



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

PREZentační WEB

PRESENTATION WEB

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ KOVAŘÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAROSLAV ROZMAN, Ph.D.

BRNO 2022

Zadání bakalářské práce



Student: **Kovařík Lukáš**
Program: Informační technologie
Název: **Prezentační web**
Presentation Web

Kategorie: Web

Zadání:

1. Nastudujte audiovizuální weby organizací, prezentujících historické události a osoby.
2. Navrhněte web, ve kterém budou moci uživatelé zadávat fotografie a různé další historické materiály, přidávat jejich popisky a různě je mezi sebou párovat (např. fotografie, kde je stejná osoba) a vyhledávat.
3. Navržený web implementujte.
4. Implementovaný web otestujte na skupině minimálně 5 uživatelů a navrhněte případná vylepšení do budoucna.

Literatura:

- TATROE, Kevin a Peter MACINTYRE. *Programming PHP: creating dynamic web pages*. 4th edition. Sebastopol, CA: O'Reilly, 2020, xxiv, 516 stran : grafy. ISBN 978-1-492-05413-9.

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Rozman Jaroslav, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 3. listopadu 2021

Abstrakt

Cílem této práce je navrhnout a implementovat webovou aplikaci pro vizualizaci historických informací týkající se historických událostí a osob. Webová aplikace je napsaná v programovacím jazyce PHP s využitím frameworku Nette a databázového systému MySQL. Začátek práce je věnován technologiím potřebným k vývoji, následně návrhu uživatelského rozhraní a způsobům užití aplikace. Poté budou popsány postupné kroky vedoucí k realizaci celé aplikace. Webová aplikace umožňuje správcům systému vytvářet historické záznamy, jejichž obsah si návštěvníci mohou prohlížet. V případě registrace do systému aplikace jim bude umožněno podílet se na tvorbě obsahu tohoto webu. Závěr práce popisuje průběh testování a zamýšlení se nad budoucím zlepšením aplikace.

Abstract

The aim of this thesis is to design and implement a web application for the visualization of historical informations related to historical events and persons. The web application is written in the PHP programming language using the Nette framework and the MySQL database system. The beginning of the work is devoted to the technologies needed for development, then the design of the user interface and ways to use the application. Then the gradual steps leading to the implementation of the entire application will be described. The web application allows system administrators to create historical records that visitors can view. If they register in the application system, they will be allowed to participate in the creation of the content of this website. The conclusion describes the course of testing and thinking about future improvement of the application.

Klíčová slova

PHP, Nette, MySQL, Bootstrap, CKEditor 5, Video.js, Webová aplikace, Audio-video web

Keywords

PHP, Nette, MySQL, Bootstrap, CKEditor 5, Video.js, Web application, Audio-video web

Citace

KOVAŘÍK, Lukáš. *Prezentační web*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D.

Prezentační web

Prohlášení

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

.....
Lukáš Kovařík
8. května 2022

Poděkování

Rád bych poděkoval vedoucímu práce panu Ing. Jaroslavu Rozmanovi, Ph.D. za vedení práce, odbornou pomoc a přátelský přístup. Dále pak děkuji paní Mgr. Lence Kopřivové za možnost využít textů a obrázků z webu Moravští Chorvati pro tuto práci.

Obsah

1	Úvod	3
2	Analýza existujících řešení	5
2.1	Webové stránky	5
2.1.1	Paměť národa	5
2.1.2	Ústav pro studium totalitních režimů	6
2.2	Shrnutí	7
3	Princip tvorby webu	8
3.1	Architektura informačních systémů	8
3.1.1	Architektura klient-server	8
3.1.2	Třívrstvá architektura	9
3.2	Webová stránka	10
3.2.1	Statické stránky	10
3.2.2	Dynamické stránky	11
3.3	Webová aplikace	12
3.4	Programovací jazyky	12
3.4.1	PHP	13
3.4.2	ASP.NET	14
3.4.3	Ruby	14
3.5	Technologie	14
3.5.1	HTML	14
3.5.2	CSS	15
3.5.3	Bootstrap	16
3.5.4	AJAX	17
3.5.5	MySQL	17
3.5.6	PHP framework	18
4	Návrh webové aplikace	21
4.1	Specifikace požadavků	21
4.2	Diagram případů užití	22
4.3	Návrh uživatelského rozhraní	25
4.4	ER diagram	28
5	Implementace	30
5.1	Framework Nette	30
5.1.1	Architektura MVP	30
5.1.2	Composer	31

5.1.3	Ladící nástroj Tracy	32
5.1.4	Šablony Latte	32
5.2	Lokální vývoj	33
5.2.1	WampServer	33
5.2.2	Adresářová struktura	34
5.2.3	Databáze	36
5.2.4	Vývojové prostředí	36
5.3	Nasazení aplikace	36
5.4	Uživatelé	38
5.4.1	Registrace	38
5.4.2	Přihlášení	39
5.4.3	Obnova hesla	40
5.5	Tvorba webového obsahu	41
5.6	Zobrazení webového obsahu	42
5.7	Administrace	43
5.8	Vyhledávání	43
6	Testování	45
6.1	Způsob testování	45
6.2	Test nepřihlášeného uživatele	46
6.3	Test přihlášeného uživatele	47
7	Závěr	49
7.1	Dosažené výsledky	49
7.2	Možná budoucí vylepšení	50
	Literatura	51
	A Obsah příloženého paměťového média	54

Kapitola 1

Úvod

Tato bakalářská práce se zabývá vytvořením webové aplikace pro zobrazení historických informací popisující historické události, osoby nebo například skupiny osob, které se odlišují od okolní většinové společnosti. To bude zajišťovat s využitím textového, fotografického a audio-video materiálu. Taková aplikace může být využita pro prezentaci záznamů historických organizací. Tyto organizace jsou například muzea a sdružení osob, k nimž patří menšiny a historické spolky. Výsledná webová aplikace je veřejně dostupná na adrese¹ uvedené v poznámce pod čarou.

Webová aplikace umožní především zobrazovat záznamy návštěvníkům, ať už pamětníkům, kteří si chtějí připomenout svoji minulost nebo i novým zájemcům o historii. Návštěvníkovi aplikace nabídne zobrazení článků a životopisů určitých osob specifické historie, u kterých si bude moct prohlížet fotografie nebo přehrávat dobové audio-video nahrávky osob či událostí. Dále si může vytvořit uživatelský účet a podílet se tak společně se správci na tvorbě obsahu webu. Prostřednictvím uživatelského rozhraní bude správce moci vytvářet, upravovat a mazat záznamy a udržovat tak systém do budoucna.

Protože se jedná o historii těch, kteří tu již mezi námi nejsou, ale i těch, kteří ji ještě žijí, je zapotřebí, aby byl web uživatelsky přívětivý a jednoduchý v jeho použití. K tomu přispěje také automaticky přizpůsobivý vzhled stránky podle velikosti obrazovky daného zařízení a bude tak možné přehledně procházet webovou aplikací i na mobilních zařízeních. Přizpůsobivý vzhled bude možné zajistit s využitím technologie Bootstrap.

Obsah práce je členěn do několika následujících kapitol. V každé části provede čtenáře postupem tvorby webové aplikace. V kapitole 2 se nachází rozbor existujících řešení podobných webových stránek a posléze jejich hodnocení podle přínosu pro uživatele. Následuje kapitola 3 popisující princip fungování webové aplikace a poté jaké technologie a vědomosti jsou potřeba k jejímu vývoji i běhu.

Kapitola 4 rozebírá konkrétní návrh řešení pro tuto aplikaci a zmiňuje i specifikaci, která k tomu byla využita. Implementace samotného návrhu a funkcí webu je rozebrána v kapitole 5. Řídí se specifikací a návrhem z předchozí kapitoly. Aplikace bude vytvořena v programovacím jazyce PHP s využitím webových technologií a databázového systému MySQL. Všechny tyto prvky popíše dříve zmíněná kapitola 3 Princip tvorby webu. Kapitola 6 se zabývá způsobem zvoleného testování, průběhem samotného testování implementované webové aplikace a úpravou na základě názorů poptaných uživatelů.

¹<http://presentationsite.euweb.cz/>

Poslední kapitola **7** popisuje dosažené cíle práce a nápady na možné budoucí rozšíření. Na konci tohoto dokumentu se nachází příloha s popisem obsahu přiloženého paměťového média.

Kapitola 2

Analýza existujících řešení

V rámci kapitoly Analýza existujících řešení nám budou představeny již existující webové stránky v sekci 2.1 s podobným zaměřením jako tato práce. Na závěr kapitoly v sekci 2.2 je zhodnocení těchto popsaných stránek. V následující kapitole 3 probereme technologie používané k tvorbě webu.

2.1 Webové stránky

Pro budoucí vývoj práce bylo nutné poohlédnout se po webových stránkách podobného zaměření, ať už z důvodu inspirace, ale i kvůli zjištění možných problémů při návrhu funkcionality budoucí webové aplikace. Tyto webové stránky nabízejí historické informace o událostech a postavách specifické historie. Zde je pár řešení představeno.

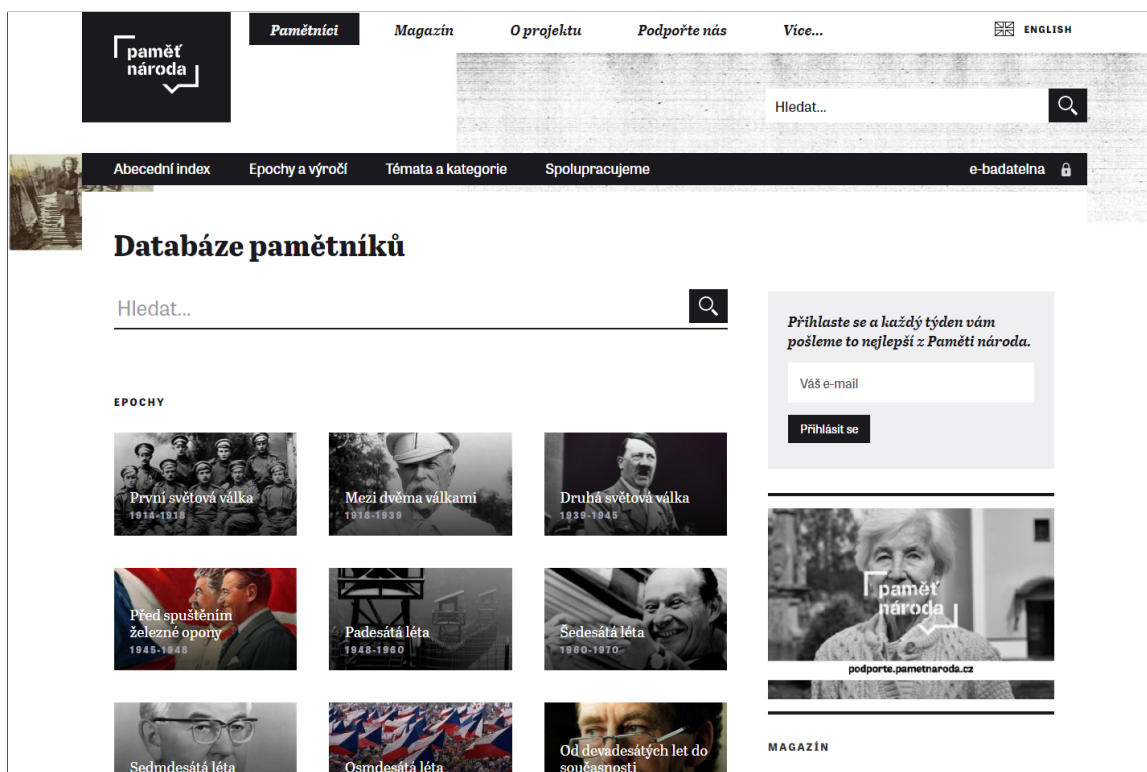
2.1.1 Paměť národa

Paměť národa [22] je zdarma dostupným webem prezentující historické informace. Byl založen jako databáze pamětníků, kteří byli přímými svědky událostí 20. století a funguje tak už od roku 2008. Jeho vzhled můžeme vidět na obrázku 2.1.

Sbírku Paměť národa je spravována neziskovou organizací Post Bellum ve spolupráci s Českým rozhlasem a Ústavem pro studium totalitních režimů poskytující jednu z nejrozsáhlejších sbírek vzpomínek pamětníků v Evropě.

Návštěvník si zde může procházet archiv obsahující příběhy pamětníků, které vycházejí z audio či video záznamu vyprávěných vzpomínek. Spolu s tím jsou přiloženy také dobové i současné fotografie a jejich životopis. Pokud si chce poslechnout celé třeba i několika hodinové nahrávky, musí se zaregistrovat a uvést k čemu hledané informace potřebuje.

Dále jsou na webu stále publikovány články inspirované příběhy pamětníků. Informace je možné hledat pomocí zabudovaného vyhledávače, abecedního seznamu, určité epochy moderních dějin, kategorie nebo také prostřednictvím štítků s odkazy (anglicky tags).



Obrázek 2.1: Snímek obrazovky webu Paměť národa [22].

2.1.2 Ústav pro studium totalitních režimů

Ústav pro studium totalitních režimů [30] je vědecká instituce vzniklá v roce 2008. Tato instituce se zabývá dvěma obdobími našich novodobých dějin, je to doba nacistické okupace 1939–1945 a doba komunistické nadvlády 1948–1989. Články, které jsou na jejím webu publikovány, se zaměřují na proti-demokratickou činnost bezpečnostních složek státu vedoucího k vzniku autoritativních režimů.

Ústav spolupracuje s velkou řadou dalších vědeckých institucí, jako je například předchozí zmíněná Post Bellum, ale také se podílí na modernizaci dějepisu tuzemských i zahraničních škol. Pro veřejnost koná konference, výstavy a přednášky. Ukázka webu je na obrázku 2.2.

Uživatel tohoto webu má na výběr několik historických témat z již zmíněného období dějin. Publikované články mohou v závislosti na obsahu poskytovat dobové fotografie, audio-video nahrávky událostí nebo samotné osobnosti či pamětníky. U každého pamětníka si můžeme přečíst stručný životopis. K vyhledávání zde slouží vyhledávací pole, které ale postrádá možnost filtrování výstupu podle našich preferencí.



Obrázek 2.2: Snímek obrazovky webu Ústav pro studium totalitních režimů [30].

2.2 Shrnutí

Po prozkoumání mnoha webových sítí je patrné, že se weby orientují vždy jen na určitou výseč historie. Jsou určeny pro prezentaci historických záznamů a z toho důvodu jsou pro neznalého člověka hůře přehledné a intuitivní. V případě zobrazení článku, oceníme obsahové možnosti, kdy je návštěvníkovi nabídnut nejen textový materiál, ale i audio-video nahrávky spolu s příloženými obrázky.

Bohužel větší množství historických webů nabízí pouze textové záznamy, případně i fotografie, ale audio-video nahrávky už nikoli. U dvou zmíněných webů je možnost si pustit nahrávku velkou výhodou. Článek pak čtenáře mnohem více zaujme. Užitečná je také propojitelnost článků-osob prostřednictvím tematických štítků, díky nimž můžeme nalézt další podobné historické osobnosti. Čím více má web článků, tím více je potřeba jejich dělení do různých tematických okruhů, kategorií, epoch a zajistit tak čtenáři přehledný a jednoduchý přístup k informacím.

Při hledání podobných webových stránek na české webové sféře si uvědomíme, jak malé množství takových stránek existuje. Proto je skvělé se vůbec pokusit vytvořit webovou aplikaci, která by čerpala z popsaných výhod a k tomu nabízela užitečné funkce navíc.

Kapitola 3

Princip tvorby webu

V této kapitole nalezneme popis síťové architektury (viz sekce 3.1), na které celá tvorba webu stojí. Dále co je to vlastně webová stránka 3.2 a jak vede k vytvoření webové aplikace v sekci 3.3. Závěrečná část kapitoly představí programovací jazyky 3.4 a technologie 3.5 využívané při vývoji webové aplikace. Následující kapitola 4 popíše konkrétní návrh aplikace pro tuto práci.

3.1 Architektura informačních systémů

Tato sekce pojednává o dvou základních typech architektur informačních systémů využívané k tvorbě webu. O architektuře klient-server a dnes nejvíce používané třívrstvé architektuře.

3.1.1 Architektura klient-server

Architektura klient-server [16] je jeden z typů síťové architektury informačních systémů. Skládá se z dvou typů koncových zařízení, z klienta a serveru. Tyto zařízení mezi sebou komunikují a předávají si vzájemně data přes počítačovou síť. Jde tedy o dvouvrstvou architekturu. Toto schéma můžeme vidět zde na obrázku 3.1.

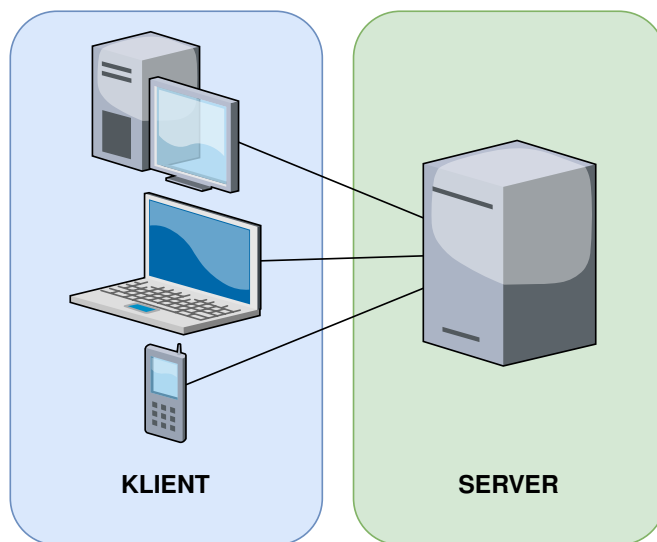
Klient přeloží uživatelův požadavek na dotaz, který odešle serveru, čeká od něj odpověď, kterou překládá zpět tak, aby ji na obrazovce prezentoval uživateli. Server tedy zpracovává dotazy v relační databázi a klient je prezentuje, zajišťuje aplikační logiku a zprostředkovává rozhraní pro uživatele. Se složitostí aplikace (aplikační logiky) rostou nároky na klienty, to vedlo ke vzniku nové koncepce s názvem třívrstvá architektura, která je popsána v následující podsekci 3.1.2.

Výhody

- Architektura umožňuje rozdělit jednotlivé úkoly a zodpovědnosti počítačového systému mezi několik klientů a umožnit snadnější údržbu.
- Data se ukládají centralizovaně na straně serveru, aktualizování údajů je poté mnohem jednodušší, než aktualizovat data na každé stanici zvlášť.
- Servery jsou bezpečnější než většina klientů, protože mohou lépe kontrolovat přístup k datům a využítí zdrojů a zaručovat tak, že přístup mají pouze oprávnění klienti [27].

Nevýhody

- Velký počet souběžných požadavků klientů na daný server může snadno přetížít síť a znepřístupnit služby serveru dalším klientům.
- Pokud dojde k výpadku serveru, žádosti klientů nemohou být splněny [27].



