



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**SYSTÉM PRO NABÍZENÍ DOMÁCÍCH PRODUKTŮ**

HOMEGROWN PRODUCT OFFERING SYSTEM

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**DANIEL DOUBEK**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. RADEK BURGET, Ph.D.**

**BRNO 2019**

## Zadání bakalářské práce



21817

Student: **Doubek Daniel**  
Program: Informační technologie  
Název: **Systém pro nabízení domácích produktů**  
**Homegrown Product Offering System**  
Kategorie: Informační systémy

Zadání:

1. Prostudujte aktuální technologie pro tvorbu informačních systémů s webovým rozhraním.
2. Seznamte se s existujícími platformami a způsoby sdílení informací o nabídkách zemědělských produktů mezi malopěstiteli a odběrateli.
3. Navrhněte systém pro nabízení domácích produktů umožňující příjem objednávek a rezervací a jejich vyřizování zahrnující podporu spolupráce s dalšími osobami.
4. Implementujte navržený systém pomocí vhodných webových technologií. Implementujte integraci nabídek s mapovými podklady a statistiky pro vyhodnocení úspěšnosti prodeje.
5. Proveďte testování vytvořeného systému.
6. Zhodnoťte dosažené výsledky.

Literatura:

- Gutmans, A., Rethans, D., Bakken, S.: Mistrovství v PHP 5, Computer Press, 2012
- Žára, O.: JavaScript - Programátorské techniky a webové technologie, Computer Press, 2015

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Burget Radek, Ing., Ph.D.**

Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 15. května 2019

Datum schválení: 16. října 2018

## Abstrakt

Tato práce se zabývá zejména návrhem, implementací a testováním informačního systému, který bude sloužit k prodeji domácích produktů. Je určen primárně pro malopěstitele a malochovatele, kterým vznikají přebytky a rádi by je nabídli dál. Zaměřil jsem se především na uživatelskou přívětivost, ale i na to, aby systém obsahoval takové funkce a možnosti, které usnadní nabídku a prodej těchto produktů.

Systém je implementován v PHP, ve frameworku Nette v kombinaci s MySQL, dále pak v HTML společně s knihovnou Bootstrap a v Javascriptové knihovně JQuery.

V rámci této práce se podařilo vytvořit uživatelsky přívětivý informační systém, který odpovídá požadavkům, které jsou na něj od uživatelů kladeny.

## Abstract

This thesis focuses on the design, implementation and testing of the information system which will be used to sell homemade products. It is intended primarily for smallholders and backyard keepers who are creating surpluses and would like to offer them. I focused especially on user-friendly design, but also on the system features and capabilities that make it easier to offer and sell these products.

This system is implemented in PHP, framework Nette combined with MySQL, HTML with Bootstrap library and in JavaScript's library JQuery.

As a part of this thesis I have created a user-friendly information system which meets the demands of users.

## Klíčová slova

informační systém, webová aplikace, PHP, Nette, HTML, CSS, Bootstrap, databáze, MySQL, inzerce produktů

## Keywords

information system, web application, PHP, Nette, HTML, CSS, Bootstrap, database, MySQL, offering products

## Citace

DOUBEK, Daniel. *Systém pro nabízení domácích produktů*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Radek Burget, Ph.D.

# **Systém pro nabízení domácích produktů**

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Radka Burgeta, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Daniel Doubek

10. května 2019

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Použité technologie</b>	<b>5</b>
2.1	Frontend . . . . .	5
2.2	Backend . . . . .	9
<b>3</b>	<b>Jiné vhodné technologie</b>	<b>13</b>
3.1	Laravel . . . . .	13
3.2	Symfony . . . . .	13
3.3	PostgreSQL . . . . .	14
3.4	MariaDB . . . . .	14
<b>4</b>	<b>Analýza požadavků</b>	<b>15</b>
4.1	Popis požadavků . . . . .	15
4.2	Konkurenční systémy . . . . .	17
<b>5</b>	<b>Uživatelské rozhraní</b>	<b>19</b>
5.1	Menu . . . . .	19
5.2	Hlavní stránka . . . . .	19
5.3	Přidání produktů . . . . .	20
5.4	Můj účet . . . . .	20
5.5	Statistiky . . . . .	20
5.6	Objednávka . . . . .	21
5.7	Uživatelská oprávnění . . . . .	21
5.8	Pravidla webu pro uživatele . . . . .	21
5.9	Propojený účet . . . . .	22
<b>6</b>	<b>Návrh informačního systému</b>	<b>23</b>
6.1	Diagram případů užití . . . . .	23
6.2	ER diagram . . . . .	23
<b>7</b>	<b>Implementace</b>	<b>28</b>
7.1	Struktura . . . . .	28
7.2	Přihlašování a registrace . . . . .	29
7.3	Hlavní strana . . . . .	30
7.4	Nový inzerát . . . . .	30
7.5	Můj účet . . . . .	32
7.6	ProductPresenter . . . . .	35

7.7	Nahlášené produkty, nahlášení uživatelé, nahlášené komentáře . . . . .	37
7.8	Statistiky . . . . .	37
7.9	Kooperace mezi účty . . . . .	38
7.10	Responzivita . . . . .	39
<b>8</b>	<b>Testování</b>	<b>40</b>
<b>9</b>	<b>Závěr</b>	<b>42</b>
	<b>Literatura</b>	<b>43</b>

# Kapitola 1

## Úvod

Hlavním cílem této bakalářské práce je vytvoření informačního systému nejen pro pěstitele či chovatele ze zahrádek a ohrádek, ale také pro obyčejné lidi. Někteří, a bohužel jich není málo, totiž nemají to štěstí a vlastní zahradu si nemohou finančně dovolit. Zahrada je pro mnohé místem, kde mohou pěstovat či chovat něco vlastního, a to převážně pro svůj vlastní užitek. Takoví lidé to dělají hlavně z toho důvodu, že jim není lhostejné jejich zdraví či zdraví někoho jim blízkého. I proto nechtějí dopustit, aby byli odkázáni na mnohdy nekvalitní produkty, které tu jsou běžně k dostání ve všech supermarketech. Někdy to dělají samozřejmě i proto, že je to pro ně finančně výhodnější, nicméně pravdou je, že vypěstovat si něco svého je z velké části případů mnohem nákladnější než za jakou cenu by sehnali daný produkt v obchodě. V dnešní době je stále narůstající poptávka po kvalitních potravinách, které neobsahují žádné přídatné škodlivé chemické látky.

Moje řešení propojuje lidi s domácími produkty s těmi, kteří takové produkty poptávají. Stránka bude sloužit jako zprostředkovatel kontaktů s možností o zboží projevit zájem a domluvit se s prodejcem na dalších podrobnostech. Lidé dostanou přímý kontakt s daným farmářem a nebudou tak muset využívat služeb překupníků.

Většina aktuálně fungujících webů specializovaných na prodej domácích produktů neumožňuje přímý kontakt s prodejcem. Na těchto webových stránkách jsou většinou pouze překupníci, kteří objíždí farmáře. Od nich berou produkty a ty nabízí v bedýnce za jednotnou, předem určenou, cenu. Mnohdy tedy člověk neví, odkud zboží pochází, v jakých podmínkách vyrůstalo či jakým způsobem bylo vytvořeno.

Je však nutno zmínit, že i dnes existuje místo, kde lze sehnat domácí produkty přímo od farmářů. Tím místem je Facebook, kde vznikají skupiny se zaměřením na domácí produkty bez chemie. Jednou takovou skupinou je pro Brno například „Domácí produkty Brno a okolí“. Problém však je, že ne každý má dnes Facebook a ne každý si ho chce kvůli jedné skupině zakládat.

I z toho důvodu jsem se rozhodl zpracovat systém jako webové plně responzivní rozhraní, které tak bude přístupné všem, kteří mají připojení k internetu. Práce využívá architektury klient/server. Je zaměřená na pomoc menším farmářům, kteří spadají především do 4. sektoru podnikatelů, čili takových podnikatelů, kteří nemusí za prodané zboží odvádět daně.

V kapitole 2 Vám představím použité technologie a prostředí samotné, ve kterém jsem aplikaci vyvíjel. Technologie blíže popíši a v kapitole 3 je porovnáím i s dalšími technologiemi, které by šly použít. Další kapitola (kapitola 4) se bude zabývat požadavky potencionálního uživatele a požadavky vycházející z konkurenčního, již existujícího, řešení. Další část práce (kapitola 6) je samotný návrh systému, kde budou popsány nejen vazby a vzájemé vztahy komponent, ale také jejich grafické vyjádření v podobě ER-diagramu, diagramu případů

užití use-case a nebo také pár modelových příkladů použití aplikace. Následující kapitola (kapitola 7) je pak samotná implementace, která bude reprezentovat nejzajímavější funkce aplikace a blíže je specifikovat do té míry, jakých prostředků bylo použito pro zprovoznění požadované funkcionality. Aby byla aplikace použitelná pro cílovou skupinu, je třeba ji řádně v průběhu implementace testovat, tomu se věnuje kapitola 8. Poslední kapitolou 9 je závěr, kde bude krátce shrnut výsledek práce, budou nastíněna možná rozšíření aplikace do budoucna a možné optimalizace.

Výsledkem práce by měla být uživatelsky přívětivá stránka, která bude přehledná i pro člověka, který počítače jinak běžně neovládá. Tato vlastnost je potřeba především proto, že uživatelé budou moci vytvářet nabídky produktů sami, a proto je důležité, aby jim žádný dílčí krok nečinil problémy.



## Kapitola 2

# Použité technologie

V této kapitole popíšeme technologie, které byly vybrány pro práci k tvorbě tohoto informačního systému.

Každá dílčí technologie bude charakterizována, aby bylo jasné, jakého významu v systému nabývá a jaký účel přesně plní.

Kapitola je rozdělena pro větší přehlednost na dvě části – na technologie použité na frontendu<sup>1</sup> a na technologie použité na backendu<sup>2</sup>.

### 2.1 Frontend

V následující části budou popsány jednotlivé technologie, které se zpravidla podílejí na implementaci uživatelského rozhraní, tedy takového rozhraní, které je veřejně viditelné.

#### 2.1.1 HTML

HTML neboli hypertextový značkovací jazyk (Hypertext markup Language), umožňuje vytváření webových stránek, které si pak může zobrazit prakticky kdokoli, kdo je připojen k internetu. První verze HTML 1.0 byla představena v roce 1991. Autory byli Tim Berners a Robert Caillau. Nejnovější verzí pro rok 2019 je verze HTML 5.3.

Používá předdefinované tagy (značky) pro vytváření základní kostry celého webu. Každý tag je označen pomocí znaků menšího (zleva) a většího (zprava). Tyto tagy nejsou pro koncového uživatele viditelné, slouží pouze k vytvoření základní struktury a rozložení jednotlivých prvků na webu.

Správnost syntaxe je možné ověřit například na stránce organizace W3C<sup>3</sup>, což je organizace, která je pověřena navrhováním a údržbou jazyka.

Mezi základní elementy (tagy) patří `<h1>...<h6>`, které jsou určeny především pro nadpisy. Dále mezi základní tagy patří `<span>` a `<div>`. Tag `<div>` je tzv. blokový tag, který se roztahuje na celou šířku elementu, ve kterém se nachází. Oproti tomu tag `<span>`, což je tzv. inlinový tag, je vždycky tak široký, jak široký je obsah v něm. Pro text je určen tag `<p>`, pro tučný text `<b>` či `<strong>`. Na tabulky je využíván tag `<table>`, který v sobě může nést další tagy jako `<thead>`, což je hlavička tabulky (typicky nadpisy jednotlivých sloupců) nebo `<tbody>`, který zahrnuje hlavní obsah tabulky. Pro odrážkový

<sup>1</sup>Pojem frontend slouží k označení části webu, která je viditelná běžným návštěvníkům.

<sup>2</sup>Pojem backend se používá pro tu část webové aplikace, která slouží k administraci webu a ke zpracování dat.

<sup>3</sup>Dostupné na: <https://www.w3.org/>

seznam je určen tag `<ul>`, naopak pro číselný seznam je tag `<ol>`. Pro jednotlivé položky seznamu se využívá tagu `<li>`. Pro vytvoření odkazu se použije tag `<a>`. Na formuláře jsou určeny vstupy, tzv. inputy, které se značí jako tag `<input>`. Pokud je potřeba, aby v nich šla zapisovat pouze čísla, text, hesla atd., přidává se k inputu parametr „type“, například `<input type="number">`. Pro založení webové stránky je potřeba deklarovat typ dokumentu pomocí tagu `<!DOCTYPE html>`, poté následuje párový tag `<html>`, který zaobaluje celý dokument. Další značkou je pak tzv. hlavička, která se značí jako `<head>`. Do ní patří všechna metadata, odkazy na externí soubory, formát kódování, název stránky atd. Po uzavření hlavičky `</head>` následuje značka `<body>` neboli tělo dokumentu. Ten již obsahuje veškerý informační obsah dané stránky.

Samotný HTML se dnes ale nepoužívá, protože v současně velkém konkurenčním prostředí je důraz na zaujetí návštěvníka webu, takže se klade velký důraz na grafické zobrazení webu. Pro dosažení určitého stylu či formátu využíváme tzv. kaskádové styly, více v kapitole 2.1.2.

## 2.1.2 CSS

Jak již bylo nastíněno v předchozí podkapitole, pokud chceme dodat jednotlivým HTML tagům vzhled, jako je barva či rozmístění daného prvku, využijeme k tomu právě CSS, což je anglická zkratka: „Cascading Style Sheets“, v překladu do českého jazyka: „kaskádové styly“. První verze byla představena v roce 1996. V současnosti je nejnovější verzí CSS3.

Prvky na stránce se rozdělují buď podle identifikátoru ID či třídy („class“). ID se používá v takových případech, kdy je jasné, že se daný prvek se vyskytne v celé práci jen jednou. ID totiž musí být vždy unikátní v rámci celého HTML dokumentu. Pokud bychom měli na stránce více stejných ID, jednalo by se o chybu a dané prvky by nemusely fungovat správně.

