

ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA EKONOMICKÁ

Bakalářská práce

**Návrh a implementace webového informačního
systému**

**Design and implementation of a web-based
information system**

Michaela Teřlová

Plzeň 2018

Čestné prohlášení

Prohlašuji, že jsem bakalářskou/diplomovou práci na téma

„Návrh a implementace webového informačního systému“

vypracovala samostatně pod odborným dohledem vedoucího bakalářské práce za použití pramenů uvedených v příložené bibliografii.

V Plzni dne 20. dubna 2018

.....
podpis autora

Poděkování

Děkuji panu doc. RNDr. Mikuláši Gangurovi, Ph.D. za cenné rady, věcné připomínky a za pomoc při zpracování této bakalářské práce.

Obsah

Úvod.....	8
1 Webový informační systém	9
2 Databázová vrstva – databázový systém	10
2.1 Databázový systém MySQL.....	10
2.2 Struktura databáze	10
2.2.1 Tabulka	10
2.2.2 Sloupce	10
2.2.3 Řádky.....	10
2.2.4 Hodnoty	11
2.2.5 Klíče	11
2.3 Dotazy SQL nad jednou tabulkou	11
2.3.1 Vytvářecí SQL dotaz	11
2.3.2 Výběrový SQL dotaz	12
2.3.3 Aktualizační SQL dotaz	13
2.3.4 Odstraňovací SQL dotaz.....	13
3 Statické a dynamické webové stránky	14
3.1 Statické stránky	14
3.1.1 Výhody	14
3.1.2 Nevýhody	14
3.2 Dynamické stránky	14
3.2.1 Výhody	15
3.2.2 Nevýhody	15
4 Programovací jazyky	16
4.1 HTML.....	16
4.1.1 Zápis HTML.....	16
4.2 CSS	17
4.3 PHP.....	17
4.4 Python.....	18
4.5 Perl.....	18
4.6 JavaScript	19
5 Nástroje.....	20
5.1 MAMP.....	20
5.2 CodeIgniter	20

5.2.1	Architektura MVC.....	20
5.3	Django	21
5.4	Dancer.....	21
6	Programovací jazyk a nástroje použité pro vývoj.....	22
6.1	PHP.....	22
6.2	MAMP.....	22
6.3	Framework CodeIgniter.....	23
6.4	GIT	23
7	Představení problému	24
7.1	Současná řešení	24
7.2	Požadavky na systém.....	24
7.2.1	Cíl	24
7.2.2	Funkcionality	25
8	Metodika	26
9	Design	28
10	Implementace	30
10.1	Prerekvizity.....	30
10.1.1	Instalace MAMP.....	30
10.1.2	Instalace frameworku CodeIgniter	30
10.1.3	Vytvoření databáze	30
10.1.4	Připojení k databázi	31
10.1.5	Návrh databáze	32
10.2	Statické stránky.....	33
10.2.1	Stránka O nás.....	33
10.2.2	Stránka Kontakt	34
10.3	Uživatelé	34
10.3.1	Registrace	34
10.3.2	Přihlášení	38
10.4	Novinky	39
10.4.1	Vytvoření článku	40
10.4.2	Výpis článku.....	44
10.4.3	Úprava článku.....	46
10.4.4	Smazání článku.....	47
10.5	Naše nabídka.....	48

10.5.1	Vytvoření nabídky	48
10.5.2	Výpis nabídky	49
10.5.3	Úprava nabídky	50
10.5.4	Smazání nabídky	50
10.6	Zabezpečení	50
10.6.1	MySQL injection	50
10.6.2	Hashování hesla	51
10.6.3	HTTPS	51
11	Funkcionality systému	53
11.1	Registrace nového uživatele	53
11.1.1	Registrační formulář	53
11.1.2	Uložení uživatele do databáze	54
11.2	Přihlášení uživatele	54
11.3	Menu	55
11.4	O nás	55
11.5	Naše nabídka	56
11.5.1	Vytvoření nabídky	57
11.5.2	Editace nabídky	57
11.5.3	Smazání nabídky	58
11.6	Novinky	58
11.6.1	Vytvoření novinky	59
11.6.2	Editace novinky	60
11.6.3	Smazání novinky	60
11.7	Kontakt	60
12	Testování	61
13	Návrhy na zlepšení a inovaci	62
	Závěr	63
	Seznam obrázků	65
	Seznam použitých zkratk	67
	Seznam použité literatury	68
	Seznam příloh	71
	Abstrakt	75
	Abstract	76

Úvod

Webové informační systémy jsou v dnešní době internetu velmi rozvíjené a je na ně brán velký důraz. Firmy je využívají k propagaci, firemní komunikaci, ale i k přímému prodeji zboží a služeb. Právě na návrh a implementaci webového informačního systému, který je využíván k propagaci, je zaměřena tato práce.

V současné době je předpokládáno, že je každá firma na internetu aktivní, je to hlavně způsobeno tím, že jsou technologie více rozvíjeny a technologická gramotnost obyvatelstva stoupá. Není překvapením, že valná většina obyvatel ve vyspělých zemích vlastní mobilní telefon s připojením k internetu, naopak je překvapením, pokud jej někdo nemá. Lidé si na skutečnost, že mohou najít vše na internetu velice zvykli, tím spíše to platí pro současnou generaci, která dostupnost informací na internetu vítá, pokud informace o podniku, zboží nebo službě online nejde, ztrácí o ně zájem.

I z tohoto důvodu bude vytvořený tento webový informační systém. Jedna z nejoblíbenějších a nejnavštěvovanějších kaváren v Plzni totiž do nedávné doby žádný informační systém neměla, informace byly uvedené pouze na jejich Facebookové stránce. Facebookové stránky však nemají téměř žádnou strukturu, co se týká samotné propagace a pro uživatele jsou poměrně nepřehledné, pokud hledá konkrétní informace.

Cílem práce je vytvořit webový informační systém pro kavárnu, který bude umožňovat jednoduchou editaci uveřejněného obsahu. Mezi dílčí cíle práce patří zajistit, aby byl systém zabezpečený tak, aby obsah mohli upravovat pouze přihlášení uživatelé, s tím souvisí implementace databáze. Dalším cílem je navrhnout a implementovat vhodný design.

Práce je rozdělena do celkem 13 kapitol. Každá kapitola se věnuje jinému problému, některé se však mezi sebou prolínají. V kapitole číslo 13, Návrhy na zlepšení a inovaci, jsou diskutovány další možné inovace webového informačního systému a možnost dalšího vývoje.

1 Webový informační systém

Webové informační systémy tvoří v současné době velmi rozšířenou část aplikací, které firmy využívají. Na rozdíl od návrhu softwaru, je při návrhu webového informačního systému brán větší ohled na odvětví, pro které systém vzniká. Webové informační systémy je možné rozdělit do několika základních oblastí, těmi jsou intranetové aplikace, webové aplikace a elektronické obchodování. Intranetové aplikace jsou vyvinuté pro uzavřené prostředí podniků, naopak webové aplikace jsou využívány zejména pro marketingové účely a komunikaci vně organizace. Elektronické obchodování slouží pro přímý prodej zboží.

Při vývoji webových informačních systémů není známa konkrétní skupina uživatelů, pouze lze předpokládat, že uživatelé budou různorodí a to jak z kulturního, tak i geografického hlediska. I přesto, že je internet v dnešní společnosti velmi rozšířený, není možné předpokládat vysokou úroveň odborné práce s výpočetní technikou (Šilhavý 2013).

Při návrhu webového informačního systému je vhodné zodpovědět následující otázky (Gangur 2014):

- Co je obsahem a náplní systému
- Kdo bude systém využívat
- Jaké jsou přínosy systému
- Jaké jsou náklady na vývoj a provoz
- Zda je možné upravit některá existující řešení
- Zda může být systém rozvíjen dle budoucích požadavků

Před samotným vývojem webového informačního systému je vhodné vytvořit strukturovaný popis problému a specifikovat požadavky. Jednotlivé kroky na sebe logicky navazují. Prvním krokem je specifikace cíle neboli účelu, kterého má být dosaženo, dále je třeba navrhnout všechny funkcionality systému, což jsou například registrace nebo přihlášení a v konečné fázi navrhnout prvky systému (Gangur 2014).

2 Databázová vrstva – databázový systém

Hlavní prvkem většiny webových informačních systémů jsou databáze a databázový systém. V jednotlivých kapitolách níže jsou popsány databázový systém MySQL, struktura databáze a dotazy nad jednou tabulkou.

2.1 Databázový systém MySQL

MySQL je systém řízení báze dat, který zajišťuje práci s databází a musí umožňovat práci s velkým množstvím dat. MySQL používá strukturovaný dotazový jazyk (SQL), který je využíván všemi relačními databázemi. Prostřednictvím SQL je možné vytvářet databáze a zpracovávat data v jednotlivých tabulkách dle zadaných kritérií (Gilfillan 2003).

2.2 Struktura databáze

Databáze má svou předem danou strukturu. Mezi základní prvky patří tabulka, sloupce tabulky, řádky tabulky, hodnoty a klíče.

2.2.1 Tabulka

Databáze obsahují tabulky, které mohou být spojené relacemi, tyto relace určují vzájemné vztahy (Gangur 2014). Tabulka obsahuje informace o názvu, typu a velikosti polí, dále obsahuje informace o tom, který sloupec je nutné vyplnit, a který může zůstat prázdný a v neposlední řadě údaje o primárním klíči a cizích klíčích (Čápka 2012).

2.2.2 Sloupce

Sloupce tabulky jsou také nazývány atributy či pole. Každý atribut musí mít unikátní název a musí mít přidělený datový typ. Nejčastěji používanými datovými typy jsou INT, VARCHAR, TEXT, DATE. Datový typ INT slouží pro zadání čísel, VARCHAR značí řetězec dané délky až do 255 bajtů, text může obsahovat až 65 535 znaků, datový typ DATE slouží pro zadání data ve tvaru RRRR-MM-DD (Gangur 2014).

2.2.3 Řádky

Dalším názvem pro řádky jsou záznamy, které obsahují informace o jednom objektu konkrétní entity. Každý řádek by měl být jednoznačně identifikován, to řeší primární klíče (Horváth 2007).

2.2.4 Hodnoty

Jak již bylo zmíněno výše, každý sloupec má definovaný datový typ, hodnota, která je ve sloupci pro určitý řádek zadána, je právě tohoto datového typu (Gangur 2014).

2.2.5 Klíče

Klíče v databázích jsou děleny na klíče primární (vlastní) a klíče sekundární (cizí). Vlastní klíč je unikátní identifikátor řádku tabulky. Obvykle se jedná o číselnou řadu, kdy každý další řádek tabulky má identifikátor o jednu hodnotu větší (Chapple 2018).

Prostřednictvím cizího klíče je odkazováno na vlastní klíč jiné tabulky a tím jsou vytvořeny mezi jednotlivými tabulkami relace. Cizí klíč nemusí být nutně jedinečný (Zedníček, 2017a).

2.3 Dotazy SQL nad jednou tabulkou

Pro manipulaci s daty v databázi je využíván jazyk SQL (Structured Query Language). Velkou výhodou SQL je, že je jazyk standardizovaný, díky tomu je možné psát kód v různých databázových systémech velice podobným způsobem.

Pro práci s daty jsou využívány příkazy INSERT, SELECT, UPDATE, DELETE, prostřednictvím dotazu CREATE je možné vytvořit celou databázi či jednotlivé tabulky (Roubal 2010).

V SQL je obvyklé zapisovat příkazy velkými písmeny a názvy tabulek a sloupců písmeny malými (Vrána 2010).

2.3.1 Vytvářecí SQL dotaz

Prostřednictvím vytvářecího dotazu CREATE je možné vytvořit jak celou databázi, tak i jednotlivé tabulky v ní. Jelikož je syntaxe tohoto příkazu velmi obsáhlá, je níže zobrazený pouze základní kód pro vytvoření tabulky (Lacko 2013).

Obrázek č. 1: SQL dotaz CREATE

```
CREATE TABLE nazev-tabulky (  
    nazev_sloupce1 datovy_typ,  
    nazev_sloupce2 datovy_typ,  
    ...  
);
```

Zdroj: vlastní zpracování, 2018

2.3.2 Výběrový SQL dotaz

Pro výběr konkrétních dat z databáze je využíván příkaz `SELECT`, který je nejčastěji používanou akcí, co se práce s databázemi týče. Po provedení příkazu jsou data vrácena v tabulkové struktuře (Dofactory 2018).

Základní příkaz `SELECT` je v následujícím tvaru.

Obrázek č. 2: SQL dotaz SELECT

```
SELECT *  
FROM nazev_tabulky;
```

Zdroj: vlastní zpracování, 2018

Hvězdička za příkazem `SELECT` udává, aby bylo ze zvolené tabulky vybráno vše. Znak je možné nahradit konkrétním parametrem. Pokud mají být vypsána pouze data z jednoho konkrétního sloupce, je příkaz konkretizován: `SELECT nazev_sloupce`.

Požadavky na data, která mají být vybrána je možné specifikovat ještě více. Definováním `WHERE` v příkazu, je možné vybrat například pouze data, která začínají na určité písmeno, případně je například možné nechat vypsát data, která jsou z určitého časového úseku. Data je možné následně seřadit vzestupně nebo sestupně (Zedníček 2017b).

Příkaz může vypadat následovně.

Obrázek č. 3: SQL dotaz SELECT s definováním WHERE

```
SELECT nazev_tabulky.nazev_sloupce  
FROM nazev_tabulky  
WHERE nazev_tabulky.nazev_sloupce >= 'konkretni_datum'  
ORDER BY nazev_tabulky.nazev_sloupce;
```

Zdroj: vlastní zpracování, 2018

2.3.3 Aktualizační SQL dotaz

Pro úpravu databáze je používán aktualizační dotaz UPDATE. Syntaxe dotazu pro aktualizaci je zobrazena níže.

Obrázek č. 4: SQL dotaz UPDATE

```
UPDATE nazev_tabulky  
SET nazev_sloupce = hodnota1, nazev_sloupce = hodnota2, ...  
WHERE podminky_vyhledavani;
```

Zdroj: vlastní zpracování, 2018

Nejprve je vyhodnocena podmínka WHERE, u záznamů, pro které je klauzule vyhodnocena jako TRUE, je položka aktualizována (Stephens 2012).

