

VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Katedra technických studií

Prezentační web nemocniční lékárny

Pelhřimov

bakalářská práce

Autor práce: Michal Vacho

Vedoucí práce: PaedDr. František Smrčka, Ph.D.

Jihlava 2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce:	Michal Vacho
Studijní program:	Elektrotechnika a informatika
Obor:	Aplikovaná informatika
Název práce:	Prezentační web nemocniční lékárny Pelhřimov
Cíl práce:	Redakční systém bude obsahovat klasickou prezentaci nemocniční lékárny (úvod, služby, výdejní místa, kontakt, ...). Web bude s patřičným grafickým rozhraním. Stránky dále budou obsahovat mapu s polohou lékárny pro jednodušší navigaci. Na úvodní stránce se budou zobrazovat aktuální speciální nabídky akcí a důležité informace, které se budou řadit podle data zveřejnění a zadané priority pořadí. Web bude také podporovat zobrazení v mobilním zařízení. Web bude mít dvě úrovně přístupu: administrátor a lékárník. Administrátor bude moci editovat celý web. Lékárník bude moci přidávat a upravovat články na úvodní stránce. Uživatel bude vidět informace bez přihlášení.

PaedDr. František Smrčka, Ph.D.
vedoucí bakalářské práce

doc. Ing. Zdeněk Horák, Ph.D.
vedoucí katedry
Katedra technických studií

Abstrakt

Cílem této bakalářské práce bylo vytvořit prezentační web nemocniční lékárny v Pelhřimově. V úvodní části je seznámení s úvodním problémem, vymezení cílů práce, rešerše možných řešení a použitých technologií. Další část se zabývá analýzou problému a popisem implementačních částí aplikace. Poslední částí je návod na správu aplikace a testování systému ze strany Nemocnice Pelhřimov a odstranění nedostatků. Celý systém je vytvořen v jazyce PHP s pomocí frameworku Nette. Dále využívá databázový systém MySQL.

Klíčová slova

Redakční systém; webová aplikace; PHP; MySQL; Nette

Abstract

The goal of this Bachelor thesis was to create a presentation website for Pelhřimov Hospital's pharmacy. The first part focuses on the introduction of the main problem, determining the goals and recherche of possible solutions and usable technology. The following part deals with analysis of the problem and description of implementing pieces of the application. The last part is a manual for maintenance and testing of the system from Hospital's side and the removal of shortcomings. The whole system is created in Nette framework. Database system MySQL is used as well.

Key words

Content Management System; web application; PHP; MySQL; Nette

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „AZ“).

Souhlasím s umístěním bakalářské práce v knihovně VŠPJ a s jejím užitím k výuce nebo k vlastní vnitřní potřebě VŠPJ.

Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje **AZ**, zejména § 60 (školní dílo).

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne 28. dubna 2021

.....

Podpis

Poděkování

Rád bych na tomto místě poděkoval svému vedoucímu bakalářské práce PaedDr. Františku Smrčkovi, Ph.D. za jeho cenné rady a možnost vytvářet práci pod jeho odborným vedením.

Poděkování rovněž patří řediteli Nemocnice Pelhřimov Mgr. Ing. Michalovi Kozárovi, MBA za umožnění zpracování tématu bakalářské práce a za poskytnutí zpětné vazby. Dále celému týmu nemocniční lékárny za poskytnutí materiálů při zpracování práce.

Poděkování patří také celé mé rodině za veškerou podporu poskytovanou po celou dobu mého studia.

Obsah

Úvod	10
2 Popis a cíle práce	11
2.1 Seznámení s problémem	11
2.2 Stanovení cílů a požadavků	11
3 Rešerše možných řešení	12
3.1 WordPress	12
3.2 Joomla!	12
3.3 Drupal	12
3.4 TextPattern	13
3.5 Zhodnocení možných řešení	13
4 Použité technologie a frameworky	14
4.1 PHP	14
4.2 HTML	14
4.3 CSS	14
4.4 Nette	15
4.5 Less	16
4.6 Twitter Bootstrap	17
4.7 jQuery	17
4.8 GIT	17
4.9 Vývojová prostředí	18
5 Analýza řešeného problému	21
5.1 Funkcionalita systému	21
5.2 Datový model	23
5.3 Grafický návrh webu	25
6 Popis implementace	27
6.1 Implementace databáze	27
6.2 Připojení k databázi	28
6.3 Layout webu	28
6.4 Úvodní stránka webu	29
6.5 Responzivita	30
6.6 Výpis článků	31
6.7 Výpis fotogalerie u článku	32
6.8 Vytvoření formuláře s dotazem	32
6.9 Zpracování formuláře	34
6.10 Administrace webu	35

6.11	Autentizace uživatele	35
6.12	Zobrazení přehledu dat v kategoriích.....	37
6.13	Tvorba HTML obsahu stránek.....	37
6.14	Nahrávání jednotlivých fotografií.....	38
6.15	Nahrávání většího množství souborů.....	38
7	Návod na správu webu	39
7.1	Obecná nastavení pro výpis kategorií.....	39
7.2	Vytváření menu	40
7.3	Správa článků	41
7.4	Vytvoření fotogalerie u článku	42
7.5	Správa novinek	42
7.6	Vytváření uživatelů.....	43
7.7	Správa galerie	43
7.8	Administrace akčního letáku	44
8	Testování	45
8.1	Srovnání s existujícími řešeními.....	45
	Závěr.....	46
	Seznam použité literatury.....	47

Seznam obrázků

Obrázek 1: WordPress logo (zdroj: stickpng.com)	12
Obrázek 2: Joomla logo (zdroj: brandslogos.com)	12
Obrázek 3: Drupal logo (zdroj: cs.wikipedia.org).....	13
Obrázek 4: TextPattern logo (zdroj: textpattern.com)	13
Obrázek 5: Nette logo (zdroj: nette.org)	15
Obrázek 6: Ukázka použití definovaných barev (zdroj: vlastní)	16
Obrázek 7: PhpStorm – ukázka IDE (zdroj: vlastní).....	18
Obrázek 8: Xampp – ovládací panel (zdroj: vlastní).....	19
Obrázek 9: GitHub Desktop – ovládací panel aplikace (zdroj: vlastní).....	20
Obrázek 10: Use case diagram (zdroj: vlastní)	21
Obrázek 11: Datový model (zdroj: vlastní).....	24
Obrázek 12: Grafický návrh aplikace - layout (zdroj: vlastní).....	25
Obrázek 13: Grafický návrh - titulní stránka (zdroj: vlastní).....	26
Obrázek 14: Vytvoření databáze (zdroj: vlastní)	27
Obrázek 15: Ukázka založení entity (zdroj: vlastní)	27
Obrázek 16: Připojení k databázi (zdroj: vlastní).....	28
Obrázek 17: Výpis menu (zdroj: vlastní)	28
Obrázek 18: Layout webu (zdroj: vlastní).....	29
Obrázek 19: Výpis galerie (zdroj: vlastní)	29
Obrázek 20: Začátek úvodní stránky (zdroj: vlastní)	30
Obrázek 21: Náhled webu v mobilním zařízení (zdroj: vlastní)	30
Obrázek 22: Zobrazení fotogalerie článků (zdroj: vlastní)	32
Obrázek 23: Zobrazení formuláře (zdroj: vlastní).....	32
Obrázek 24: Přihlášení do administrace (zdroj: vlastní)	35
Obrázek 25: Psaní obsahu s využitím CKEditoru (zdroj: vlastní)	37
Obrázek 26: Zobrazení výpisu dat z dané kategorie (zdroj: vlastní).....	39
Obrázek 27: Filtrování dat v kategoriích (zdroj: vlastní).....	40
Obrázek 28: Zobrazení titulku v kartě prohlížeče (zdroj: vlastní)	41
Obrázek 29: Editace menu (zdroj: vlastní).....	41
Obrázek 30: Úprava článku (zdroj: vlastní)	41
Obrázek 31: Přidání fotogalerie k článku (zdroj: vlastní)	42
Obrázek 32: Úprava aktuality (zdroj: vlastní).....	42
Obrázek 33: Vytváření nového uživatele (zdroj: vlastní)	43
Obrázek 34: Editace fotky v galerii (zdroj: vlastní).....	43
Obrázek 35: Administrace akčního letáku (zdroj: vlastní).....	44

Seznam použitých zkratek

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
SQL	Structured Query Language
URL	Uniform Resource Locator
WWW	World Wide Web
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XHTML	Extensible HyperText Markup Language
VŠPJ	Vysoká škola polytechnická Jihlava
CMS	Content Management System
HTTP	Hypertext Transfer Protocol
SMTP	Simple Mail Transfer Protocol
IMAP	Internet Message Access Protocol
POP3	Post Office Protocol
FTP	File Transfer Protocol
ORM	Object–relational mapping
LESS	Leaner Style Sheets
UTF	Unicode Transformation Format
API	Application Programming Interface
WYSIWYG	What you see is what you get

Úvod

Jedna z nejúčinnějších forem, jak v dnešní době zviditelnit firmu nebo organizaci, je její propagace pomocí internetových stránek. K tomu má nemocniční lékárně v Pelhřimově pomoci nový redakční systém, v kterém budou spravovány webové stránky tamní lékárny. Nemocnice Pelhřimov se nachází kousek od centra města. Nemocniční lékárna již provozuje dvě výdejní místa léčiv pro pacienty nemocnice a problém je, že je o nich veřejnost málo informována, a tak se stává že pacienti odcházejí jinam. Dalším problémem je malá informovanost o výhodných cenových nabídkách lékárny, a tak pacienti v domnění levnějšího nákupu léčivých přípravků odcházejí do konkurenčních velkých řetězců. Nový web s redakčním systémem by měl nedostatky špatné informovanosti odstranit. Měl by informovat o novinkách a akcích a dále ulehčit pacientům hledání provozní doby a správně je navést do výdejních míst pelhřimovské nemocniční lékárny.

Celý systém bude programován v jazyce PHP, který je nejrozšířenější programovací jazyk pro webové aplikace. K vývoji bude použit český framework Nette, který klade důraz na bezpečnost celé aplikace. Framework Nette ulehčuje a zpřijemňuje práci programátorům při vývoji webových aplikací. Nedílnou součástí při programování webových stránek je použití značkovacího jazyku HTML5 s využitím CSS3. Pro lepší práci s kaskádovými styly (CSS) bude využit preprocesor Less společně s knihovnou Twitter Bootstrap. Aplikace bude využívat šablonovací systém Latte, který je součástí frameworku Nette.

2 Popis a cíle práce

Tato kapitola se zabývá stanovením cílů na vyvíjený systém.

2.1 Seznámení s problémem

Momentálně je informovanost o nemocniční lékárně v Pelhřimově pouze z webových stránek Nemocnice Pelhřimov. To je v dnešní době nedostačující, a tak mě vedoucí lékárny po domluvě s ředitelem nemocnice požádala o tvorbu nového webu s redakčním systémem, který bude reprezentovat nemocniční lékárnou.

2.2 Stanovení cílů a požadavků

Cílem této práce je vytvoření webových stránek s redakčním systémem pro nemocniční lékárnou. Základní funkcí webu bude informovat návštěvníky webu o aktuálních informacích a změnách v lékárně a farmacii. Návštěvník webu by měl najít základní informace jednoduše, a proto má být web přehledný s moderním designem. Systém budou využívat především pacienti, aby se informovali o novinkách ve farmacii a speciálních akcích nemocniční lékárny. Každý návštěvník bude mít možnost se informovat o tom, jak lékárna funguje, kudy se do lékárny dostat, informovat se o zajišťované pohotovostní službě a číst si aktuální články a novinky z nemocniční lékárny a farmacie.

Systém bude mít plně responzivní design pro lepší uživatelskou přívětivost. V dnešní době, kdy každý používá mobilní telefon nebo tablet je tato vlastnost již nepostradatelná.