Pokud je požadováno, aby prvky dědily vzhledové vlastnosti, musí se označit daný prvek v HTML pomocí třídy, která se používá pro takové elementy, které se v dokumentu vyskytují vícekrát. Pokud chceme linkovat třídu, musíme vždy v CSS před její název napsat tečku, například `.navez_tridy`. Pokud chceme linkovat ID, píše se před název mřížka: `#unikatni_prvek`.

Dále je možné jednotlivé prvky také tzv. absolutně adresovat. Toho docílíme například tím způsobem, že udáme přímo ten konkrétní tag, pro který chceme stejné vlastnosti (`p font-size: 12px;`). Tímto zápisem bude docíleno toho, že všechny prvky, které budou obaleny do tagu `p`, budou mít velikost písma 12 pixelů. V adresování prvků se také můžeme setkat se znaky jako je „>“ či „+“.

```
<div class="link">
  <p>Toto</p>
  <p>Je</p>
</div>
<p>Ukázka</p>
```

Pokud bychom chtěli ve výše uvedené ukázce linkovat pouze první tag `p` v třídě `link`, stačí k tomu zápis:

```
link > p {libovolná css vlastnost}
```

Pokud bychom chtěli linkovat první tag `p` za divem s třídou `link`, použijeme zápis:

```
link + p {libovolná css vlastnost}
```

Základní forma syntaxe deklarace je znázorněna na obrázku 2.1. Jako selektor je označen prvek, kterému chceme změnit nebo přidat nějakou vlastnost. Selektor může být označen pomocí ID, třídy nebo to může být obecně jakýkoliv HTML element (např. „p“ pro text). Další částí deklarace v CSS je vlastnost a hodnota této vlastnosti. Vlastnosti jsou různé pro různé typy elementů. Jiné jsou např. pro font, text a jiné jsou pro tabulky či při pozicování. Hodnoty vlastnosti jsou předem dané, kodér či programátor vybírá jen tu hodnotu, kterou využije. Příkladem může být vlastnost `font-style`, jejíž možné hodnoty jsou: `normal`, `italic`, `oblique`. Jiná hodnota této konkrétní vlastnosti nebude platná. Vlastnost je od hodnoty vlastnosti oddělena dvojtečkou („:“) a mezerou.



Obrázek 2.1: CSS – ukázka deklarace. CSS zápis se skládá ze selektoru (na obrázku „p“), vlastnosti (na obrázku „color“) a hodnoty této vlastnosti (na obrázku „red“), (převzato a upraveno z [3]).

### 2.1.3 Font Awesome

Jedná se o rozšíření jazyka CSS, které vyvinul Dave Grandy v roce 2012. Díky němu můžeme do webu relativně snadno implementovat vektorové ikony, které jsou ve formátu SVG<sup>4</sup>. Toto rozšíření nám poskytuje výběr z téměř 4 845 nepoužívanějších ikon, které jsou navíc rozřazeny do specifických tematických kategorií. Ve své aplikaci používám aktuálně nejnovější verzi 5.8.1.

Hlavní výhodou tohoto rozšíření je, že se ikony chovají jako písmo. Díky tomu můžeme definovat jejich velikost pomocí `font-size`, či například barvu pomocí `color`.

Příklad zápisu ikonky telefonu:

```
<i class="fas fa-phone"></i>
```

Výsledek předchozího zápisu můžeme vidět na obrázku 2.2.

Nejčastější způsob implementace do webu je pomocí BootstrapCDN od MaxCDN. Díky tomu stačí pouze jeden řádek pro volání Font Awesome.

CDN neboli Content Delivery Network je distribuovaná síť serverů, díky které můžeme načítat CSS (více v kapitole 2.1.2), JavaScript (více v kapitole 2.2.4) atd. mnohem rychleji. Na tyto servery se stahuje obsah z původních serverů, přičemž se jejich obsah odesílá klientovi právě z toho místa, které je nejbližší jeho lokaci.

<sup>4</sup>SVG je značkovací jazyk a formát souboru, který popisuje dvojrozměrnou vektorovou grafiku pomocí XML.



Obrázek 2.2: Font Awesome – ukázka ikony

#### 2.1.4 Bootstrap

Bootstrap vyvinul Mark Otto a Jacob Thornton již v roce 2011. Nejnovější verze je s označením 4.3.1 a je z roku 2019. Tento nástroj lze označit jako sadu nástrojů, které usnadňují práci s frontendem.

Obsahuje předdefinované třídy CSS (více v kapitole 2.1.2), které usnadňují tvorbu layoutu, práci s typografií či vytváření základních uživatelských elementů. Za element můžeme považovat například jednoduché tlačítko, který se skládá pouze z HTML tagu `<button>` a CSS třídy `.btn`. Dále pak také modální okna, komponenty dropdown atd., které již také využívají JavaScript (více v kapitole 2.2.4). Další výhodou tohoto frameworku je, že za kódéra dokáže z větší části řešit responzivitu. Ve svém rozhraní totiž implementuje několik záchytných bodů pro nejtypičtější typy rozlišení.

Bootstrap využívá 12 sloupcový grid. Jednotlivé šířky sloupce definujeme pomocí CSS třídy `.col-xx-yy`. Za proměnnou „yy“ lze dosadit hodnoty od 1 do 12, kdy každé jednotlivé číslo značí, jakou část šířky stránky bude sloupec zabírat. Proměnná „xx“ může být nahrazena jednou z hodnot `lg`, `md` či `sm`. Hodnota `lg` je určena pro šířku 1200 px a větší, tedy pro větší zařízení, typicky počítače. Hodnota `md` je především pro větší tablety či menší notebooky s šířkou větší nebo rovnou 992 px, `sm` je určen pro menší zařízení, například menší tablet či větší telefon s šířkou 768 px nebo větší.

Šířka stránky, ze které se budou jednotlivé sloupce počítat, je definována buď CSS třídou `.container`, která je 1140 px široká s umístěním na střed stránky či třídou `.container-fluid`, která roztáhne šířku stránky na 100 % odsazenou vlastností „padding“ o 15 px z levé a pravé strany.

```
<div class="container">
  <div class="row">
    <div class="col-4"> <!-- 1/3 sirky --> </div>
    <div class="col-8"> <!-- 2/3 sirky --> </div>
  </div>
</div>
```

Nespornou výhodou Bootstrapu 4 je, že je vystavěn na flexboxu. *Flex v češtině znamená pružný, přizpůsobivý. Flexboxy jsou tedy pružné elementy layoutu. Jednou z hlavních předností flexboxu je totiž schopnost vyplňovat zbylý prostor bez nutnosti přepočítávání JavaScriptem [7].*

Bootstrap je rozhodně užitečný nástroj, který dokáže ušetřit spoustu času, protože se programátor nemusí zatěžovat věcmi, které již byly nespočetněkrát vytvořeny. Tento nástroj také přímo řeší spoustu věcí za uživatele, především JQuery nebo CSS, které se aplikuje pomocí jasně definovaných tříd. Jelikož je snadný na naučení, je využíván i širší

veřejností, která nemusí mít sebemenší předchozí zkušenosti s frontendem. Framework například nabízí předdefinovaná menu, včetně hamburgerového menu pro mobilní zařízení, což je komponenta, která se běžně vyskytuje snad na všech webových aplikacích.

## 2.2 Backend

V následující části budou popsány ty technologie, které slouží k administraci webu a k celkovému zpracování dat.

### 2.2.1 MySQL

MySQL je systém pro řízení databází („My Structured Query Language“) vlastněný firmou Oracle Corporation. První její vydání bylo již 23. května 1995. Aktuální verzí je verze 8.0.11, která byla vydána 19. října 2018. Jedná se o relační databázi typu DBMS<sup>5</sup>, která je využívána především pro webové aplikace. Databáze je jako celek založena na tabulkách, kde každá tabulka obsahuje položky jednoho typu. Položka se pak skládá z jednotlivých atributů.

V MySQL se setkáváme s pojmy jako je primární klíč, cizí klíč či jednotlivé akce databáze, jakými jsou například CASCADE či RESTRICT. Níže budou tyto pojmy vysvětleny.

**Primární klíč (PRIMARY KEY):** Každá tabulka v databázi musí obsahovat jedinečný a nezaměnitelný identifikátor, který nemůže nabýt hodnoty NULL. Tímto identifikátorem je primární klíč, který zajistí entitní integritu. Platí, že každá tabulka musí obsahovat maximálně jeden primární klíč. Je třeba si zvolit vhodný klíč, protože mezi jeho hlavní funkce patří rychlejší vyhledávání informací.

**Cizí klíč (FOREIGN KEY):** *Cizí klíč nám (spolu s primárním klíčem) zprostředkovává vazbu mezi tabulkami. Jeho hodnota pak odpovídá v jiné tabulce (která je odkazovaná) hodnotě primárního klíče [8].*

**Akce:** Cizí klíče umožňují vytvářet takzvané akce, které zajišťují, aby byla dodržena integrita databáze. Platí totiž, že *v databázi nesmí nikdy existovat žádné hodnoty cizího klíče, které nemají odpovídající hodnoty primárního klíče v relevantní báze tabulce [2].* Aby bylo toto zajištěno, použijí se 4 tzv. akce. Akce CASCADE, SET DEFAULT, SET NULL, RESTRICT, NO ACTION. V práci byly použity pouze akce RESTRICT a CASCADE.

**RESTRICT:** Pokud v dceřiné tabulce existují záznamy odkazující na cizí klíč, pak odpovídající záznam v rodičovské tabulce nepůjde změnit nebo smazat (SŘBD tomu zabrání).

**CASCADE:** Tato akce zajišťuje, aby se při změně dat v rodičovské tabulce změnila také data v dceřinné tabulce. Pro příklad, pokud bych změnil ID uživatele v tabulce user, změnilo by toto ID ve všech ostatních tabulkách, kde na tohoto uživatele odkazují, takže například v objednávkách, komentářích apod. Toto chování nám zajišťuje, že nedojde k nekonzistenci dat. V případě, kdy dojde k mazání dat v rodičovské tabulce pomocí klauzule ON DELETE, smažou se společně s ní všechny ostatní záznamy z dceřiných tabulek, které na tuto tabulku odkazovaly. V mé práci této akce využívám například v tom případě, kdy chce administrátor smazat uživatele. Uživatel může mít svoje objednávky, produkty či komentáře. To vše se smaže automaticky společně s uživatelem díky této akci.

MySQL vychází z deklarativního programovacího jazyka SQL. Jelikož je šířen jako open source, jedná se o jeden z nejvíce užívaných systémů [5]. Současně nejpoužívanějším nástrojem pro práci s MySQL se používá PHPMyAdmin. MySQL je v této práci využito především

---

<sup>5</sup>DBMS – database management system, česky „systém řízení báze dat“

kvůli rychlosti a spolehlivosti, kterou tato technologie rozhodně vyniká. Může se využít jak pro menší webová rozhraní až po rozsáhlé e-shopy.

Pro vytváření a testování informačního systému jsem využil nástroj XAMPP, který umožňuje vytvoření webového lokálního serveru. Zkratka XAMPP znamená: **X** - Cross-platform neboli multiplatformní, **A** - Apache, **M** - MariaDB, **P** - PHP, **P** - Perl.

### 2.2.2 PHP

PHP patří mezi zástupce skriptovacích jazyků. Skript je sada programovacích instrukcí, které jsou interpretovány za běhu. Účelem skriptů je obvykle zvýšení výkonu či provádění opakujících se úkolů v aplikaci. Zkratka PHP znamená „Hypertext Preprocessor“. Je určen převážně pro programování dynamických internetových stránek. Veškeré příkazy či komunikace s databází se provádí na straně serveru. Tento princip zajišťuje generování čistého HTML kódu. Před načtením webové stránky se totiž nejdříve vykonají všechny PHP příkazy a až poté se stránka zobrazí. Soubory, které obsahují nějaký PHP kód, je zvyklost ukládat s koncovkou „.php“. Od HTML se tento jazyk odděluje sekvencemi `<?php` a `?>`, přičemž každý PHP příkaz je ukončen středníkem. Proměnné se značí pomocí dolaru „\$“. Jelikož je PHP jazykem netypovým, proměnné nejsou definovány typem. Tento jazyk byl představen již roku 1995, existuje tedy proto již spoustu verzí této technologie. Nejnovější verzí je aktuálně 7.2.0.

V této práci bylo využito PHP především z důvodu široké základny uživatelů a s tím související podpory na webhostingových serverech, které PHP standardně podporují. Dále je pak výhodný fakt, že se jedná o skriptovací jazyk na straně serveru, což znamená, že jej stačí nainstalovat na server a klienti, kteří se pak k webové aplikaci připojují, již PHP nainstalované mít nemusí, pro zobrazení jim stačí pouhý webový prohlížeč. Dalším důvodem je podpora technologie MySQL (viz kapitola 2.2.1) a celková multiplatformita, což znamená, že můžeme aplikaci nasadit opravdu na téměř jakýkoliv běžně známý operační systém, jakým je Linux, Windows či Mac OS.

### 2.2.3 Nette Framework

Jedná se o český framework, kterého je převážným autorem programátor David Grudl. Nette je distribuován jako open-source pod licencí GNU GPL a licencí Nette, která nahrazuje původní BSD licenci. Díky tomu, že se jedná o „český produkt“, má poměrně silné zastoupení v naší zemi, kde je tak snadné získat odpověď či radu ke svému problému.

Implementace je založena na architektuře MVP<sup>6</sup>, kde:

**Model:** obsahuje samotnou logiku aplikace, převážně se jedná o SQL příkazy do databáze či výpočty. Model neví nic o tom, co se děje na výstupu, jeho úkolem je přijímat parametry, ty zpracovávat a výsledky vracet zase zpět.

**Views:** neboli pohledy jsou samotné HTML soubory, které se narozdíl od modelu přímo starají o zobrazení výstupu uživateli. Stejně jako model, ani pohled opět neví původ dat, stará se jen o správné zobrazení uživateli.