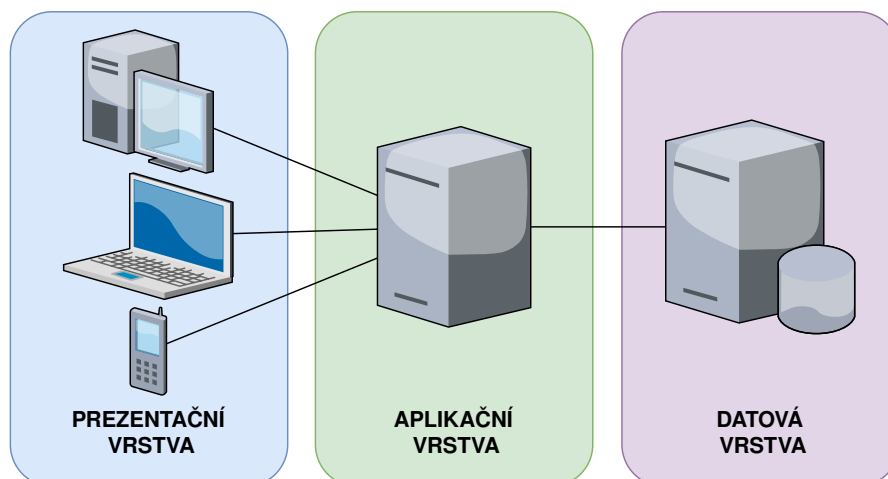
Obrázek 3.1: Schéma architektury Klient-server. Vytvořeno v [6].

3.1.2 Třívrstvá architektura

Třívrstvá architektura [17] je další z typů architektury informačních systémů a rozšiřuje předcházející architekturu 3.1.1. Je rozdělena na, to co vidí a používá uživatel (prezentační vrstva), vykonávání aplikační logiky aplikace (aplikační vrstva) a práci s daty (datová vrstva). Zde je schéma pro lepší představu 3.2.

Prezentační vrstva zajišťuje vstup požadavků od uživatele a prezentaci výsledků k uživateli. Odvíjí se od platformy nebo zařízení, na kterém vrstva běží. Aplikační vrstva provádí výpočty a operace mezi vstupně-výstupními požadavky a daty podle své aplikační logiky vykonávanými interpretem určitého programovacího jazyka nasazeného na serveru. Datová vrstva je nejnižší vrstva architektury zajišťující operace s daty. Tyto data jsou ukládána a spravována prostřednictvím systému řízení báze dat a tvoří tak databázi.

Smyslem architektury je oddělení jednotlivých vrstev tak, aby na sobě nebyly závislé. Rovnoměrnější rozdělení výkonu mezi zařízení uživatele a server vede k menší zátěži těchto zařízení a proto třívrstvou architekturu využívá většina moderních webových aplikací. Implementace takového systému může být dost složitá, kvůli kombinaci různých platform nebo operačních systémů. Tím se zvyšují i nároky na znalosti potřebné k vývoji.



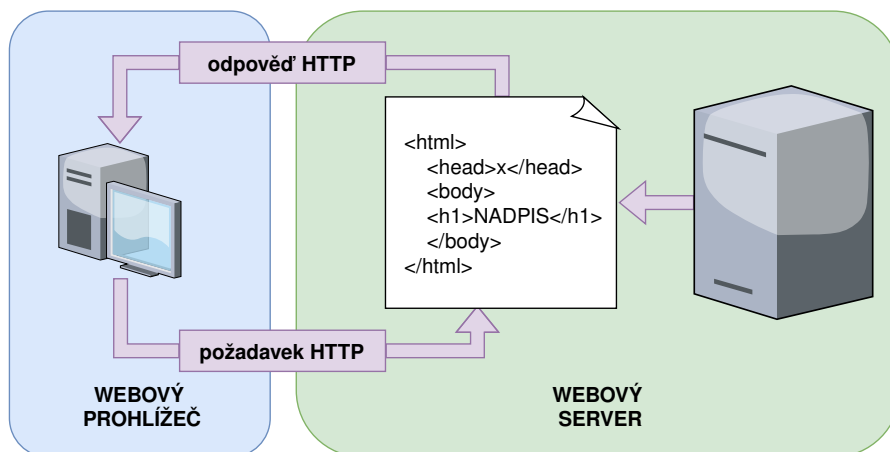
Obrázek 3.2: Schéma třívrstvé architektury. Vytvořeno v [6].

3.2 Webová stránka

Webová stránka [28] je dokument získávaný pomocí protokolu HTTP (Hypertext Transfer Protocol) prostřednictvím služby WWW (World Wide Web), která běží na webovém serveru. Tento dokument je kombinací textů, multimediálního obsahu a odkazů vytvořených v jazyce HTML a graficky upravených v jazyce CSS, které budou blíže popsány v sekci 3.5. Více webových stránek potom dohromady tvoří web. V závislosti na tom, jakým způsobem se jejich obsah zobrazuje, můžeme webové stránky rozdělit na statické nebo dynamické. Uživatel pak v praxi k těmto stránkám přistupuje přes webový prohlížeč.

3.2.1 Statické stránky

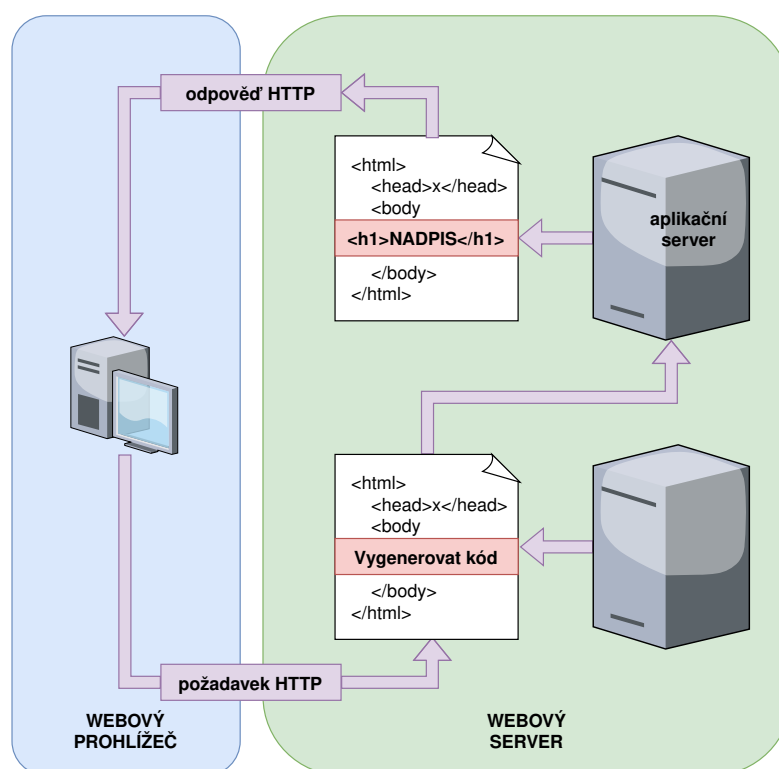
Jsou to stránky [2], ve kterých obsah zůstává nezměněn, pokud se nezmění ze zdrojového kódu. Každá stránka je reprezentována jako samostatný soubor a v případě, že je změněna hlavní stránka, vyžadují úpravu i ostatní stránky. Přenos k webovému prohlížeči probíhá tak, že webový prohlížeč požádá o webovou stránku a webový server ji odešle ve stejné podobě, ve které byl na server uložen (viz obrázek 3.3).



Obrázek 3.3: Zpracování statické webové stránky. Vytvořeno v [6].

3.2.2 Dynamické stránky

Dynamickou webovou stránku [2] webový server upraví před odesláním webovému prohlížeči, který o ni požádal. Za tuto úpravu odpovídá speciální software nazývaný aplikační server. Aplikační server najde a zpracuje instrukce kódu programovacího jazyka v požadované stránce, vloží místo něj vygenerovaný kód HTML a vznikne tak statická stránka (viz obrázek 3.4). Ta je opět předána webovému serveru, který ji odesílá webovému prohlížeči.



Obrázek 3.4: Zpracování dynamické webové stránky. Vytvořeno v [6].

3.3 Webová aplikace

Webová aplikace [2] je kolekcí statických a dynamických webových stránek poskytovaných uživateli z webového serveru. Aplikace je možné aktualizovat a spravovat bez nutnosti šířit a instalovat software na potenciálně tisíce uživatelských počítačů.

Jsou populární především pro všudypřítomnost webového prohlížeče jako klienta, který se pak nazývá tenkým klientem, neboť sám o sobě aplikační logiku nezná. Tedy jsou nejčastěji stavěny na již dříve zmíněné třívrstvé architektuře 3.1.2. Aplikační logika, to co bude aplikace vykonávat, je psána v programovacích jazycích jako jsou například PHP, ASP.NET a Ruby. Jazyky budou popsány níže v sekci 3.4.

Pro ulehčení tvorby existuje mnoho takzvaných frameworků, o kterých se dozvíme více v podsekcí technologie 3.5.6. Frameworky nabízejí díky automatizaci procesu tvorby programátorům možnost rychlejšího vývoje. Dále jednoduchost a přehlednost kódu, která vede k menší chybovosti. Některé části textu pod licencí Creative Commons 3.0 převzaty z [7].

Webové aplikace umožňují:

- Funkčnost webových stránek bez ohledu na typ operačního systému či platformu. Není třeba vyvíjet aplikaci vícekrát pro různá prostředí a můžeme ji nabídnout téměř kdekoliv.
- Shromažďování, ukládání a analyzování dat poskytnutých uživateli, kteří web navštívili. Je možné ukládat data z formuláře přímo do databáze, vyjímat data a vytvářet zprávy pro analýzu.
- Zbavit návrháře webu nutnosti neustále aktualizovat měnící se obsah webu. Ti, kteří webu poskytují obsah, například redaktoři zpráv, dodají obsah a webová aplikace si automaticky aktualizuje příslušné místo na webové stránce. [2]

3.4 Programovací jazyky

Programovací jazyky [9], o nichž se zde budeme bavit, umožňují webovým aplikacím odpovídat na interakce uživatele a využívat dynamického chování. Programovací jazyk je nástroj pro tvorbu systémů, aplikací či webových stránek. Je to prostředek mezi programátorem a počítačem, díky němuž programátor algoritmus zapíše a přenesení do počítače dostupnými technickými prostředky pro vytvoření potřebné aplikace.

Jazyky je možné dělit dle různých kritérií. Dle míry abstrakce rozlišujeme nižší a vyšší programovací jazyky. Dle způsobu spuštění a překladu jsou definovány jazyky kompilované (například C, Java) a interpretované (například PHP, Python).

Používají se dva typy jazyků podle toho, kde jsou zpracovávány. Jsou to jazyky [8] vykonávané na straně klienta (client-side programming) a na straně serveru (server-side programming).

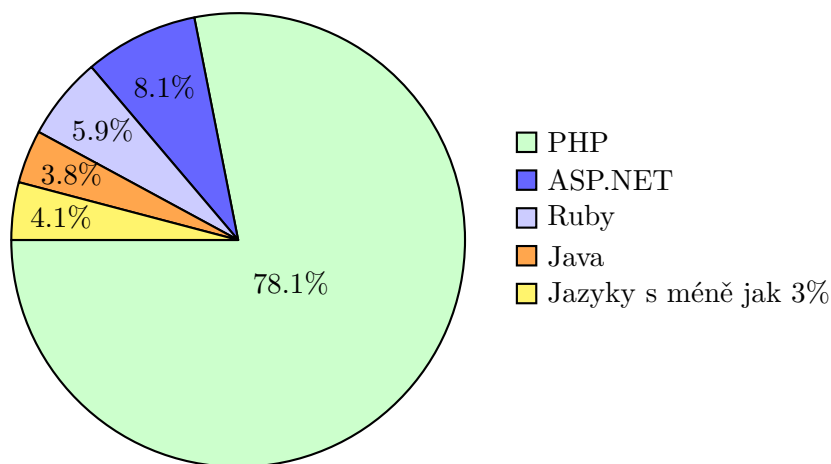
Programování na straně klienta používá jazyk, který může webový prohlížeč uživatele zpracovat. Nejstandardnějším takovým jazykem na straně klienta je JavaScript. JavaScript může být přímo součástí zdrojového kódu webové stránky nebo ve zvlášť oddělených souborech. Při nalezení kódu, webový prohlížeč kód provede a zobrazí případné změny uživateli prohlížeče.

Oproti tomu programování na straně serveru se provede na webovém serveru. Poté co uživatel vydá požadavek pomocí adresy na webovou stránku v prohlížeči, dochází k vyko-

nání kódu, následně webový server odešle kód HTML zpět do webového prohlížeče. Webové aplikace obsahující programování na straně serveru často přistupují k databázím nebo souborům na webovém serveru.

Programovací jazyk lze využít za předpokladu, že jej bude možné na webovém serveru zpracovat. K tomu je potřeba software, který aplikaci dokáže provozovat. V případě jazyka PHP to bude interpret a jazyk C využije kompilátor.

Následující obrázek 3.5 zobrazuje využití programovacích jazyků na serverové části, které jsou využívány k tvorbě webových aplikací. Z dat [26] vyplývá, že existuje velké množství webů, které při svém chodu využívají jazyk PHP. Na druhém místě se umístil ASP.NET a třetí pozici má Ruby. Tyto tři jazyky nám sekce níže popíše.



Obrázek 3.5: Využití programovacích jazyků na serverové části ze dne 6.12.2021 [26].

3.4.1 PHP

Jazyk PHP [38] (původně Personal Home Page, nyní PHP: Hypertext Preprocessor) je jedním z nejpoužívanějších programovacích jazyků pro vytváření webových aplikací. Používá se na straně webového serveru a slouží ke generování HTML kódu webové stránky a jeho dynamického obsahu, ten pak server odesílá prohlížeči.

Jazyk PHP je interpretovaný skriptovací jazyk, proto mu stačí mít na serveru instalovaný program PHP interpret, který vykonává instrukce kódu jednu za druhou a nepotřebuje celý zdrojový kód překládat. Jazyk lze využít i ke spuštění skriptů (spustitelný textový soubor ve skriptovacím jazyce) z příkazové řádky serveru. Administrátoři si tak mohou ulehčit a automatizovat práci.

Výhodou jazyka [25] je jeho nezávislost na platformě a bezplatná dostupnost. Běží na všech operačních systémech (Windows, Linux, Unix, MacOS a další). Důležitou vlastností je také podpora velkého rozsahu databází jako jsou MySQL, PostgreSQL, SQLite a MariaDB. Za dobu svého působení nasbíral početnou komunitu především díky jeho snadné naučitelnosti. Díky flexibilitě není striktně jen na generování kódu HTML, ale lze jej využít také na vytváření dokumentů XML, obrázků, Flash animací, souborů PDF a dalších.

3.4.2 ASP.NET

ASP.NET [32] je skriptovací platforma součástí .NET frameworku od společnosti Microsoft pro tvorbu moderních webových aplikací a služeb s .NET. Je nezávislý na typu operačního systému (Windows, Linux, MacOS a další). Především se využívá pro tvorbu webových aplikací v jazyce C#. Webové aplikace je možné vytvářet zdarma, ale v případě nasazení na server, jsou webové služby v tomto případě většinou zpoplatněné z důvodu licence.

Je založen na CLR [35] (Common Language Runtime) virtuálním stroji, který využívají všechny aplikace postavené na .NET Frameworku. Aplikace je tak možné psát v jakémkoli jazyce podporující CLR, jako jsou například Visual Basic, JScript.NET, C#, ale i mutace jazyku Python a dalších.

Framework .NET [18] rozšiřuje o zpracování webových požadavků v jazycích C# nebo F#. Umožňuje vytváření dynamických webových stránek za použití jazyka C#. Nabízí knihovny návrhových vzorů k vývoji webu, systémy autentizace pro ošetření přihlašovacích procesů, moduly pro práci s databází a další rozšíření zejména pro editory využívané k vývoji webových aplikací, jako je kontrola syntaxe kódu a automatické doplnění kódu.

3.4.3 Ruby

Ruby [35] je objektově orientovaný interpretovaný skriptovací jazyk čerpající inspiraci z jazyků Perl, Smalltalk, Eiffel, Ada, a Lisp. Jazyk je doporučován začátečníkům kvůli jeho snadné naučitelnosti. Do povědomí lidí přišel zejména kvůli frameworku Ruby on Rails a je tak široce využíván pro tvorbu webových aplikací. Stejně jako další dynamické jazyky pro tvorbu webu i Ruby se zaměřuje na zrychlení procesu tvorby aplikací. Jazyk podporuje téměř všechny hlavní platformy (Windows, Linux a MacOS).

Nabízí zajímavou syntaxi [31] se silným objektovým založením. Téměř vše je v jazyce objektem. Kód v jazyce Ruby se nekompile, ale překládá za běhu programu. Vyniká přehledností a efektivností, ale za cenu nižší rychlosti. Aplikace zapsané v tomto jazyce zaberou o třetinu řádků méně, než kolik by zabrala například aplikace se stejnou funkcionalitou v jazyce C. Díky tomu se hodí k psaní velkých projektů, skriptů, zpracování textových souborů, ale i GUI aplikací.

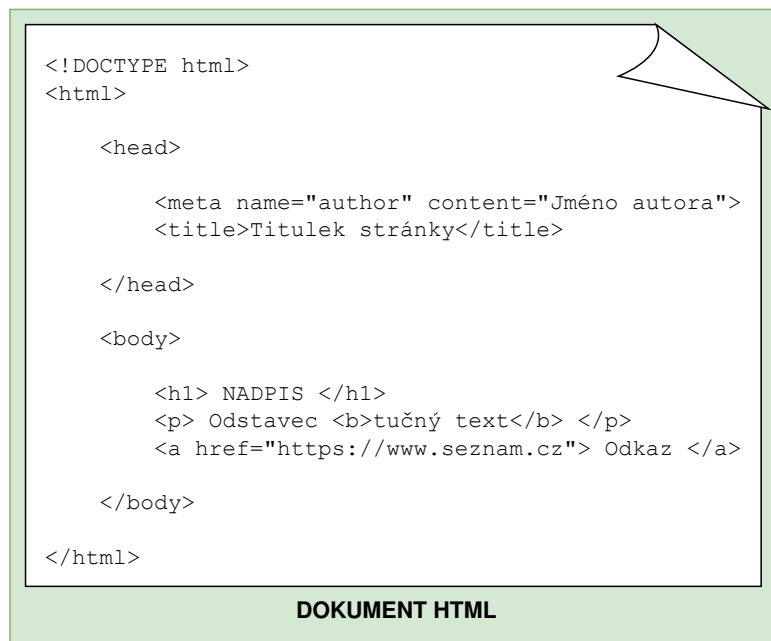
3.5 Technologie

Zde v sekci budou popsány zásadní technologie pro vývoj webové aplikace. Některé jsou důležité pro správné zobrazení a vzhled stránky, jiné umožňují celistvost a funkčnost na jakémkoli zařízení, další jako AJAX nevyžadují interakci uživatele kvůli změně obsahu. Takové výhody nejvíce ocení návštěvník webové stránky, ale existují také frameworky, technologie ulehčující práci programátorům těchto webových aplikací a systémy zaměřené pro práci s daty.