2.3.4 Odstraňovací SQL dotaz

Příkaz DELETE slouží k odstranění jednoho nebo více záznamů z tabulky, není však možné odstranit data, která jsou pro data v jiné tabulce cizím klíčem (1keydata 2018).

Obrázek č. 5: SQL dotaz DELETE

```
DELETE FROM nazev_tabulky  
WHERE podminka;
```

Zdroj: vlastní zpracování, 2018

3 Statické a dynamické webové stránky

Webové stránky se mohou dělit na statické a dynamické. Před každým vývojem je potřeba zvážit požadavky na systém a podle toho učinit rozhodnutí, která varianta bude využita. Ve zjednodušeném rozsahu jsou statické stránky takové, které není možné změnit bez úpravy zdrojového kódu, dynamické stránky jsou takové, kdy je se stejným kódem vždy zobrazen jiný obsah.

3.1 Statické stránky

Jak již bylo zmíněno v úvodu, statické webové stránky zobrazují vždy naprosto stejný obsah. Stránky jsou psány v jazyce HTML, mají tedy definovanou strukturu a mohou obsahovat například text, obrázky, audio i videa, bez úpravy přímo zdrojového je však není možné změnit (McDunnigan 2018).

3.1.1 Výhody

Jednou z mnoha výhod je ta, že statické stránky jsou při načítání velice rychlé. Jelikož stránky nejsou propojené s databází, nehrozí téměř žádné riziko útoku. S absencí databáze souvisí i nízká cena při hostování stránek. Další výhodou je to, že jsou jednoduše naprogramovatelné (Vincent 2016).

3.1.2 Nevýhody

Mezi nevýhody patří to, že pro úpravu jakéhokoliv obsahu je potřeba vývojář a samozřejmě také to, že není možné obsah personalizovat například po přihlášení (Vincent 2016).

3.2 Dynamické stránky

Dynamické webové stránky umožňují uživateli například registrovat se, následně se přihlásit a na základě toho zobrazit personalizovaný obsah, přidat komentář k příspěvku nebo vytvoření rezervace.

U dynamických webových stránek je využíváno programovacích jazyků, například PHP, k interakci s informacemi, které jsou uloženy v databázi. Z tohoto důvodu jsou dynamické stránky mnohem dražší a komplikovanější na vytvoření, viz podkapitola Nevýhody.

Jedním z termínů, který je často spojován s dynamickými webovými stránkami je CRUD, což znamená vytvoření, čtení, editace, odstranění. Jelikož je využíváno databáze, jsou tyto akce při práci s dynamickými stránkami zcela běžné. Například u blogu musí existovat možnost vytvořit článek, následně tento článek přečíst, případně jej upravit a smazat. Všechny tyto kroky probíhají v databázi (Pluralsight 2014).

3.2.1 Výhody

Mezi výhody dynamických stránek patří personalizace obsahu pro každého uživatele zvlášť, s tím souvisí i možnost přidání nových funkcí jako je přihlášení uživatele nebo platby (Vincent 2016).

3.2.2 Nevýhody

Velkou nevýhodou je možnost napadení celého systému či databáze, s integrováním databáze souvisí i to, že jsou systémy mnohem pomalejší než statické stránky a také to, že jsou náklady na jejich provoz mnohem vyšší. Vzhledem k jejich komplexnosti je i nastavení a nasazení mnohem náročnější (Vincent 2016).

4 Programovací jazyky

Pro vývoj webového informačního systému je důležité zvolit vhodné programovací jazyky. V následujících podkapitolách jsou jazyky, ze kterých je vybíráno, uvedené, výběr samotného jazyka je uvedený v kapitole č. 6, Programovací jazyk a nástroje použité pro vývoj.

4.1 HTML

HTML (Hypertext Markup Language) je hypertextový značkovací jazyk, který je používán k vytváření webových stránek. Označení hypertextový značí hypertextové odkazy, které může stránka HTML obsahovat a kterými je možné mezi sebou propojovat texty. Pojmenování značkovací jazyk pak vyjadřuje způsob, jakým se používají značky (tagy) k definování rozvržení stránky, tedy stránky tak, jak ji vidí uživatel v internetovém prohlížeči (Christensson 2015).

Všechny webové stránky jsou založeny na jazyku HTML, jinak se o webové stránky nejedná (Castro, Hyslop 2012).

4.1.1 Zápis HTML

Níže je uvedený příklad použití jazyka HTML k vytvoření jednoduché webové stránky, která obsahuje titulek, hlavní nadpis a text.

Obrázek č. 6: Zápis HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titulek stránky</title>
  </head>
  <body>
    <h1>Hlavní nadpis</h1>
    <p>Jednoduchý text.</p>
  </body>
</html>
```

Zdroj: vlastní zpracování, 2018

Jednotlivé tagy v předchozím textu mají při vývoji webových stránek velký vliv. Jejich význam je následující (W3Schools 2018):

- `<!DOCTYPE html>` definuje dokument jako HTML5
- `<html>` je kořenový prvek HTML stránky
- `<head>` obsahuje meta informace celého dokumentu
- `<title>` specifikuje titulek dokumentu
- `<body>` obsahuje viditelný obsah stránky
- `<h1>` definuje hlavní nadpis
- `<p>` definuje odstavec

4.2 CSS

CSS (Cascading Style Sheets) je kaskádový jazyk, který byl vyvinutý pro stylování HTML. Prostřednictvím tohoto jazyka je možné měnit například barvu, styl písma, velikost písma nebo také umístění textu. Kaskádový styl funguje na principu dědičnosti, čímž je kód velice přehledný, protože je odstraněna duplicita. Dědičnost spočívá v tom, že element, který je podřízený, přebírá formátování elementu, který je mu nadřazený. Například v případě stylování nadpisů, které se objevují na stránce často, je možné styl přiřadit pouze jedenkrát, není nutné definovat styl pro každý nadpis zvlášť, jako tomu bylo dříve před vývojem jazyka CSS (Čápka 2011).

I přesto, že prohlížeče implementují mnoho nových funkcí, některé jsou stále považovány za experimentální a dokud není dokončen standardizační proces, mohou se stále měnit (Gasston 2015).

4.3 PHP

PHP neboli v anglickém jazyce Hypertext Preprocessor, česky hypertextový preprocesor, je skriptovací jazyk, který je velice vhodný pro vývoj webových stránek.

Jazyk PHP patří mezi serverové technologie, to znamená, že pokud prohlížeč odešle požadavek na webovou stránku, která je spuštěna na serveru, server požadovanou stránku nejprve uspořádá a až následně ji odešle. Protože dochází ke zpracování požadavku přímo

na serveru, nejsou PHP skripty v prohlížeči vůbec zobrazeny, na použitém prohlížeči nejsou závislé a samotný výsledek je zobrazen jako HTML stránka (Zajíc 2004).

Zápis PHP

PHP kód je zapisován mezi znaky `<?php ?>`. Přičemž `<?php` značí začátek a `?>` konec kódu. Zápis jednoduchého kódu může vypadat následovně.

Obrázek č. 7: Zápis PHP

```
<?php
    echo "<h1>Toto je nadpis</h1>";
?>
```

Zdroj: vlastní zpracování, 2018

Jak je z ukázky zřejmé, kód PHP je možné kombinovat s jazykem HTML.

4.4 Python

Python je objektově orientovaný skriptovací jazyk, který nabízí rozsáhlé možnosti programování, i přesto je však zachována jednoduchá syntaxe. Kód napsaný v tomto jazyce je velmi jednoduché spojit s částmi, které jsou psány v jazyce jiném (Lutz, Ascher 2003).

Zápis Python

Pro výpis textu slouží příkaz `print`. Příklad je uvedený níže.

Obrázek č. 8: Zápis Python

```
print ("Toto je text")
```

Zdroj: vlastní zpracování, 2018

4.5 Perl

Perl je interpretační programovací jazyk, který je využíván převážně pro zpracování textu. Původně byl vyvinut pro unixovou platformu, dnes je však využíván na mnoha dalších. Základy jazyka jsou poměrně jednoduché a pro vývoj aplikace je množství kódu, oproti ostatním jazykům, malé. Jazyk má vlastní filozofii, nejedná se tak o přepisování konstrukcí jiných programovacích jazyků do syntaxe Perl. Jazyk je využíván spíše pro naprogramování kratších, zato efektivních programů (Václavík 2005).

Zápis Perl

Jak již bylo zmíněno výše, základy jazyka jsou velice jednoduché. Výpis textu `text!` je zobrazený níže.

Obrázek č. 9: Zápis Perl

```
print "text!\n";
```

Zdroj: vlastní zpracování, 2018

4.6 JavaScript

JavaScript je objektově orientovaný, multiplatformní skriptovací jazyk, pomocí kterého je možné vytvářet vyskakovací okna nebo například vytvářet interaktivní galerii obrázků s animacemi a efekty (Hrebenar 2018).

Zápis JavaScript

Jednotlivé skripty je možné vkládat přímo do kódu HTML, kód je vložen mezi tagy `<script>` a `</script>` (Hrebenar 2018).

Obrázek č. 10: Zápis JavaScript

