jsou shrnuty v kapitole páté. Tato kapitola obsahuje datový model, který slouží pro přehlednější zobrazení objektů databáze, a podrobně popisuje význam všech atributů a vztahů. Kapitola šestá se zabývá vytvářeným systémem. Podává informace o funkci jednotlivých skriptů, popisuje proces komunikace aplikace s databází a řeší problematiku zabezpečení. V závěru této práce je provedeno testování kvality. V této kapitole jsou uvedeny a vysvětleny typy testů, kterých je využito pro dosažení bezchybné funkčnosti systému. Uživatelská příručka obsahující popis procesu vkládání dat do databáze a způsob ovládání systému pro správu pedagogických úvazků je vložena jako příloha.

Výstupem této bakalářské práce je funkční aplikace sloužící jako prostředek správy pedagogických úvazků. Pro správu informací je vytvořena databáze využívající dat získaných prostřednictvím webových služeb nad IS/STAG. Spojení těchto dvou celků vytváří systém pro správu pedagogických úvazků.

Na tomto tématu mne zaujala možnost vývoje systému, který usnadní práci uživateli a vývojáři poskytne možnost zdokonalení se nejen v oblasti databázových technologií, ale také v oblasti vývoje softwarových produktů. Dalším důvodem je zcela určitě možnost práce na projektu, který se obsahem přibližuje náplni mého budoucího povolání.

## 2 Vymezení pojmů práce

Důležitou část systému tvoří databáze, z tohoto důvodu je nezbytně nutné vysvětlit, co si pod pojmy souvisejícími s tímto tématem představit. V této kapitole jsou objasněny nejdůležitější termíny, které je nutné pochopit ve spojitosti s bakalářskou prací.

### Databáze

Databáze je množina dat, která jsou uspořádána a dále využívána pro modelování organizačních procesů. Obsahuje strukturované informace, k nimž lze přistupovat a efektivně mezi nimi vyhledávat. Data v databázích jsou řízena balíkem aplikací, který se nazývá systém řízení báze dat. [6]

### Systém řízení báze dat

Systém řízení báze dat neboli SŘBD je obecný softwarový systém, který řídí přístup k databázi. Spolu s databází tvoří Databázový systém - DBS. [6]

$$DB + SŘBD = DBS [6]$$

SŘBD je složen z programů, které umožňují pracovat s daty ve smyslu jejich správy a ovládání. Data jsou uložena ve struktuře zvané datový model. Mezi datové modely se řadí model hierarchický, relační a síťový. [2, 6] Pro potřeby této práce poslouží model relační.

### Relační model

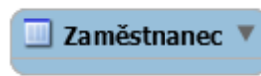
Relační model prezentuje data v podobě dvourozměrných tabulek, jež se skládají z řádků (záznamů) a sloupců (atributů). Záznamy se označují jako n-tice. Mezi dvěma tabulkami může existovat kardinalita vztahu: 1:1, 1:N, M:N. [2, 6] Vysvětlení kardinality vztahu a pojmů atribut a entita následují níže.

### Entita

Entity lze chápat jako „prvky“ z reálného světa, jejichž údaje jsou sledovány a zaznamenávány do databáze. [5] Pod pojmem entita si lze představit jakýkoliv objekt,



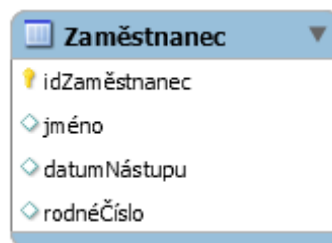
o kterém je potřeba uchovávat informace. Příklad entity ukazuje obrázek č. 1. V tomto případě se zaznamenávají informace týkající se zaměstnance.



Obrázek 1 - Entita

## Atribut

Atributy charakterizují či popisují entitu. Jsou funkcí, která entitě přiřazuje vlastnosti. [5, 6] Jako příklad takovéto vlastnosti lze uvést jméno, datum nástupu zaměstnance do zaměstnání nebo rodné číslo zaměstnance, viz obrázek č. 2. Každá entita musí obsahovat jedinečný identifikátor entity, v tomto případě atribut nazvaný „idZaměstnanec“ obsahující identifikační číslo zaměstnance.



Obrázek 2 - Entita s atributy

## Primární klíč

Jedinečný identifikátor entity se nazývá primární klíč. Takovýto atribut přiřazuje každému záznamu v entitě jedinečnou hodnotu. Na obrázku č. 2. je primární klíč označen žlutou značkou.

## Cizí klíč

Cizí klíč definuje vzájemnou souvislost mezi entitami. Určuje, které záznamy v tabulkách spolu vzájemně souvisí. [5] Cizí klíč je na obrázku č. 3. označen červenou značkou.

## Kardinalita vztahu

Data v databázi spolu souvisí a jejich propojení je jednoznačně určeno vztahy. Kardinalita určuje, v kolika vztazích se entita může vyskytovat. Vztahy mohou být typu

1:1, 1:N a M:N. V následující části jsou za účelem snazší představy používány pojmy rodičovská a dceřiná entita. Rodičovská entita vkládá svůj primární klíč do entity dceřiné, kde se z primárního klíče stává cizí klíč. [1]

- **Vztah typu 1:1** - záznam jedné entity může být přiřazen nanejvýš jednou k záznamu entity druhé a naopak. [5] Příklad vztahu zachycuje obrázek č. 3. Vztah mezi rodičovskou entitou „Manažer“ a dceřinou entitou „Oddělení“ naznačuje, že s jedním oddělením může být svázán pouze jeden manažer. V tomto vztahu i manažer může být ve vedení pouze jednoho oddělení. [1]



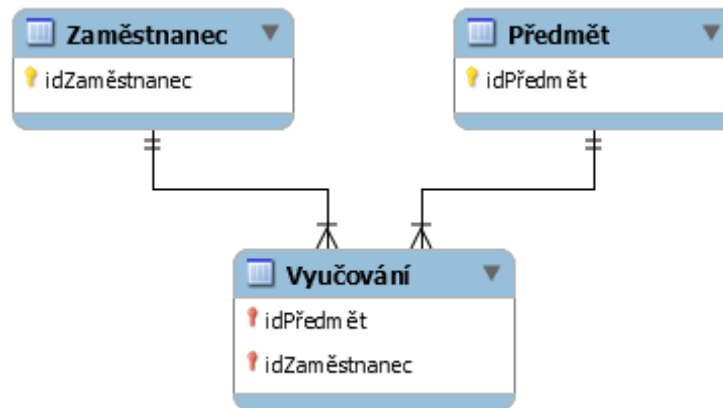
Obrázek 3 - Vztah typu 1:1

- **Vztah typu 1:N** - záznam jedné entity může být přiřazen jednomu nebo více záznamům entity druhé. Avšak každý záznam entity druhé může být svázán pouze s jedním záznamem entity první. [5] Příklad vztahu 1:N s rodičovskou entitou „Zaměstnanec“ a dceřinou entitou „Cvičení“ zachycuje obrázek č. 4. Zaměstnanec může vyučovat na více cvičeních, nicméně dané cvičení může vyučovat pouze on samotný.



Obrázek 4 - Vztah typu 1:N

- **Vztah typu M:N** - záznam jedné entity může být přiřazen jednomu nebo více záznamům entity druhé a naopak. [5] Příklad vztahu M:N s rodičovskou entitou „Zaměstnanec“ a dceřinou entitou „Předmět“ zachycuje obrázek č. 5. Zaměstnanec může vyučovat více předmětů a předmět může být vyučován více zaměstnanci. Vztah M:N se realizuje pomocí kombinace vztahů 1:N a 1:M. Tím vzniká nová tabulka, která se nazývá vazební. Tato vazební tabulka obsahuje cizí klíče dceřiné i rodičovské entity. [1] Vazební tabulka je na obrázku č. 5. nazvána „Vyučování“.



Obrázek 5 - Vztah typu M:N

## Integrita

Spolu s vytvářením relačního modelu je důležité zajistit integritu databáze. Data v databázi musí být konzistentní, tzn. že nesmějí obsahovat nereálnou informaci a musí být platná. [25]

Existují tři typy integrity: [1, 2, 25]

- **Integrita entity** - zaručí, že se v tabulce neobjeví duplicitní záznamy, a že atribut identifikující každý záznam v tabulce je jedinečný a nenabývá hodnoty NULL. Význam hodnoty NULL lze nalézt níže.
- **Doménová integrita** - zaručuje spolehlivost struktury atributu. Kontroluje, zda všechny neklíčové atributy nabývají pouze přípustných hodnot. Definuje se datovými typy.
- **Referenční integrita** - zaručí synchronizaci záznamů v propojených tabulkách během změn prováděných na datech. Referenční integritu ošetřuje databázová technologie, která určuje, zda je změna povolena či zakázána.

Způsoby ošetření referenční integrity: [25]

- RESTRICT - zabraňuje změně a smazání záznamů v rodičovské tabulce právě tehdy, když existující záznamy dceřiné tabulky odkazují na cizí klíč.
- CASCADE - umožňuje adekvátní změnu a výmaz záznamů odkazujících na smazaný či změněný cizí klíč.

## Datové typy

Mezi datové typy se řadí typy číselné, textové, výčtové a také typy ukládající datum. Druhy, jejichž znalost je v této práci potřeba, jsou v následující části vysvětleny.

### Číselný datový typ

Číselné datové typy slouží k ukládání celých a reálných čísel. Nelze do nich uložit text. Každý podtyp vyžaduje různou velikost paměti. Je proto nanejvýš vhodné předem promyslet, jak velká část paměti je pro daný celek potřebná. [4]

[X] - toto označení dovoluje nastavit maximální šířku zobrazení. Pokud není uvedeno jinak, jedná se o číslo z rozsahu 1 - 255. [4]

Používané druhy číselných datových typů: [4, 13]

- TINYINT - tento typ je vhodný pro malá celá čísla. Rozsah je mezi -128 a 127. Zabírá paměťový prostor o velikosti 1 bajt.
- SMALLINT [X] - tento typ je vhodný pro větší celá čísla. Rozsah je mezi -32768 a 32767. Zabírá paměťový prostor o velikosti 2 bajtů.
- INT [X] - tento typ je vhodný pro velká celá čísla. Rozsah je mezi -2147483648 a 2147483647. Zabírá paměťový prostor o velikosti 4 bajtů.

### Textový datový typ

Textové datové typy jsou využívány pro ukládání textu jakéhokoliv formátu. Lze do nich také uložit čísla. Avšak i v tomto případě je vhodné promyslet, které typy jsou v databázi potřebné. Nejčastěji užívanými datovými typy jsou VARCHAR a CHAR. Mezi nimi je však veliký rozdíl. VARCHAR definuje pole proměnné délky. To znamená, že pokud je nadefinováno pole o velikosti VARCHAR(15) a vloží se do něj pouze 5 znaků, délka daného pole je rovna 6 bajtům, tedy délce pole +1. CHAR oproti tomu definuje pole konstantní délky. To znamená, že pokud je nadefinováno pole o velikosti CHAR(15) a vloží se do něj pouze 5 znaků, délka daného pole je rovna 15 bajtům. Tyto typy v jedné tabulce nelze kombinovat. Výhodou datového typu CHAR je rychlejší zpracování tabulek, které jej obsahují. Výhodou datového typu VARCHAR je úspornější ukládání dat. [4]

Používané druhy textových datových typů: [4, 13]

- VARCHAR [X] - tento typ je vhodný pro textový řetězec pevné délky 0 – X. Zabírá paměťový prostor o velikosti délky hodnoty + 1 bajt, který značí délku.

- TEXT - tento typ je vhodný pro ukládání delších řetězců. Stejně jako VARCHAR definuje pole proměnné délky. Velikost řetězce může být maximálně 65535 bajtů. Zabírá paměťový prostor o velikosti délky řetězce + 2 bajty, které značí délku.

Mezi datové typy lze zařadit také typy pro zaznamenání data či výčtu položek. Jsou speciální, a proto je nelze zařadit do předchozích kategorií: [4]

- ENUM - tento typ je vhodný pro deklarování povolených hodnot, kterých má atribut nabývat. Atributu lze přiřadit pouze jednu z nabízených hodnot. Jeho tvar je ve formátu ('hodnota1', 'hodnota2'). Zabírá paměťový prostor o velikosti 1 bajt pro výčet se členy o velikosti 1 - 255 a 2 bajty pro výčet se členy o velikosti 256 - 65535.
- YEAR - tento typ se využívá pro uvedení hodnoty kalendářního roku. Jeho tvar je ve formátu YYYY. Rozsah tohoto formátu je od 1901 do 2155. Zabírá paměťový prostor o velikosti 1 bajt.

### **Hodnota NULL**

Každá položka nabývající hodnoty NULL reprezentuje neznámá nebo chybějící data. Hodnota NULL nereprezentuje nulu či řetězce skládající se z mezer. [1]

### **Upřesňující informace k datovým typům**

Upřesňující informace jsou vlastnostmi atributů. Mohou zjednodušit práci s databází a zároveň zabránit vzniku budoucích problémů. Mezi tyto vlastnosti patří: [8]

- PK - PRIMARY KEY - primární, jedinečný klíč
- NN - NOT NULL - takto zvolená hodnota nesmí být nikdy NULL
- UQ - UNIQUE - atribut nesmí obsahovat více položek, které nabývají stejných hodnot
- AI - AUTO INCREMENT - automatické přidělování hodnoty o číslo větší než v předchozím případě
- DEFAULT - hodnota, kterou položka dostane jako výchozí

### **Normalizace**

Dalším, neméně důležitým pojmem je normalizace. Normalizování dat je proces, pomocí kterého se rozkládají vztahy. Tím je zajištěna jednodušší práce s daty a zároveň

je tím zabráněno redundanci dat, tj. opakování dat v databázi. Normalizace má několik forem. S vyššími formami se zjednodušuje práce s daty. [5]

Normální formy: [5]

1. normální forma (1NF) - tabulka je v 1. normální formě právě tehdy, když neobsahuje takové atributy, které by mohly nabývat násobných hodnot.
2. normální forma (2NF) - tabulka je v 2. normální formě právě tehdy, když je tabulka zároveň v 1. normální formě a atributy, které nejsou klíčem, závisí na celém primárním klíči nikoliv na jeho podmnožině.
3. normální forma (3NF) - tabulka je v 3. normální formě právě tehdy, když je relace zároveň v 2. normální formě a data jsou závislá pouze na klíči, ne na sobě navzájem. Všechny neklíčové atributy musí být vzájemně nezávislé.