3.5.1 HTML

HTML [11] (Hypertext Markup Language) je hypertextový značkovací jazyk pro tvorbu webových stránek a dříve sloužil i k formátování vzhledu (dnes se tím zabývá jazyk CSS). Už z názvu vyplývá, že základní složkou je značka a hypertext. Tyto značky či elementy dohromady tvoří obsah webové stránky a umožňují dávat význam určitým blokům textu. Hypertext zajišťuje vzájemné propojování textů nebo jiných webových stránek pomocí odkazů.

Webová stránka musí mít jasnou strukturu dokumentu [37] (viz obrázek 3.6), aby každý klient, v tomto případě webový prohlížeč, mohl s dokumentem pracovat. Značky existují párové a nepárové, v případě párových značek je na konec daného značkového bloku přidáno navíc lomítko, například `<body> obsah_bloku </body>`. Každý soubor HTML musí začínat označením o jaký typ dokumentu jde `<!DOCTYPE html>`, následuje párová značka `<html>` označující začátek dokumentu. Poté hlavička stránky `<head>`, v jejímž bloku lze popsat meta informace o webové stránce, titulek stránky a propojení se soubory CSS definující vzhled stránky. Za hlavičkou se nachází prostor pro samotný obsah webu označený pomocí párového `<body>`.



```
<!DOCTYPE html>
<html>

  <head>

    <meta name="author" content="Jméno autora">
    <title>Titulek stránky</title>

  </head>

  <body>

    <h1> NADPIS </h1>
    <p> Odstavec <b>tučný text</b> </p>
    <a href="https://www.seznam.cz"> Odkaz </a>

  </body>

</html>
```

DOKUMENT HTML

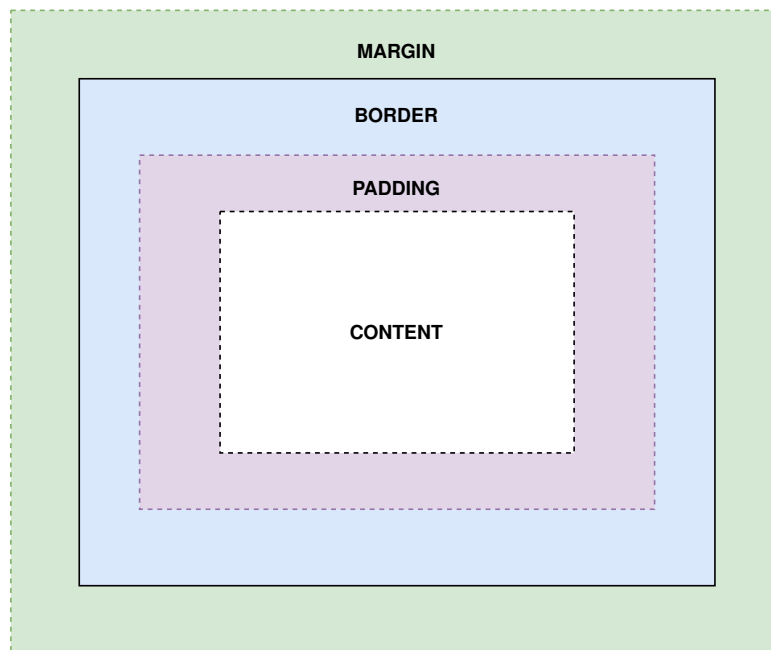
Obrázek 3.6: Struktura dokumentu HTML. Vytvořeno v [6].

3.5.2 CSS

CSS [10] (Cascading Style Sheets), v překladu kaskádové styly, jsou kolekce metod pro úpravu formátování a zobrazení webových stránek psaných v jazyce HTML, XHTML a XML. V podstatě upravuje způsob zobrazení jednotlivých elementů souboru HTML, nejen těch viditelných, ale i elementů, které tvoří grafické rozložení stránky jako je hlavička, patička, prostor pro obsah a dalších. Nevýhodou může být prohlížeč nepodporující určité vlastnosti formátování.

Důležitou vlastností rozložení obsahu na webové stránce je takzvaný model CSS Box [13] (viz obrázek 3.7). Každý element HTML na stránce je v podstatě box mající obsah (content), kterým může být například text, obrázek nebo jiný element. Kolem něj jsou tři oblasti zvané padding, border a margin. Padding je oblast od obsahu k okraji a určuje vnitřní odsazení elementu od okraje. Border vyznačuje okraj elementu a margin je opakem paddingu, tedy vnější odsazení od okraje a ostatních elementů. Tyto vlastnosti CSS umožňují definovat prostor mezi elementy a měnit tak rozložení stránky. Text publikován pod licencí Creative Commons 4.0 a převzat z [13].

V praxi se využívají tři způsoby deklarování stylů CSS. Tím prvním je zápis přímo ve zdrojovém souboru HTML u elementu použitím atributu (vlastnosti) `style=""`, tedy například `<p style="color: blue;"> text zbarvený modře </p>`. Druhou možností je umístění popisu stylů v hlavičce webové stránky do párového elementu `<style>`, například `<style> p {color: blue;} </style>`. Třetím způsobem je použití externího souboru (anglicky *stylesheet*) s příponou `.css` odkazovaného v hlavičce souboru HTML pomocí elementu `<link>`, například `<link rel="stylesheet" type="text/css" href="styly.css">`. Poté už jen stačí vytvořit soubor s názvem `styly.css` a do něj vložit `p {color: blue;}`.



Obrázek 3.7: Schéma modelu CSS Box. Vytvořeno v [6].

3.5.3 Bootstrap

Bootstrap [5] je zdarma dostupný CSS framework vytvořený pro vývoj webových aplikací. Nabízí velké množství knihoven s užitečnými a opakovaně použitelnými kódy napsanými v jazycích HTML, CSS a JavaScriptu. Poskytuje sadu nástrojů pro vývojáře frontendových aplikací. Frontend je jednoduše část aplikace zobrazující obsah uživateli, jako například formulář. Bootstrap umožňuje rychle a jednoduše vytvářet plně responzivní webové stránky. Responzivita umožňuje zobrazení webové stránky optimalizované podle velikosti obrazovky a typu zařízení.

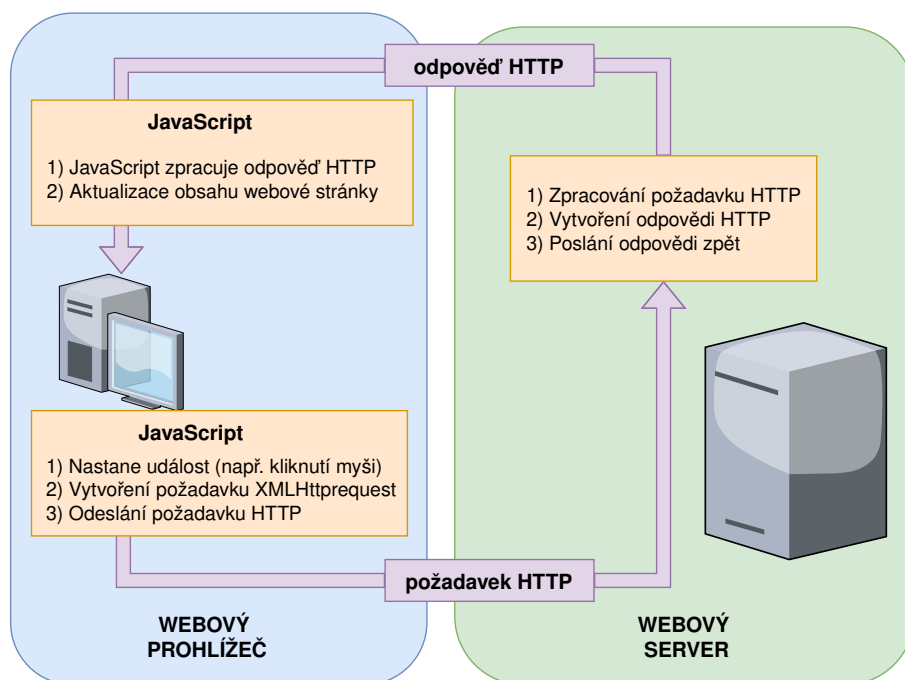
Má předdefinované rozvržení všech elementů webu a to jedním způsobem pro všechny zařízení. Je tedy možné mít stejnou stránku například jak pro počítače, tak i pro mobilní telefony. Pokud se nám rozvržení nelíbí, můžeme si vybrat z řady šablon vytvořených tvůrci Bootstrapu nebo komunitou. Framework přichází s řadou komponent rozšiřující funkce webu jako jsou navigační lišty, rozbalitelné nabídky, indikátory průběhu, náhledy a další. Součástí je množství pluginů jazyka JQuery poskytující různé animace, modální vyskakovací okna a přechody. Tyto funkce tak obohacují interakci uživatele s webovou aplikací.

3.5.4 AJAX

AJAX [36] (Asynchronous JavaScript and XML) je technologie umožňující webovým aplikacím získávat informace pro uživatele a načítat je bez nutnosti aktualizovat webovou stránku. Jde o vytváření takzvané asynchronní webové aplikace, kde se data vyměňují mezi webovým prohlížečem a serverem na pozadí. Po uživateli se nevyžaduje nekonečné obnovení stránky, to vede k větší přívětivosti aplikace a oblibě této technologie.

Proces [8] probíhá na prohlížeči klienta prostřednictvím jazyka JavaScript, data jsou poté posílána v podobě kódu XML. XML (Extensible Markup Language) je jazyk podobný jazyku HTML, kde jsou data ve formátu snadno čitelném člověkem i strojem. Interpret jazyka JavaScript vytvoří na podnět nějaké události požadavek XMLHttpRequest, který poté odešle webovému serveru. Server jej zpracuje a odešle zpátky odpověď HTTP. Odpověď převezme JavaScript a aktualizuje podle ní obsah webové stránky, jak můžeme vidět zde na obrázku 3.8.

AJAX nabízí obrovské množství využití. Jsou to především sociální sítě jako například Facebook nebo Twitter. Tam se sním můžeme setkat například při jakémkoli upozornění, psaní zpráv a posunutí na konec stránky, kde proběhne načtení dalšího příspěvku. Dnes je tato technologie často součástí frameworků pro vývoj webových aplikací.



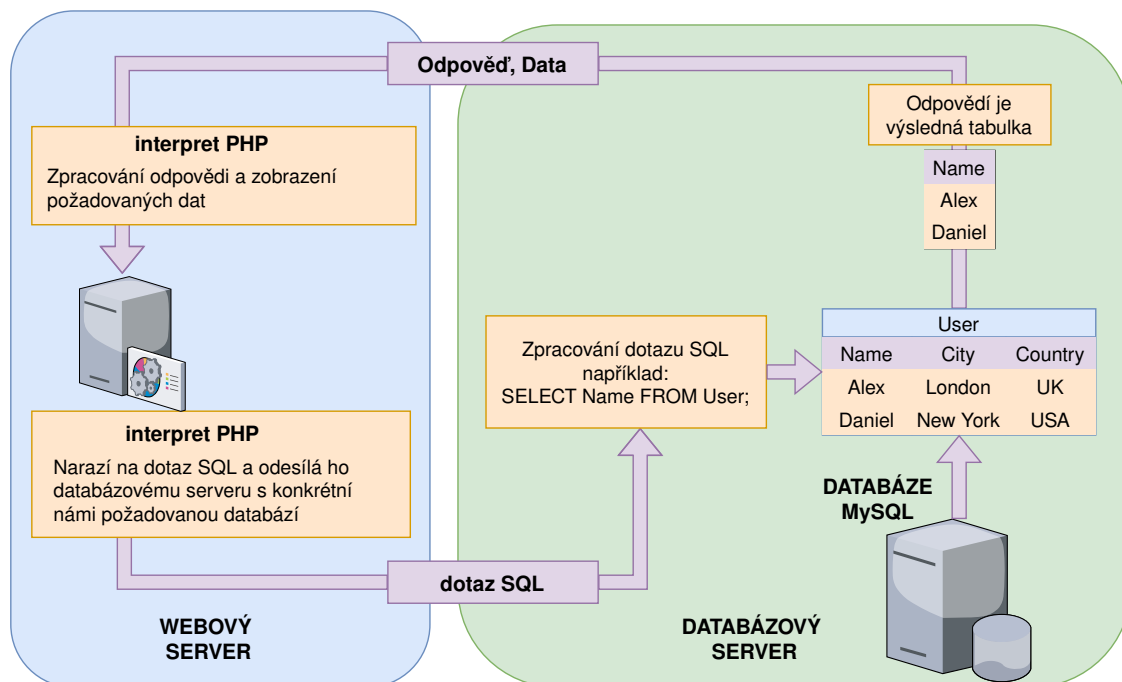
Obrázek 3.8: Zpracování požadavku/odpovědi pomocí AJAXu. Vytvořeno v [6].

3.5.5 MySQL

MySQL [12] (My Structured Query Language) je relační databázový systém typu DBMS (Database Management System, česky systém řízení báze dat) vlastněný společností Oracle. Tento systém zajišťuje práci s daty v databázi a tvoří rozhraní mezi aplikacemi a daty. Relační databáze je tvořena tabulkami, které spolu mohou souviset a tvoří tak relaci. Databázi tvoří strukturovaná data uspořádaná do tabulek, každá tabulka obsahuje řádky zvané záznamy (konkrétní data) a sloupce zvané atributy, které určují datový typ záznamu. Nad

daty respektive tabulkami se provádí takzvané dotazy pomocí strukturovaného dotazovacího jazyka SQL (Structured Query Language), který umožňuje manipulaci s daty v databázi.

MySQL je podporován na všech platformách a zdarma. Díky své jednoduché syntaxi je snadno naučitelný. V praxi jde o spolupráci webového serveru s databázovým serverem a jde tedy o architekturu klient-server. Zpracování probíhá tak, že webová aplikace (v případě jazyka PHP spouštěná PHP interpretem na webovém serveru) narazí na dotaz SQL. Dotaz se provede jako příkaz prostřednictvím příkazové řádky konzole MySQL spojené s databází MySQL. Databáze zpátky odesílá odpověď a data, která jsme požadovali. Na obrázku 3.9 můžeme vidět příklad tohoto zpracování.



Obrázek 3.9: Zpracování dotazu SQL. Vytvořeno v [6].

3.5.6 PHP framework

Tato bakalářská práce, respektive webová aplikace, bude psána v jazyce PHP, proto jsou zde uvedeny tři známé PHP frameworky [19]. Co to vlastně je framework?

Framework [34] je ucelený soubor do sebe pasujících knihoven, které ulehčují vývojářům webových aplikací proces tvorby webu. Tyto knihovny obsahují komponenty a pro ně i obsluhující funkce. Není tedy třeba stále programovat obsluhu registračního formuláře, ale jen využít již existující komponentu s tímto zaměřením. Díky vysoké míře abstrakce, kdy jsou jednotlivé komponenty zapouzdřeny a schovány před uživateli, umožňuje vývojáři soustředit se na hlavní funkce aplikace, to vede k ušetření času a menšímu množství napsaných řádků kódu. Dodržují návrhové vzory jako je model MVC blíže popsany v kapitole 5, které zajišťují oddělení prezentace a logiky. Díky přehlednosti je pak aplikace snáze udržovatelná.

Laravel

Laravel [14, 19] je nejoblíbenějším PHP frameworkem ve světě. Vychází v jeho nyní nejnovější verzi 8.x a je dostupný zcela zdarma. Umožňuje vyvíjet webové aplikace na principu softwarové architektury MVC (model-view-controller). Model zde provádí aplikační logiku, view prezentuje data a controller se stará o interakce uživatele projevující se změnou modelu a zobrazením stránky.

K důležitým schopnostem patří elegantní syntaxe, která je moderní a snadno udržovatelná, jde zde hlavně o jednoduchost a čitelnost. Jeho knihovny nabízejí často používané funkce z oblasti autentizace uživatelů, zpracování dotazů, propojení a přístupy k relačním databázím, nastavení cest webových stránek a posílání emailů.

Framework také zahrnuje lokální vývojové prostředí zvané Homestead umožňující ještě snadnější začátek projektu bez ohledu na platformu nebo systém. Jedná se o virtuální prostředí bez nutnosti instalace PHP. Obsahuje webový a databázový server pro lokální propojení s naší aplikací. Jinou možností, v případě již nainstalovaného PHP, je instalace přes program Composer a spuštění v před připraveném vývojovém serveru PHP. Laravel má vlastní šablonovací nástroj webových stránek Blade, který se liší od ostatních možností přidat kód PHP přímo do obsahu podhledu.

Díky velké aktivní a stále se rozrůstající komunitě není problém vyřešit jakoukoli záležitost. Další informace je možné čerpat z oficiální anglické dokumentace [14] a pro českou komunitu je k dispozici neoficiální česká dokumentace zde [15].

Symfony

Druhým nejpoužívanějším frameworkem také zdarma je Symfony [19, 23] ve verzi 6.x. Umožňuje zrychlit vývoj a údržbu webové aplikace zbavením se opakovaného psaní stejných funkcionalit. Architektura je postavená na modelu MVC stejně jako předchozí Laravel.

Nabízí flexibilní samostatně znovupoužitelné komponenty. Lze s nimi plnit jakékoli požadované úkoly od konfigurace objektů, vytváření formulářů, zpracování směrovaných dotazů, autentizaci až po šablonování a další.

Všechny potřebné závislosti, například knihovny obsahující komponenty, nainstalujeme prostřednictvím programu Composer. Pokud nevyužijeme Composer, je k dispozici aplikace Symfony CLI pro založení projektu webové aplikace přímo z příkazové řádky. Poskytuje všechny nástroje potřebné k vývoji a spuštění Symfony aplikace. Aplikace je pak možné spouštět na lokálním serveru PHP nebo námi zvoleném propojeném webovém serveru. Pro tvorbu šablon webových stránek je součástí frameworku zdarma dostupný nástroj Twig.

Framework využívají například projekty Drupal, což je software pro vytváření redakčních systémů pro správu obsahu nebo software phpBB pro vytváření interaktivního fóra. Symfony má rozsáhlou vývojářskou komunitu a umožňuje uživatelům nahlédnout do některých projektů provozovaných na tomto frameworku. Více informací v dokumentaci [23].

Nette

Nette [21] je český open-source framework, jejímž zakladatelem je David Grudl. Aktuální vydaná verze je 3.1.0. Zaměřuje se na eliminaci bezpečnostních rizik a přichází s novými bezpečnostními metodami, proto se jedná o nejbezpečnější PHP framework.

Nabízí vlastní šablonovací systém Latte, ladící nástroj Tracy a s jednoduchou syntaxí umožňuje ještě snazší a přehlednější vývoj webových aplikací. Framework podporuje moderní technologie HTML5 a AJAX.

Poskytuje rozsáhlou kvalitní dokumentaci jak v angličtině, tak i v češtině. Má širokou aktivní komunitu a to zejména v České republice. Díky tomu byl podle průzkumu v roce 2015 nejpoužívanějším frameworkem PHP v ČR [24]. Tento framework využívají společnosti jako například T-Systems, Fio Banka, GE Money, DHL a ESET. Framework Nette byl vybrán pro implementaci této práce a podrobněji bude popsán v kapitole 5 Implementace.