```
<script type="text/javascript">  
    kód  
</script>
```

Zdroj: vlastní zpracování, 2018

5 Nástroje

Pro další vývoj systému je vybíráno z celkem tří programovacích jazyků, těmi jsou PHP, Python a Perl. V této návaznosti jsou vybírány i potřebné nástroje. První z nich je MAMP, který slouží pro připojení serverům, tento nástroj podporuje všechny tři programovací jazyky. Dalšími nástroji jsou CodeIgniter, Django a Dancer, každý z těchto frameworků podporuje jiný programovací jazyk, níže jsou uvedené jejich výhody a další informace.

5.1 MAMP

Zkratka MAMP vychází z celkem čtyř slov, jsou to Macintosh, Apache, MySQL a PHP (případně Python nebo Perl). Všechny tyto části lze stáhnout do počítače v jednom balíčku. MAMP je vytvořen speciálně pro počítače společnosti Apple. S instalací je zprovozněn přístup k MySQL serveru a lokálnímu PHP serveru (Hopkins 2014).

5.2 CodeIgniter

CodeIgniter je bezplatný, snadno použitelný framework pro objektově orientované programování v jazyce PHP, který využívá architekturu MVC (viz níže). Objektově orientované programování umožňuje snadnější pochopení kódu, které je strukturované a koherentní, díky tomu je vývoj jednodušší a systém je možné rozšiřovat o další funkce.

Pro CI je připravena velmi rozsáhlá uživatelská příručka, díky které je orientace mnohem jednodušší. Tato příručka je mnohem užitečnější, než je tomu u jiných frameworků. Rámec navíc funguje na všech sdílených i prodejních serverech (Stones 2012).

5.2.1 Architektura MVC

MVC neboli Model-View-Controller, je architektura, která řeší aplikační vrstvu systému. Základním principem je nezávislost jednotlivých komponent, pokud tedy dojde ke změně jedné komponenty, zbylé dvě nejsou ovlivněny.

Komponenta model zabezpečuje práci s daty, která jsou v aplikaci využívána. Jedná se například o čtení z databáze i zápis do databáze.

View slouží k zobrazení uživatelského rozhraní systému, nejsou zde prováděné žádné akce.

Mezi modelem a view se nachází controller, který zprostředkovává komunikaci s uživatelem. Pokud je proveden dotaz, který se týká databáze, obdrží jej jako první controller, ten jej následně předá modelu a model jej odešle do databáze (Kottnauer 2009).

5.3 Django

Framework Django byl vyvinut v novinářském prostředí jako součást redakčních systémů. Jedná se framework, který je napsaný v programovacím jazyce Python a využívá, stejně jako CI, architektury MVC. Je zde brán velký důraz na inovaci a perfekcionismus. Jedním z hlavních principů je neopakování se, uživatel je nucen rozvrhnout svou práci tak, aby nedocházelo k duplicitním kódům (Dvořák 2009).

5.4 Dancer

Dancer je aplikační rámec pro programovací jazyk Perl a je inspirován rámcem Sinatra. Dancer je popisován jako jednoduchý a flexibilní, ale dost silný na to, aby spustil většinu webových aplikací. Framework obsahuje velké množství pluginů, což umožňuje rychlé přidání zásuvných modulů do aplikace (Holloway 2015).

6 Programovací jazyk a nástroje použité pro vývoj

Pro vytvoření webového informačního systému, byly zvoleny programovací jazyk PHP a nástroje MAMP, GIT a framework CodeIgniter. Jejich výhody jsou popsány v jednotlivých kapitolách níže.

6.1 PHP

Programovací jazyk PHP byl zvolen z důvodu, že je v České republice jedním z nejpoužívanějších, při dalším vývoji a údržbě by tak nemělo být příliš těžké najít člověka, který bude kódu rozumět a bude moci systém spravovat.

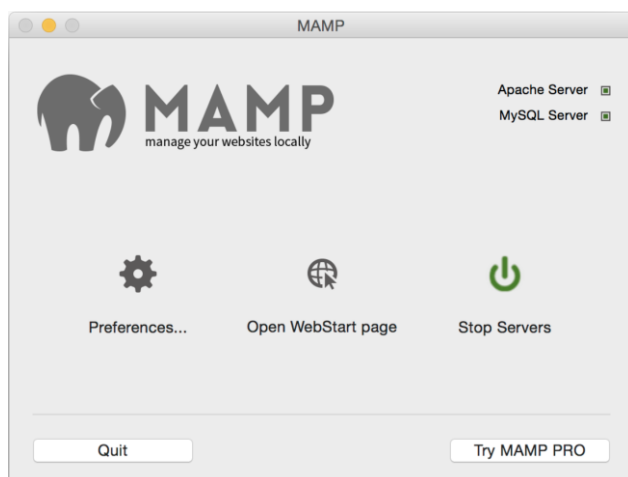
Hlavní výhodou programovacího jazyka PHP je, že umí přímo komunikovat s databází. Další výhodou je jeho rychlost a taky to, že je podporován velkou komunitou, proto má velmi dobrou podporu (Rongala 2015).

6.2 MAMP

Jak již bylo zmíněno výše v kapitole 5 Nástroje, instalace a používání nástroje pro spuštění serveru na lokálním počítači, jsou velice jednoduché, bez nutnosti měnit jakékoliv konfigurační soubory. Instalací MAMP není trvale změněno žádné původní nastavení systému a pokud již není MAMP využíván, je možné jej jednoduše odinstalovat pouhým smazáním složky. Navíc je možné základní verzi využívat zcela zdarma.

Pro zahájení využívání serveru na lokálním počítači, je nutné pouze server spustit. Toto je možné učinit v jediném kliknutí v programu MAMP. Na obrázku níže, jsou zobrazeny spuštěné servery Apache a MySQL.

Obrázek č. 11: MAMP – spuštění serveru



Zdroj: vlastní zpracování, 2018

6.3 Framework CodeIgniter

Framework CI byl zvolen v návaznosti výběru programovacího jazyka PHP. První hlavní výhodou frameworku CodeIgniter je jeho jednoduchá struktura a syntax, díky tomu je možné celý program napsat bez závažnějších chyb. Velkou výhodou jsou také funkcionality jazyka a jeho slovník, jakmile jsou jednou základy pochopeny, je možné znalosti velmi rychle rozvíjet. Mezi další výhody je řazena rozšiřitelnost programovací struktury a architektury, mezi které jsou řazeny například vzory, knihovny a znalost API, díky tomu se vývojář může soustředit převážně na vylepšování systému. Ve srovnání s jinými frameworky je velmi rychlý a před zahájením nevyžaduje žádné zdlouhavé nastavování (Peck 2013).

6.4 GIT

GIT je v současné době nejčastěji používaným systémem, co se správy jednotlivých verzí projektu týče. Při ukládání kódu je možné pracovat i offline, nejprve je práce vytvořena na lokálním počítači, následně je místní úložiště synchronizováno s kopií na serveru. Vždy, když je práce uložena, je vytvořeno potvrzení, tím je snímek všech souborů v konkrétním časovém okamžiku. Pokud některý ze souborů nebyl změněn, je použit dříve vytvořený soubor. S každým vygenerovaným potvrzením jsou vytvářeny odkazy na předchozí potvrzení, tím vzniká vývojový graf projektu. Předchozí kód je možné kdykoliv vrátit zpět, což je velice užitečné, pokud je při vývoji fungující kód přepsán kódem nefunkčním (Ngan 2018).

7 Představení problému

Dříve než byl zahájen vývoj webového informačního systému, byla zhodnocena současná řešení, která jsou na trhu dostupná, následně byly navrženy požadavky na systém, konkrétně tedy cíl celého projektu a funkcionality stránek, které by měly být do systému implementovány.

7.1 Současná řešení

Ačkoliv je v současné době na trhu mnoho dostupných řešení, která je možné upravit přímo na přání klienta, jsou cílovou skupinou těchto projektů obvykle pouze velké firmy. Jelikož velké firmy mají odlišné a větší požadavky na systém než firmy malé, může systém ve stejném rozsahu být pro malé podniky matoucí, protože obsahuje nadbytečné funkce, které nevyužijí. Tyto systémy jsou také poměrně náročné na paměť a kvůli tomu jsou pomalejší. Z tohoto důvodu bude vytvořen informační systém, který nebude zahrnovat zbytečné procesy.

7.2 Požadavky na systém

Před vývojem samotného webového informačního systému, byl nejprve definován cíl, kterého má být dosaženo a spolu s ním byly definovány i funkcionality, které by systém měl obsahovat. Funkcionality, kterých bylo při vývoji skutečně dosaženo jsou uvedené v kapitole 11 Funkcionality systému.

7.2.1 Cíl

Cílem je vytvořit funkční webový informační systém, který bude intuitivní a uživatelsky příjemný. Obsahem stránek budou informace o kavárně, které budou zákazníci informovat o novinkách, akcích, které se budou konat nebo se konaly a dalších událostech. Zároveň zde budou uvedeny informace o nabízeném sortimentu nápojů a potravin a jeho cenách.

7.2.2 Funkcionality

Jelikož se jedná o webový informační systém, který budou moci obsluhovat pouze přihlášení uživatelé, což budou primárně zaměstnanci podniku, jsou požadavky na systém následující:

- registrace uživatele
- přihlášení uživatele
- jednoduché vytvoření článku
- jednoduchá úprava článků
- jednoduché odstranění článku
- jednoduché vytvoření nabídky
- jednoduchá úprava nabídky
- jednoduché odstranění nabídky
- zabezpečení

8 Metodika

Při vývoji webového informačního systému bude po vymezení problému nejprve navržen celkový design a rozložení stránek, tím bude definován i počet položek samotného menu. Ze všeho nejdříve bude nutné zvážit, zda vytvořit jednostránkový systém, který nebude obsahovat žádné podstránky, což je v dnešní době velmi moderní a pro uživatele pohodlné, jelikož není nucen procházet mnoho textu nebo zda budou vytvořeny klasické webové stránky s navigačním menu pro přechod mezi stránkami.

Po definování obsahu jednotlivých stránek, bude předběžný návrh systému nakreslen v programu Sketch. Tento program je vhodný k vytvoření designu webových stránek a mobilních aplikací.

Statické stránky začnou být vyvíjeny poté, co budou grafické návrhy hotové. Nejprve bude vytvořena hlavní stránka O nás, která bude obsahovat základní informace o podniku, následně bude vytvořena stránka Kontakt. Na této stránce budou uvedené, jak název napovídá, kontaktní údaje a také mapa. Jelikož se data, která jsou na těchto dvou stránkách uvedena, nemění příliš často, jsou jako statické stránky ponechány. Následně budou připravené stránky Naše nabídka a Novinky. Tyto stránky budou v dalším kroku převedeny na dynamické, protože jsou do nich často přidávány nové položky. Aby bylo možné zkontrolovat, zda bude návrh po naplnění daty funkční, budou použita data testovací, která budou zadána přímo do HTML. Nejprve budou nastýlovány samotné stránky, které budou sloužit pro výpis položek, poté budou nastýlovány veškeré formuláře, které budou využívány.

Dále bude možné pokračovat ve vývoji dynamické části. Ze všeho nejdříve bude vytvořena databáze, do které budou přidány jednotlivé tabulky. Tato databáze bude propojena s frameworkem CodeIgniter. Pro stránky Naše nabídka a Novinky bude vždy naprogramováno nejprve vytvoření novinky/článku, následně jejich výpis (pro Novinky bude vypsán i detail konkrétního článku), dále úprava a nakonec odstranění.

Jelikož pro editaci bude muset být uživatel přihlášený, bude následovat vytvoření registračního formuláře. V této fázi bude nutné vyřešit i ochranu hesla, které bude ukládáno do databáze. Poté co bude registrační formulář plně funkční, bude vytvořený i formulář pro přihlášení uživatele.

Až bude v systému definovaný přihlášený uživatel, bude možné zprovoznit veškeré odkazy, které budou přístupné pouze přihlášeným uživatelům. To jsou například odkazy pro vytvoření, úpravu nebo odstranění článků.

V závěrečné fázi proběhne testování, aby bylo ověřeno, zda opravdu vše funguje dle požadavků. Poté, co bude systém fungovat bez jakýchkoliv problémů, bude zveřejněn na URL adrese www.terlova.marcuk.cz, kdy opět proběhne testování a následné odladění systému.

9 Design

Při vývoji webového informačního systému byl kladen velký důraz na design celých webových stránek, protože právě celková úprava a vzhled potenciálního zákazníka zaujme.

Celý webový informační systém je laděný do světlých šedých barev, s prvky barvy bordové. Hlavním prvkem, který každého na první pohled upoutá, jsou výrazné fotografie a zároveň jednoduchost.

Nejprve bylo potřeba vyřešit, zda systém bude jednostránkový, celý jeho obsah tak bude pouze na jedné stránce a uživatel se bude po stránce pohybovat prostřednictvím menu, které bude fungovat na principu kotev. Nebo zda budou vytvořeny klasické webové stránky s menu, které uživatele přesměruje na samostatnou stránku. Vzhledem k tomu, že do systému mělo být možné přidávat jednotlivé položky menu a články, kterých v konečné fázi může být mnoho, byla zvolena druhá z variant.

Menu obsahuje celkem čtyři položky, jsou to O nás, Naše nabídka, Novinky a Kontakt. Jelikož položek není mnoho a žádná z nich neobsahuje podmenu, je menu nastylováno na horní straně obrazovky přímo pod logem. Při rolování stránky zůstává viditelné, uživatel tak k němu má stále přístup a může se mezi stránkami pohodlně pohybovat. Aktivní položka menu je zvýrazněna, uživatel proto vždy ví, kde se právě nachází.

Nad menu se nachází logo kavárny, které je ve stejném stylu jako zbylá část stránek. Logo bylo vytvořené v programu Sketch. Dominantou loga je šálek uprostřed, který vyjadřuje vztah ke kávě. Lístky okolo značí čerstvost použitých surovin a kladný vztah k přírodě. Při rolování stránky je logo zmenšeno a umístěno do levé části k menu, které je na stránce neustále viditelné.

Stránka O nás zaujme fotografií kavárny, která je téměř přes celé okno. Déle je zde uvedený slogan “S kávou je den hezčí!”, který kavárna k propagaci využívá a v dolní části stránky je zobrazen uvítací text, který vyzývá zákazníky k návštěvě.

Nabídka kávy, dortů a jednoduchých sendvičů či bagetek je zobrazena na stránce Naše nabídka. Každá sekce obsahuje velký nadpis, pod kterým jsou zobrazené jednotlivé položky, ty jsou střídavě zobrazené vlevo a vpravo. U každého produktu je zobrazena fotografie, aby zákazník věděl, co pod uvedeným názvem může očekávat a také cena.

Pro lepší orientaci v nabídce je na pravé straně zobrazeno menu, které odkazuje na jednotlivé sekce.

Hlavní stránka novinek obsahuje seznam všech článků. Na jednom řádku jsou vždy zobrazeny celkem tři novinky, u každé je obrázek a titulek. Jednotlivé položky jsou zobrazené střídavě ve světlejší a tmavší barvě. Opět je zde brán velký důraz na výrazné a kvalitní fotografie.

Na poslední stránce Kontakt je uvedena adresa a, jak je z názvu patrné, kontaktní údaje. Tyto informace jsou zobrazené na středu stránky, zákazník je proto nemůže přehlédnout. V dolní části stránky je mapa, která byla upravena tak, aby ladila do kompozice stránek. Mapa je nastýlována do světlých barev, s šedým základem a některými barevnými prvky, které jsou převážně v cihlovém odstínu.

Formuláře, kterých je využíváno, jsou také ve velmi jednoduchém designu. Jednotlivá pole jsou ve světlé šedé barvě jako je i pozadí stránek, ohraničena jsou tmavší šedou barvou.

10 Implementace

Po návržení designu webového informačního systému byla dalším logickým krokem samotná implementace. Nejprve byla provedena instalace nástrojů a vytvořena databáze, po těchto krocích byly vytvořeny statické stránky, po kterých následoval vývoj stránek dynamických spolu s registrací a přihlašováním.

10.1 Prerekvizity

Před zahájením samotného vývoje webových stránek kavárny, je nutné nastavit několik základních funkcí, bez kterých by nemohl být vývoj zahájen. Mezi tyto prerekvizity patří instalace MAMP, instalace frameworku CodeIgniter a následné připojení k databázi.

10.1.1 Instalace MAMP

Balík, který, jak již bylo zmíněno výše, obsahuje tři programy (Apache, MySQL, PHP), byl stažen z oficiálních stránek www.mamp.info.

Po úspěšné instalaci bylo možné lokální server spustit. Port, na kterém běží webový server byl změněn na 80, pro zobrazení webových stránek kavárny ve webovém prohlížeči na lokálním počítači, bylo možné zadávat adresu <http://localhost>.

10.1.2 Instalace frameworku CodeIgniter

Zdrojové kódy pro zprovoznění samotného frameworku byly staženy z oficiálních stránek www.codeigniter.com. Tyto kódy jsou následně nakopírovány do adresáře, ze kterého čte i webový server, v tomto případě Apache, nainstalovaný v rámci MAMP.

10.1.3 Vytvoření databáze

Před připojením frameworku CodeIgniter k databázi, bylo nutné databázi nejprve vytvořit. Pro správu databáze byl zvolen programový systém phpMyAdmin, který prostřednictvím webového rozhraní umožňuje jednoduchou správu databáze.

V systému byla vytvořena nová databáze s názvem baka, ve které budou později vytvořeny tabulky potřebné pro vývoj webových stránek. Pro databázi baka byla nastavena znaková sada UTF-8. SQL příkaz, kterým byla nová databáze vytvořena, je zobrazený níže.

Obrázek č. 12: Vytvoření databáze

```
CREATE DATABASE 'baka' CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Zdroj: vlastní zpracování, 2018

10.1.4 Připojení k databázi

Pro připojení Codeigniteru k databázi, která je spuštěna na lokálním počítači, bylo nutné nastavit několik parametrů. Jedním z nejdůležitějších byla adresa lokálního počítače, následně přihlašovací jméno a heslo. Dalším důležitým parametrem bylo konkretizování databáze, v tomto případě se jedná o databázi s názvem baka.

Pro databázi je možné přenastavit několik dalších parametrů, které jsou v systému přednastaveny. Jedním z nich je například nastavení kódování, které je v této chvíli nastavené na UTF-8.

Toto nastavení slouží pouze pro databázi na lokálním počítači, pokud by byla databáze umístěna na produkčním serveru, bylo by nastavení odlišné. Vývojové nastavení je viditelné na obrázku níže.

Obrázek č. 13: Připojení k databázi

```
$db['default'] = array(
    'dsn'           => '',
    'hostname'     => 'localhost',
    'username'     => 'root',
    'password'     => 'root',
    'database'     => 'baka',
    'dbdriver'     => 'mysqli',
    'dbprefix'     => '',
    'pconnect'     => FALSE,
    'db_debug'     => (ENVIRONMENT !== 'production'),
    'cache_on'     => FALSE,
    'cachedir'     => '',
    'char_set'     => 'utf8',
    'dbcollat'     => 'utf8_general_ci',
    'swap_pre'     => '',
    'encrypt'      => FALSE,
    'compress'     => FALSE,
    'stricton'     => FALSE,
    'failover'     => array(),
    'save_queries' => TRUE
);
```

Zdroj: vlastní zpracování, 2018

10.1.5 Návrh databáze

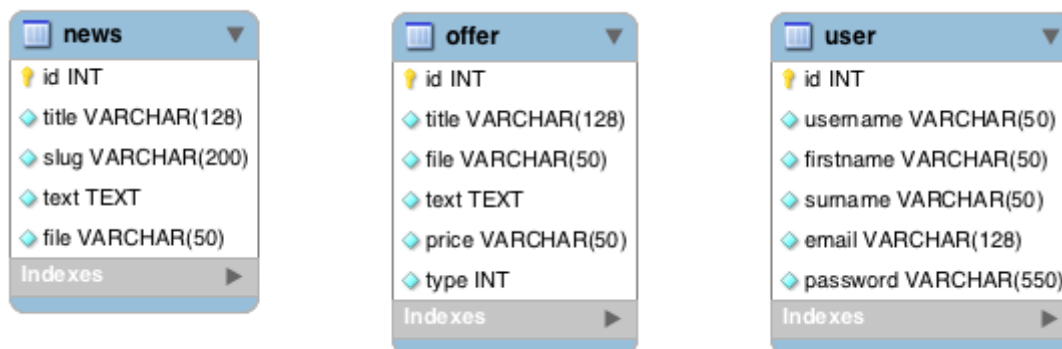
Při práci s webovým informačním systémem, bude potřeba ukládat některá data do databáze. Bude nutné ukládat informace o registrovaném uživateli, informace o článcích a informace o nabídce kavárny. Z toho důvodu byl navržen model databáze. Jednotlivé tabulky mezi sebou nemají žádný vztah, jelikož na sebe nenavazují a nevyužívají mezi sebou cizí klíče.