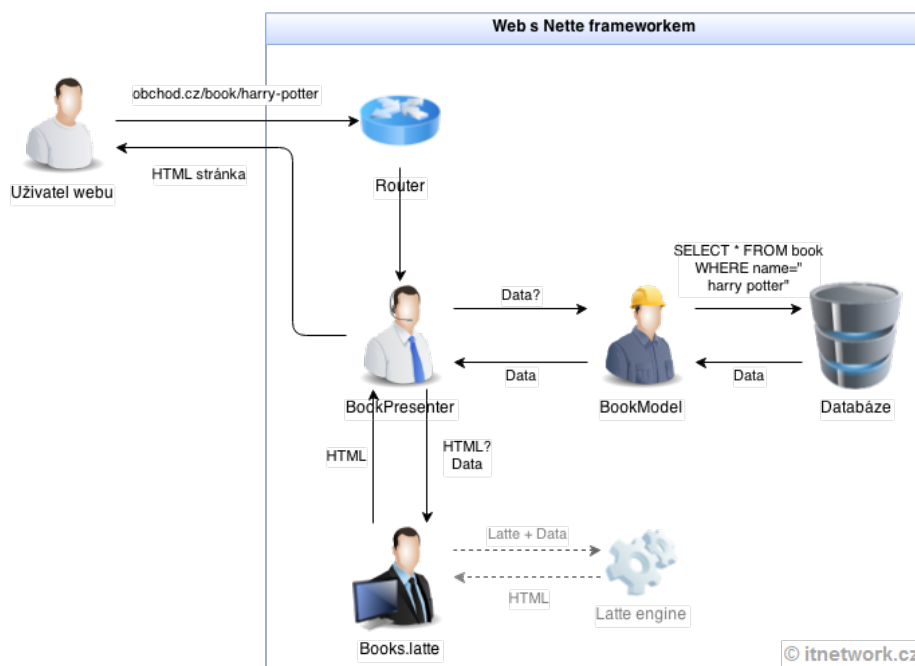
**Presentery:** Ty oddělují samotnou logiku a zobrazení. Presentery komunikují s modely a odesílají data do HTML souborů.

Princip architektury MVP je znázorněn na obrázku 2.3.

---

<sup>6</sup>MVP: Model–View–Presenter

Pro odchyťávání chyb byl implementovaný velice užitečný ladící nástroj, který dostal označení „Tracy“. Chyby zobrazuje v přehledné formě a dokáže také odhadovat, v čem uživatel nejspíše chyboval.



Obrázek 2.3: Princip architektury MVP, převzato z [1]

## 2.2.4 JavaScript

Jedná se o multiplatformní, objektově orientovaný, událostmi řízený skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape. Javascript existuje již od 4. prosince 1995[10]. Je překládán za běhu a vykonáván podle svého zdrojového kódu.

Tento jazyk funguje na straně klienta, což znamená, že všechny aplikace jsou spuštěny přímo v prohlížeči u klienta. Toto je jeden z nejhlavnějších rozdílů oproti PHP (zmíněno v kapitole 2.2.2), které neumožňuje měnit obsah webové aplikace přímo u klienta.

Pomocí JavaScriptu můžeme provádět na webu kontrolu dat v reálném čase – typicky u formulářů, kdy jsme schopni odhalit chybu na uživatelském vstupu ještě dříve, než uživatel data odešle, zobrazovat reálný čas, vytvářet animace či mít na webové stránce mapu. Většina prohlížečů má své JavaScriptové enginy, které umožňují správně fungování. U některých prohlížečů nejde tento engin ani vypnout.

Pokud chceme napsat něco v Javascriptu, můžeme to udělat dvěma způsoby. Jedním z nich je napsat kód přímo na stránku, takový kód se musí napsat do obalujícího párového tagu `<script>` či můžeme kód napsat do externího souboru, jehož název končí koncovkou „.js“.

## 2.2.5 JQuery

Jedná se o open source JavaScriptovou (viz kapitola 2.2.4) knihovnu, která je šířena pod licencí MIT. Knihovna byla představena v lednu 2006 Johnem Resigem u BarCamp v NYC a jejím hlavním cílem je zjednodušení práce s JavaScriptem.

Mezi nejvýznamnější funkce, které JQuery nabízí, patří možnost animací, manipulace s CSS či Ajax, který umožňuje načítat data ze serveru bez nutnosti aktualizace stránky. Je nutno podotknout, že vše, co umožňuje JQuery lze zvládnout i v JavaScriptu, jen se mnohdy takový kód psaný přímo v JavaScriptu až několikanásobně protáhne. Aktuálně nejnovější verzí je 3.3.1, která byla vydána 20. ledna 2018.



## Kapitola 3

# Jiné vhodné technologie

V této kapitole budou charakterizovány další vhodné a známé technologie, které by šly k implementaci této práce využít. Důvodem, proč níže zmíněné nebyly pro implementaci vybrány, byly osobní preference.

### 3.1 Laravel

Jedná se open source PHP framework pod licencí MIT, který vznikl již v roce 2011, a je určen primárně pro webové aplikace. Autorem této práce je programátor Taylor Otwell. Stejně jako Nette využívá i Laravel MVC architektury. Framework patří ke světové špičce, díky čemuž má i velice dobrou komunitu programátorů. Nejnovější verzí je prozatím 5.6, která byla vydána 7. února 2018. Framework si klade za cíl jednoduchost, které dosahuje tím, že zjednodušuje nejběžnější operace, které musí většina programátorů při vývoji webové aplikace řešit.

Mezi základní funkcionality patří autentizace uživatele, nástroje pro komunikaci s databází, mailing či správu a zpracování dotazů na jednom místě. Framework obsahuje některé komponenty či třídy, které vychází z dalšího, jiného frameworku Symfony (viz kapitola 3.2).

### 3.2 Symfony

Stejně jako Laravel či Nette je i Symfony open source webový PHP framework, který je vedený pod licencí MIT. Je postaven na webových aplikačních frameworkcích, například na Django či Spring, které se však v dnešní době již nepoužívají.

Původní název projektu byl “Sensio Framework”, který byl vyvíjen již v roce 2004, přičemž první verze byla dostupná již v roce 2005. V současnosti je nejaktuálnější verze 4.2.5, která byla vydána 2. dubna v roce 2019. Vývoj je sponzorován francouzskou firmou Sensio Labs v Paříži.

Framework je taktéž postaven na architektuře MVC a obsahuje velkou spoustu nejrůznějších komponent, které ulehčují práci programátorovi, stejně, jako tomu bylo i v případě dosud zmíněných frameworkcích. Stejně jako Laravel, tak i Symfony má za sebou velkou komunitu uživatelů, díky kterým jsou k dispozici nejen nejrůznější diskuze či rady, ale také se i díky nim může Symfony stále rozvíjet a zlepšovat.

### 3.3 PostgreSQL

Jedná se o objektově-relační databázový systém, který je dostupný jako open source pod licencí MIT. Běžně je též známý pod označením „Postgres“. Jedná se o alternativu k běžnému MySQL (více o MySQL v kapitole 2.2.1). Systém plně splňuje kritéria ACID. Zkratka ACID je odvozena od slov atomicity, consistency, isolation a durability.

**Atomicity (atomicita)** = příkazy se v transakci musí provést buď všechny nebo žádný.

**Consistency (konzistence)** = před dokončením a po dokončení transakce musí být data konzistentní, což jinými slovy znamená, že v kterýkoliv čas musí data v databázi obsahovat možnou informaci z reálného světa.

**Isolation (izolovanost)** = ačkoliv bude probíhat více transakcí zároveň paralelně, žádná z nich nesmí ovlivnit výsledek jiné současně probíhající transakce

**Durability (trvalost)** = jakmile je transakce úspěšně dokončena, pak všechny změny dat v databázi nemohou být ztraceny a jsou trvalé, tzn., že pokud dojde například k výpadku proudu či jinému přerušení provozu, vše zůstane zachováno tak, jak to bylo před tímto přerušením.

Ačkoliv se PostgreSQL těší v poslední době vysoké popularitě, stále je MySQL jeden z nejvíce používaných databázových systémů. Ani v dnešní době není PostgreSQL běžným standardem u stále velké části webhostingových poskytovatelů. Nevýhodou může být pro někoho i menší rychlost oproti MySQL a větší striktnost, co se týče datových typů.

### 3.4 MariaDB

Jedná se o relační databázi, za jejichž vznikem stojí bývalí vývojáři MySQL. Za hlavního vývojáře je považován Michael Widenus, který byl i původním zakladatelem MySQL. Důvodem odtržení se od původního MySQL bylo, že společnost Oracle koupila společnost Sun Microsystems, která mimo jiné produkty, vlastnila i databáze MySQL. Společnost Oracle pak začala vytvářet různé změny, které se však komunitě nelíbily.

MariaDB je šířena jako open source s tím, že všichni, co se podílí na vývoji, musí sdílet svá autorská práva s MariaDB Foundation, což je nadace, která je správcem veškerého kódu komunity MariaDB a je také vlastníkem serveru MariaDB.

MariaDB je plně kompatibilní s MySQL, takže přechod na tuto databázi není nikterak náročný.

## Kapitola 4

# Analýza požadavků

V této kapitole je rozepsán podrobný popis požadavků na informační systém. Funkcionalita vychází nejen z určeného zadání bakalářské práce, ale také z dalších uživatelských požadavků či obohacujících funkcí, které mě napadlo navíc implementovat.

Cílem systému je vytvoření prostředí, které bude uživatelsky přívětivé pro nabízející – farmáře, ale i pro kupující. Předpokladem je, že jak farmáři, tak kupující nejsou příliš pokročilými uživateli počítačů, proto výsledný systém musí být jednoduchý na ovládání, zároveň ale obsahovat vše, co by mohlo být potřeba a je pro uživatele užitečné.

### 4.1 Popis požadavků

Níže sepsané požadavky vycházejí z osobních rozhovorů s farmáři a dalšími lidmi, kteří nabízejí své produkty na nejrozličnějších serverech nebo na Facebooku.

#### 4.1.1 Uživatelé systému

Základem systému jsou uživatelé. Je důležité si uvědomit, že farmář nemusí nutně pouze prodávat, ale může i nakupovat. Stejně tak kupující nemusí být vždy nutně za všech okolností jen kupující. Z tohoto důvodu není vhodné určovat role uživatelům, jelikož nejsme schopni je jednoznačně rozdělit na „nakupující“ nebo „prodávající“. Typ takových uživatelů bude pouze jeden. U podobných systémů (těmi se zabývá kapitola 4.2) se tento problém řeší stejně. Je ale nutné zavést navíc typ uživatele administrátor, který bude moci smazat nahlášené uživatele, produkty apod.

Každý uživatel by měl mít svůj profil, který bude veřejně přístupný a tedy viditelný i bez nutnosti přihlášení. U každého takového uživatele by měly být kontaktní informace. Samozřejmostí je jméno, dále kontaktní telefon, e-mailová adresa a především město působení spolu s mapou. Jakmile se uživatel přihlásí do systému, je mu umožněno přidávat inzeráty, popřípadě vytvářet objednávky. Do sekce přidání inzerátu by měl být jednoduchý přístup, ideálně viditelný hned z hlavní stránky, aby tuto jednu z hlavních funkcí uživatel nemusel dlouho hledat.

V tabulce 4.1 jsou přehledně znázorněna práva jednotlivých typů uživatelů, kteří budou přistupovat do systému. Tabulka nepokrývá celý systém, pouze znázorňuje nejpodstatnější rozdíly mezi rolemi uživatelů.

	Administrátor	Registrovaný uživatel	Neregistrovaný uživatel
Přidat inzerát	Ano	Ano	Ne
Zobrazit inzerát	Ano	Ano	Ano
Vytvořit poptávku	Ano	Ano	Ne
Nahlášení závadného obsahu	Ano	Ano	Ne
Smazání produktu	Ano, všem	Ano, svůj	Ne
Statistiky	Ano	Ano	Ne

Tabulka 4.1: Porovnání uživatelských práv

#### 4.1.2 Nabízení produktů

Farmářům bude umožněno nabízet celkem širokou škálu produktů – nejen ovoce a zeleninu, ale i maso, mléčné produkty apod. Proto je nutné, aby přidávání inzerátu bylo vytvořeno tak, aby bylo použitelné pro všechny kategorie a jednotlivé vyžadované položky dávaly smysl.

Nezbytné je, aby farmář vyplnil název nabízeného produktu a také příslušnou kategorii. Různé produkty mají i různou trvanlivost, což je obzvlášť u některých kategorií (jako např. mléčné produkty) velmi podstatná informace. Samozřejmě nesmí chybět také dostupné množství (s ohledem na příslušnou jednotku, která dává smysl v kontextu daného produktu) a cena (za jednu tuto jednotku). Mezi další údaje, které by mohly spoustu kupujících zajímat, patří informace o správném skladování, informace o pěstování či chovu, případně další popis, kam farmář napíše cokoliv dalšího, co usoudí za vhodné zmínit v inzerátu. Ke každému inzerátu se také obvykle přidává fotografie pro ukázkou produktu.

Další nezbytnou součástí inzerátu musí být informace o možných způsobech dodání. Farmáři zpravidla umožňují osobní odběr na určité adrese, někteří nabízejí i rozvoz přímo na adresu kupujícího za určitý poplatek.

Pro kupujícího je podstatné vyhledávání produktů. Systém bude sdružovat všechny farmáře na území ČR, takže je pro kupující podstatné filtrovat produkty jak podle kategorií, tak podle krajů či měst.

#### 4.1.3 Inzerát produktu

U samotného inzerátu jsou pro kupujícího důležité především informace, které přidává farmář přímo při vkládání inzerátu (viz výše), ale nejen to. Kupující také zájímají zkušenosti ostatních s daným farmářem. To by mohlo být znázorněno například nějakým hodnocením farmáře. Toto hodnocení by mělo obsahovat jednak nějakou škálu, ale i možnost slovního dodatečného hodnocení. Kupující mají často i dotazy navíc, takže u každého produktu by měla být možnost komentáře, resp. dotazů. Na tyto dotazy farmář může odpovídat. Dotazy by měly být veřejné, protože většinou mívají kupující podobné otázky, kterým se takto může předejít. Pro kupujícího je také podstatné, aby snadno a rychle mohl daný produkt poptat. Přímo u každého produktu by tedy měla být možnost jednoduché poptávky – kolik jednotek kupující poptává, způsob odběru a případná další dodatečná poznámka (např. adresa v případě rozvozu).

