

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Redakční systém pro tvorbu osobních blogů

Bakalářská práce

Autor: Matěj Váňa
Studijní obor: Aplikovaná Informatika

Vedoucí práce: Ing. Jan Krunčík

Hradec Králové

duben 2024

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne

Matěj Váňa

Poděkování

Rád bych touto cestou poděkoval panu Ing. Janu Krunčíkovi za metodické vedení, cenné rady, odbornou pomoc a trpělivost při vedení této bakalářské práce.

Abstrakt

Cílem této bakalářské práce je návrh a implementace webové aplikace, konkrétně redakčního systému pro osobní blogy. Práce začíná popisem redakčních systémů. Následuje dotazníkové šetření, včetně představení výsledků. Dále pak popisuje architektury a koncepty pro vývoj webové aplikace, tento popis je zakončen shrnutím zvolených možností pro výslednou implementaci. Poté následuje popis využitých technologií pro vývoj. Popis technologií zahrnuje také představení nástrojů potřebných pro vývoj. Poslední obsahovou částí je představení výsledné aplikace. V rámci této části je popsána samotná aplikace, ale také jednotlivé části implementace, včetně řešení vybraných problematik. Závěrem práce je shrnutí výsledků a samotný závěr. Zde jsou popsány splněné cíle, navržené možnosti pro rozšíření. Výsledkem práce je funkční redakční systém využitelný pro osobní blog.

Klíčová slova: Redakční systém, blog, webová aplikace, Java, Spring, React, REST API, MySQL, backend, frontend

Abstract

Title: Content management system for creating personal blogs

The aim of this bachelor thesis is to design and implement a web application, specifically a content management system for personal blogs. The thesis starts with a description of content management systems. Next part is a questionnaire survey, including the presentation of the results. Third part then describes the architectures and concepts for the development of the web application, this description is ended with a summary of the options chosen for the resulting implementation. Following part is a description of the technologies used for development. The description of the technologies also includes an introduction of the tools needed for the development. The last content section is a presentation of the implemented application. Within this section, is describing of application itself, as well as few parts of the implementation, including solutions to selected issues. Finally, the paper concludes with a summary of the results and the conclusion itself. Here, the fulfilled objectives are presented, the proposed possibilities for extension are suggested. The result of the work is a functional content management system usable for a personal blog.

Key words: Content management system, blog, web application, Java, Spring, React, REST API, MySQL, backend, frontend

Obsah

1	Úvod	1
2	Redakční systém	2
2.1	Představení.....	2
2.2	Existující redakční systémy.....	3
3	Průzkum	6
3.1	Výzkumné otázky	6
3.2	Způsob dotazníkového šetření.....	6
3.3	Výsledky dotazníku	7
4	Vývoj webových aplikací.....	17
4.1	Architektura MVC.....	17
4.2	Rest API.....	18
4.2.1	Komunikace skrze http.....	18
4.2.2	Koncept REST.....	19
4.3	Backend.....	20
4.4	Frontend	21
4.5	Single-page aplikace.....	21
4.6	Architektura monolitu.....	22
4.7	Architektura mikroservis.....	24
4.8	Architektury použité pro vlastní řešení	25
5	Technologie.....	26
5.1	Backend – Spring.....	26
5.1.1	Java	26
5.1.2	Spring.....	28
5.2	Frontend – React	29
5.3	Databáze – MySQL	30
5.4	Vývojářské nástroje.....	31

5.4.1	Vývojové prostředí.....	31
5.4.2	Verzovací systém.....	32
5.4.3	Nástroje pro Api testování.....	33
6	Návrh a vývoj aplikace.....	34
6.1	Představení aplikace.....	34
6.2	Popis aplikace.....	35
6.3	Backend.....	39
6.4	Frontend.....	47
7	Shrnutí výsledků.....	50
8	Závěry.....	51
9	Seznam obrázků.....	52
10	Seznam tabulek.....	53
11	Seznam grafů.....	54
12	Seznam zdrojových kódů.....	55
13	Seznam použité literatury.....	56
14	Přílohy.....	1
14.1	Příloha 1.....	1
14.2	Příloha 2 – Fotodokumentace aplikace.....	1
15	Zadání práce.....	1