Kapitola 4

Návrh webové aplikace

Tato kapitola představí návrh řešení praktické části bakalářské práce. Na začátek si rozebereme klíčovou funkčnost budoucí aplikace (viz sekce 4.1), tato část je zásadní a měla by odpovídat požadavkům zadání. Na základě specifikace požadavků vytvoříme diagram případů užití v sekci 4.2 zobrazující možné interakce se systémem. Následně se budeme věnovat návrhu uživatelského rozhraní 4.3, jenž pro tuto interakci slouží. Poslední část nabídne abstraktní popis uložení dat v podobě ER diagramu v sekci 4.4. Další kapitolou následuje implementace tohoto navrženého řešení.

4.1 Specifikace požadavků

Specifikace požadavků je pro vývoj každé aplikace nesmírně důležitá. Je určována zadáním od zadavatele, definuje to, co by měla budoucí aplikace umět a splňovat. Takové informace je nutné srozumitelně popsat.

Jak již bylo řečeno, tato webová aplikace má za cíl zobrazovat historické události a osoby. Návštěvníkovi nabídne články a životopisy osob, u kterých si bude moci prohlížet fotografie a audio-video nahrávky. Mezi klíčové vlastnosti patří:

Uživatelé aplikace

Podle zadání je nutné rozdělit uživatele webové aplikace na dva typy. První typ je přihlášený uživatel, je to uživatel s oprávněním modifikovat obsah, který je prezentován ostatním návštěvníkům webu. Pouze jemu bude umožněno se přihlásit prostřednictvím přihlašovacích údajů, aby nikdo jiný nemohl poničit prezentovaný obsah webové stránky. Pokud zapomene své heslo, může využít funkce obnovení hesla. Po přihlášení může zvolit přidání nového obsahu. To zahrnuje vytvoření a editaci článku či životopisu osoby. Jakmile takový obsah existuje, může jej smazat.

Zde budou figurovat dvě systémové role, správce a běžný uživatel. Běžný uživatel uvidí pouze obsah schválený správcem k zobrazení a svůj vytvořený obsah. Editovat článek nebo životopis osoby může v případě, že je jeho autorem. Role správce umožňuje uživateli vidět veškerý obsah, který byl na webu vytvořen a vybrat jaké články či životopisy budou zobrazeny pro návštěvníka. Bude mít také možnost přehledně vidět seznam uživatelů systému, zobrazit si seznam vytvořených článků nebo životopisů konkrétního uživatele a nakonec smazání konkrétního uživatele. Uživateli je možné zvýšit jeho roli v systému z běžného uživatele na správce, nebo naopak snížit.

Druhým typem je nepřihlášený uživatel, tedy ostatní návštěvníci bez přihlašovacích údajů. Může procházet stránky webu, zobrazovat články a životopisy osob schválené správcem k zobrazení, napsat komentář k článku nebo změnit jazyk textů.

Článek, životopis osoby

Jsou to hlavní komponenty prezentačního webu. Tvoří jeho obsah a každý kdo jej navštíví, je uvidí. Každý článek zobrazí titulek, stručný popis a text. K tomu bude přidáno datum publikace a autor článku. Životopis bude co se týče informací pestřejší. Je zde nutné uvést jméno dotyčné osoby, datum a místo narození a případně i úmrtí. Dále nabídne podrobný popis osoby nebo jejího života.

Oba základní prvky pak bude možné doplnit o fotografie a audio-video nahrávky. Propojitelnosti podobných témat dosáhneme přidáním tagů a témat/kategorií. Jakmile se přihlásíme, můžeme obsah obou prvků modifikovat. V tomto případě musí být uživatel autorem nebo mít roli správce. Článek lze také navíc doplnit o odkazy na osoby zmíněné v textu nebo pod něj umístit komentář.

Vyhledávání

Vyhledávání je důležitá funkce každého webu a také nástroj k nalezení požadované věci, v tomto případě článku nebo osoby. K dispozici bude fulltextové vyhledávání podle názvu článku nebo jména osoby, případně obojího. Další možností je kliknutí na určitý tag, téma nebo odkaz na osobu s následným výpisem podobných výsledků. Poslední způsob je zobrazení nabídky veškerých témat/kategorií skrze položky hlavní nabídky.

4.2 Diagram případů užití

Diagram případů užití [33] (Use Case Diagram) zobrazuje chování systému z vnějšího pohledu tak, jak by ho viděl samotný uživatel. Účelem diagramu je přehledně a srozumitelně popsat funkcionalitu aplikace bez informací, jakým způsobem se toho dosáhne. Jde o takzvanou černou skříňku, kde skryjeme vnitřní logiku aplikace před uživatelem.

Diagramy se skládají ze tří hlavních částí, z případů užití (například zobrazit článek), aktérů (například správce a návštěvník) a vztahů mezi nimi. Na základě předchozí specifikace vznikl diagram, který můžeme vidět zde na obrázku 4.1.

V případě digramu případů užití této aplikace budeme rozlišovat tři typy aktérů:

Návštěvník

Všichni nepřihlášení uživatelé této webové aplikace spadávají do role návštěvník. Takovému uživateli bude umožněno procházet webové stránky proklikáváním odkazů na články a životopisy osob, nebo přesměrováním se na jiný obsah pomocí tagů (značek příslušící například nějakému tématu). Další možností je vyhledání článku nebo životopisu prostřednictvím vyhledávacího pole.

Tyto články a životopisy si bude moct zobrazit, v případě doplnění jejich obsahu fotografiemi a audio-video nahrávkami je také prohlížet a spouštět. Pokud bude chtít, může daný článek okomentovat. Poslední funkcí je změna jazyku textového obsahu stránky.

Běžný uživatel

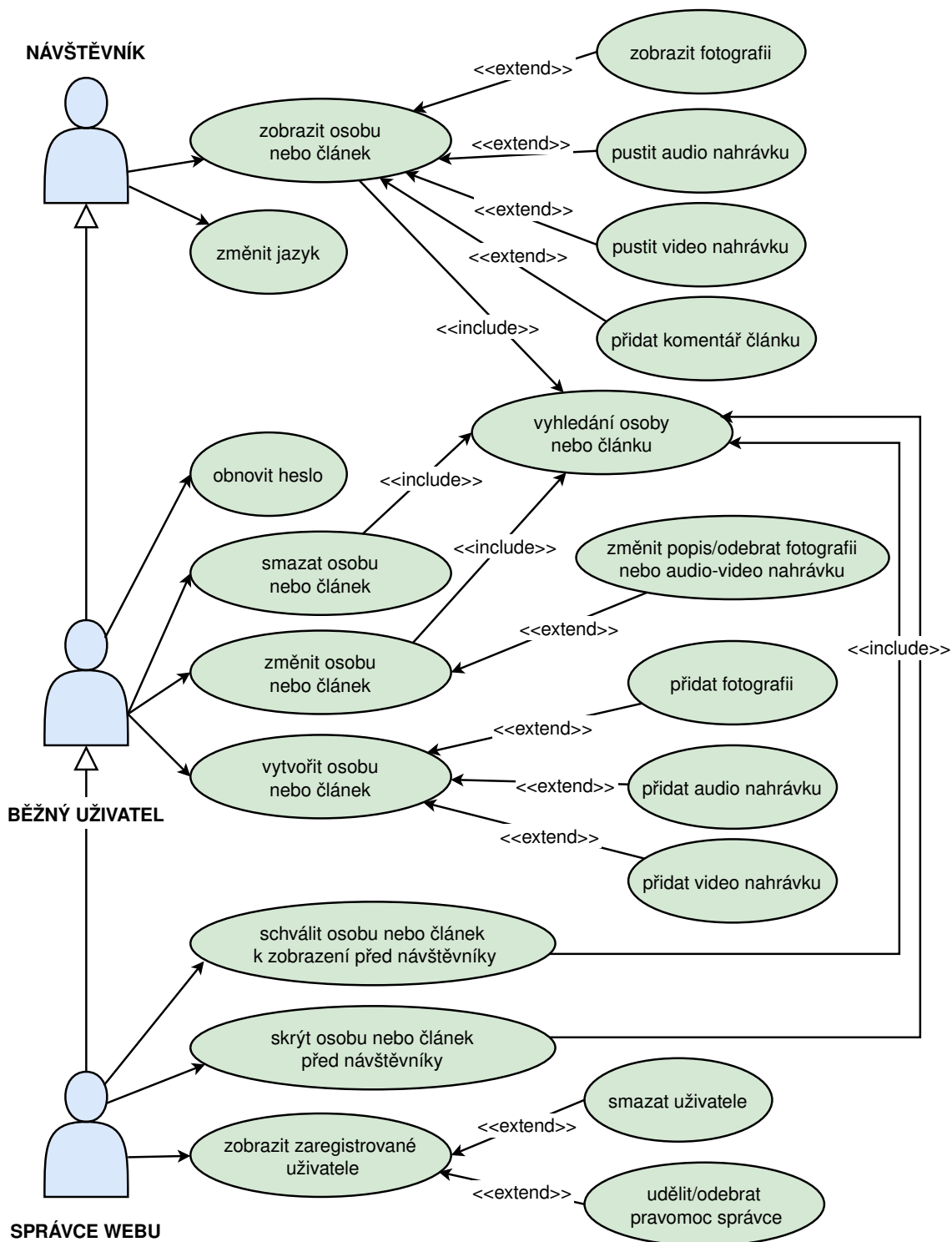
Je to jeden ze dvou uživatelů, kteří budou moci modifikovat obsah webové aplikace. K jeho pravomocem náleží přidávat, upravovat a mazat články nebo životopisy osob publikovaných na webové stránce včetně jejich obsahu. Tedy měnit textové údaje, vybírat fotografie nebo přidávat audio-video nahrávky. Navíc lze každému fotografickému nebo audio-video souboru změnit textový popis.

Aby uživatel bylo modifikování a mazání umožněno, musí být autorem tohoto obsahu. Když zapomene heslo do systému, poslední možností mu je funkce obnovení hesla. Díky generalizaci v diagramu případu užití pak běžný uživatel dědí možnosti chování návštěvníka.

Správce webu

Správce webu je uživatel systému s nejvíce oprávněními. Jde v podstatě o administrátora zodpovědného za zobrazení korektních informací pro budoucí návštěvníky webu. Dědí možnosti chování běžného uživatele.

Jeho pravomoci jsou rozšířeny o schvalování článku nebo životopisu osob, prohlížení seznamu registrovaných uživatelů, zobrazení článků či životopisů vytvořených konkrétním uživatelem a smazání konkrétního uživatele. Nakonec je mu umožněno udělit běžnému uživateli pravomoci správce, nebo snížit roli správce na roli běžného uživatele.



Obrázek 4.1: Diagram případů užití. Vytvořeno v [6].

4.3 Návrh uživatelského rozhraní

K návrhu uživatelského rozhraní je vhodné použít takzvaný WireFrame (česky drátěný model) znázorňuje základní strukturu webové stránky. Zobrazuje rozmístění statických a funkčních prvků na stránce bez grafických prvků pouze černobíle.

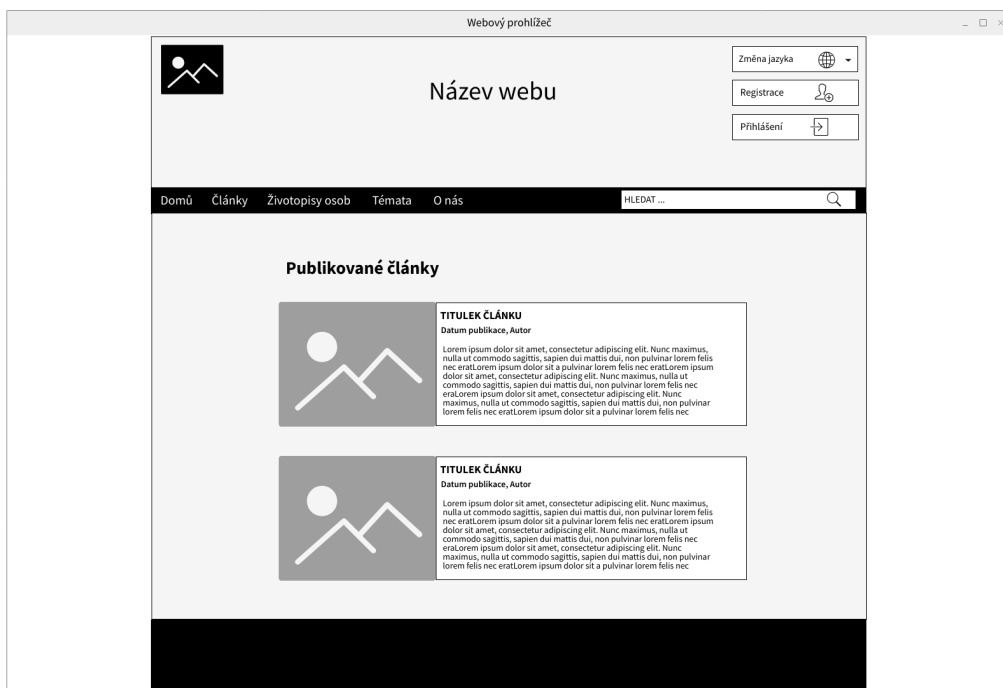
Návrh předchází samotnou grafickou a programátorskou práci a umožňuje představit i ostatním nezainteresovaným lidem, jaký bude mít web rozložení a na jaké stránky mohou při procházení narazit.

Zde jsou představeny návrhy čtyř webových stránek této aplikace:

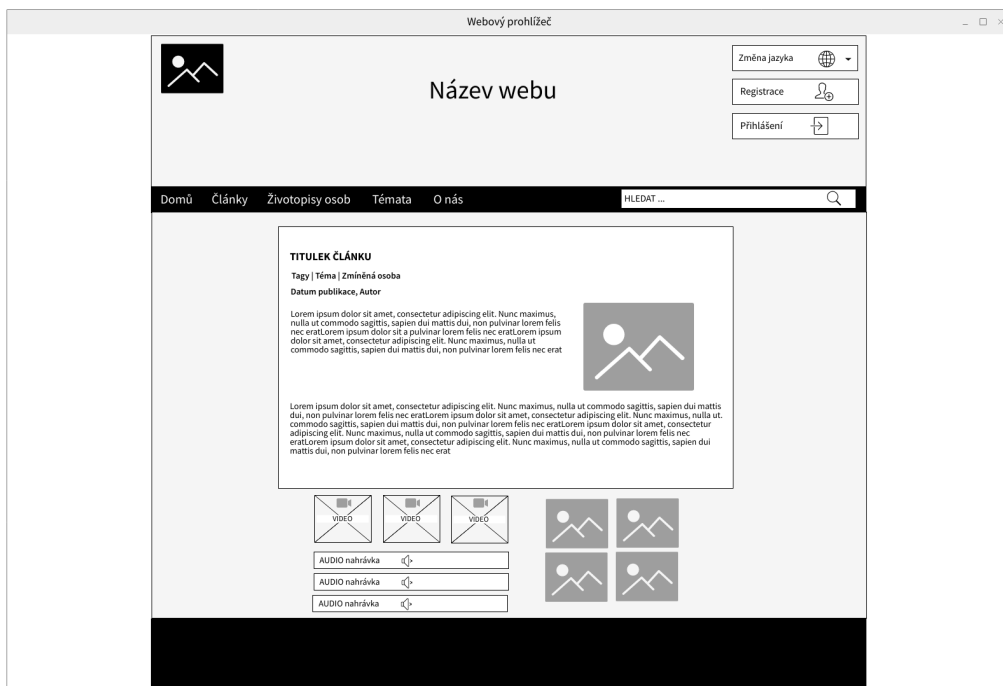
Domovská stránka

První stránkou webu, kterou návštěvník uvidí, je domovská stránka (viz obrázek 4.2). Zde budou v obsahové části zobrazovány aktuálně nejnovější publikované články řazené podle data publikace. Tady využijeme stručný popis článku. V hlavičce stránky se nachází logo webu, jeho název a možnost výběru jazyka. Pod výběrem jazyka se nachází dvě důležitá tlačítka pro vstup do informačního systému, prvním je registrace nového uživatele a druhým je přihlášení do již existujícího uživatelského účtu.

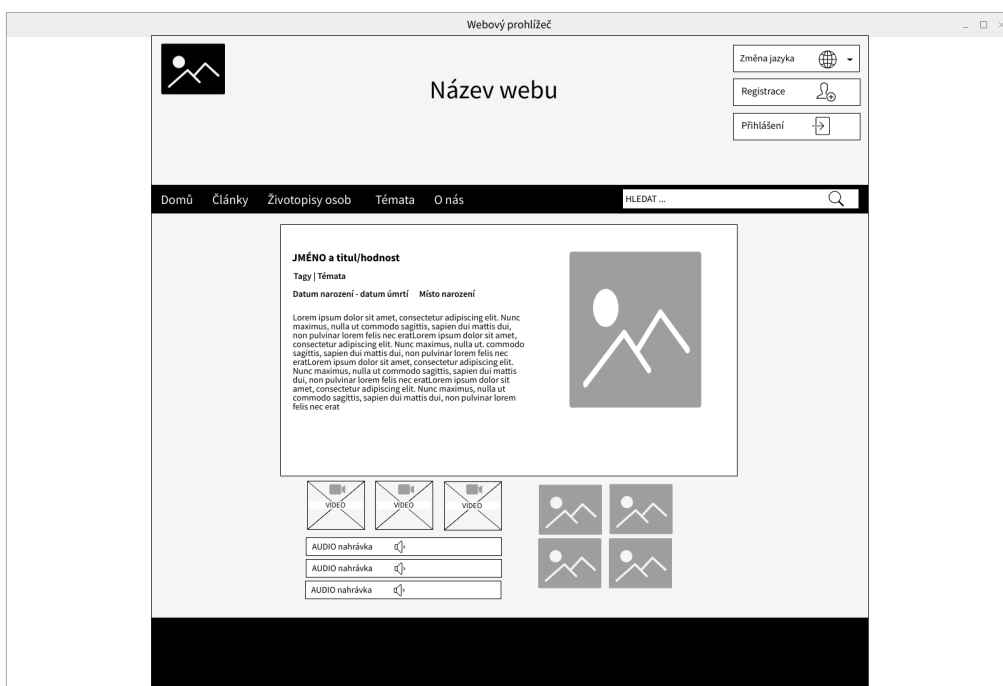
Dále bude možné nastavit hlavičce barevné pozadí či webovou ilustraci. Pod ní následuje hlavní nabídka, nastíní obsahovou strukturu webu a jednotlivé odkazy zajistí průchod do jiné požadované části. Vyhledávat můžeme přes vyhledávací pole taktéž umístěné v nabídce. Rozložení bude pro každé další zobrazení stejné, měnit se bude pouze obsah pod hlavní nabídkou.