#### 4.1.4 Nahlašování

V souvislosti s komentáři je vhodné přidat i možnost nahlášení nevhodného komentáře. Nevhodný ale může být i produkt nebo například samotný profil uživatele může být třeba falešný, tedy nevhodný.

Samotné nahlášení by ale bylo zbytečné, pokud by nebyl nikdo, kdo by mohl nahlašování dotazů, profilů či produktů spravovat. Proto je nutné vytvořit dalšího uživatele, administrátora, který bude moci dělat tuto údržbu. Měl by mít umožněno případně i danou nahlášenou věc smazat, včetně uživatele, pokud by to bylo oprávněné.

#### 4.1.5 Poptávky/objednávky

Jelikož systém slouží pro objednávání produktů, pro přehlednost by zde měly být uvedeny vlastní nákupy (neboli provedené poptávky) uživatele, případně nákupy, které u tohoto uživatele provedli jiní uživatelé. Aby si byl nakupující jistý, že farmář objednávku přijal a počítá s ní, měl by mít farmář možnost objednávku buď přijmout nebo zamítnout. Kupující by pak měl mít ve svém účtu tento stav (přijato/zamítnuto) také k dispozici.

Farmář by měl objednávku ve svém účtu vidět jako soupis základních důležitých informací nutných k objednání. Mezi ně patří samozřejmě jméno a kontaktní údaje kupujícího, dále o jaký produkt se jedná a v jakém množství. Farmář také musí vědět, jaký si kupující vybral způsob dodání. Nakonec je pro farmáře podstatné vědět i celkovou cenu za daný objednaný produkt.

Pro farmáře by navíc mohly být užitečné informace o prodejkách. Například jaký uživatel u něj nakupuje nejvíce, jaký produkt se nejvíce prodává a kolik za prodeje utržil.

### 4.2 Konkurenční systémy

V následující části budou uvedeny již existující systémy, které se svým zaměřením podobají systému, který je předmětem této práce. Každý systém bude krátce charakterizován.

Jsou zde uvedeny jak všeobecné bazary, tak portály se zaměřením na prodej farmářských výrobků, žádný z nich ale není kombinace obého – na českém trhu se mi nepodařilo nalézt takový portál, který by umožňoval nejen nákup farmářských produktů, ale i jejich prodej. Na všeobecných bazarech je možné prodávat cokoliv, ale z toho důvodu je zde horší možnost vyhledávání a nejsou zde uvedeny podstatné informace, které jsou specifické pro potraviny a podobné produkty (např. trvanlivost, specifika pěstování/chovu apod.). Oproti tomu na zaměřených stránkách většina takových informací je, ale zato zde není možnost prodávat produkty jako např. přebytky ze zahrádky. Na takových stránkách nabízejí pouze farmáři a živnostníci zabývající se výrobou takových produktů, ale „obyčejní“ lidé nikoliv.

#### 4.2.1 Všeobecné bazary

Mezi hlavní konkurenty patří portál *Bazoš*<sup>1</sup>, který se však nesoustředí přímo na potraviny.

Jedná se o klasický všeobecný prodejní portál, kde lze nalézt prakticky jakýkoliv sortiment. Aby se však produkty zobrazily ve správné kategorii, musí je uživatelé správně zařadit, což jim mnohdy dělá problémy a pak se stane, že jsou například pneumatiky zařazeny v sekci osobní automobily místo v sekci náhradní díly a příslušenství. Takže ačkoliv

---

<sup>1</sup>Dostupné na: <https://www.bazos.cz/>

je tento portál v České republice nejoblíbenější, není vždy snadné nalézt to, co hledáme. Stejně funguje i například *Sbazar*<sup>2</sup> od společnosti Seznam.cz, a.s.

#### 4.2.2 Farma na dlani

Dalším konkurenčním webem je již konkrétněji zaměřený portál na prodej domácích produktů. Jeho název je *Farma na dlani*<sup>3</sup>. Tento web je graficky i obsahově velice povedený. Nabízí spoustu funkcí, nicméně je jich už tolik, že z uživatelského testování vyplynulo, že se na webu ztrácí i lidé mladších ročníků, především ale však ti starší, pro které je web také určen.

Přes tento portál je možné vyhledávat navíc farmářské trhy v okolí. Jednou z hlavních funkcí ale zůstává nabídka a vyhledávání výrobků. Uživatel si zde může vybrat kategorii, kterou chce vypsát, ale může i zadat, zda má zájem zobrazit výrobky nabízené v ověřené prodejně, nabízené jako prodej ze dvora nebo jako domácí přebytky. Dále může upřesnit město, případně zadat klíčové slovo pro upřesnění vyhledávání. Web je koncipován tak, že se po vyhledání nevypisují konkrétní produkty, ale jakési položky, které jen jako nabídky produktů vypadají. Tyto položky odkazují do tzv. „obchodu farmáře,“ což je v podstatě profil farmáře, kde jsou uvedeny všechny jím nabízené produkty. Zde již produkty (po tom, co je nutné je znovu najít, tentokrát již bez jakéhokoliv formuláře pro vyhledání) lze přidat do košíku a případně objednat. Tento způsob je ale poměrně matoucí a v systému, kterým se zabývá tato práce, by mělo být vyhledávání vyřešené tak, jak uživatel očekává, tzn., že se mu vyhledají konkrétní produkty, které může rovnou popsat, nikoliv odkazy na všechny nabídky farmáře.

U každého farmáře je veřejná adresa (lze zobrazit i mapu) a další kontaktní údaje, krátký popis. V sekci u jeho profilu je možné vidět hodnocení od jiných uživatelů.

#### 4.2.3 Svět bedýnek

Druhým přímým konkurentem je stránka *Svět bedýnek*<sup>4</sup>. Tento web je uživatelsky více přívětivý, než předchozí, bohužel však nenabízí pouze domácí produkty, ale i klasické komerční produkty z velkoprodukce, které jsou chemicky ošetřené. Mnohdy chybí u produktů informace o způsobu pěstování nebo o původu. Toto a další maličkosti sráží celkovou kvalitu jinak povedeného prostředí.

Uživatel si přes tento portál objedná nejružnější produkty, v logistickém centru portálu mu nákup připraví do bedýnky a další den doručí. Tento portál tedy nefunguje jako inzertní portál, ale spíše online supermarket, kde si uživatelé mohou na jednom místě objednat produkty zároveň od více producentů, farmářů či vybraných obchodů

---

<sup>2</sup>Dostupné na: <https://www.sbazar.cz/>

<sup>3</sup>Dostupné na: <https://www.farmanadlani.cz/>

<sup>4</sup>Dostupné na: <https://www.svetbedyneku.cz/>

## Kapitola 5

# Uživatelské rozhraní

V této kapitole budou popsány jednotlivé důležité dílčí prvky uživatelského rozhraní. U vytváření uživatelského rozhraní jsem se inspiroval především již existujícími podobnými systémy (konkurenční systémy byly popsány v kapitole 4.2).

Jak jsem ale v kapitole o konkurenčních systémech zmínil, nebyl nalezen takový portál nebo systém, který by již implementoval to, co je zadáním této práce. Totiž propojit funkce inzertního portálu s funkcemi portálu pro nabídku farmářských produktů. Návrh uživatelského rozhraní tedy vychází z obou typů těchto systémů a snaží se zahrnout to nejlepší z obou, zároveň však udržet nekomplikovanost a co největší uživatelskou přívětivost.

### 5.1 Menu

Stránka bude obsahovat celkem dvě navigační lišty. V té první budou položky jako jsou „Pravidla pro prodej ze dvora,“ kde naleznou prodejci stručný přehled toho, co musí dodržet, aby stále mohli svůj prodej nazývat prodejem přebytků a nemuseli se strachovat případných problémů se zákonem. Další položka bude mít název „Kontakt,“ kde budou moci nalézt kontaktní údaje na správce domény, v případě potřeby řešit reklamní a podobné požadavky. Posledními položkami v menu jsou, v případě nepřihlášeného uživatele, tlačítka pro registraci a přihlášení do stránky. Pokud je uživatel přihlášený (jako řadový uživatel, ne administrátor), pak je v této části tlačítko na odhlášení a pro vstup do osobního účtu. V případě administrátora webu zde bude pak navíc tlačítko „Správa webu,“ kde bude moci administrátor spravovat nahlášené uživatele, příspěvky či komentáře.

Další podmenu (druhá navigační lišta) obsahuje výčet všech kategorií, do kterých mohou spadat prodávané suroviny v systému. Slouží tedy pro rychlé vyfiltrování pouze těch produktů, které zákazník hledá. Dostupnými kategoriemi myslíme ovoce, zeleninu, bylinky a houby, pečivo, maso a ryby, mléčné produkty a domácnost. Položka vše vypíše všechny nabízené produkty v systému bez ohledu na kategorie. Poslední položkou v této liště je barevně odlišené tlačítko „Přidat inzerát“. Umístění je zvoleno schválně mezi kategoriemi, protože se předpokládá, že na tuto část se lidé budou dívat nejvíce, tudíž toto tlačítko nepřehlédnou.

### 5.2 Hlavní stránka

Hlavní stránka bude obsahovat jednoduchý úvodní slider, který může sloužit buď pro prezentaci zajímavých či reklamních fotografií. Při návrhu jsem se inspiroval knihou *A project*

*guide to UX design*[9]. Další komponenta na stránce bude vypisovat kolik uživatelů je již registrovaných, kolik nabídek je v systému k dispozici a kolik je v systému komentářů. Na hlavní stránce budou výpisy nejnověji přidaných nabídek, patřících do nejzákladnějších kategorií, jako jsou mléčné produkty, ovoce a zelenina + masové produkty. Tento výpis se bude zobrazovat ve 4 sliderech rozdělených právě podle těchto zmíněných kategorií. Každá položka ve slideru musí obsahovat základní informace, tzn. o co se jedná, za jakou cenu se produkt prodává a kdo jej prodává. Dále pak bude u každého produktu tlačítko na proklik na detail.

### 5.3 Přidání produktů

Uživatel smí přidávat libovolné produkty (splňující pravidla webu) společně s několika povinnými atributy. Těmi jsou cena za produkt v předem definované měně, počet kusů zboží (uživatelé budou mít na výběr ze základních měrných jednotek), které je určeno pro prodej, trvanlivost produktu, cena, způsob předání (na výběr jsou dva typy: osobní odběr či dovoz na adresu), zařazení nabízeného produktu do odpovídající kategorie, popis produktu, způsob pěstování, popis toho, jak by se měl produkt ideálně skladovat. Dále je pak možnost přidat 5 fotografií k produktu, přičemž první fotografie se bude zobrazovat jako titulní (zároveň bude tato fotografie u náhledu produktu). U každého produktu bude zaznamenáno datum vytvoření, jméno prodejce s odkazem na jeho osobní profil, kde lze nalézt i výpis všech jím nabízených produktů.

### 5.4 Můj účet

Na svém osobním účtu lze nalézt soupis všech objednávek od uživatelů, u kterých je možné se rozhodnout, zda danou poptávku schválit či nikoliv. Pokud nebude schválena, přičte se poptávané množství zpět na sklad. Dále je možné vidět i stav svých vlastních objednávek, které uživatel odeslal ostatním farmářům. V tomto soupisu je možné vidět, zda byla jeho objednávka schválena či nikoliv.

V osobním účtu lze zpětně editovat zadané osobní údaje, které byly uvedeny při registraci, dále je možné doplnit profil o profilovou fotografii. V účtu je k vidění také soupis všech uživatelem nabízených nabídek. Z tohoto soupisu může nabídku smazat či editovat. K těmto akcím slouží dvě připravená tlačítka - „editovat“ a „smazat“.

Účet dále nabízí možnost propojení účtu s jiným účtem. Tato možnost je především pro větší farmáře, resp. ty, kteří mají tolik objednávek a práce, že nemají sami osobně čas být na počítači, schvalovat objednávky a nabízet produkty. Mohou si proto přidat účty, které uvidí všechny jeho nákupy, nabídky či objednávky od zákazníků. Tento účet v kooperaci pak provádí všechny úkony (zamítání a schvalování přijatých objednávek, editace a mazání nabídek produktů) pod jménem dotyčného farmáře.

### 5.5 Statistiky

Jelikož se na stránce nabízí produkty, které pak zákazníci objednávají, hodí se prodejcům vědět i základní statistiky o jejich prodejkách. Tyto statistiky bude možné nalézt také v rámci sekce „Můj účet“.

Statistiky zahrnují grafy, které znázorňují, jací uživatelé nejčastěji u tohoto prodejce nakupují, jaké kategorie produktů jsou u něj nejčastěji poptávané a jaký je výdělek z jed-



notlivých kategorií produktů. Bude zde i informace o celkovém výdělku, kterého za dobu existence na stránce dosáhl.

## 5.6 Objednávka

Každý registrovaný uživatel bude moci vytvořit objednávku na konkrétní zboží nabízené v systému. U každé objednávky se musí zadat požadované množství, o jaké má zájem a zvolit si případně místo odběru, je-li na výběr z vícero míst. Zákazník má také možnost napsat doplňující otázku či vzkaz do poznámky. U každého produktu bude ale možnost napsat dotaz i přímo ke zboží, bez nutnosti vytvářet objednávku. Pokud je odeslána poptávka, automaticky se odečte množství zboží k dispozici, které prodejce uvedl.

Prodejce je upozorněn systémem na novou poptávku formou e-mailu. Prodejce bude mít pak možnost, po přihlášení se na web, objednávku zrušit, čímž opět dojde k přičtení původně poptávaného zboží na sklad. Pokud poptávku schválí, zobrazí se toto v systému, jak u prodejce, tak u zákazníka, který si zboží objednal.