## **Databázový jazyk**

Důležitou součástí problematiky databází je komunikace s nimi. Jazyk SQL (Structure Query Language) lze považovat za univerzální jazyk relačních databází, jenž zprostředkovává styk uživatele a databáze. Je příkazově orientovaný a jeho příkazy jsou formulovány pomocí klíčových slov a parametrů. [2, 5]

Druhy: [2, 3, 13]

- DQL (Data Query Language) - jazyk, který umožňuje dotazování dat. Takovéto příkazy se databáze pouze dotazují, nemají však žádná práva provádět změny. Do této skupiny patří SELECT.
- DML (Data Manipulation Language) - uživatelský prostředek, který umožňuje manipulovat s daty. DML příkazy mají dostatečná práva pro modifikaci dat v databázi. Do této skupiny patří: DELETE, INSERT, UPDATE.
- DDL (Data Definition Language) - jazyk pro definování dat. Tyto příkazy manipulují přímo s databázovými objekty. Jsou využívány správci databází, jelikož umožňují vytvářet pravidla pro ukládání dat či přístup k nim. Do této skupiny patří: CREATE, DROP, ALTER.
- DCL (Data Control Language) - jazyk určený k řízení dat. Tyto příkazy jsou určené k přidávání nebo odebrání oprávnění jednotlivým uživatelům. Do této skupiny patří: GRANT, REVOKE.

## **Znakové sady**

Databáze podporují různé druhy znakových sad. Znakové sady jsou skupiny znaků, jejich číselných kódů a způsobů řazení. Čeština využívá tři znakové sady: [18]

- ISO Latin 2 (latin2\_czech\_cs)
- Windows 1250 (cp1250\_czech\_cs)
- UTF-8 (utf8\_czech\_ci)

Tyto sady nelze kombinovat. Je velice důležité, aby tvořená databáze a vkládaná data byly ve shodné sadě. V opačném případě mohou vznikat problémy se zobrazením českých znaků. [18]

## **Open Source**

K databázovým technologiím, ale nejenom k nim, se váže také pojem Open Source. Vznik pojmu Open Source se datuje do roku 1998, kdy se firma Netscape rozhodla pro zveřejnění zdrojového kódu prohlížeče Navigator. Na základě tohoto podnětu vznikla iniciativa OSI (Open Source Initiatives). Postupem času programátoři začali přispívat svými opravami a novými nápady. OSI se stalo velmi důležitým pojmem v počítačovém světě. Programy typu Open Source jsou dostupné veřejnosti, jejich užívání není zpoplatněno a nejsou závislé na platformách. [3]

## 3 Analýza požadavků

Kapitola analýzy požadavků podává informace o situacích spojených se správou úvazků. Obsahuje stručně popsané stavy, ke kterým při správě dochází a specifikuje požadavky zadavatele, které slouží jako cíle vývoje systému. Tato fáze představuje velmi podstatnou část vývoje, jejíž podcenění může mít za následek nenaplnění účelu systému.

### 3.1 Problematika

V období standardního předzápisu si student vytváří rozvrh z nabízených rozvrhových akcí. Mezi tyto akce se řadí: cvičení, přednášky a semináře. Každá z těchto položek má nastavenou platnost, která určuje, zda je možné se na ni přihlásit. Má také přiřazeného pedagoga, který ji bude vyučovat. Student může svůj rozvrh upravovat do chvíle ukončení předzápisu. Po jeho ukončení vznikají tyto situace:

1. Obsazenost na rozvrhové akci je nízká, tudíž nejsou splněny podmínky pro otevření.
2. Pedagog přesahuje svůj pracovní úvazek počtem přidělených rozvrhových akcí.

Zmíněné situace byly doposud řešeny zdlouhavou manuální kontrolou pedagogických úvazků. Tento proces bude nahrazen jednoduchou správou systému.

### 3.2 Funkční požadavky

Mezi funkční požadavky spadají operace, které jsou pro správný chod systému zásadní. Základ bude tvořit databáze obsahující nejnnutnější informace, které uživatel potřebuje k efektivnímu rozhodování při úpravě rozvrhových akcí. Data, jež budou do tohoto systému vkládána z již existující databáze Západočeské univerzity v Plzni, budou následně přizpůsobena vytvořené databázi a vložena do vybrané správcovské aplikace. Pro částečnou správu a zobrazení informací je třeba vytvořit uživatelsky přívětivý systém, který bude využíván na operačním systému Windows.

Systém uživateli nabídne:

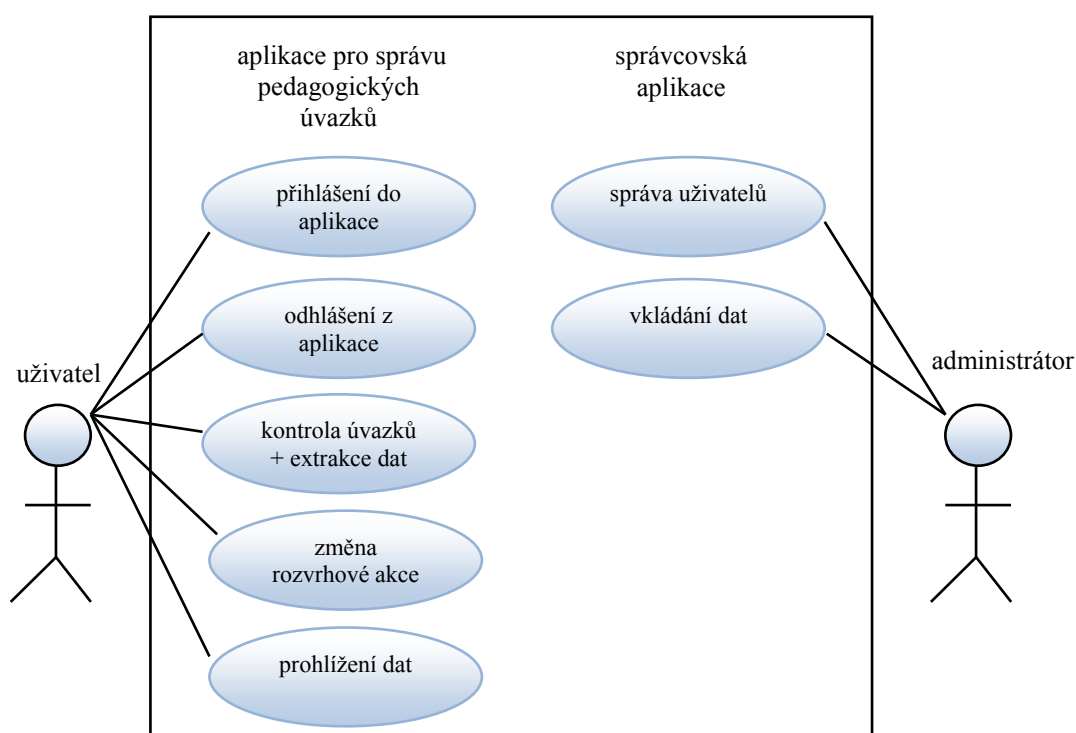
- prohlížení databázových dat
- možnost výběru pedagoga vztahujícího se k rozvrhové akci
- možnost změny platnosti akce



- přehled úvazků jednotlivých pedagogů
- možnost tisku a extrakce dat týkajících se úvazků

### 3.3 Identifikace uživatelů

V rámci zpřehlednění požadavků je vytvořen diagram případu užití, viz obrázek č. 6. Systém je určen pro dva typy uživatelů: administrátora a uživatele. Administrátor řídí přístup k systému a vkládá data do vybrané správcovské aplikace. Uživatel prostřednictvím tvořené aplikace pro správu pedagogických úvazků pracuje s daty.



Obrázek 6 - Diagram případu užití

Služby poskytované systémem, tedy správcovskou aplikací a aplikací pro správu pedagogických úvazků jsou využívány zadavatelem osobně. Jeho rolemi jsou tedy jak administrátor, tak uživatel. Přesto je však nutné vytvořit omezení přístupu k tvořené aplikaci, a to prostřednictvím přihlašovacího formuláře.

## 4 Používané technologie

Tato kapitola obsahuje informace o vybraných technologiích, kterých je využito k tvorbě aplikace, jež slouží jako prostředek správy pedagogických úvazků. Pro tvorbu je potřeba zvolit vhodný systém pro řízení báze dat a aplikační programovací rozhraní. Volba technologií se odvíjí od cílů stanovených ve fázi sběru požadavků.

### 4.1 Systém řízení báze dat

Výběr vhodného systému pro řízení báze dat je klíčovou položkou při tvorbě aplikace nad databází. Proto je nutné důkladně prozkoumat možnosti, které software nabízí. V rámci této práce je vybrána databáze MySQL, jejíž vlastnosti jsou popsány níže.

#### MySQL

MySQL je databázový software, který byl vyvinut firmou My SQL AB v roce 1995 ve Švédsku. V roce 2010 byla tato společnost převzata společností Oracle Corporation. MySQL pracuje na operačních systémech Windows, Linux i OS X. Nabízí dvě verze. První je Open Source, ale lze si zakoupit komerční rozšířenou verzi od společnosti Oracle. [17]

Hlavními výhodami tohoto databázového softwaru jsou rychlost, spolehlivost a nenáročnost. MySQL umožňuje velmi rychlé nastavení a snadnou správu. Dokáže pojmout velké množství dat, a to bez velké ztráty na výkonu. Tento software je využíván velmi rozsáhlou komunitou uživatelů, což má za výsledek četné množství návodů na internetu pro začátečníky i pokročilé. Umožňuje nasazení na více serverech a zprostředkovává přístup z mnoha programovacích jazyků, například PHP, C, C++, Java, Perl či .NET. Dokumentace tohoto nástroje je velice obsáhlá. [22]

Vlastnosti databázové technologie MySQL jsou klíčem pro tvorbu kvalitní, nenáročné databáze. V této práci využít databázový nástroj MySQL verze 5. 6. 17.

### 4.2 Aplikační rozhraní

V kombinaci se zvolenou databází MySQL se k tvorbě aplikací využívají skriptovací jazyky. Skriptovací jazyky umožňují rychlý vývoj a jednoduchou instalaci aplikací. Používají se především při tvorbě grafických uživatelských rozhraní či webových

aplikací. Webové aplikace se tvoří pomocí databáze uchovávající data a skriptů, které zajišťují funkčnost. [7]

Dnešním nejrozšířenějším standardem pro vytváření webových aplikací je kombinace MySQL a PHP. Tato dvojice je proslulá a důvodů pro jejich použití je několik. Objasnění volby skriptovacího jazyka se věnuje část o technologii PHP. [7]

## **PHP**

PHP je skriptovací jazyk, který byl dříve známý jako Personal Home Pages. Tento jazyk umožňuje vytváření webových aplikací a také dynamických webových stránek. [7]

Byl vytvořen v roce 1994 a původně se využíval jako jazyk pro tvorbu formulářů na webových stránkách. O rok později byl pozměněn na PHP/FI, neboli Personal Home Page/Forms Interpreter. Změna byla nutná z důvodu narůstajících požadavků na webové stránky. V tuto dobu byla připojena i podpora pro MySQL. V průběhu času se ustálil pojem Hypertext Preprocessor. [19]

Mezi největší výhody tohoto jazyka lze zařadit podporu nejen široké řady souvisejících technologií, formátů a standardů, ale i internetových protokolů například: HTTP, FTP či POP3. Má rozsáhlou podporu komunity a komunikace s databází MySQL, ale i mnoha dalšími, je také velice snadná. [19] V dnešní době existuje četné množství hotových řešení pro okamžité použití, což je velikou výhodou pro začínající vývojáře.

Z důvodů shrnutých v této části je pro tvorbu aplikace využit jazyk PHP verze 5.5.12.

Pro práci s PHP je zapotřebí tří technologií. Databázového systému, vývojového prostředí pro editaci skriptů a také webového serveru s podporou PHP skriptů. [7] Jako vhodný systém pro řízení báze dat je již vybrána technologie MySQL. Volba vývojového prostředí a webového serveru je popsána níže.

### **4.2.1 Podpůrné technologie**

Jako hlavní skriptovací jazyk je zvolen jazyk PHP, nicméně pro optimální chod systému je potřeba využít dalších technologií. Mezi tyto technologie patří: HTML, CSS, JavaScript a AJAX. Kombinace těchto komponent umožňuje tvorbu vysoce interaktivních aplikací.

## **HTML**

HTML (HyperText Markup Language) je sadou značkovacích symbolů, které slouží pro popis struktury webových stránek. Patří mezi značkovací jazyky, které definují strukturu dokumentů. Dokument se skládá ze značek, které jsou nazývány tagy. [24]

## **CSS**

CSS (Cascading Style Sheets), neboli kaskádové styly slouží pro vytvoření stylu webových stránek psaných v jazyce HTML. Udávají styl i formu a nabízí širokou škálu grafických prvků, a to pro více stránek najednou. [24]

## **JavaScript**

JavaScript je populárním objektově orientovaným, interpretovatelným, skriptovacím jazykem pro vytváření interaktivních webových stránek. Ačkoli vychází z programovacích jazyků C a Javy, společnou mají pouze podobnou syntaxi. [24]

## **AJAX**

AJAX (Asynchronous JavaScript and XML) je technologie velmi často využívaná webovými aplikacemi. Je kombinací technologií HTML, JavaScript, XML a XMLHttpRequest<sup>1</sup>. AJAX umožňuje získávání a ukládání dat ze serveru bez nutnosti znovu nahrávání celé stránky. [10]

### **4.3 Webový server**

K testování skriptů PHP je zapotřebí webového serveru. Server je programem, jenž poskytuje služby klientům. Zpracovává požadavky prohlížečů a zprostředkovává odeslání požadovaného souboru, tedy příslušné webové stránky. [7] Velmi známým řešením v kombinaci technologií PHP a MySQL je server Apache.

## **Apache**

Apache je Open Source webový server dostupný pro operační systémy Linux i Windows. Vznik se váže k roku 1993, kdy Rob McCool na univerzitě v Illinois vytvořil projekt NCSA HTTPd. O rok později, v roce 1994, McCool vývoj opustil. Ostatní

---

<sup>1</sup> Objekt umožňuje odeslat XML dokument v těle HTTP požadavku.

vývojáři přispívali do projektu svými softwarovými záplatami, a tím vytvořili webový server s názvem Apache. [11]

Mezi jeho výhody patří především nulová cena a multiplatformost. [11]  
Z těchto důvodů je Apache využit jako webový server, a to ve verzi 2. 4. 9.

## 4.4 Instalace

Aplikace je tvořena na operačním systému Windows 7 v kombinaci komponent MySQL, PHP a Apache. Instalace a konfigurace technologií v dnešní době hotových řešení není nutná. Proto je v této práci využito řešení, jež umožňuje nastavení automatické.