Databáze bude obsahovat celkem 3 tabulky – news, offer a user. Tabulka news bude sloužit práci se článkem a bude nutné zadat titulek, slug, který bude vygenerován z titulku, text článku a obrázek. Všechny atributy s výjimkou textu jsou datového typu VARCHAR s definovaným rozsahem. U tributu text je předpokládán větší rozsah, proto je datového typu TEXT.

Tabulka offer bude uchovávat informace o jednotlivých položkách nabídky a bude obsahovat opět titulek, obrázek a text, tyto atributy budou mít stejný datový typ jako mají tyto položky v tabulce news. Dále bude nutné uložit cenu a také typ od 1 do 3, v závislosti na typu položky (káva, dort, bageta), pro tyto atributy bude nastaven datový typ INT.

Do poslední tabulky user budou ukládány informace o právě registrovaném uživateli, podle této tabulky budou později kontrolovány údaje při přihlášení. Do této tabulky bude uloženo přihlašovací jméno, jméno, příjmení, email a heslo, které však nebude uloženo v původní podobě, aby nedošlo ke zneužití. Veškeré atributy jsou typu VARCHAR s nastavenou maximální velikostí. Pro heslo je tento rozsah největší, jelikož bude nejprve upraveno.

Obrázek č. 14: Datový model



Zdroj: vlastní zpracování, 2018

10.2 Statické stránky

Při vývoji webových stránek byly implementovány dvě statické stránky, které nebude možné upravovat, jelikož to pravděpodobně nebude ani nutné. Jsou to stránky O nás a Kontakt.

10.2.1 Stránka O nás

Stránka O nás obsahuje velkou fotografii kavárny, slogan, který je k propagaci využíván a krátký text, který zve zákazníky na návštěvu.

Fotografie kavárny byla nastavena jako pozadí stránky s nastavenou konkrétní maximální výškou, aby se fotografie zobrazovala vždy téměř celá a byla automaticky uzpůsobována velikosti okna, byla nastýlována přes kaskádové styly.

Obrázek č. 15: CSS pro stylování fotografie na hlavní stránce

```
.home-background{
    content: "";
    background-image: url(bak_obr/home.jpeg);
    width: 100%;
    height: 600px;
    background-size: cover;
    background-position: center;
    margin-top: 20px;
}
```

Zdroj: vlastní zpracování, 2018

10.2.2 Stránka Kontakt

Na stránce Kontakt jsou uvedeny základní informace o lokalitě a kontaktních údajích. Na úplném konci stránky je mapa, která vyznačuje adresu, na které se kavárna nachází. Mapa byla upravena tak, aby ladila ke zbytku stránek.

Stylování je provedeno prostřednictvím JavaScriptu. Pro zobrazení byly využity mapy společnosti Google. Pro mapu je nastavena výchozí poloha, která je po načtení stránky zobrazena a přepínač, který tuto polohu zvýrazňuje, dále je nastaven zoom a hlavně samotný styl mapy. Jelikož je kód pro stylizaci příliš rozsáhlý, je uveden v Příloze A.

10.3 Uživatelé

Jelikož byl jeden z požadavků na systém vytvořit jednoduché a uživatelsky intuitivní rozhraní pro správu článků a nabídky, bylo nutné umožnit registraci nového uživatele. Přihlášený uživatel má následně možnost tyto sekce spravovat.

10.3.1 Registrace

Pro registraci nového uživatele byl nejprve vytvořený formulář v jazyce HTML, který obsahuje celkem sedm textových polí. Tato pole slouží pro zadání přihlašovacího jména, jména, příjmení, emailu a zadání 2x stejného hesla pro porovnání. Prostřednictvím frameworku CI je zajištěno, že pole zůstane vyplněné, pokud registrace nebude provedena úspěšně. Příklad kódu pro vytvoření pole pro přihlašovací jméno včetně výpisu předchozí zadané hodnoty a chybových hlášek, pokud pole není vyplněné dle instrukcí, je zobrazený níže.

Obrázek č. 16: Vytvoření pole pro přihlašovací jméno

```
<?php echo form_open('register'); ?>
    <div class="error"><?php echo form_error('username'); ?></div>
    <label for="username">Přihlašovací jméno</label>
    <input type="text" name="username"
        value="<?php echo set_value('username'); ?>"
        placeholder="Přihlašovací jméno musí být 5 až 12 znaků dlouhé."
        size="50"><br />
```

Zdroj: vlastní zpracování, 2018

Registrační formulář je možné zobrazit na adrese „/user/register“. Díky definování této cesty v souboru routes.php, který je uložený ve složce config, je možné stránku zobrazit pouhým zadáním „/register“.

Obrázek č. 17: Definování cesty pro registrační formulář

```
$route['register'] = 'user/register';
```

Zdroj: vlastní zpracování, 2018

Ve funkci register, která je definována v controlleru v souboru User.php, je nejprve načtena knihovna pro validaci samotného formuláře, která zprostředkovává jednoduchou validaci polí.

Obrázek č. 18: Načtení knihovny pro validaci formuláře

```
public function register()
{
    $this->load->library('form_validation');
    ...
}
```

Zdroj: vlastní zpracování, 2018

Následně je tato knihovna využita pro nastavení pravidel pro jednotlivá pole. V kódu níže jsou pro příklad definována pravidla pro nastavení uživatelského jména. Uživatelské jméno musí být zadané, unikátní a musí být v rozmezí od 5 do 12 znaků. Pokud jsou pravidla porušena, je u pole zobrazena konkrétní hláška, ve které je použit zástupný znak %s, který vypíše název pole, v tomto případě Přihlašovací jméno.

Obrázek č. 19: Nastavení pravidel pro přihlašovací jméno

```
$this->form_validation->set_rules('username', 'Přihlašovací jméno',  
'trim|required|min_length[5]|max_length[12]|is_unique[user.username]',  
    array(  
        'required'    => 'Zadejte %s.',  
        'is_unique'   => 'Toto %s již existuje.',  
        'min_length'  => 'Minimální délka je 5 znaků.',  
        'max_length'  => 'Maximální délka je 12 znaků.'  
    )  
);
```

Zdroj: vlastní zpracování, 2018

Výstrahy se liší dle konkrétního pole, každé pole však musí být vyplněné. Pro email a heslo, které je zadáváno pro kontrolu znovu, jsou definována další dvě pravidla. Email musí být vyplněný ve tvaru, který je pro emailovou adresu obvyklý, tedy [jméno]@[doména druhého řádu].[doména prvního řádu], tento požadavek je definován jako 'valid_email'. Heslo, které slouží pro kontrolu, se musí shodovat s heslem, které bylo zadáno původně, tento požadavek je zapsán jako 'matches[password]', password je v tomto případě první zadané heslo.

Pokud je stránka formuláře zobrazena poprvé nebo je formulář vyplněn chybně, je stránka registrace načtena znovu. Pokud naopak vše proběhne v pořádku, jsou data prostřednictvím funkce set_user uložena do databáze, funkce je blíže popsána v obrázku číslo 20. Uživatel je o této skutečnosti informován prostřednictvím zprávy, kdy je na obrazovce zobrazeno "Registrace proběhla úspěšně.", zároveň je uživatel přesměrován na hlavní stránku.

Obrázek č. 20: Výpis stránky při úspěšné/neúspěšné registraci

```
if ($this->form_validation->run() == FALSE)
{
    $this->load->view('templates/header', $data);
    $this->load->view('user/register', $data);
    $this->load->view('templates/footer');
}
else
{
    $this->user_model->set_user();
    $this->session->set_flashdata('msg', 'Registrace proběhla
úspěšně.');
```

Zdroj: vlastní zpracování, 2018

Když je registrace provedena úspěšně, jsou data uložena do databáze s názvem baka. Pro tento účel byla v databázi vytvořena tabulka user. Zápis do databáze je proveden prostřednictvím funkce set_user, která je uvedena níže.

Obrázek č. 21: Zápis uživatele do databáze

```
public function set_user()
{
    $data = array(
        'username' => $this->input->post('username'),
        'firstname' => $this->input->post('firstname'),
        'surname' => $this->input->post('surname'),
        'email' => $this->input->post('email'),
        'password' => $this->hash_password($this->input->post('password')),
    );
    return $this->db->insert('user', $data);
}
```

Zdroj: vlastní zpracování, 2018

Do proměnné data, jsou nejprve uložena data, která byla zadána do formuláře, následně jsou tyto údaje uloženy do databáze prostřednictvím Active record, což slouží

pro zjednodušení komunikace s databází. Funkce hash_password je detailněji popsána v podkapitole 10.6 Zabezpečení.

10.3.2 Přihlášení

Pro přihlášení uživatele musí být správně zadané přihlašovací jméno a heslo, to je kontrolováno prostřednictvím funkce login. Přihlašovací formulář je dostupný na stránce „/login“ („/user/login“), zkrácená cesta je opět zabezpečena nadefinováním cesty v souboru routes.php. Nejprve je zkontrolováno, zda jsou údaje v polích zadána, pokud nejsou, není provedena žádná akce. Zadané údaje jsou porovnány s tabulkou user, která je uložena v databázi. Pokud se data v databázi nenachází, není možné uživatele přihlásit, je zobrazena hláška “Špatně zadané uživatelské jméno nebo heslo.” a formulář je zobrazen znovu.

Obrázek č. 22: Nastavení session

```
$data['logged_in'] = $this->session->logged_in;
$this->load->view('templates/header', $data);
```

Zdroj: vlastní zpracování, 2018

Obrázek č. 23: Funkce login

```
public function login()
{
    if ($this->input->post('username') === FALSE ||
        $this->input->post('password') === FALSE)
    {
        return false;
    }

    $query = $this->db->get_where('user', array(
        'username' => $this->input->post('username'),
        'password' => $this->hash_password($this->input->post('password'))
    ));
    return $query->row_array();
}
```

Zdroj: vlastní zpracování, 2018

V případě, že jsou data zadána správně, jsou informace o uživateli uloženy do session spolu s ukazatelem, že je uživatel přihlášen, tedy proměnná `logged_in` je nastavena na `TRUE`, což je následně využíváno v šablonách.

Obrázek č. 24: Uložení do session

```
$user = $this->user_model->login();
$newdata = array(
    'username' => $user['username'],
    'email' => $user['email'],
    'logged_in' => TRUE
);

$this->session->set_userdata($newdata);
$this->session->set_flashdata('msg', 'Přihlášení proběhlo úspěšně.');
```

```
redirect('/');
```

Zdroj: vlastní zpracování, 2018

10.4 Novinky

Před každou z akcí, které jsou uvedeny v této kapitole, je nejprve zkontrolováno, zda je uživatel přihlášen. Toto je zajištěno prostřednictvím definované hodnoty proměnné `logged_in` při přihlášení. Tato proměnná je dostupná v šablonách po nastavení do pole `data`.

V následujícím příkladu je zobrazen odkaz pro vytvoření nového článku, pokud je uživatel přihlášen.

Obrázek č. 25: Zobrazení odkazu pro přihlášeného uživatele

```
<?php if($logged_in): ?>
    <div class="navigation">
        <a href="/news/create">Přidat nový článek</a>
    </div>
<?php endif; ?>
```

Zdroj: vlastní zpracování, 2018

Pokud uživatel přihlášen není a URL adresa je do adresního řádku zadána ručně, je uživatel přesměrován na přihlašovací formulář.

Obrázek č. 26: Přesměrování nepřihlášeného uživatele

```
if($this->session->logged_in == FALSE){  
    redirect('/login', 'refresh');  
}
```

Zdroj: vlastní zpracování, 2018

10.4.1 Vytvoření článku

Pro snadné vytvoření článku byl opět vytvořen formulář, do kterého jsou zadávány požadované hodnoty. Formulář pro vytvoření novinky je dostupný na stránce „/novinky/create“ (případně „/news/create“). Formulář obsahuje titulek článku, text a obrázek, všechny tyto položky musí být vyplněné. Pro vložení textu článku, je implementován textový editor, díky kterému je možné definovat nadpisy, vkládat odkazy, seznamy a měnit styl písma. Zároveň je možné vložit blokovou citaci. K této činnosti je využíván CKEditor, jelikož je jeho implementace velmi jednoduchá a je možné jej využívat prakticky ihned. Editor je stažen z CDN. Ta slouží ke stažení knihovny z internetu bez nutnosti vložení knihovny přímo do projektu. Použití editoru pro pole, do kterého je vkládán text článku, je definováno prostřednictvím id.

Obrázek č. 27: Implementace CKEditoru

```
<script src="https://cdn.ckeditor.com/ckeditor5/1.0.0-alpha.2/classic/ckeditor.js">  
</script>  
...  
<textarea name="content" id="editor"><?php echo set_value('content'); ?>  
</textarea><br />  
...  
<script>  
    ClassicEditor  
        .create(document.querySelector('#editor'))  
        .catch(error => {  
            console.error(error);  
        });  
</script>
```

Zdroj: vlastní zpracování, 2018

V této kategorii jsou využívány obdobné postupy jako při registraci, hlavní rozdíl je patrný v případě vkládání obrázku. Pole pro vložení obrázku je ve formuláři zobrazeno prostřednictvím HTML prvku pro vkládání souborů.