## 5.7 Uživatelská oprávnění

V aplikaci budou existovat dva typy uživatelů. Jedním z nich je admin, druhým pak řadový uživatel, který má povoleno, jak prodávat, tak i nakupovat zboží v aplikaci. Tím to však nekončí, na webu je mu umožněno také psát dotazy prodejci k produktu, hodnotit farmáře, schvalovat či zamítat objednávky na své nabízené produkty, upravovat a mazat své produkty, dále nahlašovat produkty ostatních prodejců, nahlašovat komentáře či přímo uživatele na webu.

Neregistrovaný návštěvník má přístup k obsahu prakticky celého webu, může si vše ale jen prohlížet. Nemůže provádět nákup zboží, inzerovat své produkty či komentovat jakékoliv příspěvky nebo produkty ostatních uživatelů, ani je jakýmkoliv způsobem hodnotit.

Při registraci bude muset návštěvník webu vyplnit několik povinných údajů o své osobě. Za povinné údaje v systému je považováno celé jméno registrujícího, platná adresa bydliště (město, ulice, poštovní směrovací číslo), telefonní kontakt + libovolné heslo a konečně souhlas se zpracováním osobních údajů dle platné legislativy.

Uživatel typu administrátor bude mít oproti běžnému uživateli povoleno vkládat do úvodního slideru na hlavní stránce fotky. Má taktéž možnost zobrazit nahlášené inzeráty, komentáře nebo profily ostatními uživateli. Mezi jeho další práva spadá celková správa uživatelů a jejich komentářů. Celkovou správou se rozumí možnost odstranění nahlášeného elementu na stránce. Smazat je však možné i ty nenahlášené.

## 5.8 Pravidla webu pro uživatele

Každý uživatel musí dodržovat slušné chování, nejsou respektovány ani vulgarismy a ani snahy o podvod. Za podvod se považuje, pokud někdo nevyzvedne poptávané zboží bez předchozí omluvy nebo pokud nabízené zboží neodpovídá realitě. Ochranou proti neodpovídající realitě jsou vlastní fotografie, nikoliv stažené z internetu. Dále je zakázána jakákoliv propagace vlastní firmy či jiné propagační odkazy, které vedou mimo web. Pokud jakýkoliv uživatel takové chování odhalí, je možné nahlásit administrátorovi. Na něm pak je, jak daný případ vyřeší.

## 5.9 Propojený účet

Pro ušetření farmářova času by bylo vhodné umožnit přístup k jeho nabídkám i dalším vybraným účtům. Tyto přiřazené účty by měly umožňovat správu objednávek – přijetí nebo zamítnutí objednávky, dále správu nabízených produktů – jejich editaci, smazání, ale i přidávání. Mělo by být ale ošetřeno, aby tyto propojené účty nemohly editovat soukromé údaje. Takto přiřazený účet by měl sloužit např. pro brigádníky.

## Kapitola 6

# Návrh informačního systému

V této kapitole bude představen vytvořený ER diagram, u kterého budou vysvětleny základní pojmy, které je pro pochopení celku nutné znát. Dále bude představen vytvořený use-case diagram, u kterého bude opět popsáno několik základních pojmů, které se k tomuto diagramu vážou.

### 6.1 Diagram případů užití

Diagram případů užití, neboli use-case diagram, popisuje chování systému z pohledu uživatele. Každý dostupný typ uživatele je běžně značen jako panáček, u kterého je vždy napsáno, o jakého typového uživatele se v systému jedná. Diagram v sobě nese jednotlivé případy užití, čímž i zároveň vytváří hranice systému. Každý případ užití nese své identifikační jméno, může mít i textovou specifikaci.

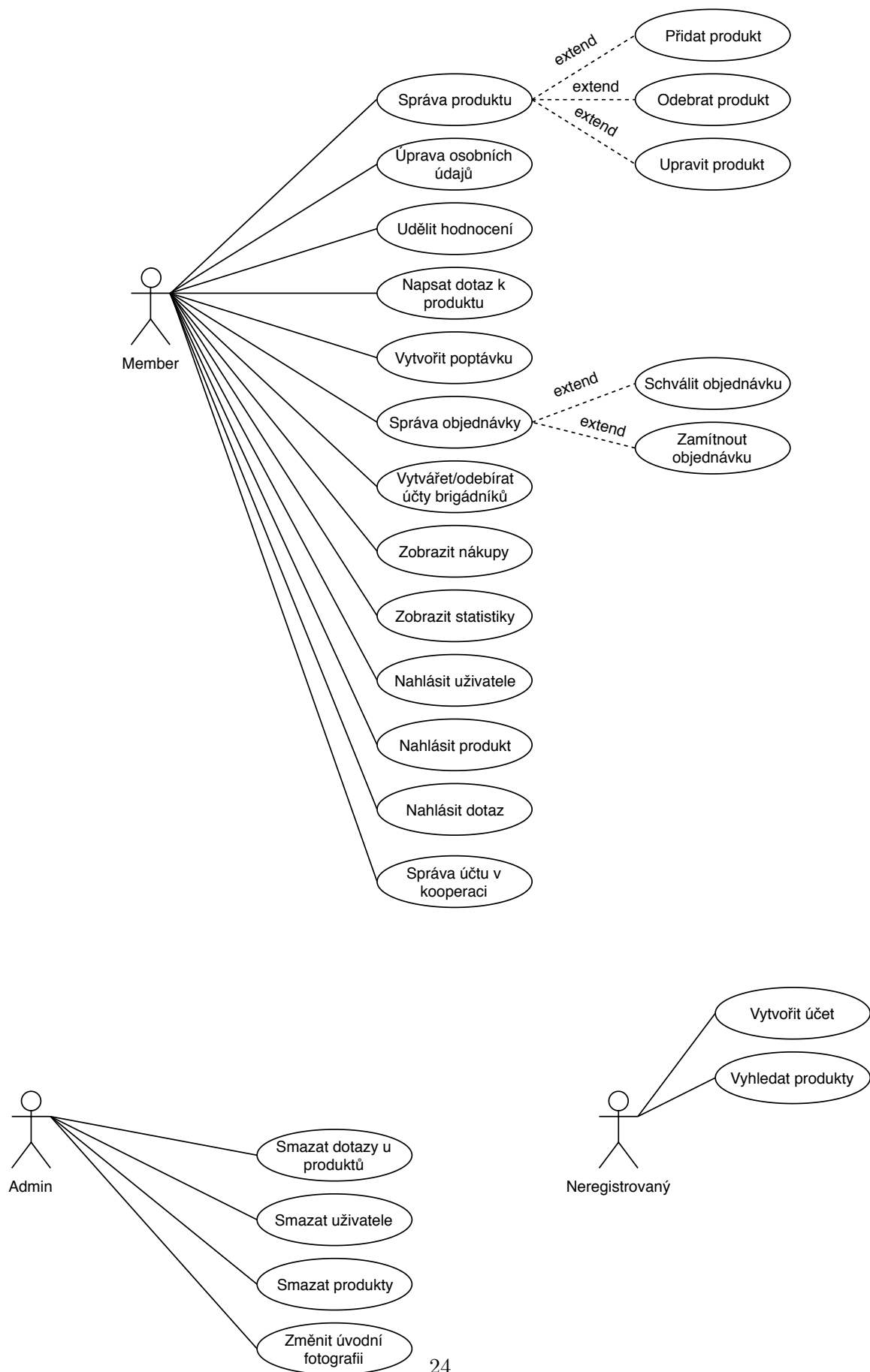
V Use-case se můžeme setkat i s pojmy „include“ a „extend“. „Extend“ se používá v případě, že případ použití je rozšířen o nějakou další funkci. „Include“ se aplikuje, pokud se nějaká sekvence případu užití opakuje.

Níže na obrázku 6.1 lze vidět Use-case diagram vytvořený pro tento informační systém.

### 6.2 ER diagram

Entity Relationship Diagram neboli entitně vztahový model se používá pro datové modelování. Pomocí něj je možné graficky popsat entitní typy a jednotlivé vztahy mezi nimi formou grafu. Data jsou popsána pomocí entit, k nimž náleží odpovídající atributy.

Každá entita musí být jednoznačně identifikovatelná vhodným identifikátorem UID, který slouží zároveň jako unikátní/primární klíč pro danou tabulku. Entity jsou nejčastěji vyjádřeny pomocí podstatného jména. Entita je popsána množinou svých atributů, které blíže specifikují vlastnosti entity. Příkladem může být entita lidé, kdy jejími vlastnostmi, neboli atributy, jsou jejich adresy, jména, stáří atd. Mezi vlastnosti atributů patří tzv. volitelnost neboli optionality. Ta určuje, zda musí mít konkrétní atribut nějakou hodnotu nebo nikoliv a je tak tedy tolerováno i NULL. Dalším pojmem je kardinalita, která definuje množství výskytu. Toto množství se zaznačuje k relacím mezi jednotlivými entitami. Relace je spoj, typicky čára, mezi dvěma entitami.



Obrázek 6.1: Use-case diagram

Kardinalita může být zaznačena jednou ze tří možností:

- **1:1** Jedné entitě A může odpovídat jen jedna entita B a naopak.
- **1:M** Jedné entitě A může odpovídat více jak jedna entita B, entita B odpovídá jen jedné entitě A.
- **M:M** Entitě A může odpovídat více jak jedna entita B, entitě B může odpovídat více jak jedna entita A [4].

V souvislosti s ER diagramem se můžeme setkat i s pojmy množina entit či množina vztahů. Množina entit definuje množinu entit stejného typu, které sdílí stejné vlastnosti (atributy) a množina vztahů naproti tomu definuje množinu vztahů stejného typu, které sdílí stejné vlastnosti.

ER diagram tohoto systému je možné vidět níže na obrázku 6.2.

### 6.2.1 Tabulky produkt a produkt\_photo

Vazba mezi tabulkou *produkt* a *produkt\_photo* říká, že každý produkt může být doplněn o fotografii. Každá fotografie musí být k danému produktu unikátní a jejich počet nesmí přesáhnout celkem 5 fotek na 1 produkt. U fotografie platí, že musí být ve formátu JPG, PNG či GIF a velikost jednoho obrázku nesmí přesáhnout 5 Mb. Tabulka *produkt\_photo* uchovává ID produktu v podobě cizího klíče, aby bylo jasné, k jakému výrobku se fotografie váže.

### 6.2.2 Tabulky produkt a comments

Každý produkt může obsahovat libovolný počet komentářů od přihlášených uživatelů. U komentáře jsou uchovávány 2 cizí klíče a to *id\_produkt* a *id\_user*. První z nich je určen k tomu, aby bylo jasné, k jakému produktu je komentář vázán a druhý je použit k tomu, abych mohl u každého komentáře vypsát i jméno komentujícího. Platí tedy, že produkt může obsahovat 0 až nekonečno komentářů, ale zároveň musí platit, že jeden komentář náleží pouze jednomu konkrétnímu unikátnímu produktu.

Komentář se dělí vždy na dvě části. Na komentář hlavní a na odpověď na komentář. Hlavní se uchovává v databázi ve sloupci *id\_comment* s hodnotou 0 (ta značí, že jde o hlavní komentář). Odpověď na komentář má ve sloupci *id\_comment* číslo ID primárního klíče hlavního komentáře, na který navazuje odpověď na tento hlavní komentář.

### 6.2.3 Tabulky produkt a objednávka

Uživatelé mají možnost vytvářet v systému objednávky na konkrétní inzerované produkty. Na každý produkt může existovat 0 až nekonečno objednávek. Objedávka však náleží vždy jednomu produktu. U každé objednávky se vlastník inzerce může rozhodnout, zda ji schválí či nikoliv. K tomu v této tabulce slouží atribut s názvem *potvrzeno*, který může nabývat dvou hodnot: jedničky, která potvrzuje přijetí objednávky a nebo dvojky, která naopak objednávku zamítne. V této tabulce se také v atributu *celkova\_cena\_objednavka* vypočítává celková cena objednávky podle zadaného poptávaného množství a ceny za jednotku produktu. Mezi další atributy tabulky patří mimo jiné i cizí klíč s ID produktu, aby bylo možné konkrétní produkt v systému pojmenovat.



#### 6.2.4 Tabulky produkt a report\_products

Tato vazba umožňuje zaznamenávat do tabulky *reports\_products* všechny produkty, které porušují pravidla stránky. Jedno nahlášení odkazuje vždy na jeden konkrétní produkt, přičemž na produkt jako takový může být vedeno nekonečno nahlášení. Přístup k této tabulce má pouze typ uživatele administrátor, který má i oprávnění dotyčné produkty smazat či ponechat. Pokud se rozhodne produkt smazat, má k tomu právo, nahlášený záznam se z tabulky smaže, má však i možnost produkt ponechat a odkliknout, že si dotyčné nahlášení zkontroloval. V takovém případě dojde taktéž k odstranění záznamu o nahlášení.

#### 6.2.5 Tabulky comments a reports

Entita *reports* slouží pro uchovávání záznamů o nahlášených komentech. K těmto záznamům má opět přístup pouze uživatel typu administrátor. Vyřešení daného nahlášení probíhá stejně jako v předchozím případě (viz kapitola 6.2.4), jen s tím rozdílem, že se neodstraňují produkty, ale jednotlivá vlákna komentářů pod daným produktem. Taktéž platí, že komentář může mít 0 až nekonečno nahlášení. Každé jedno nahlášení může být vždy pouze na jedno konkrétní vlákno komentářů.

#### 6.2.6 Tabulky user a objednávka

Každý registrovaný uživatel má možnost vytvářet 0 až nekonečno objednávek, přičemž ale každá objednávka náleží jednomu uživateli a zahrnuje v sobě pouze jeden konkrétní produkt. Pokud by nastala situace, kdy by byl uživatel smazán ze systému (např. za porušení pravidel) a zároveň měl v systému objednávku, smažou se i všechny jeho objednávky společně s takovým uživatelem pomocí cizího klíče s akcí RESTRICT. V tabulce *objednavka* se dále uchovává ID prodejce, aby se mohli poptávky zobrazovat v účtu vlastníků dané inzerce. Každý uživatel vidí ve svém účtu jím realizované nákupy případně poptávky jiných na vlastní nabízené produkty.