### WampServer

WampServer je platforma určená pro vývoj webových aplikací na systému Windows. Název Wamp je složen z prvních písmen technologií Windows, Apache, MySQL a PHP. Po instalaci WampServeru se vytvoří virtuální server, který umožňuje spouštění PHP skriptů a testování webových aplikací. [23]

WampServer je zvolen kvůli správcovské aplikaci phpMyAdmin, kterou poskytuje, a kvůli jednoduché instalaci s optimálním defaultním nastavením. [23]  
Z důvodů shrnutých v této části je využit balíček nástrojů WampServer verze 2. 5.

## 4.5 Vývojové prostředí

Vývojové prostředí neboli IDE (Integrated Development Environment) je sada nástrojů, pomocí kterých lze vytvářet, ladit a udržovat zdrojový kód.

### Eclipse

Eclipse je jedno z nejznámějších vývojových prostředí pro Javu. Je založeno na konceptu rozšiřitelnosti pomocí pluginů. Umožňuje kombinaci jazykové podpory pomocí nabízených balíčků, jako jsou C/C++ či PHP. [20] Eclipse stejně jako spousta jiných IDE poskytuje nabídku rozvinutí metod a funkcí. Zvýrazňuje syntaxe a napovídá názvy metod.

Z důvodů shrnutých v této části je k vývoji využit Eclipse verze 3. 9. 1.

## 4.6 Ostatní využité technologie

### phpMyAdmin

PhpMyAdmin je bezplatným nástrojem, který umožňuje správu databáze pomocí webového rozhraní. Nabízí širokou škálu funkcí. Zprostředkovává tvorbu, prohlížení, úpravu i mazání databází, tabulek a atributů. Dále povoluje export do formátů: CSV, XML PDF i SQL a import formátů CSV či SQL. Další výhodou phpMyAdmina je dostupnost v českém jazyce. [9]

Tento nástroj je využit jako správcovská aplikace, jejímž prostřednictvím jsou do databáze vkládána data a pomocí které se řídí přístup uživatelů k tvořenému systému. Využita je verze 4. 4. 2.

### Bootstrap Framework

Framework<sup>2</sup> Bootstrap je jednoduchým nástrojem pro tvorbu webových aplikací. Vývojáři usnadňuje práci s typografií, vytvářením uživatelského rozhraní a ošetřuje zobrazení na různých platformách. Podporuje webové technologie HTML, CSS a JavaScript. [12]

Framework Bootstrap je Open Source nástrojem a využita je verze 3. 3. 4.

### MySQL Workbench

MySQL Workbench je Open Source nástroj, který je vytvořen tvůrci MySQL. Program lze spustit pod operačními systémy Windows, Linux i OS X. Workbench je kvalitním vizuálním nástrojem, který zjednodušuje tvorbu databází. [16]

Nabízí prostředí pro datové modelování, vývoj SQL a nástroje pro nastavení serveru a uživatelů. V této práci je využita jeho funkce tvorby komplexního datového modelu. Využita je verze 6. 2.

---

<sup>2</sup> Softwarová struktura sloužící jako podpora pro vývoj jiného softwarového projektu.

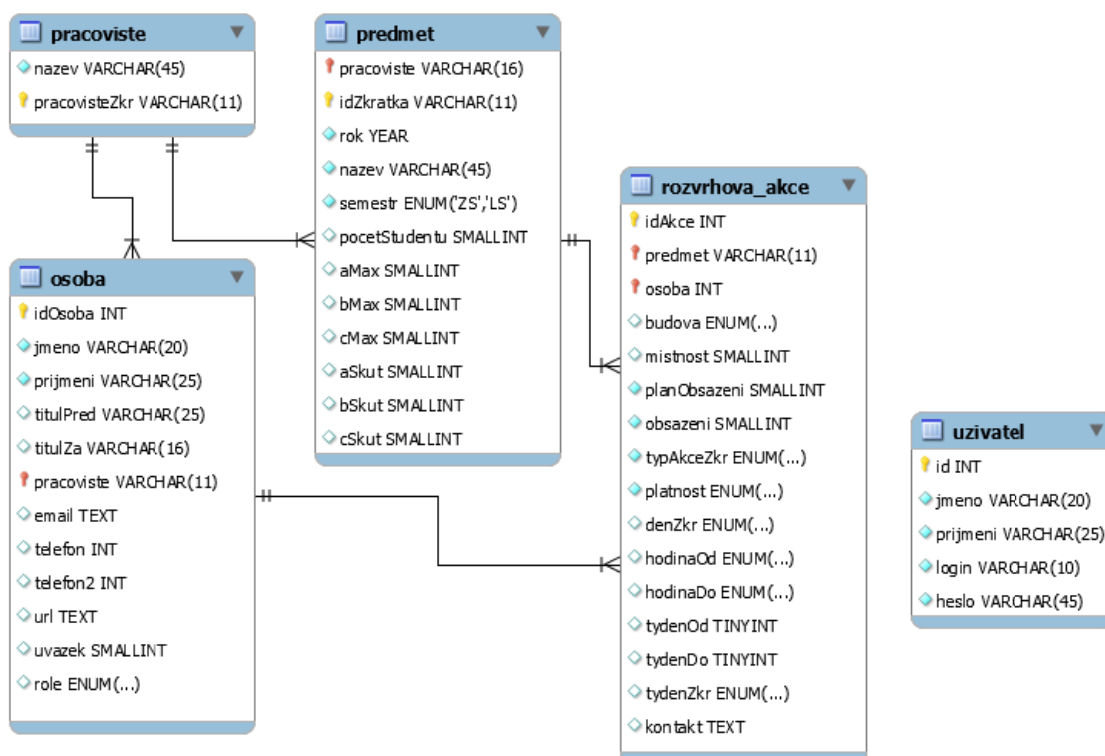
# 5 Tvorba databáze

Databáze je tvořena pomocí nástroje MySQL Workbench, který je popsán v předchozí kapitole. Tato část se věnuje tvořené databázi. Obsahuje datový model, jehož entity, vazby mezi nimi a atributy jsou podrobně specifikovány.

## 5.1 Datový model

Pro dosažení požadovaných výsledků je při tvorbě databáze nutné navrhnout vhodný datový model. Vygenerovaný návrh je velice přehledný nejen pro vývojáře, ale i pro uživatele, kteří potřebují k projektu přistupovat. Vazby mezi jednotlivými tabulkami jsou zjevné a výsledný obrázek datového modelu lze využít i v dokumentacích.

Vyvíjený datový model, viz obrázek č. 7., se skládá z pěti tabulek a čtyř vazeb mezi nimi. Databázi tvoří entity: „OSOBA“, „PREDMET“, „ROZVRHOVA\_AKCE“, „KATEDRA“ a „UZIVATEL“. První čtyři entity tvoří část, která je důležitá pro získání informací o zaměstnancích a rozvrhových akcích. Entita „UZIVATEL“ slouží pouze jako prostředek pro identifikaci uživatele a ověření jeho identity.



Obrázek 7 - Datový model

Navržená databáze splňuje 3NF pro každou tabulku. Normální formy jsou pouze doporučením, které má databáze splňovat, nicméně databáze v 3NF zajistí optimální využití systému.

### 5.1.1 OSOBA

Entita „OSOBA“ se skládá z dvanácti atributů. Data jsou stažena pomocí webových služeb z IS/STAG. Tato služba je popsána níže. Sloupce jsou nazvány, v rámci přehlednosti, podobnými či shodnými názvy. Přehled jednotlivých atributů a jejich vlastností lze nalézt v tabulce č. 1.

Název sloupce	Datový typ	PK	NN	AI	Default
<b>idOsoba</b>	INT	•	•		
<b>jmeno</b>	VARCHAR(20)		•		
<b>prijmeni</b>	VARCHAR(25)		•		
<b>titulPred</b>	VARCHAR(25)				NULL
<b>titulZa</b>	VARCHAR(16)				NULL
<b>pracoviste</b>	VARCHAR(11)		•		
<b>email</b>	TEXT				NULL
<b>telefon</b>	INT				NULL
<b>telefon2</b>	INT				NULL
<b>url</b>	TEXT				NULL
<b>uvazek</b>	SMALLINT				NULL
<b>role</b>	ENUM				NULL

Tabulka 1 - Tabulka OSOBA

**idOsoba** Atribut je primárním klíčem této entity. Je nastaven na datový typ INTEGER a nesmí nabýt hodnoty NULL. Není nastaven na AUTO\_INCREMENT, což má opodstatněný důvod: data nejsou nově vytvářena, jsou přebírána z již existující databáze.

**jmeno** Atribut obsahuje křestní jména vyučujících. Je nastaven na datový typ VARCHAR a jeho délka je po prozkoumání získaných dat odhadnuta na 20 znaků.

**prijmeni** Atribut obsahuje příjmení vyučujících. Je také nastaven na datový typ VARCHAR a jeho délka je po prozkoumání získaných dat odhadnuta na 25 znaků.



<b>titulPred,</b> <b>titulZa</b>	Tyto atributy zaznamenávají informace o titulech vyučujících. V případě stejného jména mohou uživateli aplikace pomoci identifikovat správného pedagoga. Jsou nastaveny na datový typ VARCHAR a je u nich nutno podotknout, že nejsou položkami vícenásobných hodnot. Každá položka je brána jako jedna hodnota, podobně jako složená jména osob. Délka je po prozkoumání získaných dat odhadnuta na 16 a 25 znaků.
<b>pracoviste</b>	Atribut je cizím klíčem entity „OSOBA“ a odkazuje na atribut „pracovisteZkr“ entity „PRACOVISTE“. Jednoznačně identifikuje katedru, na které vyučující pracuje. Referenční integrita je nastavena na ON DELETE RESTRICT a ON UPDATE CASCADE.
<b>email</b>	Atribut email je v databázi uveden z důvodu možné potřeby písemné komunikace s vyučujícím. Je nastaven na datový typ TEXT. Položky tohoto atributu nemusí obsahovat data.
<b>telefon,</b> <b>telefon2</b>	Tyto atributy obsahují kontakt na kancelář či mobilní telefon vyučujícího. Jsou nastaveny na datový typ INTEGER a hodnotu NULL.
<b>url</b>	Atribut obsahuje odkaz na webové stránky vyučujícího. Je nastaven na datový typ TEXT. Položky tohoto atributu nemusí nutně obsahovat data.
<b>uvazek</b>	Tento atribut je velmi důležitý pro splnění požadované funkce aplikace. Obsahuje číslo, které informuje uživatele o maximálním počtu odpracovaných hodin vyučujícího. Je nastaven na datový typ SMALLINTEGER a defaultní hodnotu NULL.
<b>role</b>	Atribut informuje o roli zaměstnance. Defaultní nastavení tohoto atributu je NULL. Dále je nastaven na datový typ ENUM a ve výčtu položek lze vybrat jednu z možností: <ul style="list-style-type: none"> <li>• doktorand</li> <li>• učitel</li> <li>• externista</li> </ul>

### 5.1.2 PŘEDMĚT

Tato entita se skládá z dvanácti atributů. Jednotlivé atributy jsou vybírány z dat stažených z IS/STAG a pojmenovány, v rámci přehlednosti, podobnými či shodnými názvy. Přehled jednotlivých atributů a jejich vlastností lze nalézt v tabulce č. 2.

Název sloupce	Datový typ	PK	NN	AI	Default
<b>pracoviste</b>	VARCHAR(16)		•		
<b>idZkratka</b>	VARCHAR(11)	•	•		
<b>rok</b>	YEAR		•		
<b>nazev</b>	VARCHAR(45)		•		
<b>semestr</b>	ENUM		•		
<b>pocetStudentu</b>	SMALLINT				NULL
<b>aMax</b>	SMALLINT				NULL
<b>bMax</b>	SMALLINT				NULL
<b>cMax</b>	SMALLINT				NULL
<b>aSkut</b>	SMALLINT				NULL
<b>bSkut</b>	SMALLINT				NULL
<b>cSkut</b>	SMALLINT				NULL

Tabulka 2 - Tabulka PŘEDMĚT

- pracoviste** Atribut je cizím klíčem, který odkazuje na atribut „pracovisteZkr“ entity „PRACOVISTE“. Jednoznačně identifikuje katedru, na které je předmět vyučován. Referenční integrita je nastavena na ON DELETE RESTRICT a ON UPDATE CASCADE.
- idZkratka** Atribut je primárním klíčem této entity. Je nastaven na datový typ VARCHAR a nesmí nabýt hodnoty NULL. Atribut je tvořen ze zkratk jednotlivých předmětů. V rámci jedné katedry je toto označení jednoznačné. Délka je po prozkoumání získaných dat odhadnuta na 11 znaků.
- rok** Tento atribut označuje rok vyučování daného předmětu. Je nastaven na datový typ YEAR a jeho hodnota nesmí být NULL.
- nazev** Atribut obsahuje celý název předmětu. Je nastaven na datový typ VARCHAR a jeho hodnota nesmí být NULL. Délka je po prozkoumání získaných dat odhadnuta na maximálně 45 znaků.
- semestr** Atribut informuje o semestru, kdy je předmět vyučován. Nesmí nabýt hodnoty NULL a je nastaven na datový typ ENUM, tudíž lze vybrat jen jednu variantu z výčtu možností:
- ZS - zimní semestr
  - LS - letní semestr
- pocetStudentu** Atribut obsahuje číslo studentů, kteří jsou na daný předmět v aktuálním roce zapsaní. Je nastaven na datový typ SMALLINTEGER a defaultní hodnotu NULL.

**aMax,**  
**bMax,**  
**cMax**

Tyto atributy slouží k nastavení kapacit studentů, pro které je daný předmět:

- a) Oborovým předmětem
- b) Povinně volitelným předmětem
- c) Předmětem zapsaným mimo studijní plán

Jsou nastaveny na datový typ SMALLINTEGER. Jejich hodnota je defaultně nastavena na hodnotu NULL.

**aSkut,**  
**bSkut,**  
**cSkut**

Tyto atributy obsahují skutečné počty studentů při zápisu na daný předmět. Rozdělení je stejné, jako u výše zmíněných. Jsou opět nastaveny na datový typ SMALLINTEGER a jejich defaultní hodnota je NULL.

### 5.1.3 ROZVRHOVÁ AKCE

Tato entita se skládá z šestnácti atributů. Jednotlivé atributy jsou vybírány z dat stažených pomocí webových služeb z IS/STAG a pojmenovány, v rámci přehlednosti, podobnými či shodnými názvy. Přehled jednotlivých atributů a jejich vlastností lze nalézt v tabulce č. 3.