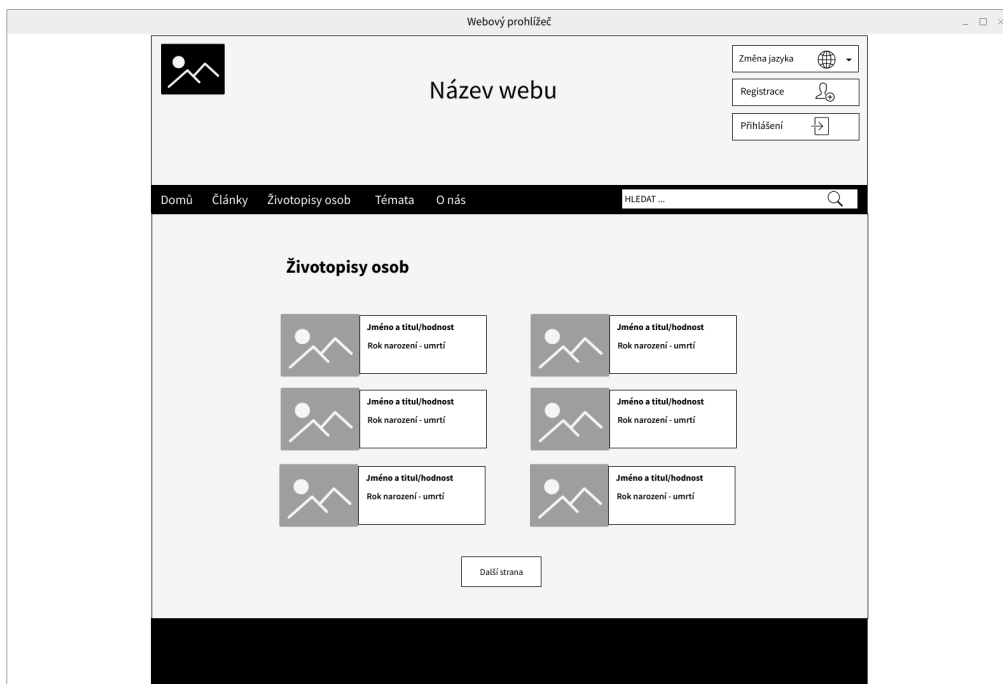
Obrázek 4.2: Návrh domovské obrazovky webu. Vytvořeno v [1].



Obrázek 4.3: Návrh zobrazení článku. Vytvořeno v [1].



Obrázek 4.4: Návrh zobrazení životopisu osoby. Vytvořeno v [1].



Obrázek 4.5: Návrh zobrazení seznamu osob. Vytvořeno v [1].

Zobrazení článku

Každý článek bude zobrazován jednotlivě s rozmístěním prvků, jak můžeme vidět v obrázku 4.3. Na začátku jsou rozmístěny informace o názvu článku, kdo článek napsal, datum publikace, vypsání osob zmíněných v obsahové části a náležící tagy nebo témata/kategorie. Následuje stručný popis článku, vedle nějž je umístěna ilustrační fotografie.

Pod tímto úvodem je místo pro hlavní obsah článku. Na konec stránky se přidají případné fotografie a audio-video nahrávky pro přehrání, které článku náleží. Pod mediální částí přibude formulář pro zadání komentáře k článku.

Zobrazení životopisu osoby

Životopis ponese jméno dotyčné osoby a hned vedle svůj vlastní portrét. Dalšími informacemi, kterými bude životopis disponovat, jsou rok a místo narození, dosažený titul a náležící tagy nebo témata/kategorie. Hlavní text je umístěn pod těmito informacemi.

Pod obsahovou částí se budou nacházet fotografie z vlastní sbírky nebo fotky se zachycenou osobou stejně jako u článku. Pokud budou k dispozici i audio-video materiály, budou zde také přiloženy (viz obrázek 4.4).

Seznam osob

Pro zobrazení seznamu životopisů osob bude nutné kliknout na položku hlavní nabídky s názvem životopisy osob, nebo se k němu dostat skrze další filtrační položky menu Témata. Pokud si návštěvník ani tak nebude vědět rady, další možností je zadat hledaného přímo do vyhledávacího pole hlavní nabídky.

Obsahová část vykreslí seznam životopisů řazených podle příjmení jako dvě položky vedle sebe, jak můžeme vidět na obrázku 4.5. Pro zobrazení dalších položek, které se nevezly na stranu, slouží tlačítko Další strana.

4.4 ER diagram

ER diagram [29] (Entity Relationship Diagram) je metoda datového modelování znázorňující abstraktní a konceptuální schéma databáze. Je postaven na množině základních objektů. Těmi jsou entita (Entity) a vztahy (Relationship) mezi nimi. Entity jsou v relačních databázích tabulky tvořené atributy neboli vlastnostmi entit. Tabulky se mezi sebou propojují vztahy a každý takový vztah má kardinalitu. Ta určuje maximální a minimální množství vztahů, na kterých se může entita podílet.

V diagramu byla pro kardinalitu využita notace (symboly na čárách) jazyka UML (Unified Modeling Language) s názvem Crow's Foot. Diagram vycházející z předchozích návrhů vytvořený pro tuto aplikaci můžeme vidět na obrázku 4.6.

Při návrhu tak vzniklo deset tabulek, každá s vlastním unikátním klíčem a seznamem atributů. Dvě z těchto entit jsou ty nejpodstatnější: Článek a Osoba. Představují hlavní prezentační komponenty webu. Obě tyto tabulky jsou propojeny s tabulkami Téma/kategorie, Tag, Záznamové médium a Uživatel. Se všemi kromě tabulky Uživatel mají vztah M:N, který je následně z důvodu implementace rozdělen na dva vztahy 1:N a mezi ně je vložena spojovací tabulka nesoucí dva cizí klíče odkazující na primární klíče první a druhé tabulky z původního vztahu. Kombinací těchto dvou cizích klíčů tak vznikne jeden primární klíč. Dále jsou zde tabulky Komentář, Typ, Role a Zapomenuté heslo.

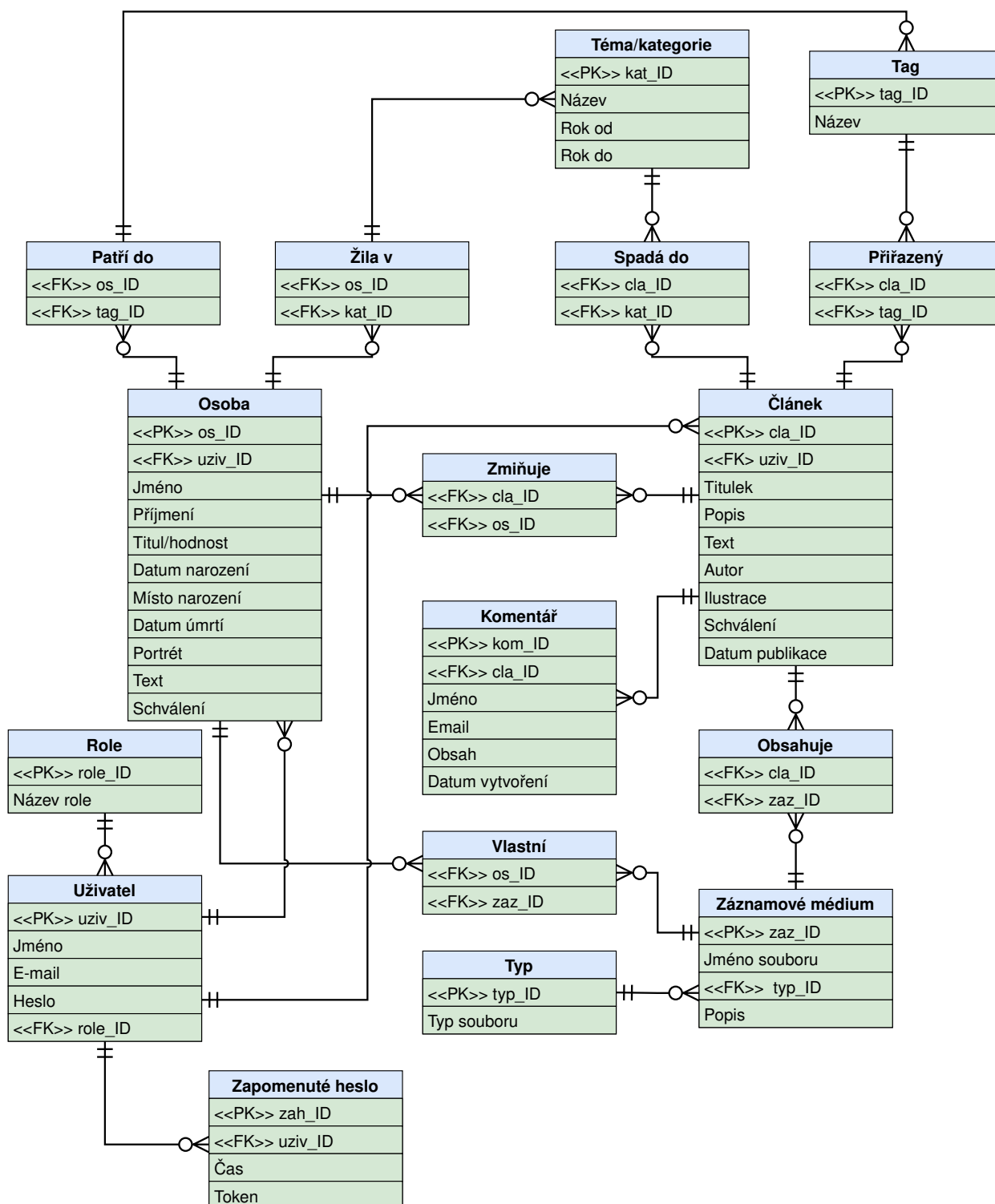
Tabulka Článek ukládá informace potřebné pro zobrazení článku publikovaném na webové stránce. Každému návštěvníkovi je umožněno publikovat komentář k článku pro jehož informace slouží tabulka Komentář. Tabulka Osoba ponese informace pro životopis. Tyto dvě tabulky spolu dále souvisí, protože článek se může zmiňovat o několika osobách a osoba může být v několika dalších článcích, tím vznikla tabulka Zmiňuje.

Tabulka Osoba ukládající informace o historických osobách ponese, jak již bylo dříve zmíněno, základní osobní údaje a texty životopisu potřebných k zobrazení. Propojením článku a životopisu s tabulkou Záznamové médium získáme přehled, jaké médium je potřeba na stránce u článku nebo osoby zobrazit. Atribut typ souboru je cizí klíč nesoucí jednoduše číselnou informaci 1 pro fotografii, 2 pro video a 3 pro zvukovou nahrávku, toto rozlišení mezi typy souboru bude uloženo v tabulce Typ. Dalšími atributy jsou jméno souboru (cesta k němu) a popis obsahu média.

Tabulka Téma/kategorie obsahuje název určitého tématu nebo kategorie upřesňující časový úsek, do kterého článek nebo osoba spadá. Pro jasné řazení mezi sebou dále obsahuje rok začátku a konce období. Tabulka Tag nabídne uložení specifických názvů, pod kterými je možné osoby hledat například zařazením některých osob mezi partyzány, disidenty a další.

Dále bude možné se jako super správce, správce nebo běžný uživatel, toto rozlišuje tabulka Role, přihlásit pomocí údajů uložených v tabulce Správa a vytvářet nebo modifikovat tak obsah webu. Pro přihlášení se využije uživatelské jméno a heslo. Pro obnovu hesla zase email zadaný při registraci. Tabulka Zapomenuté heslo slouží pro uživatele, který zapomněl heslo a vyžádal si jeho obnovení. Nese údaj o uživateli, který o obnovu požádal, záznam data a času požádání a speciální token, což je vygenerovaný řetězec znaků pomocí takzvané

hashovací funkce z uživatelského emailu. Tato funkce pracuje na principu matematické funkce, kdy přijatý vstup mění podle parametrů na jiný výstup.



Obrázek 4.6: ER diagram, kde «PK» je primární klíč a «FK» je cizí klíč. Vytvořeno v [6].

Kapitola 5

Implementace

Kapitola rozebere postupné kroky pro splnění praktické části této bakalářské práce. Řídí se specifikací požadavků a návrhem z kapitoly 4. Pro začátek je v části 5.1 nutné zjistit, jakým způsobem framework Nette funguje.

Druhým krokem je lokální vývoj webové aplikace 5.2, kde jsou popsány aplikace využívané při vývoji a proces propojení všech technologií pro první úspěšné spuštění. Následuje ostré nasazení webu na webový hosting v sekci 5.3.

Poté kapitola blíže vysvětlí jednotlivé funkce webu. Mezi ně patří vytváření uživatelů a jejich přihlášení do systému (viz sekce 5.4), správa uživatelů a hlavních obsahových komponent aplikace v sekci 5.5, zobrazení článků a životopisů (viz sekce 5.6) a jejich možná editace v sekci 5.7. Poslední částí je vyhledávání požadovaného obsahu v sekci 5.8.

5.1 Framework Nette

Jak bylo již dříve zmíněno, pro vývoj a chod této aplikace byl vybrán framework Nette [21]. Zde probereme jeho architekturu a prostředky zjednodušující průběh vývoje webové aplikace.

5.1.1 Architektura MVP

Nette je v podstatě balíček knihoven stavějících na základech jazyku PHP. Proto využívá všech vlastností a to především objektově orientované programování. Architektura, kterou se Nette řídí, se nazývá MVP [34] (Model-view-presenter) fungující na stejném principu architektury MVC (Model-view-controller). Toto schéma můžeme vidět na obrázku 5.1.

Již z názvu vyplývá, že se skládá ze tří základních komponent, které mají ve webové aplikaci rozdílné úkoly. Toto rozdělení umožňuje snadné rozšíření a zvyšuje přehlednost.

Model

Model zajišťuje aplikační logiku webu. Je to část aplikace, která je před uživatelem skryta v pozadí. Patří sem práce s databází, se soubory souborového systému daného počítače nebo různé výpočty potřebné pro chod aplikace. Každá datová entita, kterou lze oddělit, má svůj vlastní model (například článek, životopis, uživatel atd.).

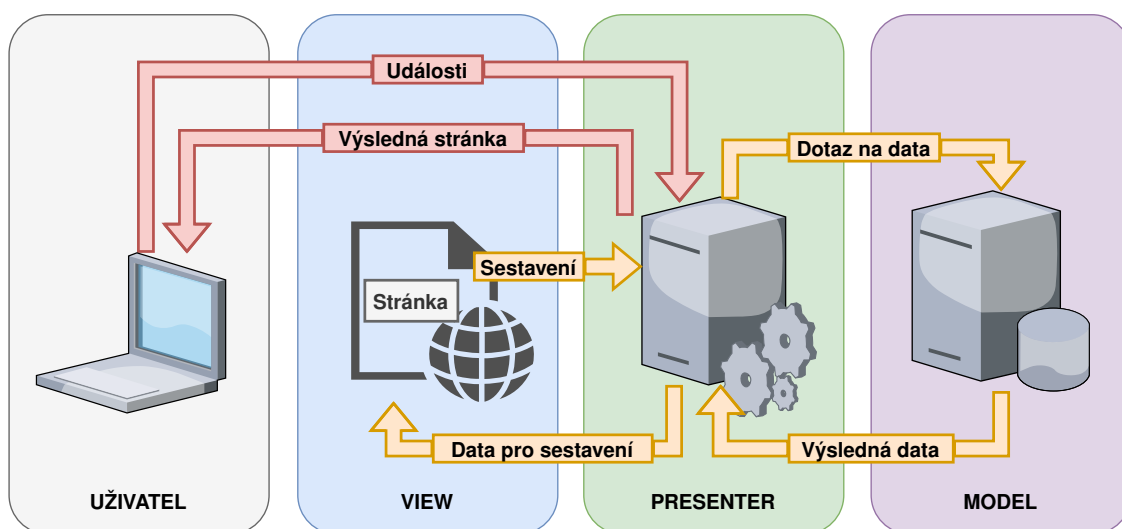
View

View (česky pohled) je v Nette reprezentován pomocí šablon. Tyto šablony jsou výstupem aplikace, uvidí je tedy jakýkoli uživatel webu. Využívají šablonovací systém Latte popsaný níže.

Presenter

Presenter má za úkol řízení komunikace mezi uživatelem dané stránky, za kterou presenter zodpovídá a modelem aplikace. Uživatel mu svým jednáním předává parametry, ty jsou předány modelu, který s nimi provede požadovanou operaci.

Presenter na základě uživatelových parametrů a výsledků modelu předává data pohledům. Pohled poté sestaví výslednou stránku v jazyce HTML s využitím obdržených dat. Následně opět presenter tuto stránku odešle uživateli.



Obrázek 5.1: Schéma architektury MVP. Vytvořeno v [6].

5.1.2 Composer

Composer je nástroj na správu závislostí projektu psaném v jazyce PHP. Je k dispozici ke stažení na adrese¹. Seskupí dohromady libovolně složité závislosti jednotlivých přidaných knihoven, modulů a balíčků. Poté je v případě, že naše stávající knihovny jsou v požadované verzi, nainstaluje do našeho projektu.

Všechny seskupené závislosti našeho projektu a metadata o něm ukládá v souboru `composer.json`, který se nachází v kořenovém adresáři projektu. Dále je možné Composer použít v novém či stávajícím projektu. Tímto způsobem lze nainstalovat i základní kostru² webové aplikace od frameworku Nette a postavit na ní budoucí aplikaci.

Nainstalovat kostru projektu je tedy možné použitím příkazu:
`composer create-project nette/web-project název-projektu`

¹<https://getcomposer.org/download/>

²<https://github.com/nette/web-project>

Pro aktualizaci knihoven se využije příkazu:
`composer update`

5.1.3 Ladící nástroj Tracy

Tracy je velice důležitý ladící nástroj pro vývoj webové aplikace. Umožňuje detekovat chyby v sintaxi a sémantice kódu, zaznamenává časovou a paměťovou náročnost proběhnutých skriptů a nakonec toto vše zobrazuje v reálném čase přehledně v okně webového prohlížeče.

Při lokálním vývoji je tento nástroj automaticky spuštěný a odhalené chyby jsou zobrazovány. V pravém dolním rohu v plovoucí liště se zobrazují stavové informace, například o použitém databázovém dotazu SQL nebo přihlášeném uživateli. Pokud webovou aplikaci nasadíme na webový server, Nette toto automaticky detekuje a vypne Tracy. V tomto případě se všechny vzniklé chyby ukládají do adresáře s názvem `log` a nezobrazují se žádné citlivé informace, které by mohli ohrozit bezpečnost webu. Vývojář určitě ocení, že i toto omezení lze při produkčním nasazení zrušit změnou v kódu.

5.1.4 Šablony Latte

Latte je šablonovací systém pro jazyk PHP. Šetří čas, zpříjemňuje a zpřehledňuje práci. Díky zabezpečení výstupu před zranitelnostmi chrání web před napadením. Má jednoduchou intuitivní syntaxi, která je zapisována do šablon vedle HTML značek a umožňuje vkládat data z presenteru pomocí speciálních značek. Výsledkem je pak sestavená webová stránka. Šablony používají koncovku souboru `".latte"`.

Na obrázku 5.2 je možné vidět jednoduché spojení značek HTML a značek Latte systému. Kód PHP se zde zapisuje do složených závorek nebo jako zkrácené atributy HTML elementu s předponou `n:tag`, kde `tag` je značka Latte. Na obrázku jsou dva bloky, kdy jeden je zobrazován v případě, že je počet životopisů 0, druhý s počtem životopisů větším než 0. Dva stavy jsou rozlišeny pomocí konstrukce `if`, tedy větvení s podmínkou. V každém bloku je informační text a element vytvářející odkaz ve stylu tlačítka. Tlačítko obsahuje dva Latte atributy.