#### 6.2.7 Tabulky user a hodnoceni

Lidé mohou hodnotit jednotlivé prodejce. Mají možnost udělit hvězdičky a ty pak případně doplnit nepovinným komentářem, který bude, společně s hvězdičkami, veřejně viditelný všem uživatelům i nepřihlášeným návštěvníkům webu. Z tabulky se vybírají hodnoty z atributu s názvem *hodnoceni\_hvezda*, které se pak průměrují a na základě této hodnoty se zobrazuje průměrné množství hvězdiček, kterých je k dispozici celkem 5 (5 hvězd je samozřejmě nejlepší hodnocení). Tabulka obsahuje cizí klíč *id\_rating\_from*, aby bylo možné u hodnocení vypisovat jméno uživatele, který tak hodnotil.

#### 6.2.8 Tabulky user a user\_cooperate

V tabulce *user\_cooperate* jsou uchovávány dva cizí klíče a to *id\_owner*, který určuje ID uživatele, který vlastní daný účet a *id\_coop*, který uchovává ID takového účtu, který má oprávnění provádět změny v účtu původního vlastníka. Vlastník účtu si ke svému účtu může přiřadit libovolné množství účtů.

## Kapitola 7

# Implementace

V této kapitole bude popsána implementace jednotlivých částí systému. Systém byl implementován v PHP frameworku Nette (viz kapitola 2.2.3).

V první části bude popsána struktura celého webu, dále bude ukázáno, jak probíhá přihlašování a registrace do administračního systému, následně, v jednotlivých podkapitolách, budou popsány zajímavé a nebo důležité části systému.

### 7.1 Struktura

V této části popíšeme základní hierarchii adresáře, tedy názvy složek včetně jejich popisu, dále pak názvy a významy jednotlivých souborů v nich.

#### 7.1.1 Adresář app

- **config** Obsahuje konfigurační soubor *config.neon*, pomocí kterého lze konfigurovat vlastní parametry, v případě tohoto systému to jsou jednotlivé uživatelské role v systému, formuláře, aplikace, databáze a jiné.

Přístup do databáze je v práci následující:

```
dsn: 'mysql:host=127.0.0.1;dbname=farmadodому'  
user: 'root'  
password:
```

Parametr DSN definuje, kde je databáze umístěna. Přesné umístění je za argumentem *host* v podobě IP adresy host serveru. Parametr *user* a *password* slouží pro určení uživatele a přístupového hesla k databázi. Abychom mohli s databází komunikovat, musíme vytvořit instanci `$database` třídy `Nette\Database\Context`. Tu jsem vytvořil v konstruktoru souboru *BaseManager.php*.

- **model** Obsahuje soubory obsahující veškerou logiku aplikace. Jsou určeny především pro komunikaci s databází.
- **modules** Tato složka obsahuje podsložku *Front*, která dále obsahuje všechny presentery, šablony a formuláře. Podsložka *forms* obsahuje soubory se základními formuláři, jejichž základ se v systému opakuje (například e-mail a heslo pro přihlášení a registraci). Proto, když jsou potřeba, jsou zavolány v Presenteru + se k nim dopíše vše ostatní potřebné, co je u daného formuláře třeba.



Presentery jsou v systému následující:

- **AdministrationPresenter.php** registrace uživatele + ověření + odhlášení uživatele.
- **BaseFrontPresenter.php** login uživatele + editace dat uživatele.
- **HomepagePresenter.php** výpis hlavní strany + výpis všech kategorií + výpis účtu uživatele + vkládání nabídky uživatele + editace nabídky uživatele + statistiky + správa objednávek + vyhledávání produktů dle navolené lokalizace + správa fotografií v úvodním slideru + správa uživatelů v kooperaci + správa nahlášených položek uživateli.
- **ProductPresenter.php** výpis detailu produktu, hodnocení u produktu, formulář pro objednání produktu + správa nahlášených položek uživateli.

### 7.1.2 Adresář www

Tento adresář obsahuje všechny obrázky, fonty, javascriptové knihovny či soubory s kaskádovými styly. Obrázky lze nalézt ve složce *images*. Soubory css ve stejnojmenné složce *css*. Složka se dále dělí na podsložku *min*, která obsahuje minifikované soubory. Soubory s koncovkou *js* lze nalézt ve složce *js*, která je, stejně jako složka *css*, rozdělena na složku *min*. Fonty jsou uloženy ve složce *fonts*. V práci jsou použity fonty Montserrat.

## 7.2 Přihlašování a registrace

Pro registraci uživatele slouží tlačítko Registrovat v menu v horní části stránky. Uživatele přesměruje na stránku šablony *register*. Uživatel pak musí postupně vyplnit povinné položky a v závěru musí souhlasit se zpracováním osobních údajů.

Prvně vyplňuje svůj email a heslo. Dále pak musí znovu opsat své heslo, pro kontrolu, zda se shoduje s původně zadaným. Pokračuje svým jménem a krajem, který vybírá z předpřipraveného selectboxu obsahující všech 14 krajů ČR. Poté mu již zbývá vyplnit město, ulice, PSČ a telefonní číslo. Žádná z těchto hodnot se ale neodešle dříve, než uživatel potvrdí, že souhlasí se zpracováním osobních údajů. Jakmile toto potvrdí, tlačítkem „Registrovat“ provede registraci. V tu chvíli se provádí dodatečná kontrola, zda již e-mail v systému náhodou neexistuje. Pokud ne, je uživateli poslána na e-mail URL adresa, kterou když otevře, dojde k ověření uživatele a nový uživatelský účet se tímto plnohodnotně zpřístupní. Heslo se do databáze pro větší bezpečnost ukládá šifrovaně pomocí příkazu `Nette\Security\Passwords::hash('heslo')`.

Do systému se uživatel přihlašuje pomocí svého e-mailu a hesla, které zadal při registraci. Pomocí zadaného e-mailu je v databázi nalezen příslušný uživatel, vůči kterému se pak porovnává zadané heslo se zašifrovaným heslem uvnitř databáze. Jakmile je vše úspěšně ověřeno, dojde k přihlášení uživatele v roli *member*, díky které získá větší oprávnění pro práci se stránkou (viz kapitola 5.7). V systému existuje i další typ uživatele – *admin*. Tímto typem se ale uživatel stane pouze ruční změnou přímo v databázi. Jak bylo zmíněno v kapitole 5.7, tento typ uživatele má ještě více možností, do systému se přihlašuje stejně jako řadový člen.

## 7.3 Hlavní strana

Zobrazení hlavní strany zajišťuje *HomepagePresenter.php*, který předává data do šablony *index.latte*. Na hlavní stránce se zobrazují 2 typy sliderů, z čehož prvním typem je hlavní slider v horní části stránky, který má vždy viditelnou pouze jednu fotografii na jednom slidu. Druhým typem je slider, ve kterém jsou vidět nejnověji přidané produkty ve čtyřech nejoblíbenějších kategoriích (zelenina, ovoce, maso a mléčné výrobky). Takových sliderů je na stránce 4 (pro každou kategorii jeden) a každý z nich zobrazuje na jednom slidu až 4 inzerované produkty. Jednotlivé slidery jsou implementovány pomocí knihovny OwlCarousel a každý z nich má své vlastní ID.

Ve vrchní části stránky se zobrazuje slider s úvodními fotografiemi, jak již bylo výše zmíněno. Tento slider slouží převážně pro reklamní či sdělovací účely. Pod správou ho má vždy pouze administrátor webu. Pokud by chtěl měnit jeho obsah, může tak učinit v administrátorském profilu pod tlačítkem „Správa webu“, kde je možnost změnit tuto fotografii. Odkaz vede na stránku *slider-photo* (šablona *slider-photo.latte*), kde jsou zobrazeny všechny akutálně použité fotografie. Ty je možné smazat pomocí handle `DeletePhoto`, který si jako parametr bere ID dané fotografie. Fotografie je možné i přidávat pomocí komponenty v *HomepagePresenter.php* s názvem `ComponentSliderForm`. V té je možnost fotku přidat a uložit ji. Ještě před samotným uložením probíhá kontrola, zda je fotografie ve správném formátu (povolené formáty jsou JPG, PNG nebo GIF). To se ověřuje pomocí validačního pravidla pro Nette, metodou `addRule`.

Pod sliderem je malý souhrn toho, jak si stránka vede. Tento krátký souhrn vyjadřuje číselná hodnota počtu celkově registrovaných uživatelů, počet nabídek v systému a počet komentářů. O tyto číselné výpisy se starají funkce `getProducts()`, `CountRegisterPeople()` a `CountComments()`.

V dalších sliderech se zobrazují nejnovější nabídky ze specifických kategorií produktů (více v kapitole 5.2). Slider však není jen pouhou prezentací, zobrazované náhledy obsahují kromě fotografie daného produktu i cenu a jméno prodejce. Všechny tyto informace k danému inzerátu, včetně fotografie, se nastavují při přidávání inzerátu. Zobrazení správné kategorie zajišťuje funkce `getProductsbyCat()`, která přijímá jako svůj parametr název žádané kategorie pro výpis. Kategorie ve sliderech na hlavní stránce byly vybrány na základě nejčastěji hledaných.

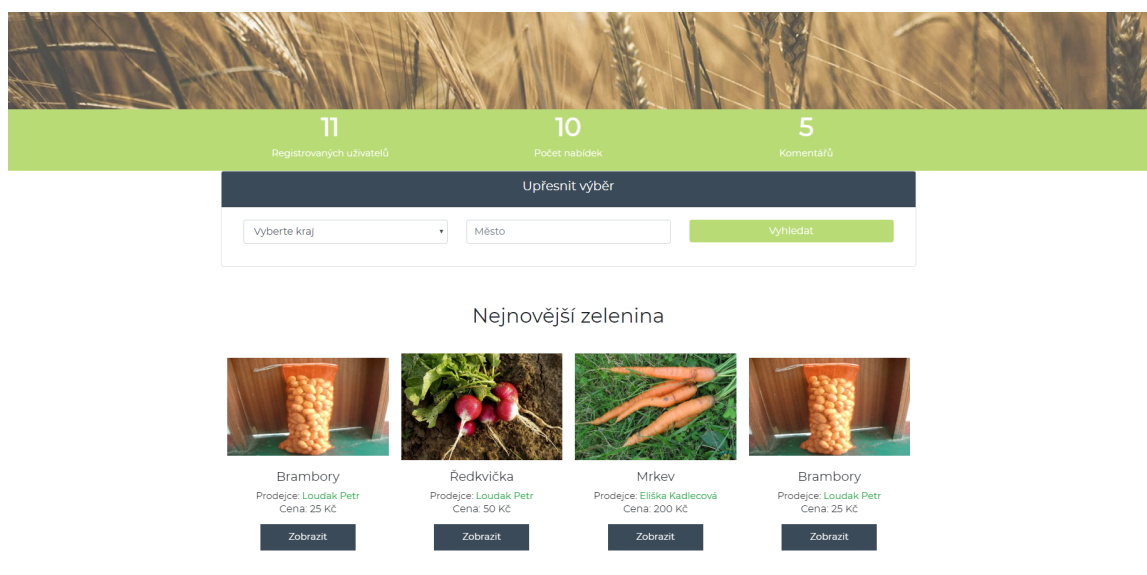
Stránka dále obsahuje možnost filtrování produktů podle zadaného města nebo kraje. Zadané hodnoty jsou předávány komponentou `ComponentHomepageSearch`, která vše zvolené ukládá do proměnné `$form`.

Ukázku výše uvedeného je možné vidět na obrázku 7.1.

## 7.4 Nový inzerát

Stránka s vytvořením nového inzerátu obsahuje několik položek, které lze k produktu vyplnit. Postupně to jsou následující položky: název produktu, dostupné množství produktu (zde je prováděna kontrola, zda není zadávána záporná hodnota pomocí metody `addRule`), dále náležité jednotky (ty uživatel vybírá v elementu `<select>`, v drop-down seznamu, jsou jimi: kusy, kilogramy, gramy, litry, mililitry), které odpovídají typu produktu, trvanlivost produktu, cena produktu, kde je opět kontrolováno, zda není hodnota záporná.

V následující části si může inzerent vybrat, zda je produkt třeba osobně vyzvednout či je možný dovoz na adresu. U osobního odběru je možné vyplnit dostupné dny (výběr se provádí pomocí elementu `<input>` typu checkbox) plus poznámku pro zákazníka. Rozvoz



Obrázek 7.1: Ukázka hlavní strany systému – část úvodního slideru, data o stránce, upřesnění výběru dle kraje a města a především slider s výpisem nejnovějších produktů z kategorie „Zelenina“

na adresu umožňuje vyplnit cenu rozvozu a opět poznámku pro zákazníka. Při vkládání inzerátu je implicitně pomocí CSS deaktivován blok inputů, do kterých se zaznamenávají informace ohledně způsobu převzetí. Aktivují se kliknutím na checkbox. Pokud uživatel zaklikne osobní odběr, aktivuje se možnost zadat dostupné dny + poznámku k odběru, pokud klikne na checkbox s rozvozem na adresu, aktivuje se input s cenou dovozu a poznámkou. Tato funkcionality je zajištěna pomocí JQuery:

```
if ($('#navez_checkboxu').is(':checked')) {
    $('#.navez_tridy').addClass("active");
} else {
    $('#.navez_tridy').removeClass("active");
}
```

Platí, že musí být alespoň jedna z možností dovozu vyplněna, jedná se o povinný atribut. Pokud není ani jedna z možností zaškrtnuta, data se do databáze neodešlou a zobrazí se uživatelská chyba oznamující povinnost zaškrtnutí tohoto checkboxu. Produkt je dále potřeba zařadit do kategorie, které jsou ve formě selectboxu k dispozici. Mezi další informace, které jsou však již nepovinné, patří tipy na ideální skladování nabízeného produktu, informace o tom, jak byl produkt chován, pěstován či vyroben, dále pak obecný popis produktu. V závěru lze přidat fotografie produktu, maximální počet je 5 fotografií, přičemž první fotografie je zároveň fotografií titulní, která se zobrazuje vždy v náhledu produktu a to i například ve slideru na hlavní stránce.