Název sloupce	Datový typ	PK	NN	AI	Default
<b>idAkce</b>	INT	•	•	•	
<b>predmet</b>	VARCHAR(11)		•		
<b>osoba</b>	INT		•		
<b>budova</b>	ENUM				NULL
<b>mistnost</b>	SMALLINT				NULL
<b>planObsazeni</b>	SMALLINT		•		
<b>obsazeni</b>	SMALLINT		•		
<b>typAkceZkr</b>	ENUM		•		
<b>platnost</b>	ENUM		•		
<b>denZkr</b>	ENUM				NULL
<b>hodinaOd</b>	ENUM				NULL
<b>hodinaDo</b>	ENUM				NULL
<b>tydenOd</b>	TINYINT				NULL
<b>tydenDo</b>	TINYINT				NULL
<b>tydenZkr</b>	ENUM				NULL
<b>kontakt</b>	TEXT				NULL

Tabulka 3 - Tabulka ROZVRHOVÁ AKCE

<b>idAkce</b>	Atribut je primárním klíčem této entity. Je nastaven na datový typ INTEGER a nesmí nabýt hodnoty NULL. Atribut je nastaven na AUTO_INCREMENT.
<b>predmet</b>	Tento atribut je cizím klíčem, jenž odkazuje na atribut „idZkratka“ entity „PREDMET“. Jednoznačně identifikuje předmět, kterého se daná rozvrhová akce týká. Referenční integrita je nastavena na ON DELETE RESTRICT a ON UPDATE CASCADE.
<b>osoba</b>	Atribut je cizím klíčem, který odkazuje na atribut „idOsoba“ entity „OSOBA“. Jednoznačně identifikuje zaměstnance, který danou rozvrhovou akci vyučuje. Referenční integrita je nastavena na ON DELETE RESTRICT a ON UPDATE CASCADE.
<b>budova</b>	Atribut informuje uživatele o tom, v jaké budově je akce vyučována. Je nastaven na datový typ ENUM. Ve výčtu položek lze vybrat jednu z nabízených možností. Ve výběru jsou využity zkratky získané prostřednictvím zdroje [21]. Tento atribut je defaultně nastaven na hodnotu NULL.
<b>planObsazeni</b>	Tento atribut obsahuje číslo maximálního počtu studentů, pro které je rozvrhová akce naplánována. Je nastaven na datový typ SMALLINTEGER a nesmí nabýt hodnoty NULL.
<b>obsazeni</b>	Atribut obsahuje počet studentů, kteří jsou na danou rozvrhovou akci zapsaní. Je opět nastaven na datový typ SMALLINTEGER a nesmí nabýt hodnoty NULL.
<b>typAkceZkr</b>	Atribut informuje uživatele, o jaký typ rozvrhové akce se jedná. Je nastaven na datový typ ENUM a nesmí nabýt hodnoty NULL. Ve výčtu položek lze vybrat jednu z nabízených možností: <ul style="list-style-type: none"> <li>• PŘ - přednáška</li> <li>• CV - cvičení</li> <li>• SE - seminář</li> </ul>
<b>platnost</b>	Atribut určuje, zda je rozvrhová akce v danou chvíli platná. Je nastaven na datový typ ENUM a nesmí nabýt hodnoty NULL. Ve výčtu položek lze vybrat jednu z možností: <ul style="list-style-type: none"> <li>• A - akce je platná a aktuální, studenti se na tuto akci mohli přihlásit</li> <li>• B - akce je blokována, na tuto akci není povoleno přihlášení</li> </ul>

- N - akce není platná v době předzápisu, avšak je viditelná pro studenty
- P - akce je v přípravě
- Z - akce je uzavřená

**denZkr** Tento atribut informuje o dni, kdy výuka rozvrhové akce probíhá. Je opět nastaven na datový typ ENUM, nesmí nabýt hodnoty NULL a ve výčtu položek lze vybrat jednu z možností:

- Po - pondělí
- Út - úterý
- St - středa
- Čt - čtvrtek
- Pá - pátek
- So - sobota

**hodinaOd, hodinaDo** Atributy určují, ve kterých vyučovacích hodinách probíhá výuka akce. Jsou nastaveny na datový typ ENUM a ve výčtu možností lze vybrat jednu položku z čísel 1 - 14. Jsou defaultně nastaveny na hodnotu NULL.

**tydenOd, tydenDo** Atributy informují, v jakých týdnech se výuka odehrává. Jsou nastaveny na datový typ TINYINTEGER a defaultní hodnotu NULL.

**tydenZkr** Atribut obsahuje informace o týdnu konání rozvrhové akce. Je defaultně nastaven na hodnotu NULL a datový typ ENUM. Ve výčtu položek lze vybrat jednu z možností:

- K - každý týden
- J - jiný týden
- L - lichý týden
- S - sudý týden

**kontakt** Tento atribut je nastaven na datový typ TEXT z důvodu jeho možné délky. Obsahuje data, která dalšími informacemi obohacují rozvrhovou akci. Položky tohoto atributu nemusí obsahovat data.

## 5.1.4 PRACOVNÍSTĚ

Tato entita se skládá ze dvou atributů. Jednotlivé atributy jsou vybírány z dat stažených z IS/STAG a pojmenovány, v rámci přehlednosti, podobnými či shodnými názvy. Přehled jednotlivých atributů a jejich vlastností lze nalézt v tabulce č. 4.

Název sloupce	Datový typ	PK	NN	AI	Default
nazev	VARCHAR(45)		•		
pracovisteZkr	VARCHAR(11)	•	•		

Tabulka 4 - Tabulka PRACOVISTĚ

**nazev** Atribut obsahuje celý název daného pracoviště. Je nastaven na datový typ VARCHAR a nesmí nabýt hodnoty NULL. Délka je po prozkoumání získaných dat odhadnuta na maximálně 45 znaků.

**pracovisteZkr** Atribut je primárním klíčem této entity. Je nastaven na datový typ VARCHAR a nesmí nabýt hodnoty NULL. Je tvořen ze zkratek jednotlivých pracovišť. Toto označení je v rámci školy jednoznačné. Délka je po prozkoumání získaných dat odhadnuta na maximálně 11 znaků.

### 5.1.5 UŽIVATEL

Tato entita neslouží jako sběrnice dat pro správu úvazků nýbrž jako nástroj pro správu přístupu do aplikace. Entita „UZIVATEL“ se skládá z pěti atributů. Atributy jsou popsány v tabulce č. 5.

Název sloupce	Datový typ	PK	NN	AI	Default
id	INT	•	•	•	
jmeno	VARCHAR(20)		•		
prijmeni	VARCHAR(25)		•		
login	VARCHAR(10)		•		
heslo	VARCHAR(40)		•		

Tabulka 5 - Tabulka UŽIVATEL

**id** Atribut „id“ je primárním klíčem entity. Je nastaven na datový typ INTEGER a nesmí nabýt hodnoty NULL. Atribut je nastaven na AUTO\_INCREMENT.

**jmeno** Atribut obsahuje křestní jména osob, které mají přístup k aplikaci. Je nastaven na datový typ VARCHAR a nesmí nabýt hodnoty NULL. Délka je odhadnuta na 20 znaků.

**prijmeni** Atribut obsahuje příjmení osob, které mají přístup k aplikaci. Je nastaven na datový typ VARCHAR a také nesmí nabýt hodnoty NULL. Délka je odhadnuta na 25 znaků.

<b>login</b>	Atribut „login“ slouží jako přihlašovací jméno při identifikaci uživatele. Je nastaven na datový typ VARCHAR a nesmí nabýt hodnoty NULL. Maximální počet znaků v loginu je 10.
<b>heslo</b>	Atribut „heslo“ slouží jako přihlašovací heslo při ověření identity uživatele. Je nastaven na datový typ VARCHAR, nesmí nabýt hodnoty NULL a maximální počet znaků je 40.

## 5.2 Vztahy mezi entitami

Mezi entitami existují 4 vazby. Dvě z těchto vazeb jsou typu 1:N. Vazba mezi tabulkami „OSOBA“ a „PREDMET“ je typu M:N. U této vazby je nutné provést dekompozici vztahu pomocí vazební tabulky „ROZVRHOVA\_AKCE“, která obsahuje odkazy obou entit. Tím vznikají vazby 1:N mezi entitami „ROZVRHOVA\_AKCE“ - „PREDMET“ a „ROZVRHOVA\_AKCE“ - „OSOBA“.

### **OSOBA - ROZVRHOVA\_AKCE – 1:N**

Každá osoba, ať už učitel, doktorand či externista může vyučovat na více rozvrhových akcích, na cvičeních, přednáškách či seminářích. Avšak daná rozvrhová akce může být v konkrétní čas a den vyučována pouze jednou osobou.

### **PREDMET - ROZVRHOVA\_AKCE - 1:N**

Každý předmět může mít několik rozvrhových akcí, tzn. že může mít několik cvičení, přednášek či seminářů. Avšak konkrétní rozvrhová akce spadá pouze pod jeden předmět.

### **PRACOVISTE - PREDMET – 1:N**

Každé pracoviště může mít několik předmětů, které pod něj spadají. Konkrétní předmět však může patřit pouze pod jedno pracoviště.

### **PRACOVISTE – OSOBA – 1:N**

V této databázi pracujeme s informací, že zaměstnaná osoba pracuje pouze pod jedním pracovištěm. Osoba může pracovat pod více katedrami, avšak jedna musí být hlavní. Pracoviště může zaměstnávat mnoho osob.

### **5.3 Omezení pravomocí uživatelů**

Databáze MySQL umožňuje uživatelům přiřadit pravomoci pomocí příkazu GRANT. V rámci vyvíjené aplikace není potřeba řešit problematiku přístupu více uživatelů. Aplikace je vyvíjena pro potřeby zadavatele a je využívána pouze lokálně.

### **5.4 Webové služby nad IS/STAG**

Data, která jsou do databáze vkládána, nejsou nově tvořena, nýbrž stažena pomocí dostupných webových služeb nad IS/STAG. Postup stažení a následná úprava jsou uvedeny v uživatelské dokumentaci, kterou lze nalézt v příloze B.

IS/STAG je celouniverzitním systémem určeným, jak pro evidenci, tak i pro správu studijní činnosti. Je vyvíjený Centrem informatizace a výpočetní techniky Západočeské univerzity v Plzni. [14]

Tento systém je využit pro stažení základních informací a dat ve formátu CSV. Formát CSV je zvolen z důvodu snadné úpravy, zpracování a podpory nástrojem phpMyAdmin, který zprostředkovává import dat.

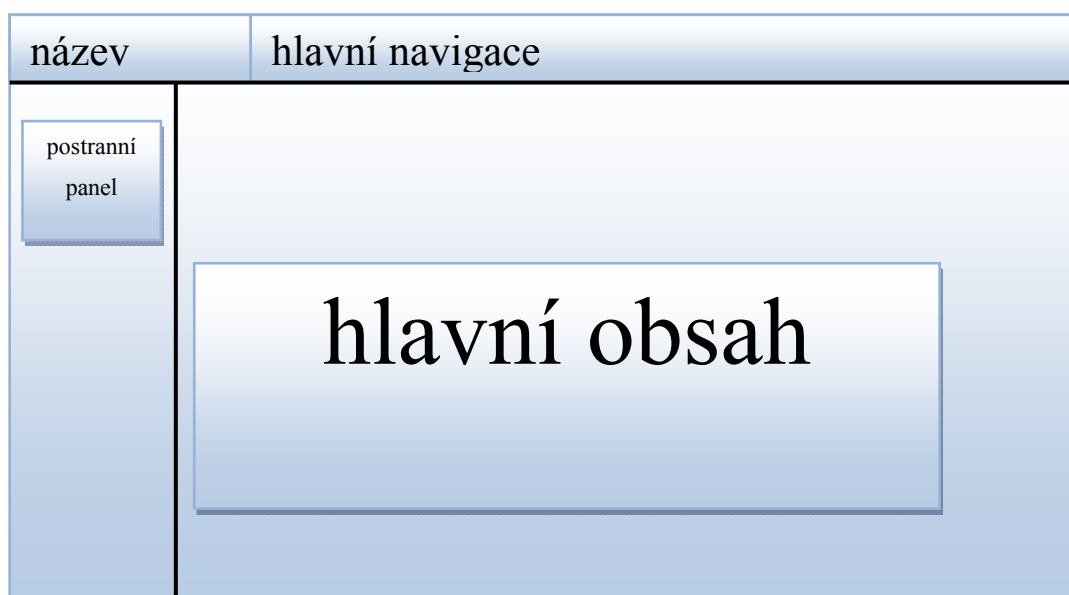


## 6 Tvorba aplikace

Tato kapitola obsahuje návrh aplikace a popis jejích nejvýznamnějších částí. Z důvodu přehlednosti je vytvořena adresářová struktura systému pro ukládání skriptů. Tuto hierarchii lze nalézt v příloze A. V závěru této kapitoly jsou vysvětleny funkce jednotlivých skriptů.