Správa webu bude mít dvě úrovně přístupu. První bude administrátor, který bude mít přístup ke správě celého webu. A druhý přístup bude lékárník, který bude pomoci přidávat a spravovat novinky, aktualizovat akční leták a dále bude mít právo editovat galerie na úvodní stránce.

Systém bude sloužit pouze lékárně Nemocnice Pelhřimov, ale jeho použití by bylo možné využít i v jiných společnostech.

3 Rešerše možných řešení

Na českém trhu je možné najít mnoho firem zabývajících se tvorbou internetových webových stránek. Pořizovací náklady jsou však příliš vysoké pro lékárnu v okresní nemocnici. Je tedy možné využít nějaký systém pro správu obsahu jako je například WordPress, Joomla nebo Drupal.

3.1 WordPress

Je nejpoužívanější Open-source redakční systém pro tvorbu webových aplikací. Je napsaný v PHP a MySQL, který dodržuje standardy XHTML, CSS a XML. Obsahuje mnoho rozšiřujících pluginů, podporuje widgety, témata vzhledu, správu galerií, možnost zařazení příspěvků do kategorií, přidávání štítků a mnoho dalšího. (WordPress 2021)



Obrázek 1: WordPress logo (zdroj: stickpng.com)

3.2 Joomla

Joomla je jeden z nejpopulárnějších Open Source redakčních systémů. Je napsán v jazyce PHP a využívá databázi MySQL. Podporuje caching, indexaci stránek, vícejazyčné verze. Obsahuje kalendář, diskusní fóra, hlasování, blogy, vyhledávání v rámci webu a RSS. (Joomla 2021)



Obrázek 2: Joomla logo (zdroj: brandslogos.com)

3.3 Drupal

Třetím nejrozšířenějším redakčním systémem je Drupal. Je naprogramovaný v PHP, podporuje databáze MySQL, PostgreSQL a SQLite. Lze využít rozšiřujících modulů pro využití dalších

typů databází jako MSSQL, Oracle a MongoDB. Je postaven modulárním způsobem. Podporuje rozšiřující balíčky s předem definovanou funkcionalitou, nachystanými doplňkovými moduly a tématy vzhledu. (Drupal 2020)



Obrázek 3: Drupal logo (zdroj: cs.wikipedia.org)

3.4 TextPattern

Je menší redakční systém založený na PHP. Jedná se o vysoce flexibilní systém, který je snadno použitelný. Podporuje šablony a téma, kterých není tak velké množství jako pro WordPress, Drupal a Joomla. Nabízí téma pro administraci za účelem přizpůsobení uživatelského prostředí. Kategorie pluginů zahrnují galerii obrázků, elektronické obchody, archivy, články, navigaci po webu a funkce administrátora webu. (TextPattern 2004)



Obrázek 4: TextPattern logo (zdroj: textpattern.com)

3.5 Zhodnocení možných řešení

Na internetu je možné najít a použít velké množství redakčních systémů, ale žádný není příliš vhodný pro cíle našeho zadání. A tak jsem se rozhodl pro vlastní vývoj webové aplikace.

4 Použité technologie a frameworky

Tato kapitola se zabývá použitými technologiemi a frameworky při tvorbě závěrečné práce. Dále se zabývá použitými vývojovými prostředími.

4.1 PHP

PHP je skriptovací programovací jazyk. Je určený pro programování dynamických webových stránek a aplikací. Skripty jsou prováděny na straně serveru a uživateli je zobrazen pouze výsledek činnosti skriptu. Jazyk PHP je multiplatformní tedy lze přenášet vytvořený kód mezi operačními systémy a podporuje internetové protokoly jako jsou například HTTP, SMTP, IMAP, POP3, FTP. Podporuje rozšiřující knihovny pro práci s grafikou, textem, souborem a přístupy k databázovým systémům jako jsou MySQL, MSSQL, SQLite, MariaDB, PostgreSQL a další. Pro běh PHP je potřeba mít nainstalovaný Apache HTTP Server (webový server). (PHP 2020)

4.2 HTML

Je značkovací jazyk určený pro tvorbu statických webových stránek a publikací na internetu. HTML je základní programovacím jazykem pro tvorbu hypertextových dokumentů v prostředí WWW. Je vyvíjen společností W3C. Jeho poslední verzí je HTML5, jež rozšiřuje předchozí verze o nové prvky. Ve verzi pět přibýly tagy pro multimediální obsah <video>, <audio> a také nové tagy pro vzhled stránek například: <article>, <header>, <main>, <footer>, <nav>. (HTML 2020)

4.3 CSS

Je jazyk pro formátování jednotlivých elementů webových stránek psaných v jazycích HTML a XML. Pomocí CSS definujeme vzhled a vlastnosti chování jednotlivých prvků pro webové stránky. Je vyvíjen společností W3C. Jeho poslední verzí je CSS3, která přináší nové vlastnosti jednotlivým elementům v HTML jazyce. Verze tři přidala například stínování prvků, zaoblené rohy, rotace a transformace objektu a nové barevné moduly. (CSS 2020)

4.4 Nette

Nette je český open-source framework pro vývoj webových aplikací, je nadstavbou nad PHP. Autorem Frameworku je David Grudl. Jeho hlavní výhodou je, že jsou vydávány bezpečnostní záplaty i po ukončení vývoje dané verze systému. Existují spousty pluginů a doplňků, které usnadňují práci programátorům při vývoji.



Obrázek 5: Nette logo (zdroj: nette.org)

Presenter

Je třída, která představuje konkrétní stránku například pro článek, nebo pro domovskou stránku. Stará se o řízení dané části stránky. Přijímá požadavky od uživatele, které dále zpracovává a následně předává data šabloně, která je uživateli zobrazí.

Šablonovací systém Latte

Latte je šablonovací systém pro PHP, který šetří a zlepšuje práci, také zabezpečuje výstup dat, pro které využívá automatické escapování zobrazovaných dat. Zajišťuje zobrazení samotných webových stránek. Šablony využívají klasické HTML znaky, které jsou doplněny o Latte makra.

Makra

Makra zajišťují výpis a práci s daty, které jsou předávány z presenteru. Umožňují nám vytvářet cykly a podmínky, díky nim můžeme jednoduše provádět formátování textu, čísel a datumu. Dále umožňují vkládat do šablony jiné šablony, které zajišťují dědičnost šablon. Definují proměnné, starají se o řízení výjimek, vykreslují návrhy Nette formulářů.

Model-View-Controller

Je architektura, která rozděluje aplikaci na tři komponenty, a to na model, pohledy a řadiče. Nette tuto architekturu využívá jen s malou obměnou, a to že namísto slova Controller používá Presenter. Model je vrstva, která obstarává práci s daty aplikace a řídí celou logiku aplikace. View neboli pohled se stará o vykreslování obsahu uživateli.

Databáze

Nette obsahuje i knihovnu pro práci s databází. Pro vytvoření spojení je potřeba nadefinovat nové spojení jako instanci třídy Nette\Database\Context. Tato třída mapuje celou databázi, uživatel tak může pracovat s databází jako s objektem. Pro spojování se používají cizí klíče propojované pomocí metody ORM, tato metoda si poté vyřeší další závislosti a propojení samotných tabulek v SQL. Pro práci s databází se nechají využít rozšiřující knihovny jako například Dibi nebo Doctrine, které usnadňují práci s daty.

4.5 Less

Je jazyk pro tvorbu dynamických CSS, do kterých je generován. Oproti CSS umožňuje využívání lokálních proměnných, což se může využít například pro definování barev, které se stále opakují. Dále umožňuje využívat vnořování definic a vnořených pravidel. Využívání předem nadefinovaných funkcí a tzv. mixinů, to je způsob, jak zahrnout spoustu vlastností z jiné sady pravidel do té aktuální. Podporuje také základní matematické operace (sčítání, odčítání, násobení, dělení, ale i některé složitější funkce). Do výsledné podoby souboru .css je generován podle pravidel CSS, při generování výsledného CSS souboru se ještě mohou zahrnout již dříve vytvořené CSS soubory. (LESS 2020)

Definování barev v samostatném souboru .less:

```
@black: #000;  
@dark-green: #014801;  
@green: #42b41e;
```

```
color: @green;  
border-color: @dark-green;
```

Obrázek 6: Ukázka použití definovných barev (zdroj: vlastní)

4.6 Twitter Bootstrap

Bootstrap je sada nástrojů pro tvorbu webových aplikací. Rozšiřuje základní použití HTML a CSS o předem definované šablony. Tyto šablony slouží pro úpravu vzhledu tlačítek, formulářových prvků, vlastností textu, navigace po stránkách. Ve svém základu zavádí možnost využít grid (mřížkový) systém. Tento systém nám dovoluje lépe upravovat vzhled webové stránky tím, že jednotlivým elementům řekne, jak velkou šířku mají zabírat vůči rodičovskému elementu. Toho se dá využít při vytváření responzivního designu. (Bootstrap 2020)

4.7 jQuery

Je javascriptová knihovna s podporou prohlížečů, která klade důraz na propojení mezi HTML a JavaScriptem. Jedná se o sadu funkcí, které velmi rychle a elegantně umí pracovat s událostmi, animacemi a efekty. (jQuery 2020)

4.8 GIT

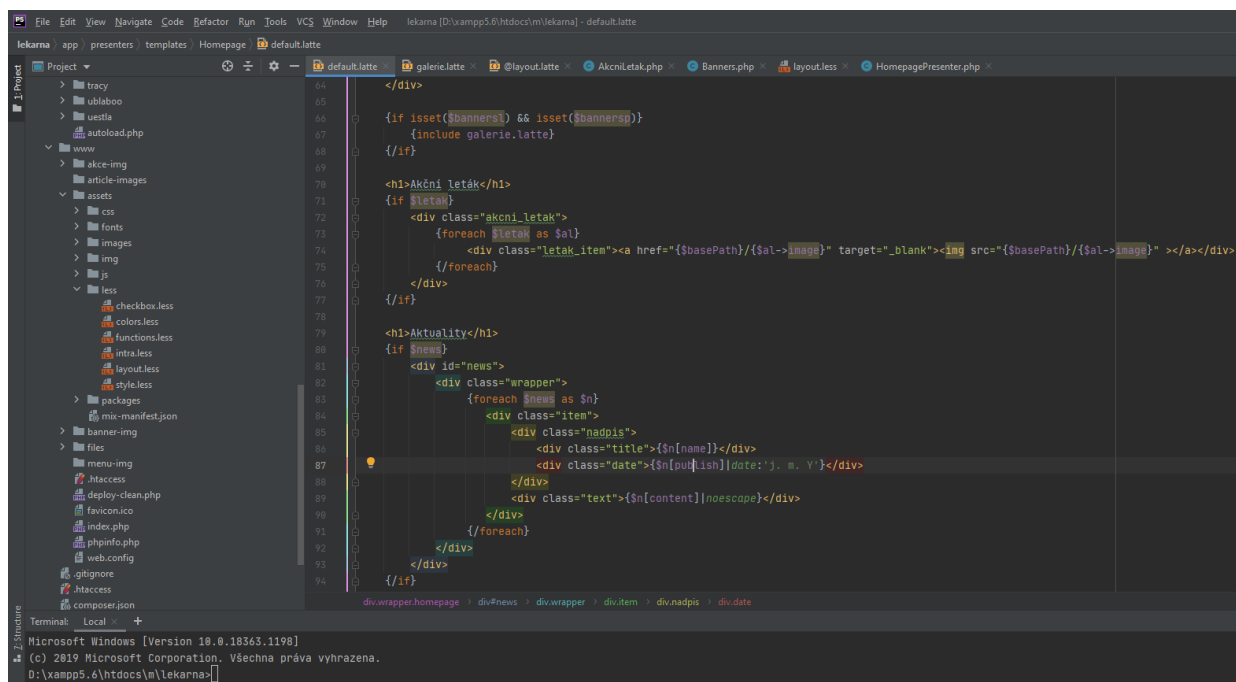
GIT je systém pro správu verzí vyvíjeného systému. Slouží k uchovávání informací o změnách zdrojového kódu aplikace. Jeho výhodou je paralelní vývoj aplikace více uživateli, kteří si navzájem nemohou přemazávat upravovaný kód za předpokladu, že každý z nich pracuje ve své vlastní větvi programu. Větev je pojmenovaná sada oprav, nebo část nové implementace systému. (Git 2020)

4.9 Vývojová prostředí

Vývojové prostředí je software, který programátorovi usnadňuje práci. Umožňuje efektivně psát kód s využitím našeptávače implementovaných funkcí, vyhledáváním textu v souborech, zvýraznění klíčových slov.