Všechny tyto informace se odesílají do databáze přes „submit button“, který volá funkci `insertProductFormSucceeded ($addproduct, $values)`, kde parametr `$addproduct` je název formuláře a pomocí `$values` se předávají hodnoty z inputů tohoto formuláře. Pro checkboxy, které slouží pro označení dostupných dnů využívám php funkci `implode`, která umožňuje oddělovat jednotlivé hodnoty v databázi, v mém případě čárkou. Pro samotné vložení dat do databáze využívám příkazu `INSERT`.

Ukázku celé stránky s přidáváním nového produktu je možné vidět na obrázcích 7.2 a 7.3.

### Pro přidání inzerátu vyplňte následující údaje

Název Produktu:

Dostupné množství:

Zvolte jednotku:

kusy ▼

Trvanlivost:

dd.mm.rrrr

Cena produktu:

### Níže vyberte preferovaný způsob dodání

☐ Osobní odběr

Po zaškrtnutí osobní odběru se Vám zpřístupní další možnosti

Zaškrtněte dostupné dny odběru:

☐ Pondělí ☐ Úterý ☐ Středa ☐ Čtvrtek ☐ Pátek ☐ Sobota ☐ Neděle

Poznámka:

☐ Rozvoz na adresu

Cena dovozu:

Obrázek 7.2: Ukázka formuláře pro přidání inzerátu - 1. část

## 7.5 Můj účet

Osobní účet každého přihlášeného uživatele má na starost *HomepagePresenter.php*. Na této stránce se nachází možnost upravit údaje přihlášeného uživatele, které byly zadány při registraci, včetně přidání profilové fotografie. Pro aktualizaci hodnot v databázi je vykreslován uživateli formulář, ve kterém jsou zobrazeny všechny aktuální hodnoty z databáze. Má tedy přímo možnost vidět své údaje a ty pak podle toho upravit. Pro provedení aktualizace dat v databázi jsem využil příkazu `UPDATE`.

Ukázku z této části je možné vidět na obrázku 7.4.

V účtu je dále možné zobrazit všechny uveřejněné nabídky, které je možné z tohoto místa také mazat či upravit jejich hodnoty a údaje. K mazání využívám funkci `handleDelete`, který na vstup přijímá jako parametr ID produktu. Při úpravě produktu bylo nutné myslet na to, že pokud si uživatel při vytváření produktu zvolil jako formu předání i osobního

☐ Rozvoz na adresu

Cena dovozu:

Poznámka:

Zvolte kategorii:

Ovoce

Jak Váš produkt ideálně skladovat?:

Způsob pěstování/chovu produktu:

Popis produktu:

Přidat fotografie Soubor nevybrán

Přidat inzerát

Obrázek 7.3: Ukázka formuláře pro přidání inzerátu - 2. část

odběr, byly dostupné dny ukládány do databáze pomocí funkce `implode` s čárkou mezi jednotlivými dny. Při úpravě tedy musím zpětně čárky odstranit, aby byly zaškrtnuty přesně ty checkboxy, které si uživatel původně při vytváření určil. Toho dosahuji reverzní funkcí oproti `implode`, funkcí `explode`, která přidané čárky odstraňuje.

Uživatel ve svém účtu může vidět všechny jím uskutečněné nákupy, včetně toho, zda byl jejich nákup farmářem schválen či nikoliv. Pokud farmář zatím nijak nezareagoval (v databázi není žádná hodnota), zobrazuje se nápis „Čekajte“. Jakmile farmář objednávku schválí, do databáze se uloží číslo 1. To způsobí, že se zájemci o produkt zobrazí v jeho účtu informace o tom, že objednávka byla schválena, zamítnutí posílá do databáze číslo 2, v tom případě je pak u objednávky v účtu napsáno „Zamítnuto“, zároveň se na sklad přičte zpět počet poptávaného zboží. Farmář své rozhodnutí může kdykoliv dodatečně změnit, tzn. zamítnutou poptávku může potvrdit, stejně tak naopak. Tato rozhodnutí může farmář provádět taktéž na stránce „Můj účet“ pod záložkou „Zájemci o nákup“. Jednotlivé stavy zamítnutí/potvrzení/čekání jsou graficky odlišeny tak, aby byly maximálně jednoznačné. O výpis objednávek v inzerentově účtu se stará tento dotaz do databáze:

```
public function getMyShop($id_user)
{
    return $this->database->query
    ("SELECT * FROM 'objednavka' INNER JOIN user ON
    objednavka.id_prodejce = user.id WHERE
    objednavka.id_prodejce = ?", $id_user);
}
```

Prvním příkazem v předchozím dotazu na databázi je příkaz **SELECT**. Ten slouží k definování sloupce v tabulce. V tomto případě je místo názvu sloupce znak hvězdička, která

Změna údajů

Moje nabídky  
Moje nákupy  
Zájemci o nákup  
Statistiky  
Propojení účtů

Vaše jméno:

Vaše příjmení:

Váš email:

Vaše telefonní číslo:

Město bydliště:

Ulice bydliště:

Číslo popisné:

PSC:

Obrázek 7.4: Ukázka změny zadaných údajů v sekci „Můj účet“

značí, že vybírám všechny sloupce z dané tabulky. Pro vypsání konkrétních sloupců by příkaz vypadal následovně: `SELECT sloupec1, sloupec2...`. Další klauzulí, která je společně s příkazem `SELECT` při dotazování povinná, je klauzule `FROM`.

Pomocí klauzule `FROM` jsou určeny tabulky, na které je v databázi dotazováno. V uvedeném příkladě výše to je tabulka s objednávkami.[6] Další částí dotazu na databázi je tzv. vnitřní spojení pomocí klauzule `INNER JOIN`.

`INNER JOIN` vrací vždy jen takové záznamy, u kterých se hodnoty spojovacích atributů přesně shodují. Záznamy, které nejsou shodné, jsou z výsledků vyloučeny. Syntaxe pro `INNER JOIN` je

```
SELECT [*] FROM jmeno_prvni_tabulky INNER JOIN
jmeno_druhe_tabulky ON podminka_spojzeni
```

Posledním příkazem v příkladu je hojně využívaná nepovinná klauzule `WHERE`. Ta slouží k definování podmínek při dotazování se z databáze.

Pokud se tedy původní dotaz shrne, jde vlastně o výběr ze všech sloupců z tabulky *user* a z tabulky *objednavka*, kde jsou spojena stejná ID prodejce v objednávce s ID uživatele v tabulce *user*, který produkt nabídl v inzerci. Podmínka je, že musí být ID prodejce shodné s ID userem, který je předáván parametrem `$id_user`.

Poslední možností v rámci osobního účtu uživatele je zobrazení osobních statistik, které udávají informace o úspěšnosti prodeje (více o statistikách bylo zmíněno v kapitole 5.5).

V dolní části v rámci stránky osobního účtu přihlášeného uživatele lze vidět krátký výpis toho, co se veřejně zobrazuje ostatním uživatelům i nepřihlášeným návštěvníkům. Na tento veřejný profil se ostatní uživatelé mohou dostat přes proklik z kteréhokoliv produktu, který daný uživatel nabízí. Pro zobrazení celého detailu, tak, jak ho vidí ostatní, slouží tlačítko „Zobrazit profil“. Uživatelé vidí základní údaje o uživateli (ID, jméno, telefon, e-mail, datum registrace, město působení a PSČ), dále je zde mapa (vložená Google mapa), která zobrazuje přesnou lokalitu farmáře, jež byla vyplněna při registraci. Pod těmito informacemi jsou vypsané všechny nabídky tohoto uživatele.

## 7.6 ProductPresenter

*ProductPresenter.php* se stará především o výpis detailu produktu. Konkrétní produkt se zobrazuje pomocí volání funkce `getProductByID($id_produkť)`, který ve svém parametru požaduje konkrétní ID produktu. Každý detail obsahuje informace, jako je název produktu, kategorie, do které produkt spadá, průměrné hodnocení farmáře, datum přidání inzerátu, jméno prodejce, které má zároveň proklik na profil farmáře, dále pak cena, množství, které je k dispozici, trvanlivost, dostupné dny odběru, pokud je k dispozici osobní odběr u farmáře, popis produktu, pokud je nějaký vyplněn, způsob skladování, pokud je uveden a způsob pěstování, pokud je k dispozici. Pokud farmář vložil fotografie produktu, zobrazuje se jedna velká hlavní titulní fotografie a pod touto náhledy ostatních 4 fotografií. Po kliknutí na fotografii se fotografie zobrazí pomocí nástroje *Fancybox*.

Fancybox je nástroj, který je založen na jQuery knihovně (více v kapitole 2.2.5). Díky němu se můžou zobrazit obrázky, html obsah či jiná multimédia v pop-up okně. Nejčastěji je využíván právě pro obrázky, stejně tak, jako je tomu v této práci. Tím, že má každý obrázek v detailu produktu v atributu `data-fancybox` napsán stejný název, tvoří tak obrázky jednu souvislou galerii, kterou lze celou procházet pomocí šipek, které se zobrazí po otevření fotografie.

Jak bylo zmíněno dříve, ostatní uživatelé mohou vidět průměrné hodnocení daného prodejce. Průměrné hodnocení využívá funkci `avgRating($id_produkť)`, kde je v parametru opět potřeba ID konkrétního produktu. V SQL příkazu se využívá přímo funkce `AVG`, která průměr sloupce v databázi automaticky spočítá.

Detail produktu obsahuje taktéž tlačítko pro nahlášení produktu, které může využít jakýkoliv přihlášený uživatel a tlačítko pro smazání uživatele, které je přístupné pouze pro administrátora webu. To, zda je uživatel administrátor rozpoznáváme následujícím příkazem:

```
{if $user->isInRole('admin')} {/if}
```

Na stránce s detailem produktu lze pak dále nalézt formulář pro objednání produktu, kde si uživatel může vybrat formu odběru, pokud je k dispozici jak osobní odběr, tak i dovoz na adresu. Forma dostupného odběru je ověřována a podle toho se do selectboxu u objednávky automaticky vloží jen ten způsob odběru, jaký je dostupný. Je to proto, aby nebyl uživatel zbytečně zatížen chybovými hláškami o tom, že si zaklikl formu dovozu, která není dostupná. Ve formuláři musí dále uvést počet poptávaných jednotek a dále má i možnost napsat doplňující poznámku k objednávce. Hodnoty se odesílají opět pomocí „send“ tlačítka (typu „submit“), které volá funkci `insertProductFormSucceeded($reqproduct,$values)`, kde `$reqproduct` je název formuláře a `$values` hodnoty z inputů či selectboxu. Ukázku poptávkového formuláře je možné vidět na obrázku 7.5.



## Možnosti odběru zboží:

Prodejce nabízí dovoz na Vaši adresu

Cena dovozu zboží na adresu:

Doplňující informace k dovozu:

Zdarma dovoz

Chci poptat!

### Poptávkový formulář

Počet jednotek:

Způsob odběru:

Poznámka k objednavce:

Poptat u farmáře!

Obrázek 7.5: Ukázka formuláře pro poptání produktu

Jakmile je poptávka úspěšně odeslána, přijde inzerentovi upozornění na jeho email, který zadal při registraci o tom, že má novou poptávku na své zboží a měl by jít tedy na stránku, aby na objednávku zareagoval buď jejím schválením či zamítnutím.

Doplňující komponentou u každého detailu produktu je možnost pokládat otázky farmáři nebo psát hodnocení farmáře, které se pak promítá do výše zmíněného průměrného hodnocení.

Hodnocení jsem vyřešil klasickou cestou hvězdiček – čím méně hvězdiček, tím větší nespokojenost. Obrázky hvězdiček řeším přes Font Awesome (viz kapitola 2.1.3). Každá hvězdička je skrytý „radio button“, kde každý input obsahuje podle hvězdičky hodnoty od 1 do 5. Jakmile tedy uživatel hvězdičku zaškrtně, zaškrtně tím vlastně radio button o konkrétní hodnotě, která se pak posílá do databáze a z kterých se pak tvoří celkový průměr. Radio button má v tomto případě tu výhodu, že v rámci jedné skupiny radio buttonů, může být vždy zaškrtnut maximálně jeden, což je hlavní rozdíl oproti checkboxu, kde v rámci jedné skupiny mohou být zaškrtnuty klidně všechny, které jsou k dispozici. Pokud uživatel přejíždí přes hvězdičky, vybarvuje se žlutě nejen ta, na které má kurzor, ale i všechny hvězdičky předchozí. Toho jsem docílil využitím JQuery události `.on(change)`. Při najetí se k hvězdičkám přidává třída `.checked`, která se zase automaticky odstraní po sjetí kurzorem mimo hvězdičky.

Každý dotaz lze také nahlásit administrátorovi, pokud by měl někdo pocit, že je jeho obsah nevhodný. Komentáře může uživatel sám vytvořit nebo může odpovědět na již existující. Taková odpověď se automaticky zarovná pod hlavní komentář. Pokud se jedná o hlavní komentář či o odpověď na něj se pozná podle hodnoty v tabulce `comments` v databázi. Hlavní komentář se totiž ukládá do atributu `id_comment` s hodnotou 0, kdežto pokud



se jedná o odpověď, uloží se do tohoto atributu místo hodnoty 0 ID hodnota primárního klíče původního hlavního komentáře.