### 6.1 Návrh systému

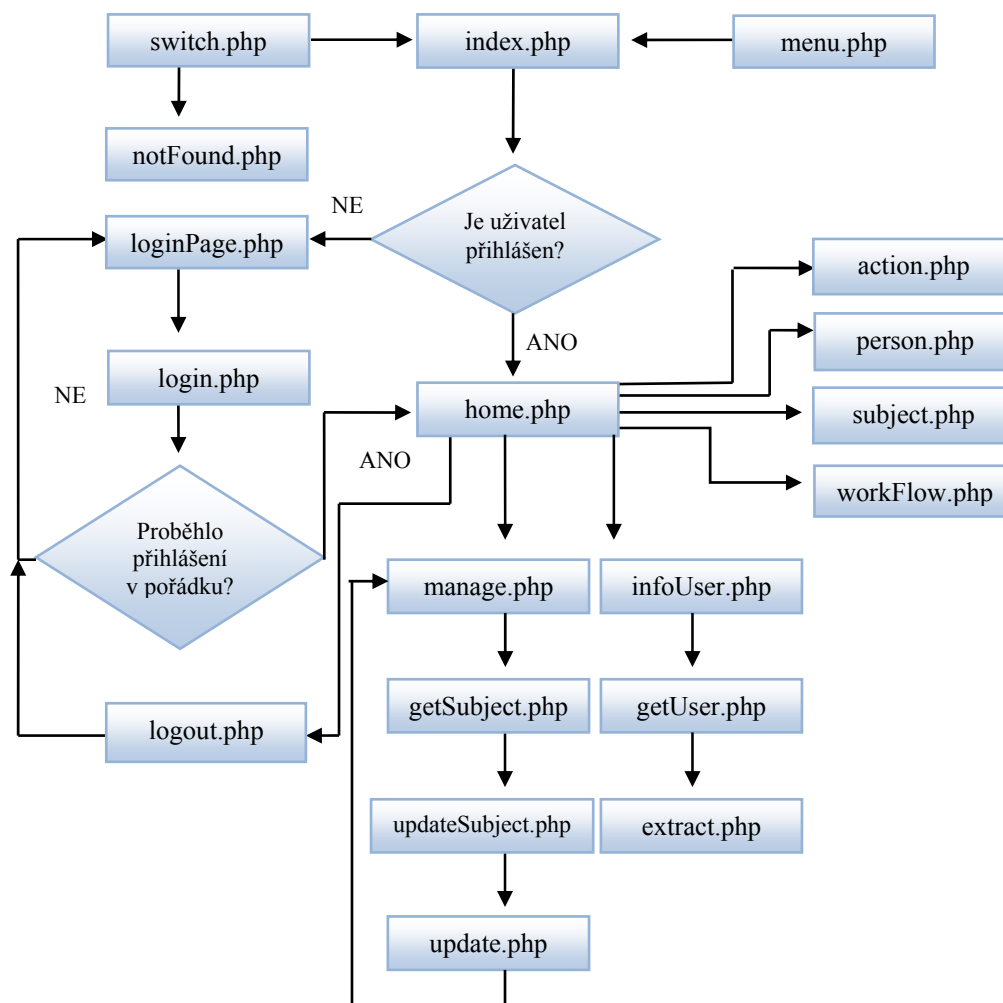
Návrh se skládá ze tří částí, viz obrázek č. 8. První část tvoří záhlaví systému. Tato část obsahuje název, ale především hlavní navigaci. Navigace slouží jako nástroj, jehož prostřednictvím uživatel získá informace o úvazcích pedagogů a vykoná potřebné změny. Další částí je postranní panel. Tento panel nabízí funkci, která uživateli umožní tisk aktivní stránky. Dále zprostředkovává přesměrování na správcovskou aplikaci phpMyAdmin. Poslední částí je hlavní obsah. Ten je v rámci navigace proměnlivý.



Obrázek 8 - Návrh systému

### 6.2 Vývojový diagram

Pro snazší pochopení funkce jednotlivých skriptů je vytvořen jednoduchý vývojový diagram, viz obrázek č. 9. Podrobný popis funkcí jednotlivých skriptů lze nalézt v kapitole 6. 7.



Obrázek 9 - Vývojový diagram

### 6.3 Připojení k databázi

Pro připojení k databázi je využíván centrální soubor connect.php. Toto řešení je praktické, protože v případě změny parametrů připojení postačí změnit pouze jeden soubor. Pro připojení k databázi je využita funkce mysql\_connect(). Kvůli zachycení případných chyb je kód rozšířen o chybný výpis na obrazovce. Pro výběr databáze je využívána funkce mysql\_select\_db().

### 6.4 Komunikace s databází

Databázový server komunikuje pouze prostřednictvím SQL jazyka. V rámci aplikace je využito funkce mysql\_query(), která slouží jako prostředek dotazování. Aplikace využívá dvou druhů dotazů DQL (SELECT) a DML (UPDATE). I tyto dotazy jsou v rámci zachycení případných chyb rozšířeny o chybný výpis na obrazovce.

## 6.5 Přihlášení k aplikaci a odhlášení

Pro zajištění bezpečnosti je důležitá identifikace a autentizace. Identifikace formuluje identitu uživatele prostřednictvím přihlašovacího jména. Autentizace slouží k ověření identity uživatele. V této práci je využito nejjednoduššího druhu autentizace, a to ověření prostřednictvím hesla. Hlavním úkolem těchto metod je omezit přístup do systému. Seznam uživatelů, jejich přihlašovacích jmen a hesel lze nalézt v tabulce „UZIVATEL“. V rámci ochrany hesel je využito jednosměrné šifrování. Řetězec, který je vložen do přihlašovacího formuláře, se přetvoří na řetězec znaků na základě neznámého klíče. Není proto možné v reálném čase heslo rozšifrovat. V aplikaci je využita funkce sha1().

Přihlašovací formulář na stránce loginPage.php směruje přihlašovací údaje do skriptu login.php. Zde jsou data šifrována a porovnávána s databází. Pro přenos dat je implementováno SESSION. Ve skriptu index.php je pomocí funkce session\_start() vytvořeno „sezení“, které umožňuje přistupovat k zadaným informacím. V případě shodných dat je uživatel přesměrován na hlavní stranu aplikace home.php. V případě odlišných dat je uživateli oznámeno, že přihlášení neproběhlo v pořádku a poté je přesměrován na stranu s přihlašovacím formulářem.

Odhlášení probíhá prostřednictvím skriptu logout.php. Tento skript má jednoduchou úlohu, prostřednictvím funkce session\_destroy() ukončí veškerá „sezení“ a uživatele přesměruje na přihlašovací stránku.

## 6.6 Mapa aplikace

Mapa umožňuje uživateli vidět celou strukturu aplikace. Hierarchii zprostředkovává skript menu.php, jehož prostřednictvím struktura vypadá následovně:

### Aplikace pro správu úvazků

- Domů
- Správa
- Info
- Prohlížení
  - Osoba
  - Rozvrhová akce
  - Předmět
  - Pracoviště
- Odhlásit

Skript menu.php vytváří hlavní navigaci a postranní panel. Navigace má podobu horizontálního pruhu a je umístěna v hlavičce.

## 6.7 Funkce skriptů

Tato kapitola popisuje funkci skriptů vyvíjené aplikace. Uspořádání se odvíjí od hierarchie skriptů, viz příloha A.

### kořenový adresář

<b>loginPage.php</b>	Skript vytváří formulář pro přihlášení a směřuje data do skriptu login.php.
<b>login.php</b>	Skript porovnává data získaná z přihlašovacího formuláře spolu s daty z databáze. V případě shodných dat přesměruje uživatele na domovskou stránku aplikace. V opačném případě odešle uživatele zpět k přihlašovacímu formuláři.
<b>logout.php</b>	Skript zprostředkovává odhlášení a ukončuje SESSION. Ukončení přihlášení lze také uskutečnit zavřením webového prohlížeče.

### includes adresář

<b>action.php</b>	Skript zprostředkovává zobrazení rozvrhových akcí seřazených dle identifikačního čísla. Zobrazuje všechna data, která jsou v rámci entity „ROZVRHOVA_AKCE“ uložena v databázi. Je určen pouze pro prohlížení.
<b>connect.php</b>	Skript je prostředkem pro navázání spojení s databází. Využívají jej všechny skripty adresáře „includes“.
<b>extract.php</b>	Skript slouží jako prostředek extrakce dat týkajících se pedagoga a rozvrhových akcí, jež vyučuje. Data ukládá do formátu CSV.
<b>getSubject.php</b>	Skript získává data z manage.php. Ta se stávají parametrem pro výběr konkrétní rozvrhové akce. Výběr probíhá prostřednictvím select boxu <sup>3</sup> .
<b>getUser.php</b>	Skript poskytuje nabídku pedagogů. Na základě výběru učitele technologie AJAX přenáší zvolená data a uživatel je přesměrován na getUser.php.

---

<sup>3</sup> Výběrová nabídka.

<b>home.php</b>	Skript zobrazuje domovskou stránku aplikace.
<b>index.php</b>	Skript vytváří SESSION, základní strukturu systému a pomocí funkce include() vkládá a vykonává soubory switch.php a menu.php.
<b>infoUser.php</b>	Skript zprostředkovává přehled všech rozvrhových akcí vybraného pedagoga. Prostřednictvím atributů „hodinaOd“ a „hodinaDo“ počítá, zda byl úvazek překročen. Pro výpočet je využit vzorec č.1.  $\text{výpočet} = [(\text{„hodinaDo“} - \text{„hodinaOd“}) + 1] \quad (1)$
<b>manage.php</b>	Skript poskytuje nabídku předmětů. Na základě tohoto výběru technologie AJAX přenáší zvolená data a uživatele směřuje na getSubject.php.
<b>menu.php</b>	Skript tvoří mapu aplikace. Vytváří hlavní navigaci a postranní panel pro lepší orientaci uživatele.
<b>notFound.php</b>	Skript se zobrazí v případě, že uživatel vyhledá stránku, která neexistuje nebo ji nelze nalézt.
<b>person.php</b>	Skript zprostředkovává zobrazení všech pedagogů seřazených dle příjmení. Zobrazuje všechna data, která jsou v rámci entity „OSOBA“ uložena v databázi. Je určen pouze pro prohlížení.
<b>subjekt.php</b>	Skript zprostředkovává zobrazení předmětů seřazených dle zkratk. Zobrazuje všechna data, která jsou v rámci entity „PREDMET“ uložena v databázi. Je určen pouze pro prohlížení.
<b>switch.php</b>	Skript slouží jako prostředek směrování.
<b>update.php</b>	Skript mění stav databáze. Po ukončení změn uživatele přesměruje zpět na manage.php.
<b>updateSubject.php</b>	Skript získává data z getSubject.php, která se stávají parametrem pro změnu pedagoga či platnosti rozvrhové akce. Potvrzení změn zprostředkovává potvrzovací tlačítko.

<b>workspace.php</b>	Skript zprostředkovává zobrazení pracovišť seřazených dle zkratk. Zobrazuje všechna data, která jsou v rámci entity „PRACOVISTE“ uložena v databázi. Je určen pouze pro prohlížení.
<b><u>css adresář</u></b>	Obsahuje css soubory Bootstrap Frameworku a soubory tvořené pro potřebu úpravy grafiky vyvíjené aplikace.
<b><u>design adresář</u></b>	Obsahuje obrázkové soubory, které jsou vytvořeny v rámci vylepšení designu systému.
<b><u>fonts adresář</u></b>	Obsahuje komponenty Bootstrap Frameworku.
<b><u>js adresář</u></b>	Obsahuje komponenty JavaScriptu Bootstrap Frameworku.

## 6.8 Systémové požadavky

Před nasazením systému je důležité zkontrolovat, zda počítač splňuje minimální systémové požadavky na provoz. Pokud tyto požadavky nejsou splněny, mohou nastat situace, kdy aplikace nevykoná požadované úkoly.

Aplikace pro správný běh vyžaduje operační systém Windows XP či vyšší a internetový prohlížeč. Je optimalizována pro prohlížeče Google Chrome verze 42, Firefox verze 35 a Internet Explorer 11. Prohlížeč musí mít zapnutý JavaScript a podporu cookies<sup>4</sup>.

---

<sup>4</sup> Soubory vytvářené webovým serverem a ukládané u klienta prostřednictvím prohlížeče.

# 7 Testování aplikace

Velmi podstatnou částí vývoje systému je ověření jeho funkčnosti. Softwarový produkt musí splňovat nejen požadavky, které byly na začátku vývoje vymezeny, ale zároveň musí být před nasazením odstraněny nežádoucí projevy a anomálie.

Pro kontrolu kvality webových aplikací existují speciální testy. Druhy těchto testů se odvíjí od stavu vývoje, ve kterém se aplikace v danou chvíli nachází. Mezi tyto druhy patří testování programátorem, testování funkční, testování použitelnosti, testování serverového rozhraní, testování kompatibility a testování akceptační. Jejich cílem je nalezení a odstranění nežádoucích projevů. Tyto testy jsou popsány níže. [15]

## 7.1 Druhy testování

### Testování programátorem

V této fázi testů je program kontrolován na úrovni zdrojového kódu. Kontrola je prováděna programátory osobně a zaměřuje se na funkčnost skriptů. Tím si vývojář ověřuje kvalitu své práce. Využívá se tzv. jednotkových testů, které prostřednictvím velmi malých částí kódů (jednotek) podrobně testují kvalitu. Cílem těchto testů je ověření funkčnosti prostřednictvím izolování testovaného kódu od ostatních celků. Mezi nevýhody tohoto testování patří neschopnost detekce rozsáhlejších chyb. [15]

### Funkční testování

Prostřednictvím funkčních testů se zjišťuje, zda aplikace reaguje dle očekávání, tedy zda uživateli vrací adekvátní výstupní informace. V této části vývojář kontroluje správnost chodu systému v pozici uživatele. V rámci tohoto testování probíhá kontrola správného směřování a funkčnosti formulářových prvků. [15]

### Testování použitelnosti

Fáze testování použitelnosti zahrnuje testy prováděné uživateli. Tento druh testů se významným vlivem podílí na snížení pravděpodobnosti výskytu poruch. Cílem je objevit překážky, které zabraňují efektivnímu využití aplikace. Testujícím subjektům jsou zadány úkoly, které aplikace musí bezchybně vykonat pro splnění požadavků

zadavatele. Toto testování zprostředkovává přímý pohled na interakci uživatele s aplikací a pomáhá odhalit nedostatky především vizuální. Například špatnou viditelnost prvků, nevhodnou stylistiku navigace či nelogické navržení aplikace. [15]

## **Testování serverového rozhraní**

Testování serverového rozhraní se zaměřuje na kontrolu komunikace serverově - aplikačního rozhraní a dále databázově - aplikačního rozhraní. [15]

## **Testování kompatibility**

V této fázi je nutné ověřit kompatibilitu aplikace s technologiemi, které jsou využity. Testování probíhá pomocí různých webových prohlížečů a na různých operačních systémech. Je nanejvýš vhodné zvolit prohlížeče a operační systémy, které uživatelé běžně používají. Není třeba testovat kompatibilitu na technologii, kterou uživatel nezná a nepoužívá. [15]

## **Akceptační testování**

Na akceptačním testování se podílí přímo zadavatelé. Jejich cílem je porovnat požadavky, které byly vytvořeny na počátku vývoje s konečným stavem softwaru. Tato závěrečná etapa testování určuje, zda je stav systému akceptovatelný. [15]

## **7.2 Výsledky testování**

Vyvíjený kód byl kontrolován a testován v průběhu samotného psaní. Pomocí jednotkového testování bylo z kódu odstraněno mnoho nefunkčních celků, jež mohly po nasazení systému způsobit potíže. Kontrolovány byly využívané funkce a tvořené proměnné.

V rámci funkčního testování bylo zkontrolováno chování ovládacích prvků, které prokázaly svoji funkčnost. Proběhla také kontrola cookies. Pro správný chod aplikace je nutné, aby webový prohlížeč povolil serverům nastavit cookies. V opačném případě aplikace nezprostředkuje uživateli přihlášení. Dále byl pomocí W3C validátoru optimalizován statický kód. Validátor odhalil chybu ve vstupních datech, jež je popsána v Uživatelské dokumentaci B. V neposlední řadě byla zkontrolována funkčnost JavaScriptu a AJAXu. Tato fáze testování neodhalila žádné další chyby.