První je `n:href="Edit:createPerson $control->storeRequest()"`. Sestavuje odkaz, po jehož odkliknutí dorazí do presenteru s názvem `Edit` žádost o vytvoření Uživatele a s tím související sestavení šablony `createPerson` se stejným názvem.

Druhý atribut `n:if="$user->isLoggedIn()"` podmiňuje zobrazení celého elementu. V případě, že je uživatel přihlášen, zobrazí element. Pokud není přihlášen, element zůstane skrytý.

```

{if ($size == 0)}
  <div class="container mx-0 mx-md-5 mt-5 mb-5">
    <h2>Nemáte vytvořený žádný životopis</h2>
    <a class="btn btn-success btn-outline-info mt-2"
      n:href="Edit:createPerson {$control->storeRequest()}"
      n:if="{$user->isLoggedIn()}">
      Vytvořit novou osobu
    </a>
  </div>
{else}
  {if (sizeof($persons) != 0)}
    <div class="container mx-0 mx-md-5 mt-5 mb-5">
      <h2>Moje životopisy osob ({$size}):</h2>
      <a class="btn btn-success btn-outline-info mt-2"
        n:href="Edit:createPerson {$control->storeRequest()}"
        n:if="{$user->isLoggedIn()}">
        Vytvořit novou osobu
      </a>
    </div>
  {/if}
{/if}

```

Obrázek 5.2: Snímek obrazovky části šablony Latte této aplikace.

5.2 Lokální vývoj

Tato sekce pojednává o náležitostech potřebných ke správnému spuštění této webové aplikace. Představí lokální webový server a adresářovou strukturu webu. Popíše vývojové prostředí, které bylo při implementaci využito a zmíní způsob propojení aplikace s databází.

5.2.1 WampServer

Prvním krokem pro úspěšné spuštění při lokálním vývoji je vytvořit si prostředí pro běh PHP skriptů. Technologie jazyka HTML, grafických stylů CSS a JavaScript lze jednoduše spouštět pouze s využitím webového prohlížeče. Toto u kódu jazyka PHP neplatí. Je zapotřebí si vytvořit lokální server PHP, ale nejprve je nutné samotné PHP instalovat na náš počítač.

Struktura této webové aplikace je převzata z kostry aplikace frameworku Nette zvaného WebProject, který je dostupný na adrese³. Tato kostra a knihovny frameworku Nette ve verzi 3.1 v ní obsažené vyžadují pro správnou kompatibilitu verzi jazyka PHP 7.4.

Nejjednodušší cestou je instalace softwarového balíčku, který nám vytvoří lokální prostředí webového serveru. Tyto balíčky jsou zpravidla zdarma a velice výhodné. Poskytují nejen server PHP, ale také přehledné uživatelské rozhraní spouštěné v okně webového prohlížeče a k tomu potřebné nástroje pro správu databáze.

³<https://github.com/nette/web-project>

Takovým nástrojem je například phpMyAdmin⁴. Poskytuje vývojáři rozhraní pro snadnou manipulaci s databázovými daty a umožňuje vytvářet, editovat nebo mazat tabulky včetně jejich atributů.

Při vývoji byl využit balíček WampServer dostupný na adrese⁵. Tento balíček se skládá, jak můžeme vidět z nezkráceného názvu Windows-ApacheWebServer-MySQL-PHP, z webového serveru Apache, aplikací pro práci s databází a jazyka PHP. Byla zvolena verze WampServer 3.2.6 s jazyky PHP a MySQL obou ve verzi 8.0.

5.2.2 Adresářová struktura

Adresářová struktura webové aplikace je položena na základech již výše zmíněné kostry od frameworku Nette. Pro potřeby této práce je navíc doplněna o několik dalších souborů a složek. Tato podsekcce vysvětlí význam jednotlivých složek a souborů zde uvedených.

Zde můžeme vidět podobu struktury této webové aplikace.

```
webova_aplikace/
├── app/
│   ├── Models/
│   ├── Presenters/
│   │   ├── HomepagePresenter.php
│   │   └── templates/
│   │       ├── @layout.latte
│   │       └── Homepage/
│   │           └── default.latte
│   ├── Router/
│   │   └── RouterFactory.php
│   └── Bootstrap.php
├── config/
├── db/
├── log/
├── temp/
│   ├── cache/
│   └── upload_images/
├── vendor/
│   └── autoload.php
├── www/
│   ├── assets/
│   ├── index.php
│   └── .htaccess
└── .htaccess
```

Každá složka tohoto kořenového adresáře kromě `www/` vlastní soubor `.htaccess`, jenž zabírá vstup do těchto složek a brání tak návštěvníkům webu v prohlížení jejich obsahu. Dále je v kořenovém adresáři umístěn další soubor `.htaccess`, který přeměrovává zobrazování webu do složky `www/`, kde se nachází soubor `index.php`, jenž je vstupním bodem k programové aplikaci.

⁴<https://www.phpmyadmin.net/>

⁵<https://wampserver.aviatechno.net/>

Mimo jiné jsou tu soubory `composer.json` a `composer.lock`, které uchovávají informace o závislostech obsažených knihoven a metadata o této aplikaci.

app/

Adresář `app/` obsahuje všechny zdrojové texty v jazyce PHP. Soubor `Bootstrap.php` je zaváděcí třídou aplikace. Zde se spouští ladící nástroj Tracy, nastavuje časová zóna a zavádějí konfigurační soubory. Adresář je dále členěn do podsložek `Models/` – aplikační logika, `Presenters/` – presentery a šablony, `Roter/` – zajišťuje správné přesměrování uživatelských dotazů.

config/

Tato složka obsahuje konfigurační soubory, jejichž změna může velice změnit chování aplikace. Jsou zde údaje potřebné pro přihlášení k databázi. Je to místo pro napojení potřebných rozšíření, které v aplikaci využíváme.

db/

Ve složce `db/` se nachází soubor `presentation_web_db.sql`, jenž slouží pro inicializaci databáze tohoto webu.

log/

Zde webová aplikace soustřeďuje soubory s příponou `.log` nesoucí záznamy o vzniklých chybách. Toto probíhá pouze za předpokladu, že je web v produkčním režimu.

temp/

Složka `temp/` je místem pro uložení dočasných souborů. Podsložka `upload_images/` slouží aplikaci pro dočasné nahrávání souborů, které chceme přidat k článku nebo životopisu osob. Složka `cache/` ukládá například konfigurační nastavení, data navštívených stránek a výsledky databázových dotazů. Tímto způsobem Nette zrychluje aplikaci a šetří její výpočetní náročnost.

vendor/

Tady jsou uloženy veškeré knihovny a komponenty frameworku Nette. Pokud si pomocí Composeru přidáme další rozšíření, bude uloženo zde. Soubor `autoload.php` načítá všechny nainstalované balíčky.

www/

Je jediný veřejně přístupný adresář aplikace. Obsahuje soubor `index.php`, kterým se aplikace spouští. Dále je zde ikona aplikace `favicon.ico` a podsložka `assets/`. Ta obsahuje veškeré styly CSS graficky upravující celý web, veškeré kódy JavaScript oživující chování elementů HTML aplikace a webové obrázky. V případě, že k publikovaným článkům a životopisům nahrajeme soubory, potom zde budou ukládány ve složce `files/`.

5.2.3 Databáze

Aby mohla na lokálním serveru spuštěná aplikace fungovat správně, je nutné ji propojit s databází. Propojení se vykonává jednoduchým způsobem v konfiguračním souboru `local.neon`, jak je ukázáno na výpisu 5.1. Nastavení lze provést i přímo v kódu aplikace, ale takto je to nejjednodušší.

Zde se vyplní IP adresa nebo textové jméno hostitelského zařízení, port naslouchající aplikace a název požadované databáze. Dále je nutné zadat přihlašovací údaje jako je uživatelské jméno a heslo. U lokálního vývoje přes WampServer se udává adresa localhostu, tedy IP adresa 127.0.0.1 pro využití místními aplikacemi našeho počítače.

V aplikaci se pak k databázi přistupuje přes objekt třídy `Nette\Database\Explorer`. Ve všech souborech končících na `Model.php` se tento objekt ukládá do privátní proměnné `$database` a přes ni se provádí dotazy SQL. Na výpisu 5.2 můžeme vidět sestavení dotazu SQL pro smazání uživatele s číslem uloženým v proměnné `$userID`.

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

Výpis 5.1: Výšeč souboru `local.neon` – napojení databáze.

```
$this->database->table('user')
    ->where('id', $userID)
    ->delete();
```

Výpis 5.2: Ukázka sestavení dotazu SQL pro smazání uživatele.

5.2.4 Vývojové prostředí

Pro efektivnější práci při vývoji je důležité vybrat vývojové prostředí. Programátorovi zjednodušuje a zrychluje psaní kódu aplikace. Zpravidla obsahuje editor zdrojových textů, kompilátor nebo interpret a ladící nástroj pro hledání a řešení chyb. Umí nejen obarvovat syntaxi kódu, ale i napovídat konstrukce, hlídat chyby, automaticky zarovnávat kód a další.

U této aplikace bylo využito vývojové prostředí NetBeans dostupné zdarma ke stažení na této adrese⁶. Největší výhodou je integrovaná podpora jazyka PHP a s tím souvisejícího frameworku Nette společně s komponentami Latte a NEON, které využívá. Proto jej také Nette na svých stránkách doporučuje.

5.3 Nasazení aplikace

Tato sekce popíše, jakým způsobem byla výsledná webová aplikace nasazena na webový server, aby byla pro kohokoli dostupná přes internet. Jak bylo řečeno v sekci Lokální vývoj 6.2, webová aplikace potřebuje prostředí pro svůj běh. Na našem počítači si vše pro toto

⁶<https://netbeans.apache.org/>

prostředí musíme zařídit sami, ale pokud chceme poskytnout přístup na náš web i dalším uživatelům, není nic jednoduššího, než si zvolit poskytovatele veřejného webového prostoru (anglicky zvaný webhosting).

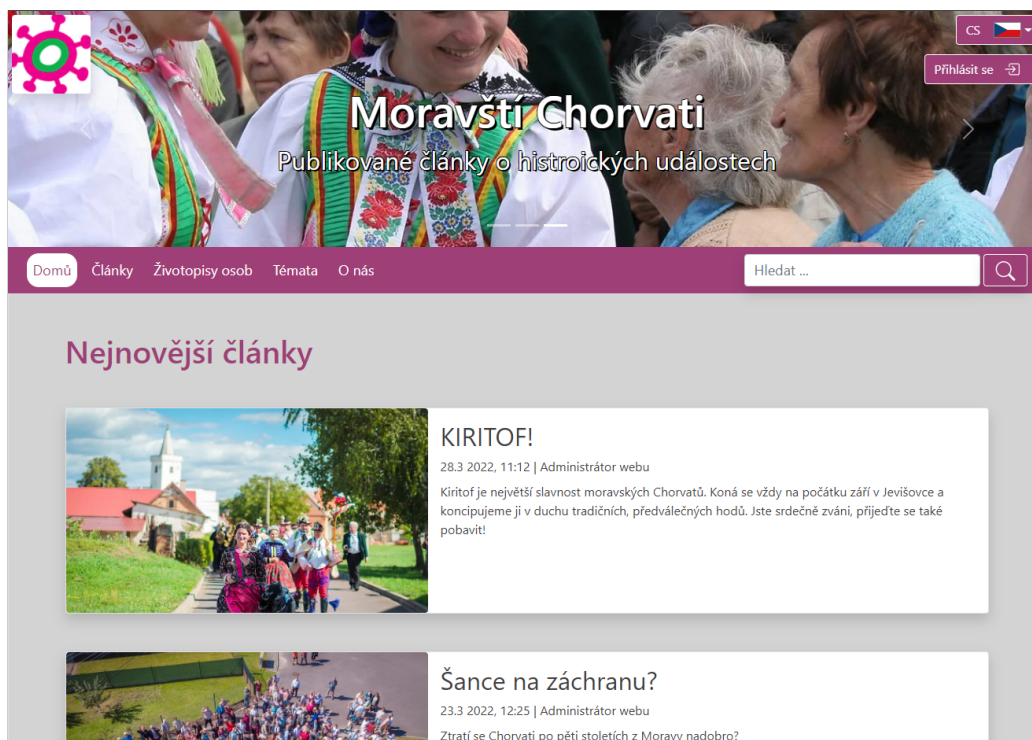
Tento poskytovatel nabízí propůjčení svého webového prostoru. Poskytovatelé se mohou lišit různými parametry. Některé webhostingy jsou zdarma, jiné zpoplatněny nebo nabízí určité funkce a nástroje za určitý poplatek.

Pro aplikaci byl vybrán poskytovatel Webzdarma⁷. Kvůli zprovoznění webu je nutné vybrat si název domény, pod kterou bude aplikace dostupné skrze internetové vyhledávání. Byla zvolena varianta s doménou třetího řádu, která není zpoplatněna.

Výhodou tohoto webhostingu je možnost využití vysoké verze jazyků PHP a SQL u obou ve verzi 8.0. Nevýhodou je zobrazení reklam ve vyhrazeném prostoru pod patičkou webové stránky. Poté už je potřeba pouze importovat databázi s daty a nahrát celou adresářovou strukturu aplikace do nám přiděleného prostoru. K tomuto úkonu je možné využít klienta FTP (FTP – protokol pro přenos souborů), který je většinou součástí každého systému.

Pro tuto práci bylo možné využít textů a obrázků z webu Moravští Chorvati⁸ se souhlasem paní Mgr. Lenky Kopřivové, místopředsedkyně Sdružení občanů chorvatské národnosti v ČR. Tato data byla využita pro naplnění obsahu webu.

Webová aplikace je po vložení na webhosting a propojení s databází ihned dostupná na adrese⁹ uvedené v poznámce pod čarou. Vzhled úvodní domovské stránky aplikace můžeme vidět na obrázku 5.3.



Obrázek 5.3: Snímek obrazovky domovské stránky aplikace.

⁷<https://www.webzdarma.cz/>

⁸<http://www.moravstichorvati.cz/>

⁹<http://prezentsationsite.euweb.cz/>

5.4 Uživatelé

Důležitou součástí prezentační aplikace je uživatel, který vytváří její obsah. Tato část vysvětlí, jakým způsobem dochází k tvorbě uživatelského účtu, jak se ověřuje již existující uživatel a jak proběhne obnova hesla, pokud dojde k jeho ztrátě.

5.4.1 Registrace

Jestliže nemáme přístup do informačního systému webu, je nutné provést registraci a vytvořit tak uživatelský účet. Tlačítko `Registrovat se` bylo odebráno, kvůli zjednodušení. K registračnímu formuláři se tedy dostaneme přes stránku s přihlášením. Pro vytvoření účtu, musí návštěvník zadat jméno, email a dvakrát stejné heslo. Jméno a heslo potom budou sloužit jako přihlašovací údaje pro vstup do systému.

Formulář se na této stránce vytváří metodou presenteru `createComponentRegisterInForm()` v souboru `RegistrationPresenter.php`. Formulářové prvky se vytváří jednoduchým způsobem, jak můžeme vidět na výpisu 5.3. Nette zde umožňuje zadání různých hlášení, které budou vidět v různých případech. Kliknutím na tlačítko `Registrovat se` se provede nejprve kontrola v prohlížeči, jestli jsou všechna pole vyplněná a hesla mají minimální délku, poté dojde k odeslání dat do presenteru. Tam metoda pouze předá obdržená data podobně nazvané metodě `registerInSucceeded($thisP, $form, $data)` nacházející se v modelu `LoginModel.php`.

V metodě probíhá kontrola na rovnost obou hesel a na jedinečnost jména a emailu zadaného uživatelem. Pokud dojde k chybě, zobrazí se chybové hlášení a uživatel bude muset provést registraci opět znovu. Když je vše správně, uloží se údaje do databáze. Nejdříve je ale potřeba zašifrovat heslo tak, aby se v databázi nevyskytovalo v lehce čitelné textové podobě. To se provádí metodou `hash($heslo)` objektu třídy `Nette\Security\Passwords` prostřednictvím algoritmu BCrypt. Vzhled registrace je na obrázku 5.4.

```
$form->addText('username', 'Jméno:')
->setHtmlAttribute('placeholder', 'Nové uživatelské jméno')
->setRequired('Prosím zadejte své nové uživatelské jméno.')
->setAttribute('class', 'my-input');

$form->addText('email', 'Email:')
->setHtmlAttribute('placeholder', 'Nový uživatelský email')
->addRule(Form::EMAIL, 'Email není zadaný správně.')
->setRequired('Prosím zadejte svůj email.')
->setAttribute('class', 'my-input');

$form->addPassword('password', 'Heslo:')
->addRule($form::MIN_LENGTH, 'Heslo musí mít alespoň %d znaků', 8)
->setHtmlAttribute('placeholder', 'Nové uživatelské heslo')
->setRequired('Prosím vyplňte své nové heslo.')
->setAttribute('class', 'my-input');
```

Výpis 5.3: Ukázka přidání kolonek do formuláře spolu s nastavením jejich parametrů. Jsou to kolonky pro jméno, email a heslo.

The image shows a registration page with a purple header containing navigation links: Domů, Články, Životopisy osob, Témata, and O nás. The main content area has a light gray background. At the top, the word "Registrace" is written in a large, bold, purple font. Below it, the text "Vytvoření nového uživatelského účtu" is displayed, followed by a blue link "Přihlásit se již k existujícímu účtu". The registration form consists of four input fields, each with a label on the left and a placeholder text inside the field:

- Jméno:** Input field with placeholder "Nové uživatelské jméno".
- Email:** Input field with placeholder "Nový uživatelský email".
- Heslo:** Input field with placeholder "Nové uživatelské heslo".
- Heslo znovu:** Input field with placeholder "Znovu uživatelské heslo".

Below the input fields is a purple button with the text "Registrovat se".

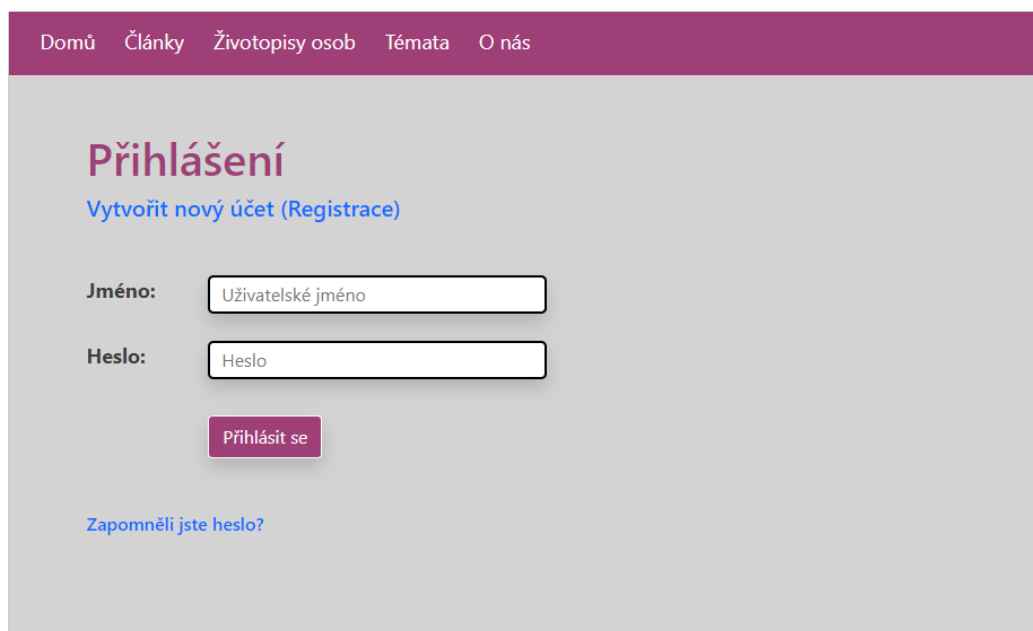
Obrázek 5.4: Snímek obrazovky stránky pro registraci.