Obrázek č. 28: Pole pro vložení obrázku

```
<label for="photo">Obrázek</label>
    <div class="error">
        <?php echo form_error('file'); ?>
    </div>
    <input type="file" name="file" value="<?php echo set_value('file'); ?>">
    <br />
```

Zdroj: vlastní zpracování, 2018

Pro uložení obrázku po odeslání formuláře je nutné nejprve načíst a nastavit CI knihovnu upload. Je nastavena složka, do které bude obrázek uložen, v tomto případě se jedná o složku „./uploads/original/“, kam je možné uložit obrázek typu gif, jpg, jpeg a png v maximální velikosti až 10 Mb, kdy maximální výška a šířka mohou být až 10 000 px. Název obrázku po uložení je generován automaticky. Pokud jsou omezení, která jsou definována splněna, je obrázek uložen.

Obrázek č. 29: Definování obrázků

```
$config['upload_path']          = './uploads/original/';
$config['allowed_types']        = 'gif|jpg|jpeg|png';
$config['max_size']             = 10000000;
$config['max_width']            = 10000;
$config['max_height']           = 10000;
$config['encrypt_name']         = TRUE;

$this->load->library('upload', $config);
```

Zdroj: vlastní zpracování, 2018

Všechna pole musí být ve formuláři vyplněna, v opačném případě není možné formulář uložit. Pro titulek a textové pole jsou použity stejné postupy jako v případě registrace, pro obrázek je nastavení varovných zpráv zobrazeno níže.

Obrázek č. 30: Nastavení pravidel pro obrázek

```
if (!$this->upload->do_upload('file'))
{
    $this->form_validation->set_rules('file', 'Obrázek', 'required',
        array(
            'required' => 'Vložte %s.'
        )
    );
}
```

Zdroj: vlastní zpracování, 2018

Jestliže validace formuláře proběhne úspěšně, jsou data pro obrázek načtena z knihovny. Obrázek je následně možné upravit dle zadaných požadavků. Pro úpravu obrázků je zvolena knihovna GD2. Dále je definován konkrétní obrázek, který má být změněn a následně složka, do které bude uložen nový upravený obrázek, který bude mít zachovaný poměr stran a jeho výška bude 300 px, šířka pak 450 px. Ke zmenšení a oříznutí obrázku je použita knihovna image_lib. Data jsou následně uložena do databáze pomocí funkce set_news, která je uvedena v textu dále.

Obrázek č. 31: Změna velikosti obrázku

```
$imageData = $this->upload->data();

$configImage['image_library'] = 'gd2';
$configImage['source_image'] = './uploads/original/'.$imageData['file_name'];
$configImage['new_image'] = './uploads/thumb/'.$imageData['file_name'];
$configImage['create_thumb'] = FALSE;
$configImage['maintain_ratio'] = TRUE;
$configImage['width'] = 450;
$configImage['height'] = 300;

$dim = (intval($imageData["image_width"]) /
        intval($imageData["image_height"])) - ($configImage['width'] /
        $configImage['height']);
$configImage['master_dim'] = ($dim > 0)? "height" : "width";

$this->load->library('image_lib', $configImage);
$this->image_lib->resize();

$this->news_model->set_news($imageData);
$this->session->set_flashdata('msg', 'Článek byl úspěšně vytvořen.');
```

```
redirect('/novinky');
```

Zdroj: vlastní zpracování, 2018

Z titulku obrázku je vytvořena URL adresa, dle které je možné článek později na stránce zobrazit. Adresa je následně spolu s titulkem, textem a názvem obrázku uložena do databáze baka, tabulky news.

Obrázek č. 32: Uložení článku do databáze

```
public function set_news($filedata)
{
    $this->load->helper('url');

    $slug = urlencode(url_title($this->input->post('title'), 'dash', TRUE));

    $data = array(
        'title' => $this->input->post('title'),
        'slug' => $slug,
        'text' => $this->input->post('content'),
        'file' => $filedata['file_name']
    );

    return $this->db->insert('news', $data);
}
```

Zdroj: vlastní zpracování, 2018

10.4.2 Výpis článku

Pro výpis článků je v modelu definována funkce `get_news`. Pokud není definován slug, jsou vypsaný všechny články z databáze, výpis článků je na stránce následně seřazen od nejnovějšího po nejstarší. Pokud je slug definován, je zobrazen detail konkrétní novinky.

Obrázek č. 33: Výpis článků z databáze

```
public function get_news($slug = FALSE)
{
    if ($slug === FALSE)
    {
        $query = $this->db->order_by("id", "desc")->get('news');
        return $query->result_array();
    }

    $query = $this->db->get_where('news', array('slug' => $slug));
    return $query->row_array();
}
```

Zdroj: vlastní zpracování, 2018

Seznam všech článků je zobrazen na stránce „/novinky“ (případně „/news“) prostřednictvím příkazu foreach, kdy je pro každý článek zobrazena fotografie a titulek. Detail článku je možné zobrazit pomocí odkazu Číst.

Obrázek č. 34: Zobrazení seznamu článků na stránce

```
<?php foreach ($news as $news_item): ?>
    <div class="news-item">
        <div class="news-item-img">
            
        </div>
        <div class="news-item-text">
            <h2><?php echo $news_item['title']; ?></h2>
            <a href="<?php echo site_url('novinky/'.$news_item['slug']); ?>">Číst</a>
        </div>
    </div>
<?php endforeach; ?>
```

Zdroj: vlastní zpracování, 2018

Konkrétní článek lze zobrazit na adrese „novinky/[slug]“, to je nastaveno v souboru routes.php.

Obrázek č. 35: Definování cesty konkrétního článku

```
$route['novinky/(:any)'] = 'news/view/$1';
```

Zdroj: vlastní zpracování, 2018

Detail článku je možné zobrazit díky definovanému slugu. Pokud je odeslán požadavek na zobrazení článku, který v databázi neexistuje, je zobrazena chyba 404, tedy stránka nenalezena.

Obrázek č. 36: Zobrazení chyby, pokud článek neexistuje

```
public function view($slug = NULL)
{
    $data['news_item'] = $this->news_model->get_news($slug);

    if (empty($data['news_item']))
    {
        show_404();
    }
    ...
}
```

Zdroj: vlastní zpracování, 2018

10.4.3 Úprava článku

V případě, že je potřeba článek upravit, je tak možné učinit opět prostřednictvím formuláře, který je dostupný na adrese „/novinky/edit/slug“.

Obrázek č. 37: Definování cesty pro úpravu konkrétního článku

```
$route['novinky/edit/{:any}'] = 'news/edit/$1';
```

Zdroj: vlastní zpracování, 2018

Editáční formulář obsahuje hodnoty titulku a textu, které jsou pro konkrétní článek uloženy v databázi. Výpis hodnoty pro textové pole Titulek je zobrazeno níže v obrázku č. 37.

Obrázek č. 38: Výpis hodnoty z databáze při editaci

```
<?php echo form_open_multipart('novinky/edit/'.$news_item['slug']); ?>

<label for="title">Titulek</label>
<div class="error"><?php echo form_error('title'); ?></div>
<input type="text" name="title"
    value="<?php echo $news_item['title'];?>"><br />
```

Zdroj: vlastní zpracování, 2018

Úprava článku probíhá z hlediska controlleru stejně, jako jeho vytvoření, pouze je pro práci s databází využíváno modelové funkce edit_news. Pokud je upraven titulek, slug zůstává stejný, aby bylo možné článek identifikovat v databázi. Titulek i text článku

se zobrazují v editačním formuláři, obrázek se však do formuláře negeneruje, proto pokud není vybrán nový obrázek, zůstává ten původní.

Obrázek č. 39: Uložení upraveného článku do databáze

```
public function edit_news($slug = FALSE, $filedata)
{
    $this->load->helper('url');

    $data = array(
        'title' => $this->input->post('title'),
        'text' => $this->input->post('content')
    );

    if ($filedata != NULL) {
        $data['file'] = $filedata['file_name'];
    }

    $this->db->where('slug', $slug);
    return $this->db->update('news', $data);
}
```

Zdroj: vlastní zpracování, 2018

10.4.4 Smazání článku

Pokud je z jakéhokoliv důvodu nutné článek smazat, jeho odstranění z databáze zabezpečuje funkce `delete_news`, která je uložena v modelu. Článek je konkretizován dle slugu.

Obrázek č. 40: Odstranění článku z databáze

```
public function delete_news($slug = FALSE)
{
    if ($slug != NULL){
        $a = $this->db->get_where('news', array('slug' => $slug))
            ->row_array();
        $this->db->where('slug', $slug);
        $this->db->delete('news');
        return $a;
    }
}
```

Zdroj: vlastní zpracování, 2018

10.5 Naše nabídka

V případě vytváření a úpravy položek nabídky, je, stejně jako u editace článků, vyžadováno přihlášení. Postup implementace byl stejný jako v předchozí kapitole.

10.5.1 Vytvoření nabídky

Pro snadné vytvoření nové položky nabídky, byl opět vytvořený formulář, který je dostupný na adrese „/nase-nabidka/create“. Pro jednotlivé položky menu je nutné zadat jejich typ (káva, dortíčky, bagetky), název, krátký popis, obrázek a cenu. Pro krátký popis a jeho snadnou úpravu je znovu využito CKEditoru. Aby nebylo možné vyplnit typ nabídky, který není zadán v systému, byl implementován rozbalovací seznam, který obsahuje pouze umožněné hodnoty (Káva, Dortíčky, Bagetky).

Obrázek č. 41: Rozbalovací seznam

```
<label for="type">Druh</label>
  <div class="error"><?php echo form_error('type'); ?></div>
  <select name="type">
    <option value="1">Káva</option>
    <option value="2">Dortíčky</option>
    <option value="3">Bagetky</option>
  </select>
```

Zdroj: vlastní zpracování, 2018

Všechna pole formuláře musí být vyplněna, v opačném případě není proveden zápis do databáze a jsou zobrazena chybová hlášení. Postup pro úpravu velikosti obrázku a následně samotného vytvoření položky je obdobný jako vytvoření článku, viz kapitola číslo 10.4.1.

10.5.2 Výpis nabídky

Samotný výpis položek z databáze je proveden stejně, jako při výpisu článků, tentokrát jsou dílčí položky identifikovány dle id. Jednotlivé položky menu jsou zobrazeny na stránce „nase-nabidka“ a jsou seříděné dle typů. Jako první je zobrazena nabídka kávy, následuje nabídka dortů a za nimi nabídka baget a sendvičů. Typ je přiřazen dle hodnot 1-3.

Obrázek č. 42: Definování typu

```
foreach($data['offer'] as $offer_item)
{
    if ($offer_item['type'] == 1){
        array_push($data['kava'], $offer_item);
    }
    else if ($offer_item['type'] == 2){
        array_push($data['dorticky'], $offer_item);
    }
    else if ($offer_item['type'] == 3){
        array_push($data['bagetky'], $offer_item);
    }
}
```

Zdroj: vlastní zpracování, 2018

Jak již bylo zmíněno výše, jednotlivé položky menu jsou řazeny dle definovaného typu a každá sekce je následně uspořádána. V každé sekci jsou položky řazeny na přeskáčku nejprve vlevo a následně vpravo, aby toho bylo dosaženo, je dle počtu položek automaticky přidělena třída, která je přes CSS nastýlována.

Obrázek č. 43: Automatické přidělení třídy a následný výpis

```
<?php $count = 0; ?>
    <?php foreach ($kava as $offer_item):
    echo '<div class="' . (++$count%2 ? "left" : "right") . "'>'; ?>
    
        <div class="text">
            <h3><?php echo $offer_item['title']; ?></h3>
            <p><?php echo $offer_item['text']; ?></p>
            <p><?php echo $offer_item['price'];?></p>
        </div>
    </div>
<?php endforeach; ?>
```

Zdroj: vlastní zpracování, 2018

10.5.3 Úprava nabídky

Pokud položka obsahuje chyby nebo dojde ke změně například v ceně, je vyžádována možnost úpravy. Položky jsou identifikovány dle svých id, u každé položky je po přihlášení uživatele zobrazen odkaz, který vede na editační formulář, který je již opět předvyplněný. Princip pro úpravu položky nabídky je obdobný jako úprava článku, viz podkapitola číslo 10.4.3.

10.5.4 Smazání nabídky

Jestliže je třeba některou z položek odstranit, je to velice jednoduché. Základ opět zůstává naprosto stejný jako je tomu při odstraňování článku, jediný rozdíl je při identifikaci, kdy článek je definován prostřednictvím slugu, kdežto položka nabídky je definována prostřednictvím id (zobrazeno v kapitole č. 10.4.4).

10.6 Zabezpečení

Jelikož se jedná o editovatelný webový informační systém, do kterého se uživatel může přihlásit, je zde mnoho potenciálních hrozeb z hlediska proniknutí do systému. Jsou ošetřena rizika změny celé databáze, ukládání hesla a HTTPS zabezpečení.

10.6.1 MySQL injection

MySQL injection využívá provádění škodlivého kódu na serveru. Takový kód může být vložen do formuláře před jeho odesláním, čímž systém namísto hodnot přijme kód, který

by se v případě neošetření problému provedl. Takový kód může například smazat celé tabulky nebo dokonce celé databáze. Problém je ošetřen použitím vnitřních funkcí CI, které takovému útoku zabraňují. Mezi takové funkce patří funkce `$this->input->post()` pro manipulaci s daty přijatých z formuláře nebo Active Record pro manipulaci s databází.

10.6.2 Hashování hesla

Při ukládání hesla do databáze je heslo nejprve zahashováno, aby konkrétní hesla nebyla přímo viditelná a nemohlo tak dojít k jejich zneužití. K tomuto účelu je definován řetězec v souboru `config.php`, který obsahuje náhodnou posloupnost znaků.

Obrázek č. 44: Nastavení náhodné posloupnosti znaků

```
$config['encryption_key'] = 'sdfghvjbknmlnbhjfhdrtzzfgjbhviohugzt';
```

Zdroj: vlastní zpracování, 2018

Tento řetězec je dále využíván v modelu `User_model.php` ve funkci `hash_password` jako klíč. Heslo je zahashováno postupně celkem 3x v různých kombinacích. Tímto způsobem je zajištěno, že ze zahashovaného hesla již není možné získat heslo původní. Pokud je však potřeba porovnat znovu zadané heslo (při přihlášení) je výsledný hash vždy stejný.