Vyvíjená aplikace byla v rámci testování použitelnosti předána testujícím subjektům. Výsledky a zhodnocení stavu systému lze nalézt v příloze C. Toto testování se prokázalo jako velice hodnotné. Upozornilo na grafické nedostatky, které způsobily dezorientovanost uživatele. V rámci zlepšení grafiky aplikace byla s testujícími subjekty konzultována možnost změny. Barvy navigace byly následně změněny na výraznější, konkrétně z šedé na černou a žlutou. Tlačítka pro potvrzení i extrakci dat byla zvětšena a zvýrazněna modrou barvou.

Testování serverového rozhraní kontrolovalo reakci na chyby způsobené restartem serveru. Restart serveru způsobil, že aplikace neobdržela data. Chyba byla uživateli oznámena chybným výpisem na domovské stránce aplikace.

V rámci testování kompatibility byly vybrány 3 webové prohlížeče, na kterých se s pozitivním výsledkem prokázala funkčnost aplikace. Testování na různých operačních systémech je v tuto chvíli bezpředmětné. Aplikace bude využívána na operačním systému Windows, proto v rámci testů proběhla kontrola kompatibility pouze s tímto systémem. V průběhu testování kompatibility s webovými prohlížeči bylo využito:

- Google Chrome verze 42
- Firefox 35
- Internet Explorer 11

Testování se zadavatelem proběhlo také v pořádku. Zadavateli byla předvedena finální verze aplikace a vysvětlena funkčnost jednotlivých prvků včetně objasnění jejich organizace. Poté proběhla názorná ukázka ovládání. Zadavatel vyzkoušel funkce, které aplikace nabízí a zkontroloval, že byly splněny stanovené cíle.

## 8 Závěr

V rámci této práce byla prozkoumána agenda personálního zajištění výuky na Katedře informatiky a výpočetní techniky Západočeské univerzity v Plzni. Prostřednictvím získaných poznatků byl navržen systém, který usnadňuje proces interní správy pedagogických úvazků.

Kapitola použitých technologií shrnula informace o nástrojích, které byly v této práci využity. První volbou byl databázový systém MySQL. Tento výběr ovlivnil další využití technologie, s jejichž pomocí byl vytvořen datový model a webová aplikace. Spolu s tvorbou modelu bylo třeba prozkoumat způsob získání dat. Jako prostředek byly zvoleny webové služby, které umožnily stažení dat z IS/STAG. Tato data byla uložena v určité struktuře, proto v rámci co nejlepší přehlednosti, byly názvy atributů upraveny do shodného či podobného tvaru. Pro vývoj programového řešení byl vybrán jazyk PHP, který umožnil tvorbu přehledné a uživatelsky přívětivé webové aplikace. Tvorba se odvíjela od stanovených cílů, jež byly postupně naplňovány a testovány. Kontrola kvality proběhla v závěru vývoje. Testy poukázaly na grafické nedostatky, které byly odstraněny.

V rámci požadavků byly splněny všechny body. Aplikace obsahuje informace, které jsou nezbytné ke správě pedagogických úvazků. Dále umožňuje měnit potřebné vlastnosti rozvrhových akcí a zprostředkovává prohlížení všech entit s výjimkou entity určené pro správu přístupu. Závěrem uživateli nabízí přehled úvazků pedagogů, a tím umožňuje jejich kontrolu. Tato data je možno extrahovat do formátu CSV, který umožňuje další manipulaci. Systém pro správu pedagogických úvazků funguje dle zadaných předpokladů a je připraven k nasazení.

Tato práce mi umožnila získat řadu cenných zkušeností. Poskytla mi přehled technologií, s jejichž pomocí se vytvářejí webové aplikace založené na databázi. Umožnila získání zkušeností pro mne s dosud neznámým programovacím jazykem a zprostředkovala mi praktickou zkušenost v oblasti vývoje softwarového produktu.

# Přehled zkratek

<b>Zkratka</b>		<b>Význam</b>
AJAX	<i>Asynchronous JavaScript and XML</i>	technologie pro vývoj interaktivních webových aplikací, aktualizuje obsah webových stránek bez nutnosti znovu načítání celé stránky
CSS	<i>Cascading Style Sheets</i>	jazyk pro popis vzhledu a formátování dokumentu psaného ve značkovacím jazyce
CSV	<i>Comma-separated values</i>	souborový formát oddělující hodnoty středníky
DCL	<i>Data Control Language</i>	jazyk pro řízení dat, umožňuje nastavování přístupových práv
DDL	<i>Data Definition Language</i>	jazyk pro definici dat, umožňuje tvorbu struktury databáze
DML	<i>Data Manipulation Language</i>	jazyk pro manipulaci s daty, umožňuje úpravu dat
DQL	<i>Data Query Language</i>	jazyk zprostředkovávající dotazování
FTP	<i>File Transfer Protocol</i>	protokol pro přenos souborů mezi počítači pomocí počítačové sítě
HTML	<i>HyperText Markup Language</i>	značkovací jazyk pro tvorbu webových stránek
HTTP	<i>Hypertext Transfer Protocol</i>	internetový protokol určený pro výměnu hypertextových dokumentů

IDE	<i>Integrated Development Environment</i>	vývojové prostředí usnadňující tvorbu zdrojového kódu
OSI	<i>Open Source Initiatives</i>	organizace podporující software Open Source
PDF	<i>Portable Document Format</i>	souborový formát pro ukládání dokumentů vytvořený firmou Adobe Systems
PHP	<i>Hypertext Preprocessor</i>	skriptovací jazyk pro tvorbu dynamických webových stránek a webových aplikací
POP3	<i>Post Office Protocol</i>	protokol, používaný ke stahování emailových zpráv ze vzdáleného serveru na klienta
SHA1	<i>Secure Hash Algorithm</i>	kryptografická hashovací funkce
SQL	<i>Structure Query Language</i>	jazyk zprostředkávající komunikaci s databázemi
SŘBD	<i>Systém řízení báze dat</i>	softwarové vybavení pro práci s databázemi
W3C	<i>World Wide Web Consortium</i>	mezinárodní standardizační organizace, kontrolor validity webových stránek
XML	<i>Extensible Markup Language</i>	značkovací jazyk definující pravidla kódování dokumentů, který je vytvořený W3C konsorciem

## Seznam obrázků

Obrázek 1 - Entita .....	3
Obrázek 2 - Entita s atributy .....	3
Obrázek 3 - Vztah typu 1:1 .....	4
Obrázek 4 - Vztah typu 1:N .....	4
Obrázek 5 - Vztah typu M:N .....	5
Obrázek 6 - Diagram případu užití .....	11
Obrázek 7 - Datový model .....	17
Obrázek 8 - Návrh systému .....	27
Obrázek 9 - Vývojový diagram .....	28
Obrázek 10 - Vstupní parametry tabulky PRACOVÍŠTE .....	47
Obrázek 11 - Vstupní parametry tabulky OSOBA .....	48
Obrázek 12 - Vstupní parametry tabulky PŘEDMĚT .....	49
Obrázek 13 - Vstupní parametry tabulky ROZVRHOVÁ AKCE .....	51
Obrázek 14 - Import dat do databáze .....	53
Obrázek 15 - Přihlašovací formulář .....	54
Obrázek 16 - Navigace a postranní panel .....	55
Obrázek 17 - Výběr předmětů a rozvrhových akcí .....	55
Obrázek 18 - Změna pedagoga a platnosti rozvrhové akce .....	55
Obrázek 19 - Informace o úvazku pedagogů .....	56

# Seznam tabulek

Tabulka 1 - Tabulka OSOBA .....	18
Tabulka 2 - Tabulka PŘEDMĚT .....	20
Tabulka 3 - Tabulka ROZVRHOVÁ AKCE .....	21
Tabulka 4 - Tabulka PRACOVIŠTĚ .....	24
Tabulka 5 - Tabulka UŽIVATEL .....	24

# Literatura

## Knižní zdroje

- [1] HERNANDEZ, Michael J. *Návrh databází*. 1. vyd. Praha: Grada, 2006, 408 s. ISBN 80-247-0900-7.
- [2] KLEČKOVÁ, Jana. ZÁPADOČESKÁ UNIVERZITA V PLZNI. *Základy informačních systémů*. Plzeň: Tiskové středisko ZČU, 2002. ISBN 80-7092-931-1.
- [3] LACKO, Ľuboslav. *PHP a MySQL: hotová řešení*. Vyd. 2. Překlad Bogdan Kiszka. Brno: Computer Press, 2006, 299 s. K okamžitému použití (Computer Press). ISBN 80-251-1249-7.
- [4] MASLAKOWSKI, Mark. *Naučte se MySQL za 21 dní: hotová řešení*. Vyd. 1. Překlad Bogdan Kiszka. Praha: Computer Press, 2001, xxviii, 478 s. Databáze. ISBN 80-722-6448-6.
- [5] OPPEL, Andrew. *Databáze bez předchozích znalostí*. Vyd. 1. Brno: Computer Press, 2006, 319 s. ISBN 80-251-1199-7.
- [6] POKORNÝ, Jaroslav a Ivan HALAŠKA. *Databázové systémy: Vybrané kapitoly a cvičení*. 1. vyd. Praha: ČVUT v Praze, 1993, 189 s. ISBN 80-706-6814-8.
- [7] PONKRÁC, Miloslav. *PHP a MySQL: bez předchozích znalostí : [průvodce pro samouky]*. Vyd. 1. Brno: Computer Press, 2007, 221 s. ISBN 978-80-251-1758-3.

## Online zdroje

- [8] 6. díl - MySQL krok za krokem: Datové typy a NULL. *ITnetwork: Sociální síť pro IT profesionály* [online]. © 2015 [cit. 2015-03-07]. Dostupné z: <http://www.itnetwork.cz/mysql-tutorial-datove-typy-a-null>
- [9] About. *PhpMyAdmin: Bringing MySQL to the web* [online]. © 2003 - 2015 [cit. 2015-03-09]. Dostupné z: [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)

- [10] AJAX. *Adaptic: tvorba webu, webdesign* [online]. © 2005–2015 [cit. 2015-03-28]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/ajax/>
- [11] About: What is the Apache HTTP Server Project. *Apache: HTTP SERVER PROJECT* [online]. © 1997-2014 [cit. 2014-11-06]. Dostupné z: [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html)
- [12] Bootstrap Tutorial. *Tutorialspoint* [online]. © 2015 [cit. 2015-04-18]. Dostupné z: <http://www.tutorialspoint.com/bootstrap/>
- [13] Databázové systémy. *Fei.mtrakal: Materiály ke stažení* [online]. 2008, 1 - 47 [cit. 2015-03-07]. Dostupné z: [http://fei.mtrakal.cz/materialy\\_public/\\_szz/Datab%e1zov%e9%20syst%e9my.pdf](http://fei.mtrakal.cz/materialy_public/_szz/Datab%e1zov%e9%20syst%e9my.pdf)
- [14] KUTKOVÁ, Veronika. *Nástroje pro analýzu dat v prostředí tabulkového procesoru* [online]. Plzeň, 2014 [cit. 2015-03-09]. Dostupné z: <https://portal.zcu.cz/StagPortletsJSR168/KvalifPraceDownloadServlet?typ=1&adipidno=59320>. Bakalářská práce. Západočeská univerzita v Plzni
- [15] MAZOCH, Břetislav. *Funkční testování webových aplikací* [online]. Brno, 2012 [cit. 2015-04-19]. Dostupné z: [http://is.muni.cz/th/325233/fi\\_b/bmazoch-bp.pdf](http://is.muni.cz/th/325233/fi_b/bmazoch-bp.pdf). Bakalářská práce. Masarykova univerzita.
- [16] MySQL Workbench. *MySQL* [online]. © 2015 [cit. 2015-03-09]. Dostupné z: <http://www.mysql.com/products/workbench/>
- [17] Oracle: *MySQL ENTERPRISE EDITION* [online]. © 2015, 5 s. [cit. 2015-03-09]. Dostupné z: <https://www.mysql.com/products/enterprise/mysql-datasheet.en.pdf>
- [18] PERNICA, Tomáš. Migrace MySQL 4.0 na 4.1 – jak na češtinu. *Tomáš Pernica – Služby IT* [online]. 2008 [cit. 2015-03-01]. Dostupné z: <http://www.tomp.eu/2008/06/29/migrace-mysql-40-na-41-jak-na-cestinu/>
- [19] PHP (1) - Historie a budoucnost. *Linuxsoft* [online]. (2003-2014) [cit. 2014-11-06]. Dostupné z: [http://www.linuxsoft.cz/article.php?id\\_article=171](http://www.linuxsoft.cz/article.php?id_article=171)
- [20] What is Eclipse. *Tutorialspoint* [online]. © 2015 [cit. 2015-04-18]. Dostupné z: [http://www.tutorialspoint.com/eclipse/eclipse\\_overview.htm](http://www.tutorialspoint.com/eclipse/eclipse_overview.htm)
- [21] SEZNAM BUDOV ZČU. *Fakulta Filozofická Západočeské univerzity v Plzni* [online]. © 1991 - 2015 [cit. 2015-03-08]. Dostupné z: <http://www.ff.zcu.cz/files/2012/seznam-budov.pdf>



- [22] Úvod do MySQL. *Garth* [online]. 2012 [cit. 2014-11-06]. Dostupné z: <http://www.garth.cz/uvod-do-mysql/>
- [23] Wamp - WampServer: Overview. *Sourceforge* [online]. © 2014 [cit. 2014-11-08]. Dostupné z: <http://sourceforge.net/projects/wampserver/>
- [24] What are HTML, CSS, JavaScript, PHP and Perl. *TheSitewizard.com* [online]. © 2008-2014 [cit. 2015-04-17]. Dostupné z: <http://www.thesitewizard.com/html-tutorial/what-is-html.shtml>
- [25] ZÁPADOČESKÁ UNIVERZITA V PLZNI. *Databázové systémy 1: KIV/DB1* [online]. 2012, 39 s. [cit. 2015-03-22]. Dostupné z: <http://www.soubor.eu/zcu/FAV-bc/KIV/DB1/datab%C3%A1ze%20-%20full.pdf>

# **Přílohy**

# A Adresářová struktura aplikace

```
/
├── login.php
├── loginpage.php
├── logout.php
├── css/
│   ├── bootstrap.css
│   ├── bootstrap-theme.css.map
│   ├── bootstrap-theme.min.css
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   └── style.css
├── design/
│   ├── notfound.png
│   └── picture.png
├── fonts/
│   ├── glyphicons-halflings-regular.eot
│   ├── glyphicons-halflings-regular.svg
│   ├── glyphicons-halflings-regular.ttf
│   ├── glyphicons-halflings-regular.woff
│   └── glyphicons-halflings-regular.woff2
├── includes/
│   ├── action.php
│   ├── connect.php
│   ├── extract.php
│   ├── getSubject.php
│   ├── getUser.php
│   ├── home.php
│   ├── index.php
│   ├── infoUser.php
│   ├── manage.php
│   ├── menu.php
│   ├── notFound.php
│   ├── person.php
│   ├── subject.php
│   ├── switch.php
│   ├── update.php
│   ├── updateSubject.php
│   └── workspace.php
```

```
|— js/  
|   |— bootstrap.js  
|   |— bootstrap.min.js  
|   |— npm.js
```

## B Uživatelská dokumentace

### Stažení dat z IS/STAG

Všechna data je nutné stáhnout ve formátu CSV a v kódování windows-1250. Ve stažených datech je potřeba provést několik úprav. Obsahují informace v databázi nepotřebné. V rámci přehlednosti byly atributy pojmenovány shodně či obdobně.