PhpStorm

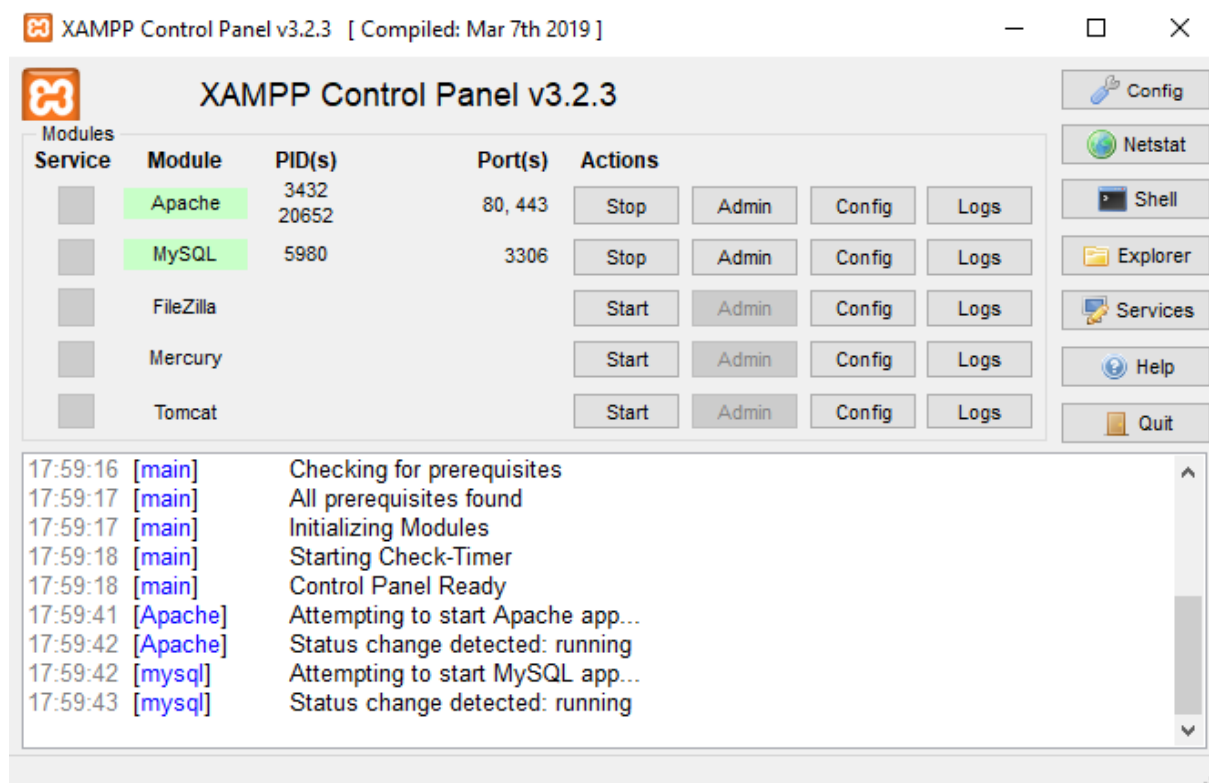
Je vývojové prostředí pro psaní webových aplikací v PHP a jeho různých frameworkách. Je vyvíjený společností JetBrains. Obsahuje automatické doplňování kódu, kontrolu chyb, debuggování aplikace, vyhledávání v kódu a mnoho dalších funkcí. Podporuje implementaci modulů pro usnadnění práce, jako je například modul pro frameworky Symfony, Nette, Laravel, Drupal, WordPress, modul pro verzovací systém GIT, LESS, CSS, HTML, JavaScript, Node.js a mnoho dalších.



Obrázek 7: PhpStorm – ukázka IDE (zdroj: vlastní)

Xampp

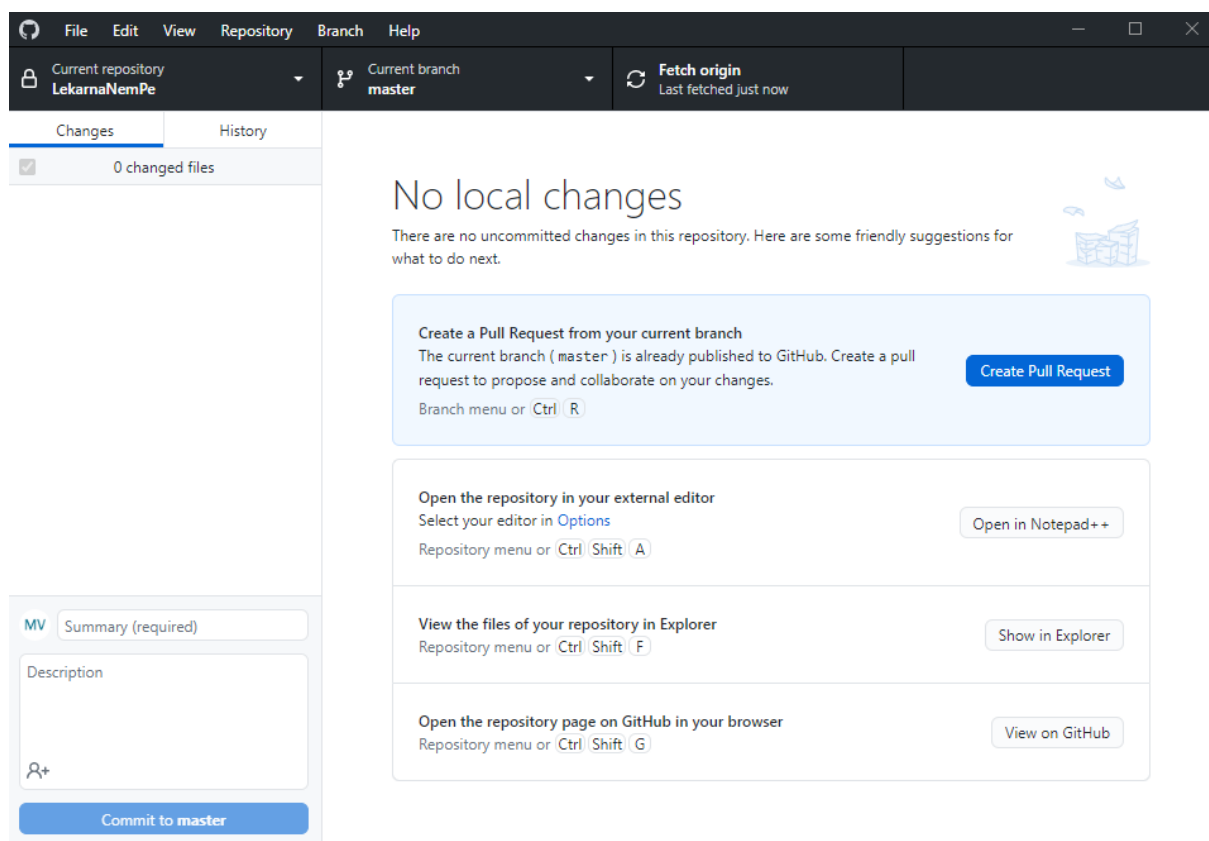
Xampp je multiplatformní webový server vyvíjený firmou Apache Friends. Obsahuje webový server Apache, databázi MariaDB, mail server, interpret programovacích jazyků PHP a Perl. Umožňuje vývojáři vytvořit lokální webový server pro vývoj a testování aplikace. Jednoduše propojí databázi s aplikací, pro správu databáze je k dispozici prostředí phpMyAdmin. Pro svůj chod nepotřebuje přístup k internetu, což využívají vývojáři pro testování i vývoj. Xampp lze snadno propojit s vývojovým prostředím. Podporuje vytváření a manipulaci s databázemi MariaDB a SQLite. (Xampp 2020)



Obrázek 8: Xampp – ovládací panel (zdroj: vlastní)

GitHub Desktop

Je to grafické rozhraní pro práci s GitHubem. Umožňuje jednoduchou správu repositářů. Repositář lze zakládat i klonovat přímo z aplikace. Poté můžeme jednoduše verzovat náš projekt. Po provedení změn ve webové aplikaci vidíme v okně, které soubory jsme upravovali, případně vytvářeli, po rozkliknutí daného souboru vidíme náhled změn. Když se jedná o úpravu již existující položky, tak vidíme v okně smazaný kód, který je zvýrazněn červeně a kód, který byl změněn, je zvýrazněn zelenou barvou. Nezměněný kód je bílý. Tyto změny kódu postupně nahráváme na GitHub do našeho repositáře s komentářem, který vysvětluje, co jsme dělali. Lze vytvářet různé větve programu, které se hodí pro vyvíjení dané části aplikace. Tyto větve po otestování a vyladění můžeme slučovat do základní větve.



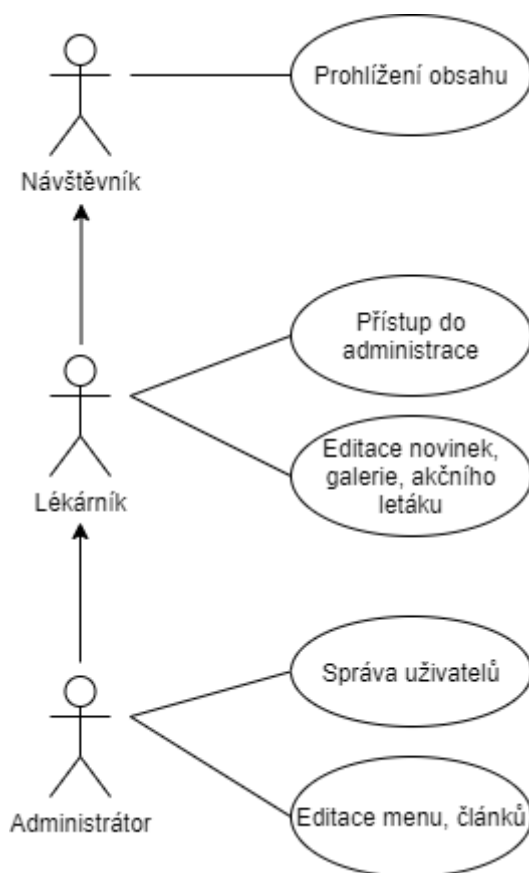
Obrázek 9: GitHub Desktop – ovládací panel aplikace (zdroj: vlastní)

5 Analýza řešeného problému

Tato kapitola se zabývá důkladnou analýzou řešeného problému. Na základě této analýzy vznikl návrh na řešení daného problému.

5.1 Funkcionalita systému

Redakční systém bude mít dvě části přístupu – uživatelskou a administrátorskou, která bude mít dvě úrovně oprávnění. Uživatelská část aplikace bude zobrazovat informace a budou do ní mít přístup všichni uživatelé bez omezení, pro uživatele je potřebné dostat se k daným informacím co možná nejrychleji. Do administrátorské části aplikace budou mít přístup pouze autorizovaní uživatelé přes administrátorské rozhraní. Pro ověření identity uživatele do administrátorské části aplikace bude vyžadováno zadání uživatelského jména a hesla. Administrátorská část bude rozdělena na dvě uživatelská oprávnění, a to na správce, který bude mít možnost spravovat veškerý obsah aplikace. Druhá role přístupu bude lékárník, který bude mít právo spravovat galerii na úvodní stránce, akční leták a aktuality. Role správce je rozšířena o správu menu, článků a uživatelů, kteří mají přístup do administrační části aplikace.