## 7.7 Nahlášené produkty, nahlášení uživatelé, nahlášené komentáře

Jedna ze sekcí, kam má přístup jen uživatel s rolí admina, se nazývá „Nahlášené produkty“. Lze nalézt v drop-down menu v tlačítku „Správa webu“. To, zda se jedná o admina, ověřuji podmínkou `if $user->isInRole('admin')`. Stránka obsahuje vždy tabulku, kde je vidět ID nahlašujícího uživatele, datum nahlášení ve formátu DMY<sup>1</sup>, odkaz na nahlášený produkt a tlačítko, kterým potvrdí, že se tím zabýval („Přečteno“). Administrátor může smazat produkt přímo ze stránky s detailem produktu. Tím se smaže i záznam v tabulce s nahlášenými produkty. Mazání probíhá příkazem `delete`. Pokud nejsou dány žádné omezující podmínky, pak se tímto příkazem smažou všechny řádky tabulky. V práci je v podmínce `WHERE` definováno, že se musí jednat o stejné ID produktu, jako je dáno na vstup z parametru `$id_produk`t. Společně s produktem se mažou související záznamy (ty, které mají v řádku ID tohoto produktu) tabulky `produkt_photo` a související záznamy tabulky `objednavky`.

Datum nahlášení vytvářím pomocí funkce `date_create()` ve formátu YMD<sup>2</sup>, ten pak zpětně v šabloně převádím do formátu DMY.

Stejným způsobem pak fungují tabulky s nahlášenými uživateli či komentáři u produktů. Pokud nahlášená záležitost nic neporušuje, stačí kliknout na tlačítko „Přečteno“ a záznam se smaže. Odstranění se provádí pomocí metody „handle“, která podle ID odstraňuje záznam z tabulky nahlášených.

Ukázku tabulky nahlášených uživatelů je možné vidět na obrázku 7.6.

Id nahlašujícího uživatele:	Datum nahlášení:	Detail produktu:	Potvrdit přečtení:
30	03.04.2019	<a href="#">Zobrazit detail produktu</a>	<a href="#">Přečteno</a>
30	13.04.2019	<a href="#">Zobrazit detail produktu</a>	<a href="#">Přečteno</a>

Obrázek 7.6: Ukázka tabulky nahlášených uživatelů

## 7.8 Statistiky

Pro přehledný výpis statistik využívám knihovnu *Chart.js*. Jedná se o open source knihovnu, která slouží pro vykreslení dat ve webových aplikacích. Pro statistiky jsem si vytvořil tabulku *statistiky*, kde uchovávám cenu objednávky, ID uživatele, který objednávku provedl a kategorii produktu, do které patří. Hodnoty se do šablony předávají z *HomepagePresenter.php* a jsou ukládány do proměnných, které se pak dále předávají pomocí JSON přímo do ovládacího skriptu, který je umístěn v souboru *custom.js*.

Pro zobrazení statistik jsou použity grafy koláčové a liniové. Liniové pro zobrazení nejčastěji nakupujících lidí, koláčové pro zobrazení nejčastěji prodávaných produktů a pro

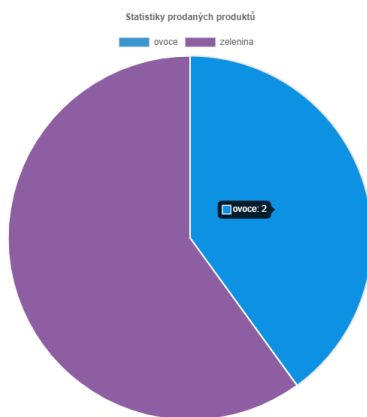
<sup>1</sup> DMY: Day-Month-Year – den/měsíc/rok

<sup>2</sup> YMD: Year-Month-Day – rok/měsíc/den

zobrazení výdělku za jednotlivé kategorie, z čehož může dotyčný farmář zjistit, o jaké produkty je největší zájem. Mimo graf se jako poslední údaj zobrazuje informace o celkovém výdělku. Každý graf má své vlastní unikátní ID, které se využívá k propojování hodnot k daným grafům.

Ukázka statistik ze systému je níže na obrázku 7.7.

Nejčastěji poptávaný produkt



Výdělek z prodaných produktů



Obrázek 7.7: Ukázka statistik prodejů uživatele

## 7.9 Kooperace mezi účty

V systému si může vlastník přidávat do svého účtu brigádníky. Tito lidé mají stejná oprávnění, jako má v systému on sám, s tím rozdílem, že brigádník nemůže měnit uživatelské údaje, které si vlastník vyplnil při registraci a dále pak nemůže zobrazovat statistiky prodeje vlastníka účtu. Uživatelé, kteří mají oprávnění nakládat s cizími účty, uchovávám v tabulce *user\_cooperate*. Pro přidání brigádníka je potřeba znát e-mail dotyčného. Pokud je e-mail brigádníka nalezen mezi e-mailovými adresami uživatelů v databázi, je přiřazen k účtu vlastníka. Pokud si pak brigádník otevře ve svém účtu stránku s názvem „Propojení účtů“, nalezne tam seznam účtů, které spravuje. Pro zobrazení účtů, které brigádník spravuje, slouží proměnná *coopAccounts*, kterou pomocí příkazu *foreach* procházím a vypisuji. Pro zobrazení brigádníků, které má vlastník účtu přiřazen, slouží proměnná *addedAccounts*. Do vlastníka účtu se brigádník dostane přes odkaz v podobě jména vlastníka, který se vypisuje pomocí *coopAccounts*. Po kliknutí na odkaz se do parametru *coop* odešle TRUE. Díky tomu může provádět již výše zmíněné operace za vlastníka účtu. Do svého účtu se brigádník vrátí po kliknutí na odkaz „Můj účet“ v menu.

Na obrázku 7.8 je ukázka ze systému, z části, kde jsou vypsané informace o propojení účtů.

Propojení účtů

Připojené účty

Vyhledat

Vyhledat

Přidání uživatele:

Eliška Kadlecová

Tvé účty, které spravuješ:

Loudak Petr, Eliška Kadlecová, Hana Doubek

Obrázek 7.8: Ukázka výpisu propojených účtů – vlevo nahoře textové pole pro vyplnění e-mailu účtu, kterému chce uživatel povolit spravování svého účtu, vpravo nahoře vypsány účty, které tento uživatel spravuje, vlevo dole výpis uživatelů, kteří mají přístup ke správě tohoto účtu

## 7.10 Responzivita

V dnešní době by měl být každý správný web responzivní, což znamená, že jeho obsah bude přizpůsobený na konkrétní rozlišení, ve kterém je web zobrazován. Dříve se totiž weby zobrazovaly na telefonech stejně, jako se zobrazovaly na desktopových zařízeních, což vedlo k tomu, že si uživatel musel obsah různě přibližovat a měl mnohdy problém s čitelností webu nebo s ovládacími prvky na stránce. Vytvořená stránka je testována na všechna běžná rozlišení či zařízení. Pro představu je web odladěn na rozlišení 4K (3840x2160 px), FullHD (1920x1080 px), HD+ (1366x768 px), HD (1280x720 px - typické rozlišení pro tablety) a na 320x480 px, což je rozlišení, které se týká starších telefonů. Stránky jsou testovány na běžným prohlížečích, jako je Google Chrome (Verze 72.0.3626.121), Mozilla Firefox (Verze 64.0) a v neposlední řadě na Microsoft Edge (Verze 42.17134.1.0).

## Kapitola 8

# Testování

Tato kapitola popisuje, jakým způsobem byl informační systém testován. Vypíšu pár základních metod testování, které vždy krátce představím a popíši, jak souvisí s vývojem tohoto systému.

### Focus Group

Metoda, který získává data od vybrané skupiny uživatelů (většinou 6 – 10 lidí) formou diskuze na konkrétní téma. Taková akce je zpravidla vedena moderátorem, který pokládá otázky či koriguje diskuzi. Typicky je prováděna na začátku vývoje, kdy se od uživatelů zjistí, jaké mají požadavky na systém či co by jim přišlo naopak nevhodné. Nevýhodou může být ostýchavost některých členů, kteří neřeknou vše, co chtějí, případně říkají, co si myslí, že chtějí ostatní slyšet. Výhodou je nízká finanční náročnost takového testu.

Tento test jsem provedl se třemi farmáři a 4 uživateli. Tato diskuze proběhla již v začátku vývoje, proto jsem měl hrubou představu toho, co od systému očekávají a jaké funkce by ocenili. Z tohoto bylo posléze vycházeno u kapitoly 4 o Analýze požadavků.

### Prototypování

K otestování správného rozložení jednotlivých prvků na stránce, alias uživatelského rozhraní, je využívána technika prototypování. Z analýzy požadavků od klienta je běžně tvořena tzv. skica, která zastává pracovní verzi finální podoby výsledné práce, typicky se jedná o výkresy na papír tužkou. Další fází je tzv. wireframe, jinak také označován jako drátěný model. Ten je typicky vytvářen v softwaru na počítači. Wireframe neobsahuje žádnou grafiku, slouží pouze k tomu, aby se o všech nápadcích, které vyplynuly na papír z předchozí fáze skicování, znovu zapřemýšlelo a odstranily se případné nedostatky. Mnoho lidí vytváří rovnou wireframe, což není dobře, protože právě skicování nám umožňuje pohotově všechny nápady realizovat. Běžně je toto rychlé zaznamenávání všech myšlenek označováno cizím slovem jako „brainstorming“. Výsledkem předchozích dvou fází je tzv. prototyp. Prototyp neobsahuje žádné funkce, slouží pro uživatelské testování. Správný prototyp obsahuje návrh a vzájemné provázání všech stránek, které se na webových stránkách budou zobrazovat. Uživatelům jsou pak zadány určité úkoly u jejichž plnění jsou pozorováni. Podle toho, jak jsou úkoly plněny intuitivně, je systém předěláván a přizpůsobováno rozmístění ovládacích prvků.

Aby se zajistilo co nejlepší rozložení jednotlivých prvků na webu, průběžně jsem prováděl testování na několika jedincích, kterým jsem zadával různé úkoly k plnění a sledoval je při

tom. Díky tomu jsem přesně věděl, zda jsou tlačítka a ovládací prvky vhodně umístěna, případně na čem je potřeba zapracovat.

## Kapitola 9

# Závěr

Cílem této práce bylo vytvořit informační systém pro malochovatele či malopěstitele, který jim umožní jednoduše inzerovat své produkty, které budou moci jiní uživatelé systému pohodlně objednat.

Nejprve jsem vytvořil analýzu požadavků, neboli požadavky reálných budoucích uživatelů (farmářů i potenciálních nakupujících uživatelů), od kterých jsem zjistil, co by v systému uvítali a co by v něm potřebovali. S těmito lidmi jsem byl v kontaktu po celou dobu své práce, takže jsem měl cennou možnost získávat přímou zpětnou vazbu, která vedla k průběžnému ladění chyb či vyjasnění nedorozumění při specifikaci požadavků. Výstupem analýzy požadavků byl digram případů užití - use-case diagram.

Na základě těchto požadavků jsem navrhl strukturu databáze pomocí ER diagramu, který obsahuje celkem 12 tabulek. Díky ER diagramu jsem znázornil vzájemné vazby mezi tabulkami v databázi.

Bylo také nezbytné vybrat technologie vhodné k implementaci takového systému. Po prozkoumání možností byl zvolen jazyk PHP, resp. jeho framework Nette. Pro implementaci uživatelského rozhraní byl zvolen nástroj Bootstrap 4.

Na základě návrhu a grafického návrhu uživatelského rozhraní, který byl průběžně konzultován s potenciálními uživateli, jsem implementoval požadovaný systém. Uživatelé typu *member* (řadoví uživatelé) se mohou sami registrovat a přihlašovat, nabízet své produkty, vytvářet vlastní objednávky, spravovat objednávky, komentovat produkty, hodnotit ostatní uživatele, případně nahlašovat produkty, komentáře nebo celé profily. Uživatelé také mohou propojovat účty s dalšími, které budou moci spravovat jejich nabídky. V systému také existuje ještě jeden typ uživatele (*admin*), který má v systému práva navíc, díky nimž může mazat cizí produkty, nevhodné komentáře nebo nahlášené profily.

Systém byl, jak jsem zmínil, v průběhu vývoje testován, aby co možná nejvíce odpovídal požadavkům reálných uživatelů.

Zajímavou možností dalšího vývoje systému by bylo vytvoření blogu, do kterého by správce webu psal různé zajímavosti o domácích produktech, upozorňoval na blížící se zajímavé akce a podobně. S tím by mohl přijít i newsletter pro uživatele. Systém v současné době zpracovává objednávky, ovšem platba v něm již neprobíhá. Do budoucna by tedy byla pro uživatele zajímavá i implementace online platby, aby zákazníci mohli zaplatit za produkty přímo v systému.

# Literatura

- [1] *Lekce 1 - Úvod do Nette frameworku pro PHP*. [Online].  
URL <https://www.itnetwork.cz/php/nette/zaklady/uvod-do-php-frameworku-nette/>
- [2] *Primární klíč, cizí klíč, referenční integrita, pravidla normalizace, relace*.  
URL [http://www.kubenka.org/PEF/1\\_rocnik/Internetove\\_technologie/tema\\_2\\_opory\\_a\\_soubory/cviceni\\_2\\_2\\_klice\\_a\\_relace.pdf](http://www.kubenka.org/PEF/1_rocnik/Internetove_technologie/tema_2_opory_a_soubory/cviceni_2_2_klice_a_relace.pdf)
- [3] *So what is CSS, really?* [Online].  
URL [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics)
- [4] Bílek, P.: *Entity-relationship diagram*. [Online].  
URL <https://www.sallyx.org/sally/psql/erd.php>
- [5] Gutmans, A.; Bakken, S. S.; Rethans, D.: *Mistrouství v PHP 5*. Brno: CP Books, 2005, ISBN 80-251-0799-X.
- [6] Luboslav Lacko: *SQL*. Brno: Computer Press, 2003, ISBN 80-722-6975-5.
- [7] Michálek, M.: *CSS3 Flexbox*. [Online].  
URL <https://www.vzhurudolu.cz/prirucka/css3-flexbox>
- [8] Skřivan, J.: *SQL – tvorba tabulek*. [Online].  
URL <https://www.interval.cz/clanky/sql-tvorba-tabulek/>
- [9] Unger, R.: *A project guide to UX design*. Berkeley, CA: New Riders, second edition vydání, [2012], ISBN 978-0-321-81538-5.
- [10] Žára, O.: *JavaScript*. Brno: Computer Press, 2015, ISBN 978-80-251-4573-9.