5.4.2 Přihlášení

Přihlášení je vstupem do informačního systému webové aplikace. Dostaneme se na něj tlačítkem **Přihlásit se** nacházejícího se v hlavičce stránky. Zobrazí se nám formulář, který je vykreslen metodou `createComponentSignInForm()` v presenteru `SignInPresenter.php`.

Zde uživatel využije dříve zadané údaje jako uživatelské jméno a heslo. Po odeslání zadaných údajů se kontroluje jejich správnost. To probíhá v metodě `signInSucceeded($thisP, $form, $data)` modelu `LoginModel.php`. Uživatelské jméno musí být nalezeno v databázi, stejně tak heslo se musí rovnat heslu uloženého u tohoto jména. K ověření se využívá metoda `verify($data->password, $row->password)` objektu třídy `Nette\Security>Passwords`. První parametr je zadané heslo a druhý je zašifrované heslo z databáze.

Chybně zadané údaje oznamuje chybové hlášení. Pokud jsou údaje správné, je uživatel přihlášen a vznikne takzvaná identita. Na začátek se nastaví doba expirace přihlášeného účtu metodou `setExpiration('30 minutes')`, po které je uživatel odhlášen. Poté se vytvoří identita jako objekt třídy `Nette\Security\SimpleIdentity`, která se předá metodě `login(new SimpleIdentity($id, $role))`. Identita se udržuje v session a předává se během procházení webu. O uživateli nese údaj jako role v systému. Díky tomu se mohou vykonávat operace vyžadující vyšší oprávnění uživatele. Informaci o roli uživatele můžeme získat metodou `getIdentity()->role`. Na obrázku 5.5 je zachycen vzhled stránky pro přihlášení.



Obrázek 5.5: Snímek obrazovky stránky pro přihlášení.

5.4.3 Obnova hesla

Uživateli s existujícím účtem se může stát, že zapomene heslo. Aby neztratil úplný přístup do systému a především svůj vytvořený obsah, existuje proto možnost obnovení hesla. Tuto funkci najdeme u přihlašování jako odkaz [Zapomněli jste heslo?](#), jak je vidět na obrázku 5.5 výše.

Po kliknutí na odkaz přesměruje na stránku s formulářem pro změnu hesla, kde je možné zadat pouze email uživatele. O formulář se stará metoda v presenteru `SignPresenter.php`, ten při kliknutí obdrží zadaný email. Předá ho metodě `forgotPasswordSucceeded()` v modelu `LoginModel.php`.

Pokud zadaný email existuje v databázi, je email zašifrován metodou `hash($email)` jako u registrace. Zašifrovaný email slouží jako jednoznačný identifikátor zvaný token, který bude sloužit pro změnu hesla. Poté se token, id uživatele a aktuální čas uloží do tabulky `Zapomenuté heslo`. Nyní přijde na řadu vygenerování a odeslání emailové zprávy.

Struktura zprávy se řídí šablonou v souboru `email.latte`. Do ní se vloží odkaz na stránku pro změnu hesla společně s tokenem. Je vytvořen email se všemi náležitostmi jako objekt třídy `Nette\Mail\Message`. Pro odeslání se použije objekt třídy `Nette\Mail\SmtMailer`. Je potřeba zadat adresu poštovního serveru, přihlašovací jméno a heslo. Pro účely této aplikace vznikl email `moravsti.chorvati.prezentacni.web@email.cz`.

Aplikace pak už konečně odešle zprávu z adresy výše na emailovou adresu uživatele. Kliknutím na odkaz v emailu se zobrazí stránka pro změnu hesla. Zadáním údajů do formuláře se vykoná validace. Pokud není token z url adresy uložený v tabulce `Zapomenuté heslo`, nebo je uplynutá doba od vyžádání obnovy delší jak 1 hodina, odkaz se ukáže jako neplatný.

5.5 Tvorba webového obsahu

V této webové aplikaci lze vytvářet dva typy obsahu a to články nebo životopisy. Obě komponenty se vytvářejí a zobrazují podobně. Největší rozdíl mezi nimi je v ukládaných informacích, které obsahují. Tato část popíše vytvoření článku.

Článek mohou vytvářet pouze uživatelé registrovaní do informačního systému. Ke stránce pro vytvoření článku vedou dvě cesty. První je tlačítko **Vytvořit nový článek** na stránce pro výpis článků a druhou možností je to stejné tlačítko v administrativní části **Můj účet** v sekci **Moje publikované články**. Tyto tlačítka vedou na stránku s formulářem pro článek. Formulář obsahuje kolonky pro titulěk, autora, popis, obsah, soubory, témata, tagy a zmíněné osoby. Poslední tři zmíněné údaje se získávají klikáním na již zobrazené hodnoty, proto mohou nabývat vícero možností. Část prvků formuláře můžeme vidět na obrázku 5.6.

Úplně stejný formulář se využívá i k editaci článku, pouze jeho kolonky jsou doplněny o data načítaná z databáze. Za vytváření, editaci a mazání obsahu zodpovídá `EditPresenter.php` a jeho metody pak předávají data do modelu `EditModel.php`. Metoda presenteru `createComponentArticleForm()` vytvoří formulář a ten se vykreslí v šabloně `createArticle.latte` pomocí konstrukce `{control articleForm}`.

Po kliknutí na tlačítko **Uložit a publikovat** odešleme data ke zpracování. Data zpracovává metoda `updateArticle($thisP, $form, $data)` v modelu `EditModel.php`. Ta slouží zároveň pro vytvoření i editaci článku. Pokud nebylo zadáno id článku, bude se vytvářet nový článek. Všechna data se uloží do příslušejících tabulek a provede se propojení mezi nimi. V případě, že má uživatel pouze roli běžný uživatel, nastaví se schválení článku na neschválený a dojde k jeho zobrazení. Když je článek neschválený, neuvidí jej nepřihlášený uživatel a ostatní s rolí běžný uživatel.

Součástí formuláře jsou dvě důležité komponenty:

Textový editor

Pro jednoduché formátování obsahu publikovaného článku byla aplikace doplněna o textový editor CKEditor 5 [4]. Umožňuje libovolně graficky formátovat text, který je potom převeden do jazyka HTML a uložen v databázi. Nabízí například zvolení úrovně nadpisu, typ písma, zarovnání textu, tučnost textu, odrážky, odkazy a speciální symboly. Změnou jazyka v hlavičce stránky se změní jazyk popisků u editoru. Editor se inicializuje souborem `initEditor.js` a vykreslí se na elementu HTML s id `editor`.

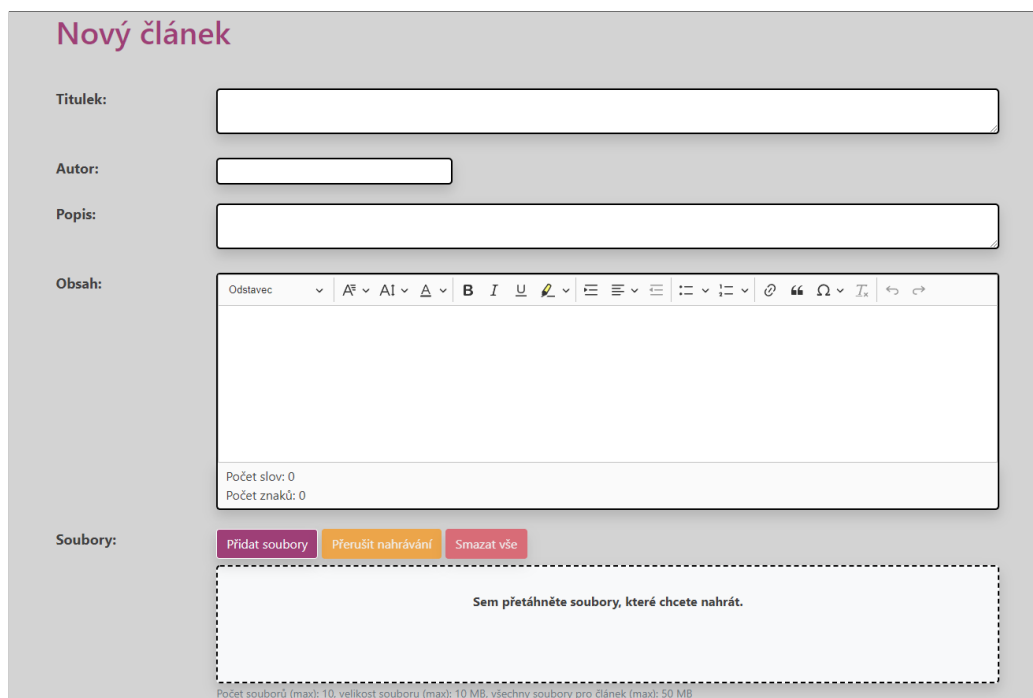
Nahrávač souborů

Další důležitou součástí je nahrání souborů k článku. Proto byla aplikace rozšířena o nahrávač souborů `nepada/file-upload-control` [20]. Důvody výběru tohoto nahrávače byly jednoduchost použití, nahrávání více souborů najednou a drag & drop (táhni a pusť) technologie. Toto rozšíření se zaregistruje v konfiguračním souboru `common.neon`. Do formuláře se jednoduše přidá takto `$form->addFileUpload('files', 'Soubory:')`. Nahrávání má několik omezení. Lze nahrát pouze soubory typu obrázků, video a audio. Dále je omezena velikost na jeden soubor v MB a celkový počet souborů, které lze nahrát najednou. Omezení se dají upravit v konfiguračním souboru `common.neon` v části parametry.

Při nahrávání se soubory ukládají na dočasné místo ve složce `upload_images/`. Když odeslaná data projdou kontrolou a vytváří se článek, provede se metoda

`insertFiles($table, $recordID, $files)` v modelu `FileModel.php`. Tato metoda v případě, že jsou soubory v pořádku, přesouvá soubory z dočasné složky do složky `files/` a ukládá záznamy o nich do databáze.

Soubory jsou ve složce `files/` řazeny podle typu tabulky, identifikátoru záznamu a typu souboru. Když nahrajeme soubor `obrazek.png` k článku s id 40, uloží se tedy v adresářové struktuře jako `files/articles/40/image/obrazek.png`. Podobná cesta se ukládá i do databáze.



The screenshot shows a web form titled "Nový článek" (New article). It contains the following elements:

- Titulek:** A text input field.
- Autor:** A text input field.
- Popis:** A text input field.
- Obsah:** A rich text editor with a toolbar containing options like bold, italic, underline, link, and list. Below the editor, it shows "Počet slov: 0" and "Počet znaků: 0".
- Soubory:** A section with three buttons: "Přidat soubory" (Add files), "Přerušit nahrávání" (Cancel upload), and "Smazat vše" (Delete all). Below these buttons is a dashed box with the text "Sem přetáhněte soubory, které chcete nahrát." (Drag and drop files you want to upload here).

At the bottom of the form, there is a small note: "Počet souborů (max): 10, velikost souboru (max): 10 MiB, všechny soubory pro článek (max): 50 MiB".

Obrázek 5.6: Snímek obrazovky stránky pro vytvoření článku.

5.6 Zobrazení webového obsahu

Prezentační web musí umět prezentovat data, která jsou na něm uložena, respektive která ukládá do databáze nebo svého přiděleného souborového adresáře. Tato část bude opět popisovat zobrazení na příkladu článku. Na této aplikaci se o zobrazení článků stará presenter `ArticlePresenter.php`, který má dvě hlavní metody s předponou `render`. Render ve frameworku Nette značí metodu, která před vykreslením šablony zpracuje obdržené parametry a připraví data pro následné vykreslení.

První metoda `renderArticles(int $page = 1)` získá všechny články, které je potřeba zobrazit a stránkuje jejich výpis pomocí proměnné `$page`. Šabloně pak předává pole článků a stranu, na které se nachází. Uživatel se k výpisu dostane pomocí tlačítka **Články** v hlavním menu. Zobrazí se první strana a předem daný počet článků. Když klikne na druhou stranu, zavolá se stejná metoda a zobrazí články odpovídající druhé straně výpisu.

Druhá metoda `renderShow(int $articleID, $backlink)` zobrazuje jeden konkrétní článek předaný proměnnou `$articleID`. Druhý parametr slouží pro návrat na předchozí stranu. Identifikátor článku je předáván metodě `getArticle($thisP, $articleID)` v modelu `ArticleModel.php`. Zde se kontroluje existence článku a připravují se data pro vy-

kreslení v šabloně. Z databáze se vyberou všechny fotografie, audio-video nahrávky, témata, tagy, zmíněné osoby a komentáře patřící článku. Poté aplikace přechází k vykreslení.

Vykreslený článek nabízí kromě svého textového obsahu výpis fotografií a audio-video nahrávek. Fotografie můžeme zvětšit kliknutím na ni a poté si v otevřeném okně procházet další fotografie. Video využívají rozšíření VideoJS [3], které pomáhá s responzivním zobrazením a upravuje jejich vzhled. Všechny soubory obsahují textový popis, který je možné si zobrazit. Pokud je uživatel autorem článku nebo správcem webu, na stránce se zpřístupní editační tlačítka. Bude možné editovat článek a popisky souborů. Pro smazání článku nebo jeho souborů se vyvolá modální okno, kde smazání potvrdíme.

5.7 Administrace

Administrace je přístupná přes tlačítko **Můj účet** pod podmínkou, že je uživatel přihlášený do systému. Na tomto místě nalezneme přehledný výpis článků, životopisů a uživatelů informačního systému. Rozlišují se zde dva typy uživatelů, uživatel s rolí správce a běžný uživatel. Běžný uživatel vidí pouze svoje články a životopisy. Přímo odsud si je může zobrazit, editovat, smazat a vytvořit nové. Články, které nemá schválené jsou zobrazeny červeně.

Správce může navíc vidět i články nebo životopisy ostatních uživatelů, schvalovat je k zobrazení a pracovat s nimi jako autor (viz obrázek 5.7). Dále se mu zpřístupňuje položka **Všichni uživatelé**, kde najde výpis všech uživatelů, jak můžeme vidět na obrázku 5.8. Pokud klikneme na uživatelské jméno, zobrazíme si uživatelovy články nebo životopisy osob. Správce zde může smazat každý účet kromě svého účtu a účtu super správce. Super správce existuje od počátku běhu aplikace, na webu rozšiřuje roli správce pouze tím, že jej nelze smazat nebo mu snížit oprávnění.

O zobrazení stránek se stará `AccountPresenter.php`. Pro smazání uživatele se vykoná metoda `actionDeleteUser(int $userID, $backlink)`. Ta předává informaci, kterého uživatele smazat, podobně zvané metodě modelu `AccountModel.php`. Všechny články a životopisy se před smazáním z databáze převedou na účet uživatele, který smazání provedl.

Když chceme delegovat oprávnění role správce na námi zvoleného běžného uživatele a rozšířit tak tým, který se o web stará, klikneme na tlačítko **Udělit správce**. Zvolený uživatel projde kontrolou, jestli se nejedná o náš účet nebo účet super správce. Pokud ano, není nám umožněno oprávnění měnit. V opačném případě, snížení oprávnění správci na roli běžný uživatel prostřednictvím tlačítka **Odebrat správce**, dojde ke stejné kontrole.

5.8 Vyhledávání

Prvním typem je vyhledávání přes vyhledávací pole umístěné v hlavním menu. Vyhledávání umožňuje najít přesně daný název nebo minimálně jeho část. Aplikace disponuje fulltextovým vyhledáváním výrazů. Pokud tedy zadáme výraz, hledá se tento výraz i v celém textovém obsahu například článku. Pro shodu hledaného výrazu a řetězce v textu postačí, že se výraz v řetězci nachází. Výrazy je možné psát bez diakritiky.

U článků lze hledat podle titulku, autora, popisu a obsahu. Stará se o to `SearchPresenter.php` s metodou `renderSearch($query, int $page = 1)`. Metoda připravuje data pro vykreslení šablony a stránkuje výsledný výpis. Hledaný výraz se předává metodám v modelu `SearchModel.php` a ty vyberou články a životopisy osob, které ho obsahují.

Dalším typem je vyhledat článek nebo životopis přes položku hlavního menu **Témata**. Tady se zobrazí témata seřazená podle roku začátku jejich období. Kliknutím na libovolné téma se vypíšou všechny články a životopisy k němu patřící.

Poslední možnost při hledání nabízí odkazy na téma a tag. Pokud má článek přidáné téma, vypíše se u jeho zobrazení společně s odkazem na toto téma. Po kliknutí se chová jako v předešlém případě.

Domů Články Životopisy osob Témata O nás Hledat ...

Můj účet

Uživatel: **superadmin (Super správce)**

Moje publikované články Moje životopisy osob Všichni uživatelé

Moje publikované články (33):

Vytvořit nový článek

#ID	Titulek	Autor	Úprava	Smazání	Schválení
70	VŠECHNO JE O LIDECH - Chorvatsky, německy a...	admin	Upravit článek	Smazat článek	Schválit článek
69	VŠECHNO JE O LIDECH - Jako Chorvat jsem se...	admin	Upravit článek	Smazat článek	Skrýt článek
68	VŠECHNO JE O LIDECH - Říkali jsme lidem, že...	admin	Upravit článek	Smazat článek	Skrýt článek
63	VŠECHNO JE O LIDECH - Mohli jsme zůstat.	admin	Upravit článek	Smazat článek	Skrýt článek
62	VŠECHNO JE O LIDECH - Češi nám nejvíc...	admin	Upravit článek	Smazat článek	Schválit článek
61	Šance na záchranu?	admin	Upravit článek	Smazat článek	Skrýt článek
60	Zánik chorvatských komunit	admin	Upravit článek	Smazat článek	Skrýt článek

Obrázek 5.7: Snímek obrazovky stránky pro administraci článků.

Domů Články Životopisy osob Témata O nás Hledat ...

Můj účet

Uživatel: **superadmin (Super správce)**

Moje publikované články Moje životopisy osob Všichni uživatelé

Všichni uživatelé (3):

#ID	Jméno	Email	Role	Smazání	Udělit správce
0	superadmin	superadmin@emailserver.cz	Super správce		
7	admin	admin@emailserver.cz	Správce	Smazat uživatele	Odebrat správce
19	user	user@emailserver.cz	Běžný uživatel	Smazat uživatele	Udělit správce

Obrázek 5.8: Snímek obrazovky stránky pro administraci uživatelů.