Obrázek 10: Use case diagram (zdroj: vlastní)

O samotných uživatelích budeme uchovávat tyto informace: jméno a příjmení, přihlašovací jméno a heslo, e-mail a skupinu oprávnění. Každý uživatel má právo si změnit heslo. Heslo bude zašifrováno pomocí hashovací funkce bcrypt.

Základní částí každého webu je menu. Ty vytvářejí jednotlivé stránky webu. O menu budeme uchovávat tyto informace: název, URL adresu, která se bude dopisovat za název webu, dále pak pořadí, podle kterého bude zobrazeno ve výpisu. Další informací, kterou budeme uchovávat je titulek, který se propíše před základní titulek stránky. Poté ještě informaci o tom, zda je menu aktivní a má být zobrazeno ve výpisu stránek.

Další nedílnou součástí každého systému je správa článků a novinek. U článků budeme zaznamenávat název článku, datum zveřejnění, obsah článku. Další informací bude pořadí na stránce, které zajišťuje to, že se více článků na stránce bude zobrazovat podle priority, kterou si administrátor zvolí. Dále pak jestli je aktivní, tedy jestli ho chceme na stránce zobrazit a jestli chceme zobrazit název při samotném výpisu. Nejdůležitější informace, kterou potřebujeme znát je v jakých menu se má daný článek zobrazovat. Dále pak bude mít administrátor možnost k článku přidat galerii, nebo soubory. Bude se jednat o jednoduchý systém, kdy se zobrazí fotky vedle sebe a po kliknutí uživatelem se mu fotky zobrazí v plném rozlišení. U dokumentů se vypíší zobrazené soubory a po kliknutí na soubor s příslušnou ikonkou se uživateli daný dokument zobrazí. U novinek pak bude uchovávat název, pořadí, datum publikování, odkaz. Dále pak jestli je aktivní a jestli se má zobrazovat název. Poslední věcí, kterou budeme o novinkách uchovávat bude obsah aktuality.

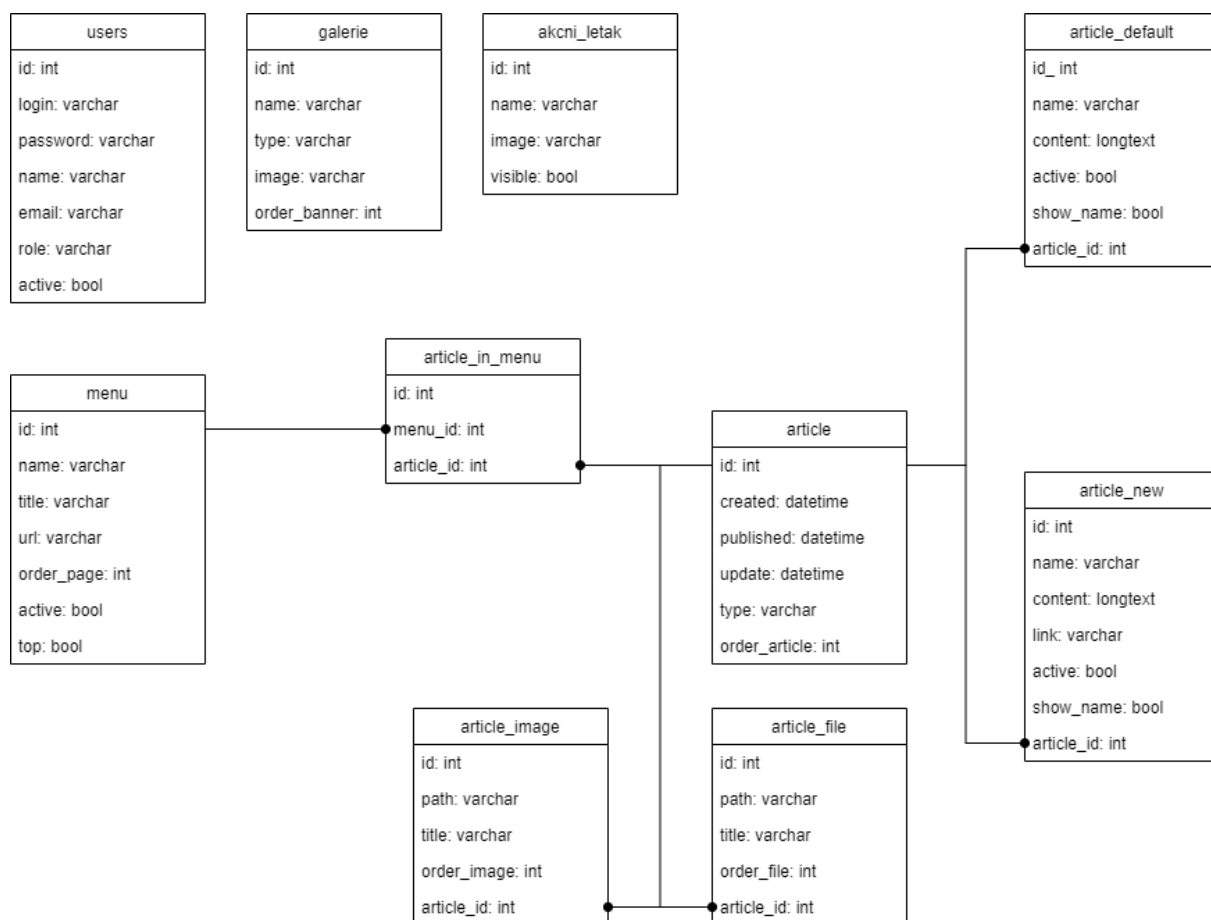
Další částí systému bude akční leták, který má návštěvníky informovat o akčně nabízených produktech nemocniční lékárny. U akčního letáku bude zaznamenávat informace o názvu letáku. Informaci o tom, zda je daný leták aktuální a jestli se má tedy zobrazovat na úvodní stránce. Dále pak bude uložena informace o samotném obrázku.

Poslední částí systému je galerie zobrazovaná na úvodní stránce. Bude se jednat o dvě samostatné fotogalerie. Každá fotogalerie bude automaticky posouvat obrázky a bude mít možnost je posouvat uživatelem samostatně. O fotografii z galerie bude uchovávat tyto informace: její název, zařazení fotky tedy to, jestli má být zobrazena v levé, anebo v pravé galerii. Dále pak pořadí tím bude administrátor určovat na jakém místě se má daná fotografie zobrazovat. Poslední informací bude samotný obrázek do galerie.

5.2 Datový model

Cílem datového modelu je navrhnout datovou strukturu pro redakční systém nemocniční lékárny. Správný návrh modelu je velmi důležitý a má vliv na funkčnost celého systému. Případné změny znamenají přepracování určitých částí aplikace.

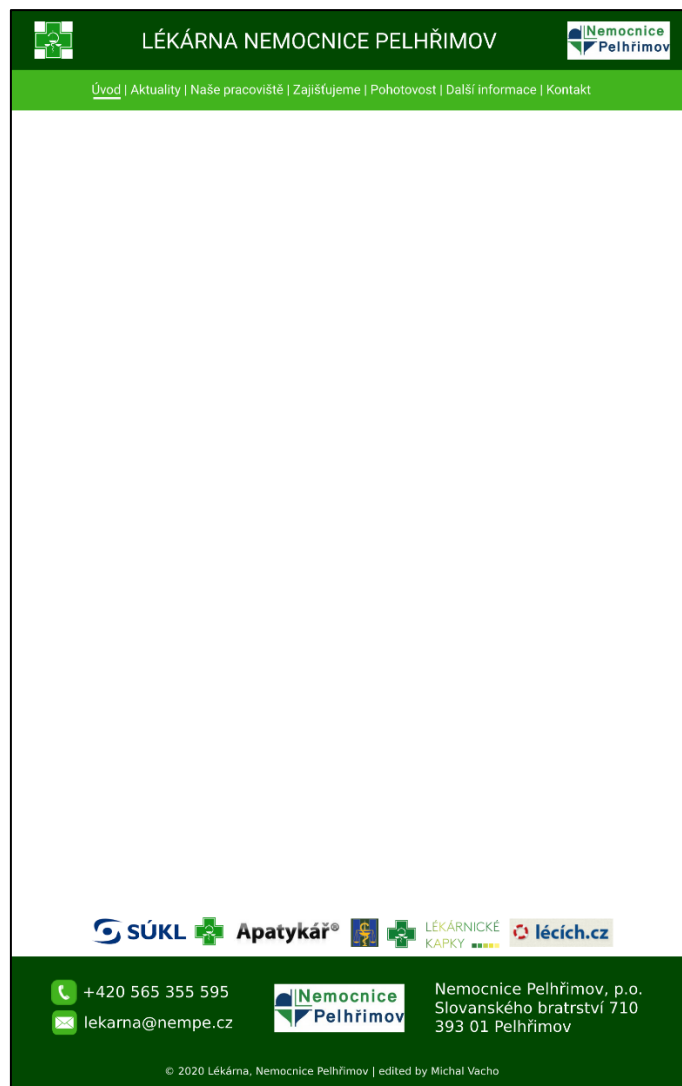
Protože se jedná o redakční systém, tak nejdůležitější tabulkou je tabulka *article*, která zajišťuje uchování základních informací o člancích. Jejími atributy jsou datumy vytvoření, publikování a poslední změny, dále pak pořadí, podle kterého má být článek zařazen při výpisu a typ článku, tedy informace o tom, zda se jedná o aktualitu, nebo o normální článek. Podle typu je pak při výpisu článek rozšířen o další parametry. Tabulka *article_default* rozšiřuje základ článku o atributy: jméno, obsah, jestli je aktivní, tedy jestli bude článek zobrazen a dále o zobrazení jména, tedy o to, zda se má jméno článku zobrazovat při jeho výpisu. Tabulka *article_new* určuje, že není článek klasický, ale že se jedná o aktualitu a má stejné atributy jako *articles_default* pouze rozšířené o odkaz. Ke každému článku je možné přiřadit galerii a pro tu slouží tabulka *article_image*, jejími atributy jsou cesta, popis, pořadí. Tabulka *article_file* umožňuje přidat k článku soubory. Atributy jsou stejné jako u tabulky *article_image*, tedy cesta k souboru, titulek a pořadí. Další je tabulka *menu*, která slouží k vytváření jednotlivých stránek webu. Jejími atributy jsou název, titulek, URL adresa, pořadí, informaci o tom, jestli je aktivní a zda se má tedy zobrazovat, a dále informaci o tom, jestli se má daná stránka zobrazit ve výpisu menu zobrazovaných na webu. Pro zařazení článků do menu je potřeba propojit tabulky *article* a *menu*. Protože je možné zařadit jeden článek do více menu a mít k jednomu menu přiřazeno více článků, vychází zde vazba mezi tabulkami na M:N. Proto je nutné mezi tyto tabulky vložit další tabulku *article_in_menu*, která již bude mít vazbu 1:N a N:1, která již umožňuje takovou práci s tabulkami. Další je tabulka *user*, která uchovává informace o tom, kdo má přístup do administrační části aplikace. Jejími atributy jsou login a heslo, pod kterými se uživatelé budou hlásit do aplikace, dále pak jméno, e-mail, informace o tom, zda je uživatel aktivní a zda má tedy možnost se přihlásit. Poslední atribut je role, podle které mu budou zobrazeny kategorie, které může spravovat. Tabulka *galerie* slouží k uchování informací pro galerii na úvodní stránce. Atributy galerie jsou cesta k obrázku, název, pořadí, na kterém má být zobrazeno a typ podle kterého bude obrázek zařazen do levé nebo pravé galerie. Poslední tabulkou je *akcni_letak*, který reprezentuje obrázek s akční nabídkou lékárny. Její atributy jsou název, cesta k souboru a zda je daný leták aktuální a má být zobrazen při výpisu na úvodní stránce.