Obrázek č. 45: Hashování hesla

```
public function hash_password($password)
{
    $key = $this->config->item('encryption_key');
    $salt1 = hash('sha512', $key . $password);
    $salt2 = hash('sha512', $password . $key);
    $hashed_password = hash('sha512', $salt1 . $password . $salt2);

    return $hashed_password;
}
```

Zdroj: vlastní zpracování, 2018

10.6.3 HTTPS

Při zpřístupnění stránek veřejnosti musí být komunikace v síti zabezpečená. K tomu slouží právě Hypertext Transfer Protocol Secure. HTTPS je založený na certifikátech,

které poskytuje certifikační autorita, to znamená, že je zdroj ověřený a je možné mu důvěřovat. HTTPS je nastavován přímo na serveru mimo informační systém. Po tomto nastavení není možné komunikaci mezi prohlížečem a serverem přechít, data jsou tak zabezpečena.

11 Funkcionality systému

V této kapitole jsou uvedené jednotlivé funkcionality systému z pohledu uživatele. Kapitola je rozdělena celkem do sedmi částí, jejichž obsahem je registrace nového uživatele, přihlášení uživatele, menu, stránky O nás, Naše nabídka, Novinky a Kontakt.

11.1 Registrace nového uživatele

Jelikož je administrativní část určena pouze pro zaměstnance kavárny, je nutné zaměstnance do systému zaregistrovat. Pokud by byl registrační formulář volně přístupný, hrozilo by riziko, že se zaregistrují i osoby, které v podniku nepracují, proto je povolena registrace nového uživatele pouze administrátorovi, který byl v systému již vytvořen. Přihlašovací údaje pro uživatele admin jsou známy pouze vedoucímu pracovníkovi (přihlašovací jméno: admin, heslo: admincafe). Po přihlášení se, se administrátorovi zobrazí odkaz, který zobrazuje registrační formulář. Pokud se osoba bez potřebného oprávnění pokusí registrační formulář zobrazit zadáním URL adresy přímo do adresní řádky, je adresa přesměrována na přihlašovací formulář a uživateli není přístup umožněn.

11.1.1 Registrační formulář

Při registraci nového uživatele, je nutné zadat jeho přihlašovací jméno, jméno, příjmení, email, heslo a pro kontrolu heslo znovu. Pokud uživatel při registraci nesplní požadavky, například zvolí nevhodné přihlašovací jméno, je na tuto skutečnost upozorněn. Zadané údaje však nejsou smazány, zůstávají vyplněné a uživatel pouze opraví chybu. Výjimkou je vyplněné heslo, to je z důvodu bezpečnosti smazáno a uživatel jej musí zadat znovu.

Přihlašovací jméno

Přihlašovací jméno každého nově registrovaného uživatele musí být v rozsahu 5-12 znaků. Pokud není dodržen rozsah, je zobrazena informace o minimální, případně maximální délce uživatelského jména. Zároveň musí být jméno unikátní a nesmí se již v databázi registrovaných uživatelů objevovat. Pokud je tato skutečnost porušena, je uživatel upozorněn, že uživatelské jméno již existuje a musí zvolit nové. V případě nevyplnění tohoto pole, není možné registraci dokončit.

Jméno a příjmení

V průběhu registrace je každý registrující se uživatel povinen zadat své jméno a příjmení. Pokud tak neučiní, není možné registraci provést. Na tuto skutečnost je po pokusu o odeslání formuláře upozorněn.

Email

Email je nutné pro registraci vyplnit také. Je kontrolováno, zda je ve správném tvaru, tudíž zda obsahuje uživatelské jméno, tedy text, následně znak @ a poté internetovou doménu včetně tečky.

Heslo

Aby se uživatel mohl přihlásit, je nutné zvolit vhodné heslo. Minimální délka hesla je osm znaků, pokud zaměstnanec tuto podmínku nedodrží, je opět upozorněn, aby vymyslel heslo nové. Jelikož text není z důvodu bezpečnosti při nastavování viditelný, uživatel musí heslo vyplnit 2x, následně je provedena kontrola, zda se obě pole s hesly shodují. Pokud se tato pole neshodují, je nutné vyplnit je obě znovu.

11.1.2 Uložení uživatele do databáze

Pokud registrace proběhne v pořádku, je nová osoba uložena do databáze uživatelů. Veškeré údaje, kromě hesla, jsou do databáze zapsány ve stejném tvaru jako byly zadány do formuláře. Vzhledem k bezpečnosti není možné do databáze ukládat hesla v původní podobě, proto je heslo nejdříve zašifrováno a do databáze uloženo až následně, viz kapitola 10.6 Zabezpečení.

11.2 Přihlášení uživatele

Registrovanému uživateli je umožněno přihlášení se prostřednictvím přihlašovacího formuláře. Uživateli je umožněn přístup do administrativní části, pokud zadá správné jméno a heslo. Pokud uživatelské jméno není uloženo v databázi nebo heslo není zadáno správně, uživateli není umožněn přístup a je zobrazena hláška “Špatně zadané uživatelské jméno nebo heslo”.

Obrázek č. 46: Úspěšně přihlášený uživatel



Zdroj: vlastní zpracování, 2018

11.3 Menu

Pro snadnou navigaci na stránce je uzpůsobené menu na horním okraji stránky, jehož prostřednictvím je uživateli umožněn pohyb mezi jednotlivými stránkami. Menu obsahuje celkem 4 položky, které uživatele přesměrují na požadovanou stránku. Jsou to O nás, Naše nabídka, Novinky a Kontakt. Pod touto nabídkou je vždy umístěn vlastní text stránky, který se liší dle aktivního odkazu.

Na první pohled je patrné, na které stránce se uživatel právě nachází. Je to zabezpečeno aktivním odkazem, který se od ostatních liší barevným podtržením textu.

Pokud je přihlášený uživatel nebo administrátor, je tato nabídka rozšířena ještě odkaz, který uživatele úspěšně odhlásí. V případě administrátora je zde umístěný i odkaz, který vede na registrační formulář.

Navigační menu je po rolování zafixováno v horní části obrazovky, uživatel tak k němu má stále přístup a prohlížení stránek je o mnoho jednodušší.

Obrázek č. 47: Menu



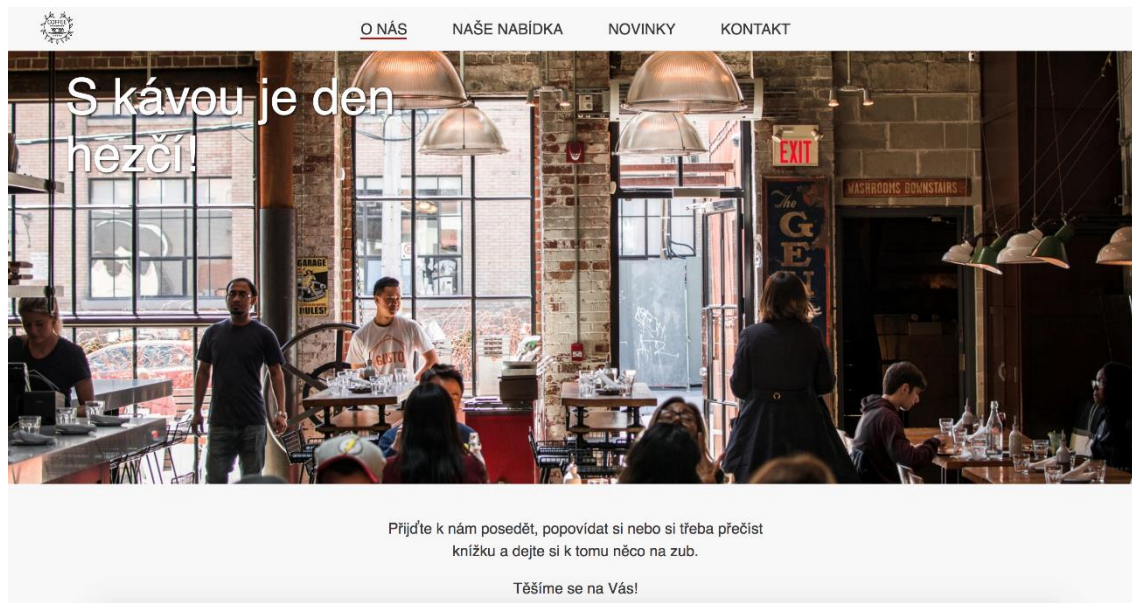
Zdroj: vlastní zpracování, 2018

11.4 O nás

Na hlavní stránce webových stránek je téměř přes celé okno umístěna fotografie kavárny, zákazníci si tak mohou udělat představu o tom, jak to v podniku přibližně vypadá.

Zároveň je zde viditelný slogan, který kavárnu vystihuje a krátký text, ve kterém jsou zákazníci vyzíváni k návštěvě.

Obrázek č. 48: Stránka O nás



Zdroj: vlastní zpracování, 2018

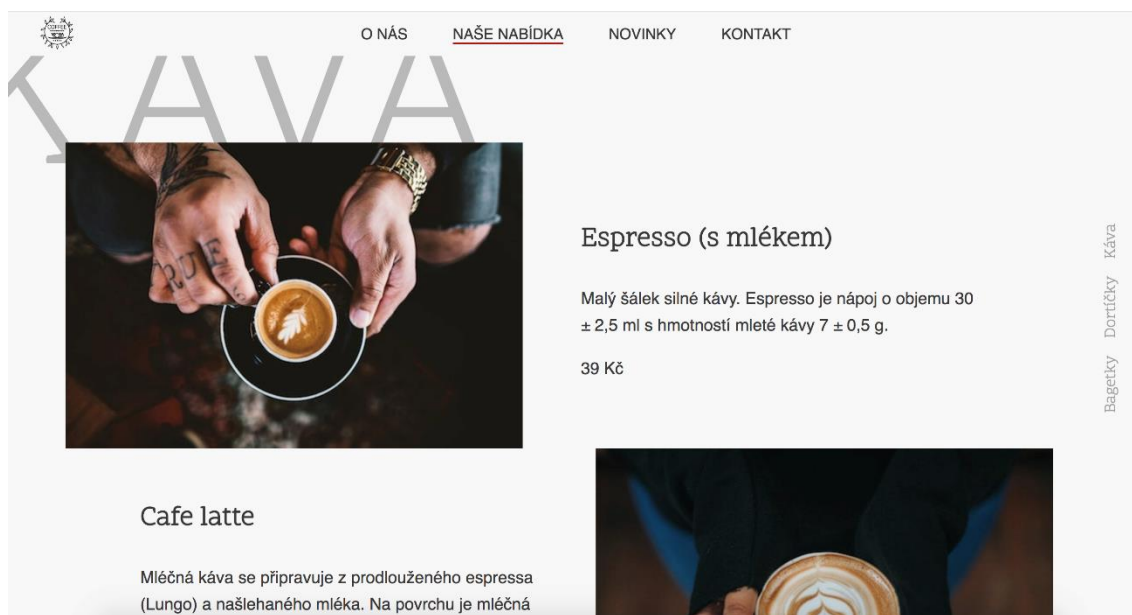
11.5 Naše nabídka

V sekci Naše nabídka je k dispozici pravidelně aktualizované menu, kde jsou uvedené i ceny a krátké popisy položek. Zákazník tak před příchodem do podniku ví, jaké občerstvení je v nabídce a do jaké cenové kategorie je podnik zařazen.

V nabídce je káva, dorty a bagety. Aby zákazník nemusel prohledávat celé menu, pokud se chce podívat pouze na nabídku baget, bylo k tomuto účelu vytvořeno navigační menu, které je umístěno na pravé straně obrazovky. Po kliknutí na jednotlivé názvy, je zobrazena přímo zvolená sekce.

Pokud je přihlášený administrativní pracovník či uživatel, může nabídky vytvářet, upravovat a mazat.

Obrázek č. 49: Stránka Naše nabídka



Zdroj: vlastní zpracování, 2018

11.5.1 Vytvoření nabídky

Jak již bylo zmíněno výše, vytvářet nové položky nabídky může pouze přihlášený uživatel. V horní části stránky je pro tento případ umístěné tlačítko, které odkazuje na formulář, který je využíván právě pro účely vytvoření nové položky.

Ve formuláři je nutné zvolit Druh nabídky, což je umožněno prostřednictvím rozbalovacího seznamu. Zde je možné volit mezi třemi typy, jimiž jsou káva, dortičky a bagetky. Dle tohoto výběru je nová položka nabídky zařazena do správné sekce v menu.

Dále je nutné zadat název položky, krátký popis a cenu. Pokud nejsou tato pole vyplněna, je uživatel upozorněn a položka není do menu přidána.

U každé položky je zobrazena fotografie produktu, tu je nutné zadat také. Fotografii je možné zadat v libovolné velikosti, pokud je obrázek větší, než je povolená velikost, je na vhodnou velikost nejdříve automaticky upraven.

Pokud jsou všechny položky vyplněné a proběhne vytvoření nové položky úspěšně, je proveden zápis do databáze a následně je nová položka zobrazena na stránce.

11.5.2 Editace nabídky

Pokud dojde ke změně například ceny, je možné položku jednoduše aktualizovat. U každé položky je totiž pro přihlášeného uživatele viditelný odkaz, který tuto činnost umožňuje.

Po kliknutí na tento odkaz je uživatel přesměrován na formulář, který obsahuje vyplněné údaje, které byly zobrazené na stránce. Po úpravě údajů a odeslání formuláře, jsou data automaticky upravena v databázi, následně je možné pozorovat změnu na stránce Naše nabídka.