Kapitola 6

Testování

Tato kapitola nás provede způsobem testování této webové aplikace v sekci 6.1. Popíše, co bylo k testování použito a kdo se na něm podílel. Testování aplikace je zde rozděleno na dvě části. První částí v sekci 6.2 je testování aplikace na uživateli, který není přihlášen do informačního systému.

Popisuje prvky, které byly testovány, jejich váhu a také reakce uživatelů. Stejným způsobem proběhlo testování na uživateli nepřihlášených do webové aplikace v sekci 6.3. To se ovšem zaměřilo na jiné prvky webu.

6.1 Způsob testování

Aplikace byla testována už od začátku vývoje až do poslední fáze implementace, to především programátorem. Nalezené chyby byly okamžitě opraveny a všechny nedostatky doplněny. Ke kontrole vzhledu aplikace na mobilních zařízeních byl využit nástroj DevTools společnosti Google, který je součástí webového prohlížeče Chrome. Finální webová aplikace byla poté představena skupině uživatelů.

Pro testování byla vybrána skupina deseti uživatelů z řad rodiny, ale i studijních kolegů. Na začátku bylo uživatelům dovoleno prohlédnout si prostředí webu, aby se tak v něm mohli alespoň základně orientovat. Každý z nich měl za úkol projít si úkony typické pro tuto aplikaci. Při plnění těchto úkonů bylo zjišťováno, jak na ně webová aplikace působila a jak si s úkony poradili. Hodnocení každého úkonu je známkováno hodnotami 1 až 5 popisující obtížnost jeho provedení.

Obtížnost úkonu:

- 1 – Lehká
- 2 – Mírně těžká
- 3 – Středně těžká
- 4 – Těžká
- 5 – Mimořádně těžká

Testování se rozdělovalo na dvě části a jejich průběh společně s využitými úkony je podrobněji popsán níže.

6.2 Test nepřihlášeného uživatele

Nepřihlášený uživatel je ten, který nemá přístup k administrativním funkcím webu. Tím je tedy každý uživatel, který právě není přihlášen. Uživatel při tomto testu přišel do styku se zobrazením článků, životopisů, témat a také s vyhledáváním.

Zjistil, jaké prvky obsahuje například konkrétní článek. Nakonec si vytvořil účet a přihlásil se do systému. Všech deset uživatelů obdrželo seznam úkonů a dotazník, ve kterém hodnotili jejich obtížnost.

Úkony, které měl uživatel při testování splnit:

1. Přejděte na stránku s životopisy a zobrazte si libovolný životopis osoby.
2. Přejděte na stránku s články a zobrazte si libovolný článek.
3. U zobrazeného článku přidejte komentář.
4. Vyberte si fotografii a zvětšete ji. Pokud je fotografií víc, prohlédněte si je všechny v již otevřeném okně.
5. Zobrazte si u článku video nebo zvukovou nahrávku a pusťte si ji.
6. U fotografie nebo zvukové nahrávky si zobrazte informační popis.
7. Přejděte na stránku s tématy a zobrazte si libovolné téma.
8. Vyhledejte pomocí vyhledávacího pole článek s názvem *Kiritof*.
9. Vytvořte uživatelský účet tím, že provedete registraci.
10. Přihlaste se již vytvořeným uživatelským účtem.

Jakmile uživatelé odevzdali vyplněné dotazníky, bylo možné vytvořit tabulku 6.1 s výslednou průměrnou obtížností každého úkonu. Z výsledků vyplývá, že v této části je aplikace pro uživatele víceméně přehledná a srozumitelná. Problém měli většinou starší uživatelé, když hledali potřebné tlačítko pro vykonání úkonu, to se ale po kratší chvíli vždycky nakonec podařilo najít.

Největší problém nastával u registrace, která je dostupná skrze stránku pro přihlášení, proto nebylo vždy lehké se k ní dostat. Uživatelé v této fázi testování ocenili zpětnou vazbu tlačítek a obrázků, které reagovali na uživatelskou interakci s myší.

Číslo úkonu	Obtížnost
1	1
2	1
3	1.1
4	1
5	1.5
6	1.2
7	1
8	1
9	1.9
10	1.1

Tabulka 6.1: Výsledek testu nepřihlášeného uživatele.

6.3 Test přihlášeného uživatele

U tohoto testování byli uživatelé přihlášení pomocí účtů vytvořených v předchozím testu. Každý z nich dostal přihlašovací údaje správce, nový seznam úkonů a dotazník, ve kterém opět hodnotili obtížnost úkonů.

Prošli si vytvářením, editací a mazáním webového obsahu. Při testu jim bylo umožněno pohybovat se v administrativní části jako správce webu. Díky tomu využili jeho pravomocí například ke smazání uživatele.

Úkony, které měl uživatel při testování splnit:

1. Zobrazte si administrativní část s názvem Můj účet.
2. Zde si zobrazte a prohlédněte Vaše životopisy osob.
3. Vytvořte novou osobu, přidejte k životopisu nějaký libovolný obrázek a publikujte ho.
4. Vytvořený životopis libovolně textově upravte a přidejte k němu navíc video.
5. Přidanému videu upravte textový popis a následně toto video smažte.
6. Smažte Vámi vytvořený životopis.
7. Odhlaste se a znovu přihlaste pomocí přihlašovacích údajů správce.
8. Zobrazte si opět Můj účet a proveďte schválení/skrytí libovolného článku.
9. Tento článek poté smažte.
10. Prohlédněte si výpis uživatelů systému.
11. Zvolte uživatele a proveďte zvýšení/snížení jeho role (oprávnění).
12. Zobrazte si články nebo životopisy osob tohoto uživatele.
13. Proveďte smazání stejného uživatele.
14. Odhlaste se.

Tento test ověřil schopnost uživatelů vyznat se v administrativní části webu. Podle dotazníku vznikla tabulka 6.2 s průměrnou obtížností úkonů hodnocených uživateli. Díky předchozímu testování se uživatelé rychle naučili orientovat na webových stránkách aplikace a mohli tak své znalosti využít u druhé fáze testování.

Výsledky obtížnosti jednotlivých úkonů a zpětná vazba od uživatelů nám poskytly informaci, že bylo dobře navrženo uživatelské rozhraní webové aplikace. Ovládací prvky jsou dostatečně dobře rozmístěny a jejich nalezení nezabere příliš mnoho času.

Všechny objevené chyby byly opraveny a nápady uživatelů využity při úpravě. Některé připomínky pro vylepšení aplikace, které uživatelé měli, byly zahrnuty do budoucího vylepšení aplikace v poslední části 7.2 této práce.

Číslo úkonu	Obtížnost
1	1.2
2	1.1
3	1
4	1
5	1.5
6	1
7	1
8	1
9	1
10	1.3
11	1
12	1.3
13	1
14	1

Tabulka 6.2: Výsledek testu přihlášeného uživatele.

Kapitola 7

Závěr

Cílem bylo vytvořit webovou aplikaci prezentující návštěvníkovi historické události a životopisy osob. Uživatelé si zde mohou prohlížet webový obsah, přidávat komentáře nebo se registrovat do systému a podílet se tak na tvorbě obsahu aplikace. Vznikla tak webová aplikace, která je dostupná na uvedené adrese¹.

Tato závěrečná kapitola popisuje v sekci 7.1 dosažené výsledky po dokončení práce. Nakonec nabídne několik možných budoucích rozšíření v navazující sekci 7.2.

7.1 Dosažené výsledky

Aplikace je psána v programovacím jazyce PHP, kde využila výhod a nástrojů frameworku Nette. Pro správu dat a manipulaci s nimi používá databázový systém MySQL. Práce byla postupně zpracována přes fázi implementace až do poslední fáze testování finální aplikace. Tato bakalářská práce vás provedla ukázkami podobně zpracovaných řešení. Představila technologie využívané k vývoji webových aplikací a nastínila vzhled a chování budoucí aplikace. Dále také popsala případy užití a ER diagram pro uchovávání dat v databázi.

Poté rozebrala postupnými kroky způsob implementace aplikace na základě předešlého návrhu. Bylo zapotřebí vytvořit šablonu webové stránky, oživit ji grafickými prvky a jejich funkčnost doplnit aplikační logikou s využitím frameworku Nette.

Zde byly ukázány nástroje a aplikace využitě při vývoji. Popisuje se tu také propojení databáze s aplikací a zobrazuje příklad komunikace s ní v případě, že potřebujeme obdržet nějaká data a pracovat s nimi. Dále kapitola pokračuje vysvětlováním nejdůležitějších komponent aplikace, s kterými přijde uživatel do styku. Mezi ně patří registrace a přihlášení do systému. Jakmile systém vytvoří uživateli účet, může si po požádání o obnovu hesla vytvořit nové uživatelské heslo.

Když už má uživatel přihlašovací údaje, může společně s ostatními uživateli webu tvořit články a životopisy. Ty může spravovat v administrativní části a pokud má vyšší oprávnění, tak i schvalovat pro zobrazení.

Na konci se dozvíme, jakým způsobem probíhalo testování této aplikace a jaké to mělo důsledky. Ve výsledné aplikaci je změna jazyku pro textový obsah omezena pouze na popisky textového editoru. Po všech vypracovaných částech se tak podařilo vytvořit webovou aplikaci, která splňuje všechny předem dané cíle.

¹<http://prezentsite.euweb.cz/>

7.2 Možná budoucí vylepšení

Výsledná webová aplikace je vytvořena tak, aby poskytovala základní nástroje a funkce, a proto nebylo již třeba v budoucnu dalších modifikací. Přesto se vždy najde vylepšení, které by aplikaci pozvedlo na novou úroveň, nebo alespoň využilo nejnovějších technologií v oblasti vývoje webových aplikací.

Takové vylepšení může být možnost zvolení si světlého nebo tmavého režimu zobrazení pro pohodlnější práci při pozdních večerních hodinách. Bude tak šetřit zrak uživatele a zpřehlední celkově web za tmy.

Dále rozšířit vytváření nebo editaci článku či životopisu osob technologií na principu drag & drop (táhni a pusť). To by umožnilo měnit pozici umístění textu, fotografií a audio-video nahrávek na stránce příslušného článku nebo životopisu a vylepšit tak vzhled webového obsahu.

Nabídka webové aplikace by mohla být doplněna o podrobnější informace týkající se historických osob. Tím je myšleno například přidání grafického zobrazení mapy s umístěním rodného bydliště. Dalším takovým rozšířením může být prohlížení rodových linií vybrané historické osoby.

Poslední možností je vytvoření mobilní verze této aplikace pro snadnější používání díky absenci webového prohlížeče.

Literatura

- [1] A PRODULE SYSTEMS (P) LIMITED. *MockFlow - Online Wireframe Tools, Prototyping Tools, Online Whiteboard, Design tool, UI Mockups, UX Suite, Remote design collaboration, UX Planning* [online]. 2021 [cit. 2022-01-10]. Dostupné z: <https://mockflow.com/>.
- [2] ADOBE. *Informace o webových aplikacích* [online]. 2021-05-19 [cit. 2021-10-24]. Dostupné z: <https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html>.
- [3] BRIGHTCOVE INC.. *Web Video Framework* [online]. 2022 [cit. 2022-04-5]. Dostupné z: <https://videojs.com/>.
- [4] CKSOURCE. *CKEditor 5 - Powerful Framework with Modular Architecture* [online]. 2003-2022 [cit. 2022-04-5]. Dostupné z: <https://ckeditor.com/ckeditor-5/>.
- [5] DEIGHT.EU. *Co je to Bootstrap a jak funguje?* [online]. [cit. 2021-11-25]. Dostupné z: <https://blog.deight.eu/web/html/co-je-to-bootstrap-a-jak-funguje/>.
- [6] DIAGRAMS.NET. *Diagrams.net: Diagram Software and FLOWchart Maker* [online]. 2005 [cit. 2021-10-21]. Dostupné z: <https://www.diagrams.net>.
- [7] FORAL, J. *WikiKnihovna – Webová aplikace* [online]. 2013-06-03 [cit. 2021-10-28]. Dostupné z: https://wiki.knihovna.cz/index.php?title=Webov%C3%A1_aplikace.
- [8] IBM CORPORATION. *Programování* [online]. [cit. 2021-10-28]. Dostupné z: <https://www.ibm.com/docs/cs/i/7.3?topic=serving-programming>.
- [9] IT SLOVNIK.CZ TEAM. *Co je to programovací jazyk?* [online]. [cit. 2021-10-28]. Dostupné z: <https://it-slovník.cz/pojem/programovaci-jazyk>.
- [10] JANOVSÝ, D. *CSS styly - úvod* [online]. 1999-2021 [cit. 2021-11-14]. Dostupné z: <https://www.jakpsatweb.cz/css/css-uvod.html>.
- [11] JANOVSÝ, D. *HTML příručka* [online]. 1999-2021 [cit. 2021-11-10]. Dostupné z: <https://www.jakpsatweb.cz/html/>.
- [12] JUNEXT.NET. *MySQL databáze - český manuál* [online]. 2002-2017 [cit. 2021-12-03]. Dostupné z: <https://www.junext.net/mysql/>.
- [13] KONEČNÝ, O. *Jak funguje CSS Box Model* [online]. 2020. 2020-11-10 [cit. 2021-11-14]. Dostupné z: <https://frontend.garden/jak-funguje-css-box-model/>.
- [14] LARAVEL TEAM. *Installation - Laravel - The PHP Framework For Web Artisans* [online]. 2022 [cit. 2021-12-20]. Dostupné z: <https://laravel.com/docs/8.x>.

- [15] LARAVELBLOG.CZ. *Vše o PHP frameworku Laravel* [online]. 2022 [cit. 2021-12-20]. Dostupné z: <https://laravelblog.cz/dokumentace/6.x/instalace>.
- [16] MANAGEMENTMANIA. *Architektura klient-server (Client-server model)* [online]. 2016-11-04 [cit. 2021-10-21]. Dostupné z: <https://managementmania.com/cs/architektura-klient-server>.
- [17] MANAGEMENTMANIA. *Třívrstvá architektura (Three-tier architecture)* [online]. 2015-12-05 [cit. 2021-10-24]. Dostupné z: <https://managementmania.com/cs/trivrstva-architektura-three-tier-architecture>.
- [18] MICROSOFT. *What is ASP.NET?* [online]. [cit. 2021-11-05]. Dostupné z: <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet>.
- [19] MONUS, A. *10 PHP Frameworks For Developers – Best of* [online]. 2018. 2018-04-11 [cit. 2021-12-13]. Dostupné z: <https://www.hongkiat.com/blog/best-php-frameworks/>.
- [20] MORÁVEK, P. *File Upload form control* [online]. 2020 [cit. 2022-04-5]. Dostupné z: <https://github.com/nepada/file-upload-control>.
- [21] NETTE. *Dokumentace Nette* [online]. 2008-2021 [cit. 2021-12-15]. Dostupné z: <https://doc.nette.org/cs/>.
- [22] POST BELLUM. *Paměť národa* [online]. 2008-2021 [cit. 2021-10-15]. Dostupné z: <https://www.pametnaroda.cz/cs>.
- [23] SAS, S. *Symfony Documentation* [online]. 2022 [cit. 2021-12-22]. Dostupné z: <https://symfony.com/doc/current/index.html>.
- [24] SKVORC, B. *The Best PHP Framework for 2015: SitePoint Survey Results* [online]. 2015. 2015-03-30 [cit. 2021-12-15]. Dostupné z: <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>.
- [25] TATROE, K. a MACINTYRE, P. *Programming PHP: Creating Dynamic Web Pages*. 4th Edition. O'Reilly Media, 2020. ISBN 978-1-492-05413-9.
- [26] W3TECHS. *Usage statistics of server-side programming languages for websites* [online]. [cit. 2021-12-06]. Dostupné z: https://w3techs.com/technologies/overview/programming_language.
- [27] WIKIPEDIE. *Klient-server – Wikipedie: Otevřená encyklopedie* [online]. 2021. 2021-06-01 [cit. 2021-10-21]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=Klient-server&oldid=19985947>.
- [28] WIKIPEDIE. *Webová stránka – Wikipedie: Otevřená encyklopedie* [online]. 2021. 2020-12-11 [cit. 2021-10-24]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=Webov%C3%A1_str%C3%A1nka&oldid=19253373.
- [29] ZENDULKA, J. *Konceptuální modelování* [online]. 2020. 2020-02-20 [cit. 2022-01-02]. Dostupné z: https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIDS-IT%2Flectures%2Fcz%2F2_kmod.pdf&cid=13296.
- [30] ÚSTR. *Ústav pro studium totalitních režimů* [online]. 2008-2021 [cit. 2021-10-16]. Dostupné z: <https://www.ustrcr.cz/>.

- [31] ČÁPKA, D. *ITnetwork – Ruby* [online]. 2020-09-03 [cit. 2021-11-07]. Dostupné z: <https://www.itnetwork.cz/programovani/ruby>.
- [32] ČÁPKA, D. *ITnetwork – Lekce 1 - Úvod do ASP.NET* [online]. 2012. 2020-11-28 [cit. 2021-11-05]. Dostupné z: <https://www.itnetwork.cz/csharp/asp-net-core/zaklady/tutorial-uvod-do-asp-dot-net>.
- [33] ČÁPKA, D. *ITnetwork – Lekce 2 - UML - Use Case Diagram* [online]. 2021. 2021-02-12 [cit. 2021-12-26]. Dostupné z: <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>.
- [34] ČÁPKA, D. a HÉL, S. *Lekce 1 - Úvod do Nette frameworku pro PHP* [online]. 2014. 2021-03-16 [cit. 2021-12-09]. Dostupné z: <https://www.itnetwork.cz/php/nette/zaklady/uvod-do-php-frameworku-nette>.
- [35] ŠAFAŘÍK, D. *ZonerCloud – Nejpoužívanější programovací jazyky pro backend webu* [online]. 2021. 2021-09-29 [cit. 2021-11-07]. Dostupné z: <https://www.zonercloud.cz/magazin/nejpouzivanejsi-programovaci-jazyky-pro-backend-webu>.
- [36] ŠTRÁFELDA, J. *Co je AJAX* [online]. [cit. 2021-11-19]. Dostupné z: <https://www.strafelda.cz/ajax>.
- [37] ŠTRÁFELDA, J. *Co je HTML* [online]. [cit. 2021-11-10]. Dostupné z: <https://www.strafelda.cz/html>.
- [38] ŠTRÁFELDA, J. *Co je PHP* [online]. [cit. 2021-11-04]. Dostupné z: <https://www.strafelda.cz/php>.

Příloha A

Obsah přiloženého paměťového média

Adresářová struktura disku:

```
CD disk/  
├─ xkovar87/  
│   ├── webova_aplikace/  
│   ├── BP_zdrojove_soubory/  
│   └─ BP_technicka_zprava/
```

- **webova_aplikace/** - Zdrojové soubory webové aplikace
- **BP_zdrojove_soubory/** - Zdrojové soubory potřebné k vytvoření technické zprávy.
- **BP_technicka_zprava/** - Technická zpráva ve formátu PDF