Pro úpravu dat lze využít jakýkoliv prostředek, který operační systém nabízí. Vhodným nástrojem jsou tabulkové procesory OpenOffice Calc či Microsoft Office Excel.

### Tabulka PRACOVISTĚ

Pro získání dat patřících do entity „PRACOVISTE“ je nutno nastavit vstupní parametry, viz obrázek č. 10.

Služba: Pracoviště - struktura a zaměstnanci pracovišť na ZČU  
getVsechnaPracoviste

**Stažení z IS/STAG** | **Nahrání do IS/STAG** | **Výsledek** | **Nápověda** | **Server webových služeb**

**i** Tato záložka umožňuje výběr a volání většiny webových služeb pomocí formuláře - mělo by to tedy být uživatelsky "příjemnější". Po výběru požadované služby se zobrazí její popis a případné parametry volání. Služby zde nabízené jsou služby typu "čtení z IS/STAG", tj. služby, které na základě nějakých jednoduchých vstupních parametrů vygenerují komplexní výstup.

Po stažení dat budete přepnuti na záložku 'Výsledek', kde budete moci výsledek volání služby stáhnout ve Vámi požadovaném formátu.

Služba:    
get/sechnaPracoviste

Popis služby: Vrátí seznam všech pracovišť.

Obrázek 10 - Vstupní parametry tabulky PRACOVISTĚ

Získaný dokument je třeba upravit, aby jeho tvar byl shodný s tvarem databázové tabulky. Ze souboru, který je pojmenován jako „getVsechnaPracoviste“ odebereme sloupce: „KOD\_PRAC“, „NAD\_KOD\_PRAC“ a „LEVEL“. Při odstranění celých sloupců se ostatní části posunou na místo odebraných. Jejich pořadí není vhodné měnit. Všechna pracoviště univerzity je nutno nahrát do databáze z prostého důvodu:

v případě, že pedagog vyučující předmět Katedry informatiky a výpočetní techniky má úvazek pod jiným pracovištěm, mohlo by dojít ke kolizi cizích klíčů v entitě „OSOBA“.

## Tabulka OSOBA

Pro získání dat patřících do entity „OSOBA“ je nutno nastavit vstupní parametry, viz obrázek č. 11.

Služba: Učitelé  
 getUciteleKatedry  
 Katedra: KIV - Katedra informatiky a výpočetní techniky  
 jenAktualni: true  
 zobrazitExternisty: true  
 lang: cs

Stážení z IS/STAG	Nahrání do IS/STAG	Výsledek	Nápověda	Server webových služeb								
<p><b>i</b> Tato záložka umožňuje výběr a volání většiny webových služeb pomocí formuláře - mělo by to tedy být uživatelsky "příjemnější". Po výběru požadované služby se zobrazí její popis a případné parametry volání. Služby zde nabízené jsou služby typu "čtení z IS/STAG", tj. služby, které na základě nějakých jednoduchých vstupních parametrů vygenerují komplexní výstup.</p> <p>Po stažení dat budete přepnuti na záložku 'Výsledek', kde budete moci výsledek volání služby stáhnout ve Vámi požadovaném formátu.</p>												
Služba	Učitelé <input type="text" value="getUciteleKatedry"/> <input type="button" value="Vybrat"/>											
Popis služby	Vrátí seznam učitelů zadaného pracoviště nebo podřízených pracovišť.											
Parametry služby	<table border="1"> <tr> <td><b>katedra *</b></td> <td>Pracoviště vyučujícího. Může být uvedena katedra, fakulta nebo REK - vrátí vždy vyučující všech podřízených pracovišť. Lze použít zástupné znaky pro vyhledávání (? = libovolný znak, % = libovolný počet znaků)</td> </tr> <tr> <td>jenAktualni</td> <td>Zda zobrazovat jen aktuální učitele. Není-li uvedeno, zobrazují se jen aktuální. Je-li false, zobrazují se i neaktuální.</td> </tr> <tr> <td>zobrazitExternisty</td> <td>Zda zobrazovat externisty a nebo nikoliv. Není-li uvedeno, JSOU externisti zobrazeni.</td> </tr> <tr> <td>lang</td> <td>Jazyk výstupu služby</td> </tr> </table>				<b>katedra *</b>	Pracoviště vyučujícího. Může být uvedena katedra, fakulta nebo REK - vrátí vždy vyučující všech podřízených pracovišť. Lze použít zástupné znaky pro vyhledávání (? = libovolný znak, % = libovolný počet znaků)	jenAktualni	Zda zobrazovat jen aktuální učitele. Není-li uvedeno, zobrazují se jen aktuální. Je-li false, zobrazují se i neaktuální.	zobrazitExternisty	Zda zobrazovat externisty a nebo nikoliv. Není-li uvedeno, JSOU externisti zobrazeni.	lang	Jazyk výstupu služby
<b>katedra *</b>	Pracoviště vyučujícího. Může být uvedena katedra, fakulta nebo REK - vrátí vždy vyučující všech podřízených pracovišť. Lze použít zástupné znaky pro vyhledávání (? = libovolný znak, % = libovolný počet znaků)											
jenAktualni	Zda zobrazovat jen aktuální učitele. Není-li uvedeno, zobrazují se jen aktuální. Je-li false, zobrazují se i neaktuální.											
zobrazitExternisty	Zda zobrazovat externisty a nebo nikoliv. Není-li uvedeno, JSOU externisti zobrazeni.											
lang	Jazyk výstupu služby											
	<input type="button" value="Stáhní z IS/STAG"/>											

Obrázek 11 - Vstupní parametry tabulky OSOBA

Ze souboru, který je pojmenován jako „getUciteleKatedry“ odebereme sloupce: „PLATNOST“, „ZAMESTNANEC“ a „PRACOVISTEDALSI“. Při odstranění celých sloupců se ostatní části posunou na místo odebraných. Jejich pořadí není vhodné měnit,

pouze na konec tabulky je nezbytné přidat atributy „UVAZEK“ a „ROLE“. Tyto položky je nutno v editoru manuálně vyplnit.

## Tabulka PŘEDMĚT

Pro získání dat patřících do entity „PŘEDMET“ je nutno nastavit vstupní parametry, viz obrázek č. 12.

Služba: Předměty  
getPredmetyByKatedra  
Katedra: KIV - Katedra informatiky a výpočetní techniky  
rok: (aktuální rok)  
semestr: LS/ZS  
lang: cs

Stažení z IS/STAG	Nahrání do IS/STAG	Výsledek	Nápověda	Server webových služeb
<p><b>i</b> Tato záložka umožňuje výběr a volání většiny webových služeb pomocí formuláře - mělo by to tedy být uživatelsky "příjemnější". Po výběru požadované služby se zobrazí její popis a případné parametry volání. Služby zde nabízené jsou služby typu "čtení z IS/STAG", tj. služby, které na základě nějakých jednoduchých vstupních parametrů vygenerují komplexní výstup.</p> <p>Po stažení dat budete přepnuti na záložku 'Výsledek', kde budete moci výsledek volání služby stáhnout ve Vámi požadovaném formátu.</p>				
Služba	Předměty getPredmetyByKatedra <input type="button" value="Vybrat"/>			
Popis služby	Vrátí seznam předmětů katedry (případně všech kateder na celé škole)			
Parametry služby	katedra: <input type="text" value="Není-li zadáno či je uveden znak %, vrátí seznam předmětů všech kateder"/> <input type="text" value="KIV - Katedra informatiky a výpočetní techniky"/> <input type="button" value="Kde mohou získat hodnotu?"/>			
	rok: <input type="text" value="Akademický rok (není-li uveden, je použit aktuální akademický rok)"/> <input type="text" value="2014 - 2014/2015"/>			
	semestr: <input type="text" value="Semestr"/> <input type="text" value="LS - Letní semestr"/>			
	lang: <input type="text" value="Jazyk výstupu služby"/> <input type="text" value="cs"/>			
<input type="button" value="Stáhní z IS/STAG"/>				

Obrázek 12 - Vstupní parametry tabulky PŘEDMĚT

Soubor pojmenovaný jako „getPredmetyByKatedra“ ponecháme ve stavu, ve kterém je. Na této tabulce není potřeba provádět změny. Je však nutné stáhnout dva soubory, jeden se semestrem zimmím a druhý s letním semestrem.

## Tabulka ROZVRHOVÁ AKCE

Pro získání dat patřících do entity „ROZVRHOVA\_AKCE“ je nutno nastavit vstupní parametry, viz obrázek č. 13.

Služba:	Rozvrhy getRozvrhByKatedra
Katedra:	KIV - Katedra informatiky a výpočetní techniky
rok:	(aktuální rok)
semestr:	LS/ZS
datumOd:	(nevyplněno)
datumDo:	(nevyplněno)
vsechnyAkce:	true
vsechnyCasyKonani:	true
jenRozvrhoveAkce	true
jenBudouciAkce:	false
lang:	cs



<a href="#">Stažení z IS/STAG</a> <a href="#">Nahrání do IS/STAG</a> <a href="#">Výsledek</a> <a href="#">Nápověda</a> <a href="#">Server webových služeb</a>																							
<p><b>i</b> Tato záložka umožňuje výběr a volání většiny webových služeb pomocí formuláře - mělo by to tedy být uživatelsky "příjemnější". Po výběru požadované služby se zobrazí její popis a případné parametry volání. Služby zde nabízené jsou služby typu "čtení z IS/STAG", tj. služby, které na základě nějakých jednoduchých vstupních parametrů vygenerují komplexní výstup.</p> <p>Po stažení dat budete přepnuti na záložku 'Výsledek', kde budete moci výsledek volání služby stáhnout ve Vámi požadovaném formátu.</p>																							
Služba	<input type="text" value="Rozvrhy"/> <input type="text" value="getRozvrhByKatedra"/> <input type="button" value="Vybrat"/>																						
Popis služby	Vrátí rozvrh všech předmětů katedry.																						
Parametry služby	<table border="1"> <tr> <td>stagUser</td> <td><b>A11B0562P</b> (Pod jakou z Vašich IS/STAG identit (rolí) službu zavolat. Máte-li více těchto identit, můžete si je přepínat v horní části této stránky.)</td> </tr> <tr> <td>katedra *</td> <td>Udává katedru či pracoviště školy  <input type="text" value="KIV - Katedra informatiky a výpočetní techniky"/>   <a href="#">Kde mohu získat hodnotu?</a> </td> </tr> <tr> <td>rok</td> <td>Akademický rok (není-li uveden, je použit aktuální akademický rok)  <input type="text" value="2014 - 2014/2015"/> </td> </tr> <tr> <td>semestr *</td> <td>Semestr. Lze použít zástupné znaky pro vyhledávání (? = libovolný znak, % = libovolný počet znaků)  <input type="text" value="LS - Letní semestr"/> </td> </tr> <tr> <td>datumOd</td> <td>Je-li vyplněno datumOd a datumDo, pak se zobrazí rozvrh mezi temito daty - vsechna data konani - a ignoruji se parametry rok, semestr a vsechnyCasyKonani  <input type="text"/> </td> </tr> <tr> <td>datumDo</td> <td>Je-li vyplněno datumOd a datumDo, pak se zobrazí rozvrh mezi temito daty - vsechna data konani - a ignoruji se parametry rok, semestr a vsechnyCasyKonani  <input type="text"/> </td> </tr> <tr> <td>vsechnyAkce</td> <td>Je-li nevyplněno nebo false, nezobrazí zrušené akce. Je-li true, zobrazí všechny akce.  <input type="text" value="true"/> </td> </tr> <tr> <td>vsechnyCasyKonani</td> <td>Je-li nevyplněno nebo false, zobrazí klasicky rozvrh (jedna RA = jedna položka). Je-li true, zobrazí všechny jednotlivé datумы konání RA.  <input type="text" value="true"/> </td> </tr> <tr> <td>jenRozvrhoveAkce</td> <td>Je-li nevyplněno nebo false, zobrazí všechny akce, například i zkouškové termíny a další součásti rozvrhu (záleží na typu rozvrhu). Je-li true, vrati pouze rozvrhové akce.  <input type="text" value="true"/> </td> </tr> <tr> <td>jenBudouciAkce</td> <td>Je-li nevyplněno nebo false, zobrazí všechny akce. Je-li true, vrati pouze akce, ktere se teprve budou konat (v ramci aktualniho dne vrati jeste vzdy vsechny).  <input type="text" value="false"/> </td> </tr> <tr> <td>lang</td> <td>Jazyk výstupu služby  <input type="text" value="cs"/> </td> </tr> </table>	stagUser	<b>A11B0562P</b> (Pod jakou z Vašich IS/STAG identit (rolí) službu zavolat. Máte-li více těchto identit, můžete si je přepínat v horní části této stránky.)	katedra *	Udává katedru či pracoviště školy <input type="text" value="KIV - Katedra informatiky a výpočetní techniky"/> <a href="#">Kde mohu získat hodnotu?</a>	rok	Akademický rok (není-li uveden, je použit aktuální akademický rok) <input type="text" value="2014 - 2014/2015"/>	semestr *	Semestr. Lze použít zástupné znaky pro vyhledávání (? = libovolný znak, % = libovolný počet znaků) <input type="text" value="LS - Letní semestr"/>	datumOd	Je-li vyplněno datumOd a datumDo, pak se zobrazí rozvrh mezi temito daty - vsechna data konani - a ignoruji se parametry rok, semestr a vsechnyCasyKonani <input type="text"/>	datumDo	Je-li vyplněno datumOd a datumDo, pak se zobrazí rozvrh mezi temito daty - vsechna data konani - a ignoruji se parametry rok, semestr a vsechnyCasyKonani <input type="text"/>	vsechnyAkce	Je-li nevyplněno nebo false, nezobrazí zrušené akce. Je-li true, zobrazí všechny akce. <input type="text" value="true"/>	vsechnyCasyKonani	Je-li nevyplněno nebo false, zobrazí klasicky rozvrh (jedna RA = jedna položka). Je-li true, zobrazí všechny jednotlivé datумы konání RA. <input type="text" value="true"/>	jenRozvrhoveAkce	Je-li nevyplněno nebo false, zobrazí všechny akce, například i zkouškové termíny a další součásti rozvrhu (záleží na typu rozvrhu). Je-li true, vrati pouze rozvrhové akce. <input type="text" value="true"/>	jenBudouciAkce	Je-li nevyplněno nebo false, zobrazí všechny akce. Je-li true, vrati pouze akce, ktere se teprve budou konat (v ramci aktualniho dne vrati jeste vzdy vsechny). <input type="text" value="false"/>	lang	Jazyk výstupu služby <input type="text" value="cs"/>
stagUser	<b>A11B0562P</b> (Pod jakou z Vašich IS/STAG identit (rolí) službu zavolat. Máte-li více těchto identit, můžete si je přepínat v horní části této stránky.)																						
katedra *	Udává katedru či pracoviště školy <input type="text" value="KIV - Katedra informatiky a výpočetní techniky"/> <a href="#">Kde mohu získat hodnotu?</a>																						
rok	Akademický rok (není-li uveden, je použit aktuální akademický rok) <input type="text" value="2014 - 2014/2015"/>																						
semestr *	Semestr. Lze použít zástupné znaky pro vyhledávání (? = libovolný znak, % = libovolný počet znaků) <input type="text" value="LS - Letní semestr"/>																						
datumOd	Je-li vyplněno datumOd a datumDo, pak se zobrazí rozvrh mezi temito daty - vsechna data konani - a ignoruji se parametry rok, semestr a vsechnyCasyKonani <input type="text"/>																						
datumDo	Je-li vyplněno datumOd a datumDo, pak se zobrazí rozvrh mezi temito daty - vsechna data konani - a ignoruji se parametry rok, semestr a vsechnyCasyKonani <input type="text"/>																						
vsechnyAkce	Je-li nevyplněno nebo false, nezobrazí zrušené akce. Je-li true, zobrazí všechny akce. <input type="text" value="true"/>																						
vsechnyCasyKonani	Je-li nevyplněno nebo false, zobrazí klasicky rozvrh (jedna RA = jedna položka). Je-li true, zobrazí všechny jednotlivé datумы konání RA. <input type="text" value="true"/>																						
jenRozvrhoveAkce	Je-li nevyplněno nebo false, zobrazí všechny akce, například i zkouškové termíny a další součásti rozvrhu (záleží na typu rozvrhu). Je-li true, vrati pouze rozvrhové akce. <input type="text" value="true"/>																						
jenBudouciAkce	Je-li nevyplněno nebo false, zobrazí všechny akce. Je-li true, vrati pouze akce, ktere se teprve budou konat (v ramci aktualniho dne vrati jeste vzdy vsechny). <input type="text" value="false"/>																						
lang	Jazyk výstupu služby <input type="text" value="cs"/>																						
<input type="button" value="Stáhní z IS/STAG"/>																							