11.5.3 Smazání nabídky

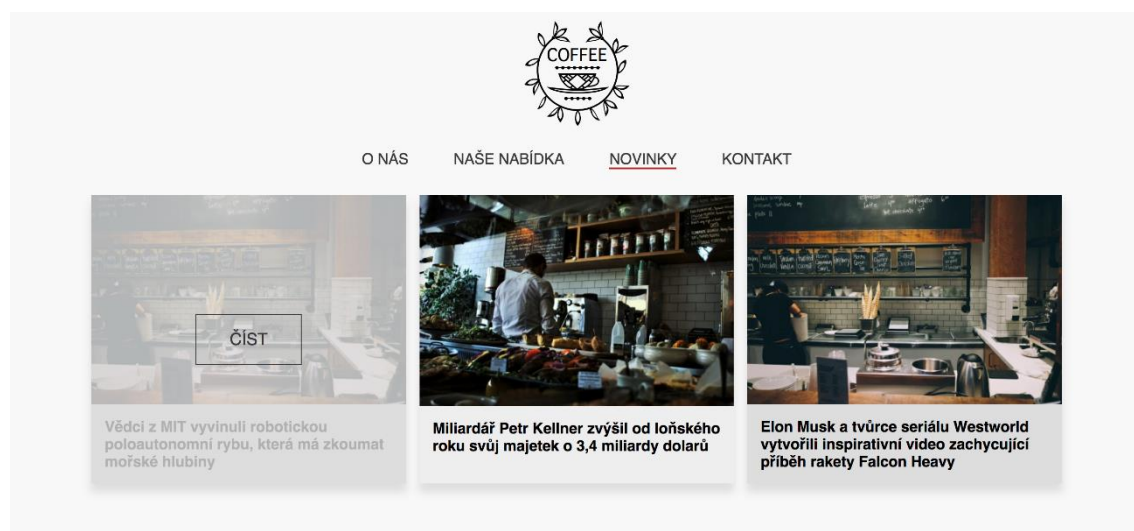
V případě, že některá položka menu již není v nabídce, je možné ji smazat. Pro tento účel je u každé položky umístěný odkaz, po jehož kliknutí, je položka odstraněna. Tato funkce je opět umožněna pouze přihlášeným uživatelům.

11.6 Novinky

Stránka Novinky obsahuje seznam všech vytvořených článků. Články jsou na řádku vždy celkem 3 a každý z nich je umístěn do boxu, jehož výška je nastavována automaticky dle délky nadpisu. Novinky jsou seřazené od nejnovější po nejstarší, čtenář má tak vždy přehled o nově přidaných článcích.

Na této stránce je zobrazena fotografie a již zmíněný titulek. Po najetí myši na jednotlivé položky, je zobrazeno tlačítko “Číst”, po kliknutí na toto tlačítko, je uživatel přesměrován na novou URL adresu, kde si může celý článek přečíst.

Obrázek č. 50: Stránka Novinky



Zdroj: vlastní zpracování, 2018

11.6.1 Vytvoření novinky

Odkaz pro vytvoření nové novinky je přímo na hlavní stránce Novinky, po kliknutí na tento odkaz je zobrazen formulář, prostřednictvím kterého je možné článek vytvořit. Tato funkce je opět zpřístupněna pouze přihlášeným uživatelům.

Pro vytvoření nového článku je nutné zadat jeho titulek. Titulek článku je velice důležitý, protože v seznamu článků vidí čtenář pouze ten a právě dle něho se rozhodne, zda si článek přečte nebo ne. Z toho důvodu musí být titulek zadán, pokud se stane, že uživatel toto pole nevyplní, není možné článek zveřejnit. Dle textu v titulku je vytvořena i URL adresa, která odkazuje na celý text článku.

Další důležitou položkou je text samotného článku. Text je možné libovolně upravovat v textovém editoru, je zde možné zvolit podnadpisy článku, napsat tučný text, vložit seznam, odkaz nebo například citaci.

V závěrečné fázi je nutné zvolit vhodný obrázek, který se, stejně jako titulek, zobrazuje v hlavním seznamu článků. Pokud je fotografie příliš velká, je automaticky uzpůsobena požadované velikosti. Obrázek je opět nutné zvolit, pokud tak není učiněno, článek opět není možné uložit.

Pokud jsou všechny položky vyplněné, je možné článek uložit do databáze a následně jej zveřejnit na webových stránkách.

Obrázek č. 51: Vytvoření novinky

Vytvořte nový článek

Titulek

Zajímavosti o kávě

Text

Choose heading **B** *I* [🔗](#) [☰](#) [☰](#) [””](#) [↶](#) [↷](#)

Pár zajímavostí o kávě

- káva je ovoce
- Brazílie je největším producentem kávy
- přibližně 100 šálků je smrtelný dávkou kofeinu
- v New Yorku se vypije asi 7x více kávy než ve zbytku světa
- káva byla původně potravina

Obrázek

Vybrat soubor photo-1486322147346-2b4d93ea6e6f.jpeg

Vložit příspěvek

Zdroj: vlastní zpracování, 2018

11.6.2 Editace novinky

V případě, že dojde k pochybení a článek obsahuje nesprávné informace, je možné jej upravit. V detailu každého článku je k tomuto účelu umístěný hypertextový odkaz, po kliknutí na něj je zobrazen formulář, který je pro editaci novinek určený.

Formulář určený pro editaci obsahuje úplně stejné položky jako formulář, který sloužil pro vytvoření článku. V této fázi je však formulář předvyplněný údaji, které jsou uloženy v databázi. Veškeré části musí být opět vyplněné, v opačném případě není možné článek změnit.

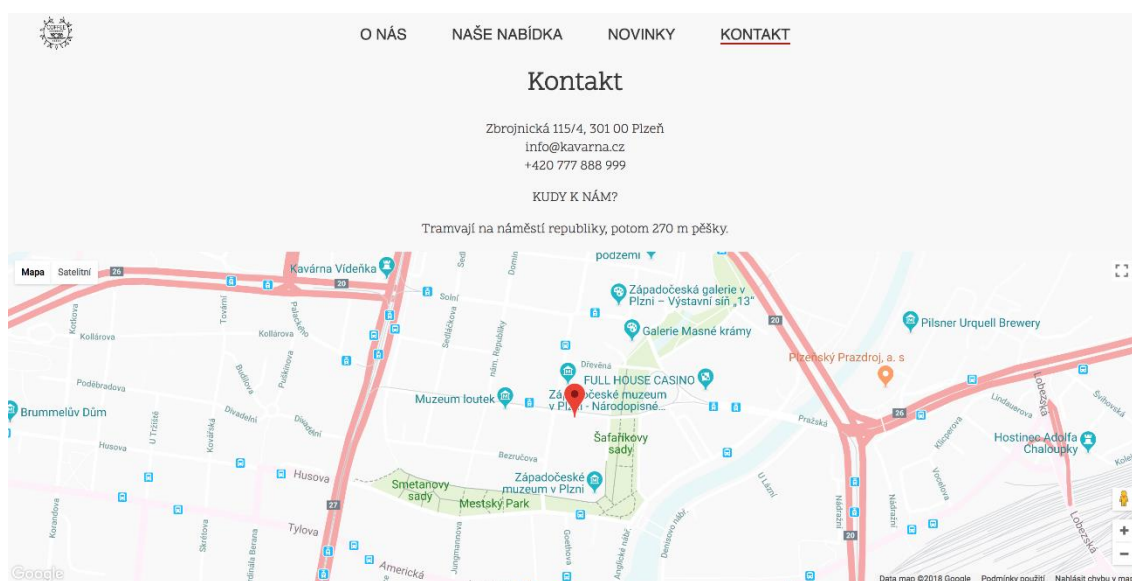
11.6.3 Smazání novinky

Pokud je z jakéhokoliv důvodu nutné článek smazat, je to možné opět přes odkaz, který se nachází v detailu každého článku. Novinka je odstraněna právě po kliknutí na tento odkaz. Tato změna je opět umožněna pouze přihlášenému uživateli.

11.7 Kontakt

Na stránce Kontakt jsou uvedené základní informace o kavárně. Zákazník zde nalezne adresu, kontaktní údaje, tedy telefonní číslo a email, dále návod, jak se na určené místo dostane a taky mapu. Mapa je nastýlována do barevného schéma, které ladí se zbytkem webových stránek.

Obrázek č. 52: Stránka Kontakt



Zdroj: vlastní zpracování, 2018

12 Testování

V závěrečné fázi bylo provedeno testování, aby bylo zjištěno, zda opravdu vše funguje tak, jak bylo na začátku vývoje definováno. Testování probíhalo v následujících krocích:

1. Přesměrování na přihlašovací formulář, pokud je vyžadováno zobrazení stránek pro vytvoření, úpravu nebo odstranění článku „/novinky/create“, „/novinky/edit/[slug]“, „/novinky/delete/[slug]“ nebo vytvoření, editaci a smazání položky nabídky „/nase-nabidka/create“, „/nase-nabidka/edit/[id]“, „/nase-nabidka/delete/[id]“, aniž by byl uživatel přihlášený
2. Přihlášení uživatele administrátor
3. Zobrazení registračního formuláře
4. Vytvoření nového uživatele – zápis do databáze
5. Odhlášení
6. Přihlášení nově vytvořeného uživatele
7. Vytvoření novinky s následným výpisem – výpis z databáze
8. Editace novinky
9. Odstranění novinky
10. Vytvoření nabídky s následným výpisem – výpis z databáze
11. Editace nabídky
12. Odstranění nabídky
13. Vložení nebezpečného kódu do formuláře pro vytvoření novinky

Jelikož bylo testování prováděno při vývoji průběžně, nebyly při konečném testování zjištěné žádné chyby, které by ohrožovaly používání systému.

13 Návrhy na zlepšení a inovaci

Požadavky na webové informační systémy se s rostoucími možnostmi neustále mění. Při dalším vývoji by mohl vzniknout požadavek na rezervační systém, protože když obsluha podniku musí při plně obsazené kavárně zvedat telefony, je zdržována od práce. V současné době je navíc tato funkce poměrně často zaváděna, jelikož mnoho lidí nerado telefonuje.

Dalším návrhem na zlepšení by mohly být editovatelné samotné sekce nabídky. Nyní nabídka obsahuje celkem tři části – káva, dortíčky a bagetky. Pokud podnik svou nabídku rozšíří, bylo by vhodné, aby bylo možné jednotlivé oddíly přidat, v krajním případě odstranit.

V současné chvíli, je nutné odstranit uživatele, který v podniku například již nepracuje, z databáze ručně. Při dalším používání systému by toto řešení nemuselo být dostačující a mohl by být vznesen požadavek na pouhé odstranění práv pro přihlášení tohoto uživatele.

Pokud je vytvořen nový článek, uživatel se tuto skutečnost dozví pouze v případě, že se na webové stránky kavárny podívá. Do budoucna by bylo vhodné umožnit registraci uživatelů k odběru nových článků

Závěr

Hlavním cílem práce bylo navrhnout a implementovat webový informační systém pro kavárnu, který bude umožňovat jednoduchou editaci obsahu.

Teoretická část práce charakterizuje webový informační systém jako takový, jsou zde také uvedené informace o tom, co by měl takový systém obsahovat. Další kapitola je zaměřena na databázovou vrstvu. V této části jsou blíže popsány databázový systém MySQL, databázová struktura a jsou vysvětleny základní SQL dotazy. Následně je vysvětlen rozdíl mezi statickými a dynamickými stránkami. V následující kapitole jsou popsány některé vybrané programovací jazyky a nakonec nástroje.

V druhé části práce bylo postupováno dle poznatků z teoretické části. Nejprve bylo nutné zvolit vhodný programovací jazyk a s tím i vhodný framework. Použití frameworku bylo zvoleno z důvodu, aby byl kód dobře organizovaný, udržitelný a znovu využitelný. Dalšími důvody bylo zabezpečení a také architektura MVC. Díky MVC je totiž možné kód udržovat separovaný a logicky seskupený, čímž je odstraněna duplicita kódu. V dalším kroku byl proveden návrh funkcionalit systému, což bylo velmi důležité pro samotný vývoj, který následoval.

V další fázi práce byl navržen samotný design, kdy bylo nutné vyřešit, zda se bude jednat o jednostránkový systém nebo o systém s klasickým menu, kdy bude uživatel přeměrováván na další stránky. V návaznosti na design byly vytvořené statické stránky, systém tak poměrně brzy získal svou podobu. Statické stránky byly později rozšířené o dynamickou část. Bylo tak postupně umožněno vytvoření článku, zobrazení článku, editace článku a odstranění. Tyto funkce byly poté implementovány i pro jednotlivé položky nabídky.

Poté, co bylo možné články a nabídku obsluhovat, byla implementována registrace uživatelů a následně i přihlášení. Když byl v systému definován přihlášený uživatel, byl umožněn přístup k funkcím, které nemohou být zpřístupněny veřejnosti, pouze přihlášeným uživatelům.

Systém byl po celou dobu vývoje průběžně testován, testování proběhlo vždy po implementaci každé nové funkce. Celkové testování proběhlo i po vývoji celého systému, poslední testování pak proběhlo po zveřejnění systému veřejnosti. Webový informační systém pro kavárnu je dostupný na adrese terlova.marcuk.cz.

Vzhledem k úspěšnému testování byl hlavní cíl práce splněn. Do systému je možné registrovat se a následně se přihlásit. Po přihlášení je možné obsah webových stránek upravovat, je tedy možné přidat nový článek a novou položku nabídky, dále je možné tyto položky editovat a také odstranit. Systém je propojený s databází a každá tato položka je v databázi uložena. Veškeré akce je možné provádět jednoduše přímo přes systém, kterému byl navržen design tak, aby byl zajímavý, ale zároveň funkční.