Obrázek 11: Datový model (zdroj: vlastní)

5.3 Grafický návrh webu

Grafický návrh webu má za cíl nastínit podobu webu a představit budoucí podobu aplikace zadavateli. Grafický návrh nastiňuje podobu titulní stránky a podobu layoutu, do kterého se bude vkládat veškerý veřejně dostupný obsah stránek.



Obrázek 12: Grafický návrh aplikace - layout (zdroj: vlastní)

LÉKÁRNA NEMOCNICE PELHŘIMOV

[Úvod](#) |
 [Aktuality](#) |
 [Naše pracoviště](#) |
 [Zajišťujeme](#) |
 [Pohotovost](#) |
 [Další informace](#) |
 [Kontakt](#)

Pracoviště

Aktuality

Pohotovost

Kontakt

AKČNÍ LETÁK

AKČNÍ NABÍDKA NAŠÍ LÉKÁRNY – ZIMA

VÁNOČNÍ NABÍDKA NAŠÍ LÉKÁRNY

AKTUALITY

Prázdninový provoz lékárny

01.08.2020

Od 1.8.2020 zůstává otevřená pouze výdejna léčiv v hlavní budově

Otevírací doba pro veřejnost : pondělí – pátek: 7.30 – 15.00 hodin

Změna otevírací doby

01.04.2020

Od 1.4.2020 dochází ke změně otevírací doby

Otevírací doba pro veřejnost : pondělí – pátek: 7.30 – 15.00 hodin

Nové výdejní místo naší lékárny

20.05.2019

Od pondělí 20.5.2019 je pacientům plně k dispozici nové výdejní místo nemocniční lékárny v hlavní lůžkové budově. Výdejna léčiv je umístěna v atriu hlavní lůžkové budovy.

Otevírací doba pro veřejnost: pondělí – pátek: 8.00 – 16.00 hodin

INFORMACE

eRecept je lékařský předpis (recept) vystavený v elektronické podobě. Lékařem vystavený eRecept je uložen do tzv. Centrálního úložiště elektronických receptů (CÚER).

Každému eReceptu je přidělen unikátní identifikátor. V lékárně pak lékárník načte identifikátor eReceptu a pokud je eRecept v CÚER nalezen, vydá předepsaný léčivý přípravek pacientovi. Informace o výdeji léčivého přípravku se zapisuje do CÚER.

Lékař bude moci eRecept (resp. jeho identifikátor) pacientovi předat v několika různých podobách - na papírové průvodce, v e-mailu, v SMS či prostřednictvím speciální aplikace v mobilním telefonu či jiným elektronickým zařízením pacienta.

LÉKOVÝ ZÁZNAM

Lékový záznam je evidence všech předepsaných a vydaných léků pacientovi. Ten se ke svému lékovému záznamu dostane díky webovým či mobilním aplikacím. Do lékového záznamu pacienta může zároveň nahlédnout jeho ošetřující lékař či lékárník ve chvíli, kdy poskytuje pacientovi zdravotní službu. Díky tomu mohou lékaři a lékárníci pacientovi správně nastavit léčbu. Lékový záznam snižuje riziko užívání duplicitních léků nebo možností jejich nevhodné kombinace, které mohou poškodit zdraví pacienta. Lékový záznam významně zvyšuje bezpečnost pacienta.

+420 565 355 595

lekarna@nempe.cz

Nemocnice Pelhřimov, p.o.
Slovanského bratrství 710
393 01 Pelhřimov

© 2020 Lékárna, Nemocnice Pelhřimov | edited by Michal Vacho

Obrázek 13: Grafický návrh - titulní stránka (zdroj: vlastní)

26

6 Popis implementace

Tato kapitola se zabývá podrobnějším popisem zajímavějších částí aplikace a jejích funkcí.

6.1 Implementace databáze

K uložení dat budeme využívat databázi MySQL. Pro správu databáze jsem použil aplikaci phpMyAdmin, která umožňuje její jednoduchou správu. Při zakládání databáze je potřeba určit pár nastavení. Pro nasazení systému v České republice musíme zajistit, aby se znaky ukládali v českém jazyce, musíme tedy nastavit znakovou sadu na UTF-8. Další nastavení, které musíme provést při vytváření databáze je vybrat formát úložiště v našem případě je potřeba zvolit formát InnoDB, protože se v našem systému vyskytují cizí klíče tabulek.



Obrázek 14: Vytvoření databáze (zdroj: vlastní)

Pro správu databáze přímo z aplikace systému bylo použito rozšíření Doctrine. Toto rozšíření využívá ORM, jež má za cíl mapovat objekty na databázi. Dále začneme ve složce `app/model/database/entity/` zakládat jednotlivé entity, které jsme si definovali v datovém modelu. Po vytvoření jednotlivých entit můžeme s využitím Entity manageru a ORM zadat v konzoli příkaz: `php index.php orm:schema-tool:update --force`. Tento příkaz zajistí, že se námi napsané entity převedou do formátu SQL dotazů a poté se tyto dotazy provedou nad naší databází, tedy vygenerují tabulky podle příslušných entit.

```
/**
 * @ORM\Entity
 */
class Menu extends ABaseEntity
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue
     */
    protected $id;

    /**
     * @ORM\Column(type="string", nullable=true)
     */
    protected $name;
```

Obrázek 15: Ukázka založení entity (zdroj: vlastní)

6.2 Připojení k databázi

Nette pro uchovávání parametrů a konfigurací spojení s databází využívá soubory typu neon. Tyto soubory obsahují veškeré konfigurační parametry potřebné pro běh aplikace. Nejprve se nahraje soubor config.neon ze složky app/config/, který obsahuje obecné konfigurace jako je například nastavení session, nebo služeb, které má systém k dispozici. Poté se nahraje další soubor .neon s danými parametry pro databázi.

```
database:
  default:
    dsn: 'mysql:host=localhost;dbname=lekarna_nempe'
    user: lekarna_nempe
    password: *****

doctrine:
  default:
    user: lekarna_nempe
    password: *****
    dbname: lekarna_nempe

dibi:
  host: localhost
  username: lekarna_nempe
  password: *****
  database: lekarna_nempe
```

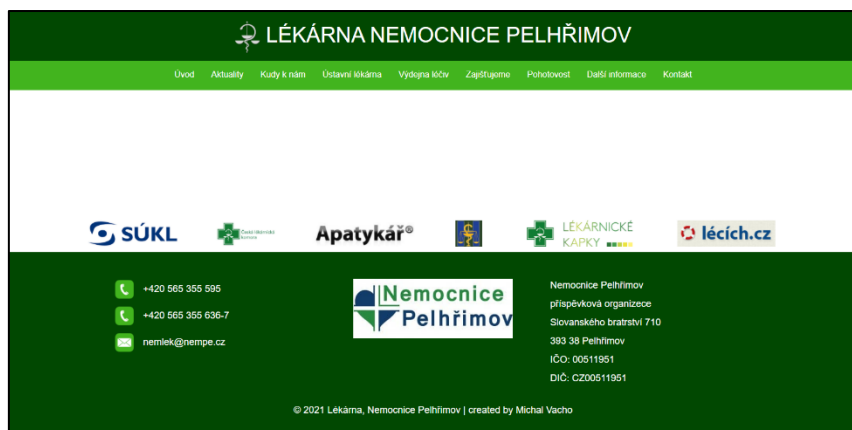
Obrázek 16: Připojení k databázi (zdroj: vlastní)

6.3 Layout webu

Cílem rozložení stránek neboli layoutu bylo o vytvoření moderního rozhraní, ve kterém se bude zobrazovat veškerý veřejně dostupný obsah webu. Layout je rozdělen do tří částí, první je hlavička s názvem a znakem lékárny. V hlavičce webu se také nachází menu, které zobrazuje dostupné stránky webu. Prostřední částí aplikace je obsahová část webu, do které se vkládají šablony, podle kterých se zobrazují informace na webu. Poslední částí je patička, které začíná výpisem doporučených webů pro pacienty a pokračuje částí ve které jsou zobrazeny kontakty do lékárny a základní informace o nemocnici.

```
<div class="content">
  <nav>
    <span class="glyphicon glyphicon-menu-hamburger"></span>
    <ul>
      {foreach $mainMenu as $m}
        {if $m->url == '/' }
          <li {if $homepage}class="active"{/if}><a n:href="Homepage:default">{$m->name}</a></li>
        {else}
          <li {if (isset($menu) && $menu->id == $m->id)}class="active"{/if}><a n:href="Page:default id => $m->id">{$m->name}</a></li>
        {/if}
      {/foreach}
    </ul>
  </nav>
</div>
```

Obrázek 17: Výpis menu (zdroj: vlastní)



Obrázek 18: Layout webu (zdroj: vlastní)

6.4 Úvodní stránka webu

Úvodní stránka je první stránka, kterou návštěvník webu uvidí po zadání URL adresy. Tato stránka se skládá z několika samostatných částí. První částí jsou navigační tlačítka, která mají za cíl urychlit hledání nejdůležitějších informací na webu a usnadnit zobrazení těchto informací. Další částí je akční leták, který zobrazuje aktuální cenovou nabídku nemocniční lékárny. Poté následuje sekce s výpisem posledních tří aktualit pro informovanost návštěvníka. Následuje výpis galerií, které představují lékárnu ve vizuální podobě. Pro výpis galerií s funkcí automatické posuvu obrázku byl použit jQuery plugin: cycle2. Poslední částí je výpis článků přiřazených na úvodní stránku.

```
<div class="cycle-slideshow"
  data-cycle-fx="scrollHorz"
  data-cycle-speed="500"
  data-cycle-timeout="2000"
  data-cycle-slides="> a"
>
<div class="cycle-pager"></div>
<div class="cycle-prev"><a><</a></div>
<div class="cycle-next"><a>>>/a></div>
{foreach $banners1 as $bl}
  <a href="{ $basePath }/{ $bl->image }" target="_blank">name }"></a>
{/foreach}
</div>
```

Obrázek 19: Výpis galerie (zdroj: vlastní)

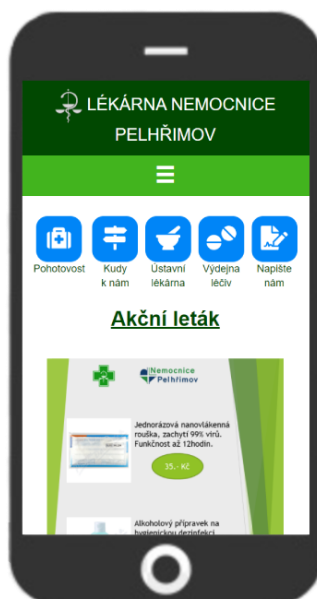


Obrázek 20: Začátek úvodní stránky (zdroj: vlastní)

6.5 Responzivita

Web je plně responzivní a podporuje různé zobrazovací šířky všech zařízení. Základem responzivního zobrazení je rozpoznávání těchto zobrazení. V první řadě je potřeba nastavit na každou ze zobrazovaných stránek meta tag pro povolení zobrazování na různých zařízeních od HTML5: `<meta name="viewport" content="width=device-width, initial-scale=1, minimal-ui">`. Další nastavení již probíhá v samotných CSS souborech, kde se pro dané příkazy napíší alternativní pro responzivní zobrazení.

```
@media (max-width: 700px) {  
    /*Alternativní CSS příkazy pro mobilní zobrazení*/  
}
```



Obrázek 21: Náhled webu v mobilním zařízení (zdroj: vlastní)

6.6 Výpis článků

Výpis článků je vytvořen pomocí komponent. Toto rozšíření má v sobě Nette již vestavěné. Volání komponenty v šabloně: {control article, \$articles, \$menu->id}. Po zavolání této komponenty se předá řízení a komponenta v cyklu projde všechny články, které jsou přiřazené ke konkrétnímu menu a sama si zajistí výpis článků pomocí námi definovaných šablon pro výpis.