Obrázek 13 - Vstupní parametry tabulky ROZVRHOVÁ AKCE

I v tomto případě je nutno stáhnout dva soubory. Jeden se semestrem zimním a druhý se semestrem letním. Obsahují velké množství nepotřebných informací, je proto nezbytné je před vložením do databáze upravit do podoby, jakou má i struktura entity.

Ze souboru, který je pojmenován jako „getRozvrhByKatedra“, odebereme sloupec: „ROAKIDNO“ a nahradíme jej atributem idAkce, který je potřeba naplnit identifikačními čísly od 1. Tato čísla poslouží jako primární klíče entity. Dále odebereme sloupce: „NAZEV“, „KATEDRA“, „STATUT“, „UCITIDNO.UCITEL“, „JMENO.UCITEL“, „PRIJMENI.UCITEL“, „TITULPRED.UCITEL“, „TITULZA.UCITEL“, „PLATNOST.UCITEL“, „ZAMESTNANEC.UCITEL“, „ROK“, „KAPACITAMISTNOSTI“, „TYPAKCE“, „DEN“, „HODINASKUTOD“, „HODINASKUTDO“, „TYDEN“, „GRUPIDNO“, „JE NADRAZENA“, „MA NADRAZENOU“, „KROUZKY“, „CASOVARADA“, „DATUM“, „DATUMOD“, „DATUMDO“, „DRUHAKCE“, „VSICHNIUCITELEUCITIDNO“, „VSICHNIUCITELEJMENATITULY“, „VSICHNIUCITELEPRIJMENI“, „REFERENCEIDNO“, „POZNAMKAROZVRHARE“, „NEKONASE“, „ZAKAZANEAKCE“. Při odstranění celých sloupců se ostatní části posunou na místo odebraných. Jejich pořadí není vhodné měnit.

V rámci kontroly validity bylo zjištěno, že položky atributu „kontakt“ obsahují znaky „<“ a „>“, které validátor vyhodnocuje jako chybné. Zmíněné značky je nutné z dat odstranit.

## Vkládání dat

Pro vložení dat do databáze je využíváno programu phpMyAdmin. Tato technologie umožňuje vkládání pomocí jednoduchého formuláře. U každé tabulky je potřeba vložit správný soubor, a to v následujícím pořadí:

- 1) PRACOVISTE
- 2) OSOBA
- 3) PREDMET
- 4) ROZVRHOVA\_AKCE

V případě jiného pořadí importovaných dat může opět nastat kolize cizích klíčů. Při importu dat je nutno nastavit parametry, viz obrázek č. 14.

Znaková sada souboru:	windows-1250
Formát:	CSV using LOAD DATA
Pole oddělené:	;
Pole uzavřené do:	„
Pole escapována:	\
Řádky ukončené:	auto

### Soubor pro importování:

---

Soubor může být komprimovaný (gzip, bzip2, zip) nebo nekomprimovaný.  
Jméno komprimovaného souboru musí končit na **[formát].[komprese]**. Například: **.sql.zip**

Procházet váš počítač:  Soubor nevybrán (Maximální velikost: 128MiB)

Znaková sada souboru:

### Částečný import:

---

Povolit přerušení importu v případě že skript pozná, že se blíží časový limit nastavený v PHP. *(Tímto způsobem můžete importovat i velké soubory, ale může to způsobit problémy s transakcemi.)*

Přeskočit daný počet dotazů (pro SQL) nebo řádek (pro ostatní formáty), počínaje od:

### Formát:

---

### Parametry pro výstupní formát:

---

Přepsat data tabulky souborem

Pole oddělené:

Pole uzavřené do:

Pole escapována:

Řádky ukončené:

Názvy polí:

Nepřerušit při chybě v příkazu INSERT

Použít klíčové slovo LOCAL

Obrázek 14 - Import dat do databáze

## Obsluha aplikace

Uživatelé, kterým je umožněn přístup k aplikaci, jsou zaznamenáni v entitě „UZIVATEL“. V této tabulce jsou uživatelé spravováni manuálně. Pro přidělení přístupu do aplikace je nutné uživateli přidělit login pro přihlášení a heslo, které je v databázi vedeno zašifrovanou formou s použitím SHA1.

## Přihlášení

Aplikace zprostředkovává vstup prostřednictvím jednoduchého přihlašovacího formuláře, viz obrázek č. 15. Do tohoto formuláře je potřeba zadat přihlašovací jméno a heslo. Po přihlášení se uživatel dostává na domovskou stránku aplikace.



The image shows a login form with the title "Přihlášení". It consists of two text input fields stacked vertically. The top field is labeled "Login" and the bottom field is labeled "Heslo". Below these fields is a prominent blue button with the text "Přihlásit" in white.

Obrázek 15 - Přihlašovací formulář

## Orientace

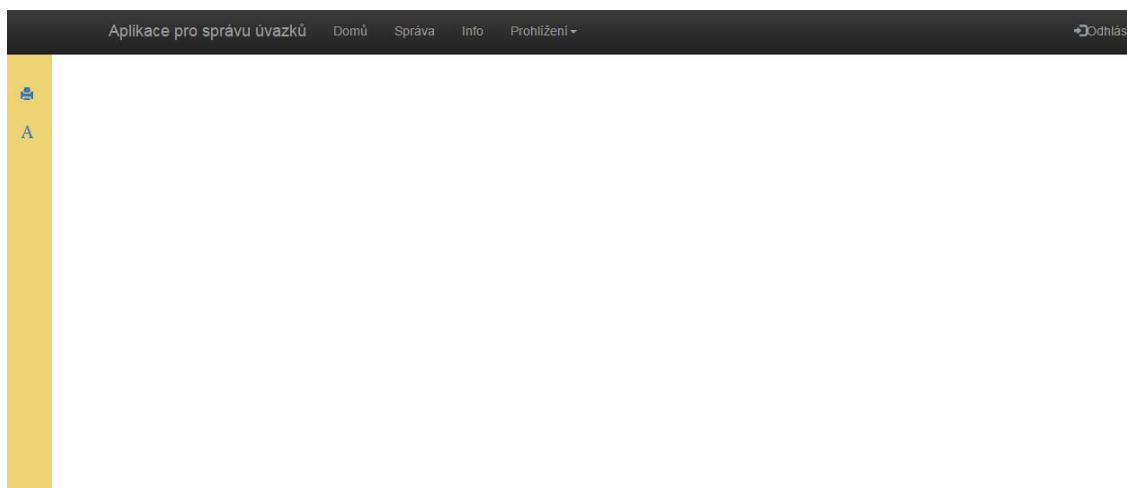
Aplikací uživatele provází hlavní navigace a postranní panel, viz obrázek č. 16. S jejich pomocí je možné rychlé zorientování.

Hlavní navigace se skládá z částí:

- Domů - domovská stránka
- Správa - možnost změny pedagogů a platností rozvrhových akcí
- Info - informace o úvazcích pedagogů
- Prohlížení - nabízí možnost prohlížení databázových tabulek
- Odhlášení - odhlášení ze systému

Postranní panel se skládá z částí:

- Tisk - umožňuje tisk aktivní stránky
- A - přesměrování na správce phpMyAdmin



Obrázek 16 - Navigace a postranní panel

## Správa

V oddílu „Správa“ je potřeba vybrat předmět, který vyžaduje změnu. Po selekci vhodného předmětu je uživateli nabídnuta další možnost, a to výběr konkrétní rozvrhové akce, viz obrázek č. 17. Po kliknutí na tlačítko OK následuje přesměrování na stránku změn, viz obrázek č. 18. V této části je možné měnit data. Po ukončení změn je potřeba potvrdit výběry tlačítkem OK. Poté následuje přesměrování uživatele na úvodní stránku správy.

Vyberte

ZI - Základy informatiky

Vyberte

14 4 - 5 Po

OK

Obrázek 17 - Výběr předmětů a rozvrhových akcí

Zkratka	Id akce	Jméno	Úvazek	Typ	Od	Do	Zkratka dne	Platnost
ZI	14	Novák Jan Ing.	100	Př	4	5	Po	A

OK

Obrázek 18 - Změna pedagoga a platnosti rozvrhové akce

## Info

V oddílu „Info“ je nutné vybrat pedagoga, u kterého je potřeba kontroly úvazků. Po zvolení vhodného zaměstnance je uživateli zobrazena tabulka se všemi rozvrhovými akcemi, viz obrázek č. 19. Nad zobrazenou tabulkou je vypsán úvazek včetně výpočtu, který udává, zda byl úvazek překročen. Po kliknutí na tlačítko „Extrahovat“ se data uloží do souboru CSV a je umožněno jejich stažení.

Vyberte:

Novák Jan Ing.,

Extrahovat

ÚVAZEK	A	B	P	N
100	2	0	0	0

Titul před	Jméno	Příjmení	Titul za	Id akce	Semestr	Zkratka předmětu	Typ	Od	Do	Týden od	Týden do	Zkratka dne	Týden	Platnost
Ing.	Jan	Novák		14	ZS	ZI	Př	4	5	38	51	Po	K	A

Obrázek 19 - Informace o úvazku pedagogů

# C Vyhodnocení nezávislých testů

## Úkoly pro testery:

- 1) Přihlaste se pomocí loginu: tester a hesla: tester.
- 2) Proveďte správu/změnu předmětu: ZI  
akce: 14  
změna učitele: Ing. Jan Novák  
změna platnosti: A
- 3) Informujte se, zda změna proběhla a Ing. Jan Novák vyučuje Předmět ZI, akci číslo 14 a akce má platnost A.
- 4) Extrahujte data Ing. Jana Nováka a zkontrolujte, zda mají všechny náležitosti.
- 5) Proveďte tisk stránky s tabulkou, která obsahuje předměty vyučované Ing. Janem Novákem.
- 6) Odhlaste se a znovu přihlaste.

**Tester č. 1** - muž 52 let, počítačová gramotnost: základní uživatelská (MS Office, internet)

- 1) Tester se úspěšně přihlásil.
- 2) Tester přehlédl navigaci a nějakou dobu trvalo, než zjistil, kde je možno provést změnu. Poté nenastal žádný problém.
- 3) Tester úspěšně zkontroloval, že změna proběhla úspěšně.
- 4) Tester přehlédl tlačítko extrahovat.
- 5) Tester úspěšně vytiskl informace o pedagogovi.
- 6) Tester úspěšně provedl přihlášení a odhlášení.

**Tester č. 2** - muž 23 let, počítačová gramotnost: programátor

- 1) Tester se úspěšně přihlásil
- 2) Tester úspěšně aktualizoval záznam.
- 3) Tester úspěšně zkontroloval, že změna proběhla úspěšně.
- 4) Tester úspěšně extrahoval záznam.
- 5) Tester úspěšně vytiskl informace o pedagogovi.
- 6) Tester úspěšně provedl přihlášení a odhlášení.

## **Závěr**

Byly zjištěny následující nedostatky:

**nevýrazná navigace** - po otevření domovské stránky je zobrazena šedá navigace, jejíž barva způsobuje zmatení uživatele

- řešení: upravení barev navigace a postranního panelu na černou a žlutou

**nevýrazné tlačítko extrakce** - po výběru pedagoga uživatel nevnímá okolí a extrakci hledá v navigaci

- řešení: zvýraznění tlačítka