Seznam obrázků

Obrázek č. 1: SQL dotaz CREATE	12
Obrázek č. 2: SQL dotaz SELECT	12
Obrázek č. 3: SQL dotaz SELECT s definováním WHERE	12
Obrázek č. 4: SQL dotaz UPDATE	13
Obrázek č. 5: SQL dotaz DELETE.....	13
Obrázek č. 6: Zápis HTML.....	16
Obrázek č. 7: Zápis PHP.....	18
Obrázek č. 8: Zápis Python.....	18
Obrázek č. 9: Zápis Perl.....	19
Obrázek č. 10: Zápis JavaScript	19
Obrázek č. 11: MAMP – spuštění serveru	23
Obrázek č. 12: Vytvoření databáze.....	31
Obrázek č. 13: Připojení k databázi	32
Obrázek č. 14: Datový model	33
Obrázek č. 15: CSS pro stylování fotografie na hlavní stránce	34
Obrázek č. 16: Vytvoření pole pro přihlašovací jméno	35
Obrázek č. 17: Definování cesty pro registrační formulář.....	35
Obrázek č. 18: Načtení knihovny pro validaci formuláře.....	35
Obrázek č. 19: Nastavení pravidel pro přihlašovací jméno	36
Obrázek č. 20: Výpis stránky při úspěšné/neúspěšné registraci	37
Obrázek č. 21: Zápis uživatele do databáze.....	37
Obrázek č. 22: Nastavení session	38
Obrázek č. 23: Funkce login	38
Obrázek č. 24: Uložení do session.....	39
Obrázek č. 25: Zobrazení odkazu pro přihlášeného uživatele	39
Obrázek č. 26: Přesměrování nepřihlášeného uživatele	40
Obrázek č. 27: Implementace CKEditoru	40
Obrázek č. 28: Pole pro vložení obrázku.....	41
Obrázek č. 29: Definování obrázků	41
Obrázek č. 30: Nastavení pravidel pro obrázek	42
Obrázek č. 31: Změna velikosti obrázku	43

Obrázek č. 32: Uložení článku do databáze.....	44
Obrázek č. 33: Výpis článků z databáze	44
Obrázek č. 34: Zobrazení seznamu článků na stránce	45
Obrázek č. 35: Definování cesty konkrétního článku.....	45
Obrázek č. 36: Zobrazení chyby, pokud článek neexistuje	46
Obrázek č. 37: Definování cesty pro úpravu konkrétního článku	46
Obrázek č. 38: Výpis hodnoty z databáze při editaci	46
Obrázek č. 39: Uložení upraveného článku do databáze	47
Obrázek č. 40: Odstranění článku z databáze	48
Obrázek č. 41: Rozbalovací seznam	48
Obrázek č. 42: Definování typu.....	49
Obrázek č. 43: Automatické přidělení třídy a následný výpis.....	50
Obrázek č. 44: Nastavení náhodné posloupnosti znaků	51
Obrázek č. 45: Hashování hesla.....	51
Obrázek č. 46: Úspěšně přihlášený uživatel	55
Obrázek č. 47: Menu.....	55
Obrázek č. 48: Stránka O nás.....	56
Obrázek č. 49: Stránka Naše nabídka	57
Obrázek č. 50: Stránka Novinky.....	58
Obrázek č. 51: Vytvoření novinky.....	59
Obrázek č. 52: Stránka Kontakt.....	60

Seznam použitých zkratk

CDN	Content Delivery Network
CI	CodeIgniter
CRUD	Create-Read-Update-Delete
CSS	Cascading Style Sheet
HTML	Hypertext Markup Language
MAMP	Macintosh-Apache-MySQL-PHP
MVC	Model-View-Controller
PHP	Hypertext Preprocessor
SQL	Structured Query Language
WIS	Webový informační systém

Seznam použité literatury

1. 1KEYDATA. *Delete From Statement* [online]. 2018 [cit. 2018-02-10]. Dostupné z: <https://www.1keydata.com/sql/sqldelete.html>
2. CASTRO, Elizabeth a Bruce HYSLOP. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. Brno: Computer Press, 2012. ISBN 9788025137338.
3. ČÁPKA, David. *MySQL krok za krokem: Vytvoření databáze a tabulky* [online]. 2012 [cit. 2018-03-15]. Dostupné z: <https://www.itnetwork.cz/mysql/mysql-tutorial-vytvoreni-databaze-a-tabulky>
4. ČÁPKA, David. *Úvod do CSS (kaskádových stylů)* [online]. 2011 [cit. 2018-04-09]. Dostupné z: <https://www.itnetwork.cz/html-css/webove-stranky/jak-psat-moderni-web-html-tutorial-uvod-do-css/>
5. DOFACTORY. *SQL SELECT Statement* [online]. 2018 [cit. 2018-02-09]. Dostupné z: <http://www.dofactory.com/sql/select>
6. DVOŘÁK, Pavel. *Django: Úvod a instalace* [online]. 14.8.2009 [cit. 2018-03-16]. Dostupné z: <https://www.zdrojak.cz/clanky/django-uvod-a-instalace/>
7. GANGUR, Mikuláš. *Základy implementace webového informačního systému: praktický průvodce*. V Plzni: Západočeská univerzita, 2014. ISBN 978-80-261-0485-8
8. GASSTON, Peter. *Moderní web*. Přeložil Ondřej BAŠE. Brno: Computer Press, 2015. ISBN 9788025143452
9. GILFILLAN, Ian. *Myslíme v MySQL 4*. Praha: Grada, 2003. Knihovna programátora (Grada). ISBN 802470661x
10. HOLLOWAY, Ruth. *Taking a spin with Dancer, the lightweight Perl web application framework*[online]. 22.9.2015 [cit. 2018-03-16]. Dostupné z: <https://opensource.com/business/15/9/taking-spin-dancer-lightweight-perl-web-application-framework>
11. HOPKINS, Callum. *PHP okamžitě*. Brno: Computer Press, 2014. ISBN 978-80-251-4196-0

12. HORVÁTH, Tomáš. *Teoretický úvod do relačních databází* [online]. 8.11.2007 [cit. 2018-04-15]. Dostupné z: <http://programujte.com/clanek/2007110801-teoreticky-uvod-do-relacnich-databazi/>
13. HREBENAR, Jiří. *Úvod do JavaScriptu* [online]. 2018 [cit. 2018-04-16]. Dostupné z: <http://www.pestujemeweb.cz/obsah/javascript/javascript-uvod.php>
14. CHAPPLE, Mike. *What Is a Primary Key?* [online]. 14.3.2018 [cit. 2018-03-15]. Dostupné z: <https://www.lifewire.com/primary-key-definition-1019179>
15. CHRISTENSSON, Per. *HTML Definition* [online]. 23.5.2015 [cit. 2018-04-09]. Dostupné z: <https://techterms.com/definition/html>
16. KOTTNAUER, Jakub. *Jemný úvod do teorie ASP.NET MVC* [online]. 3.8.2009 [cit. 2018-03-16]. Dostupné z: <http://programujte.com/clanek/2009050200-jemny-uvod-do-teorie-asp-net-mvc/>
17. LACKO, Ľuboslav. *Mistrovství v SQL Server 2012: [kompletní průvodce databázového experta]*. Brno: Computer Press, 2013. ISBN 9788025137734.
18. LUTZ, Mark a David ASCHER. *Naučte se Python*. Praha: Grada, 2003. Pohotová příručka. ISBN 802470367x.
19. MCDUNNIGAN, Micah. *The Difference Between Dynamic & Static Web Pages* [online]. 2018 [cit. 2018-03-16]. Dostupné z: <http://smallbusiness.chron.com/difference-between-dynamic-static-pages-69951.html>
20. NGAN, Kayla. *What is Git?* [online]. 2018 [cit. 2018-02-16]. Dostupné z: <https://www.visualstudio.com/learn/what-is-git/>
21. PECK, Jon. *What is CodeIgniter and why should I use it?* [online]. 7.11.2013 [cit. 2018-04-15]. Dostupné z: <https://www.lynda.com/PHP-tutorials/What-CodeIgniter-why-should-I-use-it/126122/141742-4.html>
22. PLURALSIGHT. *Static and Dynamic Websites: There's a Difference* [online]. 20.9.2014 [cit. 2018-03-16]. Dostupné z: <https://www.pluralsight.com/blog/creative-professional/static-dynamic-websites-theres-difference>

23. RONGALA, Arvind. *Benefits of PHP Programming* [online]. 5.3.2015 [cit. 2018-04-13]. Dostupné z: <https://www.invensis.net/blog/it/benefits-php-programming/>
24. ROUBAL, Pavel. *Informatika a výpočetní technika pro střední školy: [kompletní látka pro nižší a vyšší úroveň státní maturity]*. Brno: Computer Press, 2010. ISBN 978-80-251-3227-2.
25. STEPHENS, Ryan K., Ronald R. PLEW a Arie JONES. *Naučte se SQL za 28 dní: [stačí hodina denně]*. Brno: Computer Press, 2010. ISBN 9788025127001.
26. STONES, Ben. *What is CodeIgniter, and how does it work?* [online]. 2.2.2012 [cit. 2018-04-11]. Dostupné z: <https://www.eukhost.com/blog/webhosting/what-is-codeigniter-and-how-does-it-work/>
27. ŠILHAVÝ, Radek. *Vybrané aspekty návrhu webových informačních systémů*. Vsetín: Šilhavý s. r. o. ISBN 978-80-904741-3-0
28. VÁCLAVÍK, Jiří. *Dávka teorie na úvod* [online]. 8.2.2005 [cit. 2018-04-16]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=675
29. VINCENT, William S. *Static vs Dynamic Websites: Pros and Cons* [online]. 26.10.2016 [cit. 2018-04-16]. Dostupné z: <https://wsvincent.com/static-vs-dynamic-websites-pros-and-cons/>
30. VRÁNA, Jakub. *1001 tipů a triků pro PHP*. Brno: Computer Press, 2010. ISBN 9788025129401.
31. W3SCHOOLS. *HTML Introduction* [online]. 2018 [cit. 2018-02-09]. Dostupné z: https://www.w3schools.com/html/html_intro.asp
32. ZAJÍC, Petr. *Seriál PHP: Princip PHP* [online]. 28.5.2004 [cit. 2018-04-09]. Dostupné z: <http://www.linuxsoft.cz/php/Serial-PHP.pdf>
33. ZEDNÍČEK, Jan. *Cizí klíč* [online]. 25.3.2017 [cit. 2018-04-15]. Dostupné z: <http://biportal.cz/BI-slovník/cizi-klic/>
34. ZEDNÍČEK, Jan. *Základy SQL – Nejpoužívanější příkazy* [online]. 29.3.2017 [cit. 2018-04-09]. Dostupné z: <http://biportal.cz/zaklady-sql-nejpouzivanejsi-prikazy/>

Seznam příloh

Příloha A	72
Příloha B	74

Příloha A – přestylování mapy

```
<script type="text/javascript" src="https://maps.googleapis.com/maps/api/js"></script>

<script type="text/javascript">

    google.maps.event.addDomListener(window, 'load', init);

    function init() {

        var mapOptions = {

            zoom: 16,

            center: new google.maps.LatLng(49.746214, 13.378727),
```

styles:

```
[{"featureType":"administrative","elementType":"labels.text.fill","stylers":[{"color":"#5d7e9e"}]},
{"featureType":"landscape","elementType":"all","stylers":[{"color":"#f2f2f2"},{"saturation":-
100},{"lightness":-
1}],{"featureType":"landscape","elementType":"geometry.fill","stylers":[{"color":"#f9f9f9"}]},{"
featureType":"poi","elementType":"all","stylers":[{"visibility":"off"}]},{"featureType":"poi.attraction","elementType":"labels","stylers":[{"visibility":"simplified"}]},{"featureType":"poi.business","e
lementType":"labels","stylers":[{"visibility":"simplified"}]},{"featureType":"poi.park","elementTy
pe":"geometry.fill","stylers":[{"color":"#e6f3d6"},{"visibility":"on"}]},{"featureType":"poi.park","e
lementType":"labels","stylers":[{"visibility":"simplified"}]},{"featureType":"road","elementType":
"all","stylers":[{"visibility":"simplified"},{"saturation":-
84},{"lightness":"0"}]},{"featureType":"road","elementType":"labels.text.fill","stylers":[{"saturati
on":"16"}]},{"featureType":"road.highway","elementType":"all","stylers":[{"visibility":"simplified
"}]},{"featureType":"road.highway","elementType":"geometry.fill","stylers":[{"visibility":"simplifi
ed"},{"color":"#f6a8a8"}]},{"featureType":"road.highway","elementType":"labels.text","stylers":[{"
color":"#454545"}]},{"featureType":"road.arterial","elementType":"geometry.fill","stylers":[{"c
olor":"#fff0fa"}]},{"featureType":"road.arterial","elementType":"labels.text.fill","stylers":[{"color
":"#777777"}]},{"featureType":"road.arterial","elementType":"labels.icon","stylers":[{"visibility":"
off"}]},{"featureType":"transit","elementType":"all","stylers":[{"visibility":"simplified"}]},{"featur
eType":"water","elementType":"all","stylers":[{"color":"#eaf6f8"},{"visibility":"on"}]},{"featureTy
pe":"water","elementType":"geometry.fill","stylers":[{"color":"#eaf6f8"}]}
}
```



```
var mapElement = document.getElementById('map');  
  
var map = new google.maps.Map(mapElement, mapOptions);  
  
var marker = new google.maps.Marker({  
  position: new google.maps.LatLng(49.746214, 13.378727),  
  map: map,  
  title: 'Snazzy!'  
});  
  
}  
</script>
```

Příloha B

- Příložené CD
 - zdrojový kód pro vytvoření webového informačního systému
 - zdrojový kód pro vytvoření databáze
 - soubor s hesly

Abstrakt

TEŘLOVÁ, Michaela. *Návrh a implementace webového informačního systému*. Plzeň, 2018. 76 s. Bakalářská práce. Západočeská univerzita v Plzni. Fakulta ekonomická.

Klíčová slova: webový informační systém, webové stránky, PHP, model, view, controller, design, CodeIgniter

Předložená bakalářská práce je zaměřena na návrh a implementaci webového informačního systému. Hlavním cílem práce je navrhnout a implementovat webový informační systém pro kavárnu, který bude umožňovat jednoduchou editaci obsahu. Práce je rozdělena na dvě části, teoretickou a praktickou. V teoretické části je charakterizován webový informační systém, následně je popsána databázová vrstva, rozdíl mezi statickými a dynamickými stránkami. Dále jsou definovány programovací jazyky, ze kterých je následně vybrán jeden konkrétní, který je pro vývoj systému používán. S výběrem programovacího jazyka souvisí i následný výběr nástrojů pro vývoj. Po této části jsou definovány funkcionality webového informačního systému, je navržen design a následuje popis samotného vývoje systému s použitím frameworku CodeIgniter. Následně jsou popsány funkcionality systému, kterých bylo dosaženo a v konečné fázi vývoje je provedeno testování. Na závěr jsou popsány návrhy na zlepšení a inovaci.

Abstract

TEŘLOVÁ, Michaela. *Design and implementation of a web-based information system*. Plzeň, 2018. 76 p. Bachelor Thesis. University of West Bohemia. Faculty of Economics.

Key words: web information system, website, PHP, model, view, controller, design, CodeIgniter

The submitted bachelor thesis is focused on the design and implementation of the web information system. The main goal of the thesis is to design and implement a web-based information system for a café, which will allow easy editing of the content. The thesis is divided into two parts, theoretical and practical. In the theoretical part, the web information system is characterized and the database layer and the difference between static and dynamic pages are described. Then the programming languages are defined, one of them is selected subsequently and then is used for the development of the system. The choice of programming language is also related to the subsequent selection of development tools. After this part, the functionalities of the web-based information system are defined, the design is designed and the development of the system using CodeIgniter framework is described. The functionalities of the system that have been achieved and the system testing are specified after these parts. In the end, suggestions for improvement and innovation are described.