Ukázka funkce pro výpis článků:

```
public function render($articlesIDS, $menuId = null)
{
    $template = $this->template;
    $template->setFile(__DIR__ . '/templates/articles.latte');
    $articles = "";
    if ($articlesIDS) {
        foreach($articlesIDS as $idArticle => $type) {
            if ($type == 'default') {
                $articles .= $this->articleDefault($idArticle, $menuId);
            } elseif($type == 'new') {
                $articles .= $this->articleNew($idArticle);
            }
        }
    }
    $template->articles = $articles;

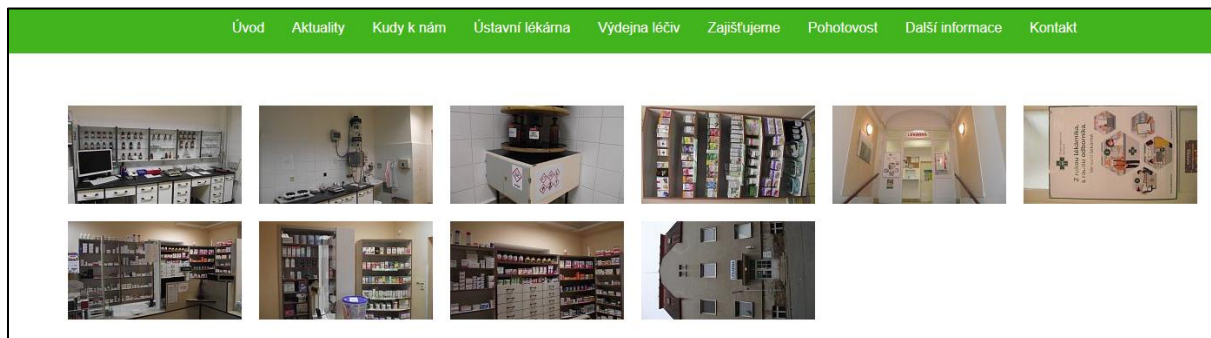
    $template->render();
}
```

Ukázka šablony pro výpis novinky:

```
<article n:if="$article" class="new">
    <h2 n:if="$article->show_name">{$article->name}</h2>
    <span class="date">{$article->publish|date:'j. n. Y'}</span>
    <div class="content">
        <div class="text">
            {$article->content|noescape}
            <div class="clearfix"></div>
        </div>
    </div>
</article>
```

6.7 Výpis fotogalerie u článku

Pro zobrazení a výpis fotogalerie přiřazené k článku byla využita jQuery knihovna fancyBox. Tato knihovna umí také samostatné zobrazení obrázků s funkcí jejich zvětšení. Dále tento plugin umožňuje přidání multimediálních souborů do webových stránek nebo různého HTML obsahu. Toto rozšíření zahrnuje také zobrazení Google map, nebo například YouTube videí do modálního (vyskakovacího) okna našeho webu.



Obrázek 22: Zobrazení fotogalerie článků (zdroj: vlastní)

Funkce pro výpis galerie:

```
{foreach $images as $img}
  <div class="item">
    <a href="{ $basePath}/{ $img->path}" data-fancybox="article{ $article->id}">
      <img n:img=" $img->path, '200x200'" alt="{ $img->alt|noescape}">
    </a>
  </div>
{/foreach}
```

6.8 Vytvoření formuláře s dotazem

Využití Nette Forms nám umožňuje efektivně a bezpečně vytvářet formuláře. Toto rozšíření nám umožňuje jednoduchou validaci dat, dále se stará o minimalizaci vzniku chyb a tím i eliminaci bezpečnostních děr. Díky tomu můžeme jednoduše vytvořit náš formulář.

 A screenshot of a web form. It has three input fields: 'Jméno' (Name) with a placeholder 'Jméno', 'E-mail' with a placeholder 'E-mail', and 'Dotaz' (Question) with a placeholder 'Váš dotaz'. Below the fields are two buttons: 'Odeslat' (Send) in blue and 'Smazat' (Cancel) in white.

Obrázek 23: Zobrazení formuláře (zdroj: vlastní)

Vytvoření formuláře v presenteru:

```
$form = new Form(); //Vytvoření nového formuláře
$form->addText('name', 'Jméno') // Přidání vstupního prvku formuláře typu text
    ->setAttribute('placeholder', 'Jméno') //Nastavení zobrazení nápovědy pro uživatele do
vstupního políčka formuláře, aby věděl co má zadat
    ->setRequired('Jméno je povinné.');//Prvek formuláře je povinný, nastavení hlášky při
nesplnění
$form->addEmail('email', 'E-mail') // Přidání vstupního prvku formuláře typu e-mail
    ->setAttribute('placeholder', 'E-mail')
    ->setRequired('E-mail je povinný.');//
$form->addTextArea('inquiry', 'Dotaz') //Přidání prvku formuláře typu textová oblast
    ->setAttribute('placeholder', 'Váš dotaz')
    ->setAttribute('style', 'resize: vertical') //Nastavení stylu
    ->setAttribute('rows', '5') //Nastavení počtu řádků textové oblasti
    ->setRequired('Dotaz je povinný.');//
$form->addSubmit('send', 'Odeslat');//Přidání tlačítka pro odeslání formuláře
$form->addButton('clear', 'Smazat') //Přidání tlačítka pro vyprázdnění formuláře
    ->setHtmlAttribute('type', 'reset');//
$form->onSuccess[] = [$this, 'successInquiry']; //Nastavení co se má stát s daným formulářem
po odeslání – tedy že se mají data odeslat funkcí 'successInquiry', která je zpracuje
return $form;
```

Volání formuláře v šabloně:

```
{form inquiryForm class => 'form-horizontal'}
    <div class="form-group">
        {label name class => 'control-label col-sm-2 col-md-offset-1'}
        <div class="col-sm-10 col-md-6">
            {input name class => 'form-control'}
        </div>
    </div>
    <div class="form-group">
        {label email class => 'control-label col-sm-2 col-md-offset-1'}
        <div class="col-sm-10 col-md-6">
            {input email class => 'form-control'}
        </div>
    </div>
    <div class="form-group">
        {label inquiry class => 'control-label col-sm-2 col-md-offset-1'}
        <div class="col-sm-10 col-md-6">
            {input inquiry class => 'form-control'}
        </div>
    </div>
    <div class="form-group">
        <div class="col-sm-12 col-md-11 text-center">
            {input send class => 'btn btn-primary'}
            {input clear class => 'btn btn-default'}
        </div>
    </div>
{/form}
```

6.9 Zpracování formuláře

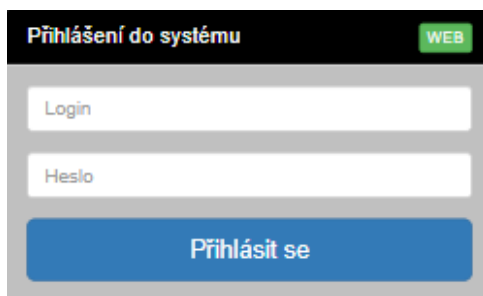
Do projektu s Nette lze přidat API pro vytváření a posílání e-mailů. Toto rozšíření bylo využito i pro náš web, rozšíření se jmenuje nette/mail. Při zpracování e-mailu využijeme hodnoty z formuláře, které si předáme parametrem do vytvořené funkce v třídě MailSender. Takto předané hodnoty odešleme šabloně pro vytvoření těla e-mailu, pro které využijeme latte šablonu. Poté si již vytvoříme samotný e-mail, kterému nastavíme odesílatele, příjemce a předmět e-mailu. Následně již takto vytvořený e-mail odešleme. K odeslání e-mailu využijeme rozhraní IMailer s metodou SendmailMailer, který pro odeslání využívá PHP funkci mail(). Další nastavení této metody není potřeba, protože tato metoda si je zajistí sama z nastavení serveru. Druhá metoda, kterou lze využít je metoda SmtplibMailer, která využívá SMTP server pro odesílání pošty. Při jeho vytváření mu musíme nastavit parametry, které má při odesílání použít. Musíme mu nastavit SMTP adresu přes kterou má odesílat e-maily, dále pak přihlašovací jméno uživatele a jeho heslo.

Funkce pro vytvoření a odeslání e-mailu:

```
$template = $this->templateFactory->createTemplate(); //vytvoření šablony
$template->values = $values; //Odeslání dat z formuláře do šablony
$template->getLatte()->addProvider('uiControl', $this->linkGenerator); //Odeslání základního odkazu na náš web do šablony
if (!$user) { //rozhodování jestli posílám e-mail uživateli anebo do společnosti
    $template->setFile(__DIR__ . '/templates/poptavka2.latte'); //Nastavení latte šablony, kterou využiji pro vytvoření těla e-mailu
} else {
    $template->setFile(__DIR__ . '/templates/poptavka.latte');
}
$mail = new Message(); //Vytvoření e-mailu
$mail->setHtmlBody($template); //nastavení těla e-mailu podle šablony
if (!$user) {
    $mail->setFrom($values->email) //Přidání odesílatele e-mailu
        ->addTo("nemlek@nempe.cz"); //Nastavení příjemce e-mailu
} else {
    $mail->setFrom("nemlek@nempe.cz")
        ->addTo($values->email);
}
$mail->setSubject('Dotaz z webu Lékárny nemocnice Pelhřimov'); //Přidání předmětu e-mailu
/*$mailer = new SmtplibMailer(['host' => 'smtp.office365.com', 'port' => '587', 'username' => '*****@*****.***', 'password' => '*****'],);*/ //Vytvoření maileru přes metodu SmtplibMailer s ručním nastavením parametrů
$mailer = new SendmailMailer; //Vytvoření maileru za pomoci metody SendmailMailer
$mailer->send($mail); //Samotné odeslání e-mailu
```

6.10 Administrace webu

Administrace je samostatné rozhraní, které má za cíl spravovat veškerý obsah webu. Do administrační části se dostaneme zadáním adresy ve tvaru: <https://web.cz/admin>. Po zadání za náš web /admin se zobrazí přihlašovací formulář do administrační části. Kromě tlačítka pro přihlášení je zde umístěno také tlačítko web, které má za cíl urychlit návrat do uživatelské sekce webu a usnadnit tak administrátorům práci.



Obrázek 24: Přihlášení do administrace (zdroj: vlastní)

6.11 Autentizace uživatele

Nette využívá pro autentizaci uživatelů službu s názvem User. Díky tomu můžeme velice snadno ověřovat přístup do administrační části aplikace. Nejprve je potřeba ověřit, zda je uživatel přihlášen, v případě že není, je potřeba ho přesměrovat na přihlašovací formulář, toto provedeme v BasePresenteru.

```
if (!$this->getUser()->isLoggedIn() && $this->getPresenter()->getName() != 'Intra:user' &&
    $this->getPresenter()->getAction() != 'login') {
    $this->redirect('User:login');
}
```

Poté co uživatel zadá své přihlašovací jméno a heslo, se ho pokusíme přihlásit do systému pomocí funkce login. Tato funkce nás buďto přihlásí, nebo vrátí výjimku, kterou potřebuje zpracovat.

```
try {
    $this->getUser()->login($values['username'], $values['password']);
    $this->getUser()->setExpiration('120 minutes');
    $this->flashMessage('Byl jste úspěšně přihlášen.', 'success');
    $this->redirect('Homepage:default');
} catch (Nette\Security\AuthenticationException $e) {
    $this->flashMessage('Nesprávné přihlašovací jméno nebo heslo.', 'warning');
}
```

V případě, že se nám povede přihlásit, tak nastavíme čas přihlášení na 120 minut. Dále uživatele přesměruje do administrační sekce a zobrazíme mu hlášku o úspěšném přihlášení.

V případě neúspěchu přihlášení zobrazíme uživateli hlášku, že zadal nesprávné jméno, nebo heslo a zůstane na přihlašovací stránce, kde se může pokusit o nové přihlášení.

Funkce login volá námi vytvořený autentikátor, který má za úkol přihlásit uživatele a vrátit jeho identitu, anebo vyvolat výjimku Nette\Security\AuthenticationException.

```
/**
 * Zajisteni autentifikace uzivatele
 *
 * @return Nette\Security\Identity
 * @throws Nette\Security\AuthenticationException
 */
public function authenticate(array $credentials)
{
    list($username, $password) = $credentials;

    $row = $this->em->getRepository(Users::class)->findOneBy(['username' => $username,
'active' => '1']); //hledání záznamu uživatele s loginem v databázi a zjištění zda je aktivní
    if (!$row) { //ověření zda jsme uživatele našli, pokud ne vyhození výjimky
        throw new Nette\Security\AuthenticationException('Uživatelské jméno je neplatné.',
self::IDENTITY_NOT_FOUND);
    } else if (!$passwords::verify($password, $row->password)) { //ověření hesla z databáze a
zadaného při přihlášení, jestli se neshodují vyhození výjimky
        throw new Nette\Security\AuthenticationException('Heslo je neplatné.',
self::INVALID_CREDENTIAL);
    } else if ($passwords::needsRehash($row->password)) { //nastavuje novou hash hesla
v případě že se v aplikaci mění uložená nastavení pro hesla
        $row->password = Passwords::hash($password);
        $this->em->flush();
    }

    $user = $row->toArray();
    unset($user['password']);

    return new Nette\Security\Identity($row->id, null, $user);
}
```

Zachování bezpečnosti všech částí administrace a zabránění přístupu neautorizovaným uživatelům lze docílit jednoduše, a to pomocí dědičnosti presenterů. Tedy v našem případě všechny presentery pro administrační část musí být potomky základní presenteru BasePresenter, který se nachází ve složce app/intra/.

6.12 Zobrazení přehledu dat v kategoriích

Pro vytvoření přehledu výpisu položek jednotlivých kategorií bylo využito rozšíření datagrid. Toto rozšíření nám umožňuje vytvořit přehlednou tabulku s výpisem dat. Dále umožňuje přidat stránkování, filtrování a řazení dat. Také umí export záznamů do excelu, nebo csv. Jako zdroj dat využívá Nette/Database, nebo pro výpis dat z více tabulek je možné využít rozšíření dibi/dibi.

Ukázka nastavení zdroje dat pro datagrid s využitím rozšíření dibi u výpisu tabulky aktualit:

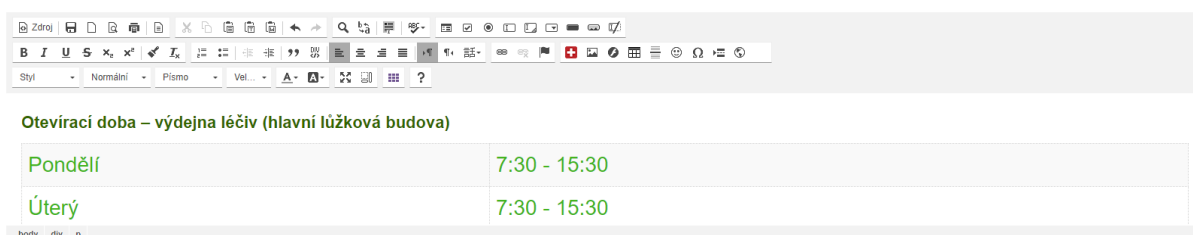
```
$datasource = $this->dibi->select('a.*', an.name AS name, an.active AS active, an.show_name AS show_name')
    ->from('articles', 'a')
    ->leftJoin('articles_new', 'an', 'ON', 'a.id = an.article_id', 'AND', 'a.type = "new"')
    ->where('a.type = "new"');
$grid->setDataSource($datasource);
```

Přidání sloupce názvu:

```
$column = $grid->addColumnText('name', 'Název'); //přidání sloupce Název do tabulky
if ($column) { //ověření sloupec na jeho existenci
    $column ->setSortable(true) //povolení řazení
    ->setFilterText('name') //nastavení filtrování
    ->setCondition(function ($dibi, $value) {
        return $dibi->where('(an.name like %~like~)', $value); //nastavení kde se má hledat
    });
}
```

6.13 Tvorba HTML obsahu stránek

Pro psaní obsahu článků a novinek byl použit WYSIWYG editor konkrétně CKEditor. Tento editor umožňuje uživatelům vytvářet obsah bez znalosti HTML tagů. Jednoduše umožňuje pro psaní textů využít nastavování velikosti, fontu, stylu, barvy a jiných vlastností písma. Mimo samotné psaní textů nám umožňuje například vkládat text z Wordu. Dále nám umožňuje vytváření tabulek, vkládání obrázků do textu a mnoho dalších funkcí pro vytváření obsahu.



Obrázek 25: Psaní obsahu s využitím CKEditoru (zdroj: vlastní)

6.14 Nahrávání jednotlivých fotografií

Nahrávání obrázku u akčního letáku a do galerie je vytvořeno pomocí klasického formulářového vstupního prvku typu file.

```
$form->addUpload('image', 'Obrázek') //přidání uploadu k formuláři
    ->setRequired(false) //nastavení na nepovinný údaj
    ->addRule(Form::IMAGE, 'Obrázek musí být JPG, JPEG, PNG nebo GIF.')
//nastavení omezení nahrávaných souborů na soubory type IMAGE
    ->addRule(Form::MAX_FILE_SIZE, 'Maximální možná velikost obrázku je 4 MiB',
4 * 1024 * 1024); //nastavení omezení maximální velikosti fotky na 4 mebibyty – zadáváno
v bytech
```

Funkce pro uložení obrázku:

```
if ($values->image->name != null) { //zjištění zda se obrázek nahrál
    if ($orig->image) { //zjištění zda už byl obrázek nahrán předtím a je měněn
        if (file_exists($orig->image)) { //zjištění zda existuje na serveru
            @unlink($orig->image); //smazání původního obrázku
        }
    }
    $mili = str_replace(".", "", microtime(true)); //definování času s nahrazením '.'
    $type = substr($values->img->name, strrpos($values->img->name, '.')); //zjištění
přípony obrázku
    if (!is_dir('akce-img')) { //zjištění, zda složka existuje
        mkdir('akce-img'); //jestliže ne, tak se vytvoří
    }
    $tmp = 'akce-img/' . $mark->id . '_' . $mili . $type; //definování cesty souboru
    $values->img->move($tmp); //přesun obrázku na definovanou cestu
    $this->facade->saveImage($tmp, $mark); //uložení cesty k obrázku do databáze
}
```

6.15 Nahrávání většího množství souborů

Pro nahrávání většího počtu souborů bylo využito jQuery knihovny DropzoneJS, která umožňuje nahrávání souborů přetažením do definovaného okna anebo klasickým výběrem.

Nastavení knihovny dropzone pro nahrání fotek:

```
var photogallery = new Dropzone('#photogallery-dropzone', { /*Vytvoření nového prvku*/
    acceptedFiles: 'image/*', /*Omezení souborů pouze na obrázky*/
    dictDefaultMessage: 'Pro nahrání obrázků je přetáhněte sem nebo klikněte zde a
vyberte.' /*Nastavení hlášky, která se zobrazí v okně formuláře*/
});
photogallery.on("queuecomplete", function(file) { /*Nastavení co se má stát po dokončení
nahrávání*/
    $.nette.ajax({ 'url': {link updatePhotogallery!} }); //Volání funkce definované v presenteru
});
```

7 Návod na správu webu

Tato kapitola se zabývá popisem, jak s aplikací zacházet a kde co nastavit.

7.1 Obecná nastavení pro výpis kategorií

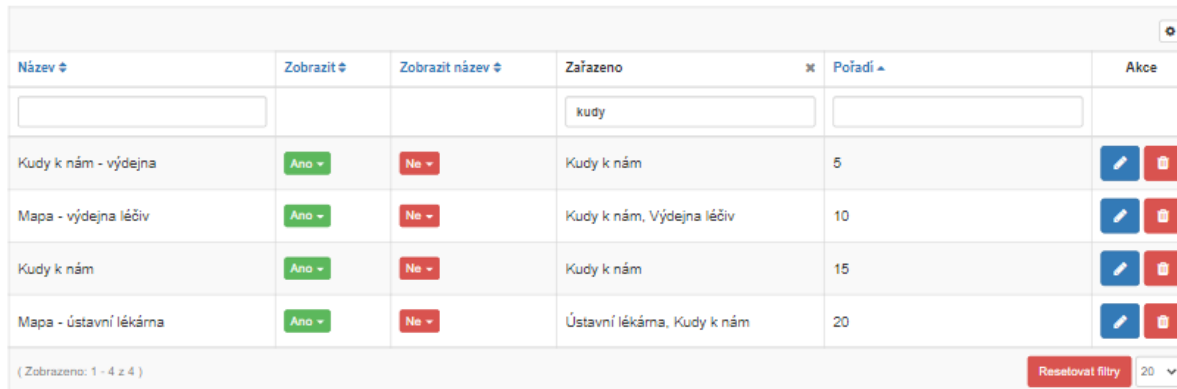
Jednotlivé kategorie se zobrazují vždy po jejich vybrání v levé části administrační části webu. Po vybrání jednotlivé sekce se zobrazí vždy tabulka s příslušnými hodnotami z dané kategorie. Nové hodnoty do těchto kategorií přidáváme kliknutím na modré tlačítko nad základní tabulkou s výpisem hodnot. Jednotlivé hodnoty lze upravovat po kliknutí na modrou tužku ve sloupečku akce na konci příslušného řádku s hodnotou. Mazání daného záznamu probíhá po kliknutí na červené tlačítko s ikonkou popelnice ve sloupečku akce vždy v příslušném řádku s hodnotou. Po kliknutí na tlačítko mazání je vždy uživatel ještě vyzván na potvrzení mazání dané položky. U položek, které mají význam zobrazit/nezobrazit je možné využít zjednodušené volby tohoto nastavení a změnit tak hodnotu přímo z tabulky. A to tak že kliknou na logickou hodnotu ,ano‘ nebo ,ne‘ a následně se jim pod touto hodnotou zobrazí nabídka s výběrem možností. Po tomto výběru se daná hodnota automaticky uloží a tabulka se znovu načte s již změněnými hodnotami. Toto má za cíl urychlit nastavování těchto hodnot a usnadnit tak administrátorovi práci s otevíráním editace dané položky z kategorie.

Název	Pořadí	Aktivní	Zobrazit v menu
Úvod	1	Ano	Ano
Aktuality	2	Ano	Ano
Kudy k nám	3	Ano	Ano
Ústavní lékárna	4	Ano	Ano
Výdejna léčiv	5	Ano	Ano
Zajišťujeme	6	Ano	Ano
Pohotovost	7	Ano	Ano
Další informace	8	Ano	Ano
Kontakt	9	Ano	Ano
Dotaz	100	Ano	Ne

Obrázek 26: Zobrazení výpisu dat z dané kategorie (zdroj: vlastní)

Další možností je nastavení stránkování. Při zobrazení velkého počtu položek se tak nebude vypisovat například všech sto záznamů dané kategorie, ale je dané množství. Toto nastavení probíhá ve spodní části tabulky, kde úplně vlevo je možné vidět které záznamy jsou aktuálně zobrazeny a z jakého počtu. V pravé straně pak můžeme provádět nastavení samotného počtu zobrazených záznamů na stránku. A v případě, že je počet záznamů překročen, tak se uprostřed spodní části tabulky zobrazí čísla stránek, přes které je možné přecházet na další záznamy z dané kategorie.

Další možností, jak pracovat s velkým množstvím dat uloženým v dané kategorii je zobrazení filtrů pro příslušné sloupce. Filtrování lze zapnout v tabulce s hodnotami v pravé horní části tabulky po kliknutí na ozubené kolečko a v nabídce vybrat možnost zobrazit filtry. Poté se nám pod názvy sloupců zobrazí nový prázdný řádek s prázdnými vstupními prvky, do kterých můžeme začít psát hledaný text a tabulka se začne aktualizovat podle zadaného textu. Další možností, jak efektivně prohledávat záznamy, je jejich řazení, které probíhá po kliknutí na název sloupce a tabulka se seřadí vzestupně nebo sestupně. Dále co se nám zobrazí po kliknutí na ozubené kolečko v záhlaví tabulky jsou názvy sloupců, ty si můžeme zobrazovat nebo skrývat podle toho o jaké hodnoty máme zájem zobrazit v daný moment.



Název	Zobrazit	Zobrazit název	Zařazeno	Pořadí	Akce
			kudy		
Kudy k nám - výdejna	Ano	Ne	Kudy k nám	5	 
Mapa - výdejna léčiv	Ano	Ne	Kudy k nám, Výdejna léčiv	10	 
Kudy k nám	Ano	Ne	Kudy k nám	15	 
Mapa - ústavní lékárna	Ano	Ne	Ústavní lékárna, Kudy k nám	20	 

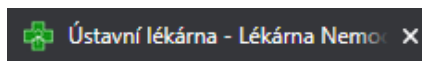
(Zobrazeno: 1 - 4 z 4) Resetovat filtry 20

Obrázek 27: Filtrování dat v kategoriích (zdroj: vlastní)

7.2 Vytváření menu

Administrátor vytváří nového menu, nebo upravuje již existující. Zadává, jak se má nové menu jmenovat, pořadí, podle kterého se bude řadit ve výpisu. Dalšími prvky, které je potřeba zadat jsou URL, to je adresa, přes kterou se do daného menu dostaneme. Například: <https://web.cz/ustavni-lekarna>. Další je titulek, který slouží k optimalizaci vyhledávačů a který se propíše do hlavičky stránky: <title>Ústavní lékárna - Lékárna Nemocnice Pelhřimov</title>.

Poslední, co administrátor nastavuje, jsou informace o tom, zda má být menu zobrazeno a zda se má toto menu zobrazit v proužku s výpisem menu.



Obrázek 28: Zobrazení titulku v kartě prohlížeče (zdroj: vlastní)

Úprava menu

Název	Ústavní lékárna	URL	ustavni-lekarna
Pořadí (priorita)	4	Titulek	Ústavní lékárna
☒ Zobrazit ☒ Zobrazit v menu			

Obrázek 29: Editace menu (zdroj: vlastní)

7.3 Správa článků

Pro články musí správce zadat zařazení článku do konkrétního menu, pro výběr zde slouží výběrové vstupní políčko s vícenásobným výběrem, protože jeden článek může být zařazen současně do více menu. Dalším parametrem je pořadí. To určuje, na jakém místě se má daný článek zobrazit při výpisu v menu. Dále si můžeme určit k jakému datumu jsme daný článek vydali a jestli ho máme zobrazit nebo ne. Další je název článku, u kterého si můžeme určit, zda se má daný název zobrazit při výpisu. A poslední, co uživatel zadává je samotný obsah článku, tedy to, co chce, aby se návštěvníci dozvěděli.

Zařazení	Další informace	Pořadí	1
Datum zveřejnění	15. 10. 2020 10:32		
Název	Další informace	☒ Zobrazit ☒ Zobrazit název	

Obsah

Platnost lékařských předpisů je 14 dní, včetně data vystavení. Výjimku tvoří recepty na antibiotika a antimikrobiální chemoterapeutika užívaná perorálně (tablety, sirupy pro děti), které platí pouze 5 dní! Recepty vystavené lékařskou službou první pomoci platí jen 1 den následující po dni jejich vystavení!

Platnost poukazů na zdravotní pomůcky je 3 měsíce.

Platnost výpisu z receptu je stejná jako platnost receptu, počítá se ode dne vystavení výpisu!

Platnost e-receptu (elektronického receptu) je stejná jako papírového receptu 14 dní.

Obrázek 30: Úprava článku (zdroj: vlastní)

7.4 Vytvoření fotogalerie u článku

U článku má administrátor možnost přidání mimo základních informací také přidat fotogalerii či soubory. Princip přidávání je u obou možností stejný. Nahoře v záložkách se přepne ze záložky základní informace do fotogalerie a zde má dvě možnosti, jak obrázky přidat. První možnost je kliknout dovnitř modrého rámečku a otevře se mu klasické vyskakovací okno pro výběr souborů. Druhou možností je přetažení souborů, které chce nahrát přímo do modrého rámečku na webu. Poté se soubory sami nahrají na web a zobrazí se pod modrým rámečkem, kde je uživatel může ještě upravovat a mazat.

Úprava článku

Obrázek 31: Přidání fotogalerie k článku (zdroj: vlastní)

7.5 Správa novinek

U aktualit bude uživatel zadávat pořadí, podle kterého se bude daná informace zobrazovat. Dále datum zveřejnění, které se bude zobrazovat při samotném výpisu. Další informací, kterou může uživatel zadat je název a zda se má zobrazit při výpisu v menu aktualit, poté může ještě určit zda se má daná novinka zobrazit. Administrátor může také zadat odkaz, který zajistí že po kliknutí na název bude návštěvník odkázán na externí web. Poslední co musí správce zadat je obsah zprávy, přes kterou chtěl návštěvníky informovat o dané události.

Obrázek 32: Úprava aktuality (zdroj: vlastní)

7.6 Vytváření uživatelů

Při vytváření nových uživatelů, kteří mají přístup do administrace musí administrátor zadat jejich přihlašovací jméno, pomocí kterého se budou do správy aplikace hlásit, k tomu jim nastaví heslo, které si můžou kdykoliv změnit. Další, co je potřeba zadat je jejich jméno a příjmení, dále pak jejich e-mail. Poslední věcí, co musí administrátor nastavit je, do jaké skupiny bude uživatel patřit. Dále je pak možné uživatele vyřazovat tím, že jim odebereme právo být aktivním uživatelem a tím se již nedostane do správy aplikace.

Úprava uživatele

Přihlašovací jméno (login)

Jméno a příjmení

Skupina

Heslo

Email

☒ Aktivní

Obrázek 33: Vytváření nového uživatele (zdroj: vlastní)

7.7 Správa galerie

U galerie správce zadává název obrázku, pořadí pro zobrazení v galerii. Dále pak zařazení, jestli daná fotografie patří do levé anebo do pravé galerie. A poslední, co je zapotřebí vybrat je samotná fotografie, která bude zobrazena v galerii. Fotografii vybere administrátor tak, že klikne na vybrat soubor a zobrazí se mu klasické okno pro výběr souboru. Po kliknutí na fotografii se zobrazí v plné velikosti.

Úprava fotky v úvodní galerii

Název obrázku

Pořadí

Zařazení

Obrázek Soubor nevybrán

Obrázek 34: Editace fotky v galerii (zdroj: vlastní)

7.8 Administrace akčního letáku

U akčního letáku správce eviduje jeho název, dále pak zda se má zobrazit na úvodní stránce. Poslední věcí, kterou administrátor nastavuje je to, jaká fotografie se má zobrazit. Po nahrání fotografie se zobrazí její miniatura, po kliknutí na danou fotografii se zobrazí v plné velikosti. Pokud se správce rozhodne smazat danou fotografii bez smazání celého záznamu může tak učinit pomocí červeného tlačítka vedle fotografie.

Editace akčního letáku

Název	leták jaro 2021	Akční leták	Vybrat soubor Soubor nevybrán
☒ Zobrazit			

Obrázek 35: Administrace akčního letáku (zdroj: vlastní)

8 Testování

Testování celé aplikace probíhalo již v rámci samotného vývoje systému na lokálním serveru. Po dokončení základní části aplikace byl celý systém nasazen na testovací server VŠPJ, kde byly zjištěny pouze drobné nedostatky, které byly ihned odstraněny. Po předvedení funkčnosti systému vedoucí lékárny a vedení Nemocnice Pelhřimov jim byl předán odkaz na provedení testování. Byl jsem velmi rád za provedení testování z jejich strany, i za návrhy na vylepšení aplikace. Testování probíhalo v prohlížečích Google Chrome, Microsoft Edge, Mozilla Firefox, Opera a Safari. Aplikace byla testována kromě klasických zařízení jako je stolní počítač nebo notebook také na mobilech a tabletech s různým operačním systémem. Validace kódu byla provedena validátorem dostupným na stránce provozované konsorciem W3C: validator.w3.org a její výsledek odpovídá standartu HTML5.

8.1 Srovnání s existujícími řešeními

V minulosti jsem se setkal s různými řešeními webů s redakčními systémy. Jedna nevýhoda těchto redakčních systémů, jako je například WordPress, nebo Joomla, je najít správnou šablonu, která je dostupná zdarma a tomu aby, daná šablona odpovídala zadání úkolu. Proto jsem se rozhodl naprogramovat si svůj vlastní redakční systém pro správu obsahu, který přesně odpovídá zadání z nemocniční lékárny. Systém bude stylově svými barvami a prvky znázorňovat klasické lékárenské prostředí. Dále se tento systém dá snadno rozšířit o další funkce.

Závěr

Cílem této bakalářské práce bylo vytvoření webových stránek nemocniční lékárny v Pelhřimově, které budou informovat pacienty o aktuálních akcích a novinkách v nemocniční lékárně.

Mimo vytvoření samotného webu byl vytvořen redakční systém pro správu celého webu. Administrace obsahuje přehlednou správu jednotlivých menu, článků, novinek, galerie, akčního letáku a uživatelů kteří mají přístup do administrace. Cíl byl především kladen na jednoduchou správu celého webu. Celá aplikace je naprogramovaná v jazyce PHP s využitím frameworku Nette, data jsou uložena v databázovém systému MySQL. Pro responzivní zobrazení webových stránek byla využita knihovna Twitter Bootstrap.

Aplikace bude přístupná veřejnosti po zaregistrování domény a připravení serveru na její nasazení. A to již záleží na nemocnici Pelhřimov. Aplikace je dočasně dostupná pro účely bakalářské práce na školním serveru VŠPJ na adrese <https://alpha.kts.vspj.cz/~vacho/bp/>. Přístup do části administrace je možný na adrese <https://alpha.kts.vspj.cz/~vacho/bp/admin/>.

Všechny cíle bakalářské práce byly splněny a vznikl tak funkční redakční systém pro nemocniční lékárnu v Pelhřimově. Tato aplikace by měla přispět pro lepší informovanost širší veřejnosti a přilákat do nemocniční lékárny nové pacienty. Celá aplikace je přizpůsobená pro různá zařízení od mobilních telefonů přes tablety až po webové prohlížeče.

Do budoucna se nabízí řada možných vylepšení, jako například přidání podstránek menu, nebo vytvoření rezervačního systému léků, problém je však v časové náročnosti propojení se stávajícím systémem skladového a výdejního hospodářství.

Seznam použité literatury

Bootstrap. Bootstrap [online]. [cit. 2020-12-27]. Dostupné z: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>

CSS. W3C [online]. [cit. 2020-12-27]. Dostupné z: <https://www.w3.org/Style/CSS/>

Drupal. Drupal [online]. [cit. 2020-12-27]. Dostupné z: <https://www.drupal.org/>

Git. Git [online]. [cit. 2020-12-27]. Dostupné z: <https://git-scm.com/about>

HTML. W3C [online]. [cit. 2020-12-27]. Dostupné z: <https://www.w3.org/MarkUp/Guide/>

Joomla. Joomla [online]. [cit. 2021-01-01]. Dostupné z: <https://www.joomla.org/>

JQuery API. JQuery [online]. 2020- [cit. 2020-11-24]. Dostupné z: <https://api.jquery.com/>

LESS. Lesscss [online]. [cit. 2020-11-24]. Dostupné z: <http://lesscss.org/>

PHP. PHP [online]. [cit. 2020-12-27]. Dostupné z: <https://www.php.net/>

TextPattern. TextPattern [online]. 2004- [cit. 2020-11-23]. Dostupné z: <https://textpattern.com/about>

WordPress. Wordpress [online]. [cit. 2021-01-01]. Dostupné z: <https://wordpress.org/>

Xampp. Xampp [online]. [cit. 2020-12-27]. Dostupné z: <https://www.apachefriends.org/about.html>