1 Úvod

Aktuálně je několik možností pro tvorbu osobního blogu. Existující řešení mají ovšem několik nedostatků. Ty jsou tvořeny primárně nutností složitých instalací různých částí systému, pro dosažení požadovaných funkcí, které je možné získat jako rozšíření od vývojářů třetích stran, případně vytvořit vlastní. Jedná se ovšem o způsob, který průměrný majitel osobního blogu nebude schopen technicky zvládnout. Nevýhodou těchto rozšíření, může být problém s kompatibilitou v rámci využití vícero rozšíření, eventuálně s neaktuální verzí. Ta nemusí odpovídat požadavkům hlavního samotného redakčního systému. Jednou z těchto funkcí je například možnost tvorby ankety pro jednotlivé články. Dalším nedostatkem, kterým disponují některé systémy využívané pro osobní blogy, je nemožnost uživatelského účtu pro návštěvníky. Kupříkladu u komentářů pod články, nemá autor ani ostatní návštěvníci jistotu, zda jsou komentující pod stejným uživatelským jménem, stále stejní lidé. Návštěvníci nemají ani možnost uložení článků do seznamů, které by mohli využít pro pozdější čtení. Více informací o nedostacích je součástí kapitoly zabývající se průzkumem, s využitím dotazníkového šetření.

Cílem práce je navrhnout a implementovat řešení, které odstraní některé nedostatky aktuálně nabízených možností a umožní snadnou instalaci celého systému. Bez potřeb dále rozšiřovat v podobě doplňků pro zpřístupnění funkcí, které jsou v rámci osobních blogů žádoucí. Vytvořené řešení pomůže majiteli stránek ve tvorbě článků, spravování celého blogu a zároveň dá nové možnosti návštěvníkům jeho stránky v rámci interakce s obsahem.

Práce se dělí do pěti částí. První část představuje pojem redakční systém, včetně existujících řešení. Druhá část se zabývá dotazníkovým šetřením. Třetí část představuje architektury a koncepty, které je možné využít, a byly využity pro vývoj výsledné aplikace. Čtvrtá část se zabývá popisem použitých technologií a nástrojů v rámci vývoje. Poslední pátá část představuje samotnou aplikaci, konkrétní problematiku z jednotlivých částí, či strukturu projektu.

2 Redakční systém

Tato kapitola se věnuje představení pojmu redakční systém. Dále také představení a porovnání existujících redakčních systémů.

2.1 Představení

[1] Redakční systém z anglického CMS neboli Content Management System, je software sloužící k jednoduché správě obsahu, většinou webových stránek. Redakční systémy by mělo být možné využívat bez znalosti programování, či jiných technických znalostí týkajících se například vývoje webových stránek. Správa obsahu v podobě článků, obrázků a ostatních typů obsahu, je tedy zjednodušená a vhodná i pro uživatele, kteří by běžné webové stránky nebyli schopni vytvořit. S využitím redakčního systému, je i průměrně zdatný uživatel počítače schopen spravovat webové stránky.

[1] Základním principem pro fungování jsou šablony. Využití šablon umožňuje dosazovat odlišná data, do jinak stejně vypadající stránky, čímž se snižuje náročnost paměti a zjednodušuje celý proces. S využitím jediné šablony, například pro článek na osobním blogu, není nutné vytvářet strukturu dané stránky opakovaně. Při požadavku na zobrazení daného článku, pouze systém automaticky doplní vhodná data, například z databáze dle zvoleného článku, a doplní je do šablony. Z pohledu technického zpracování se mohou různá řešení lišit. Zpravidla redakční systém obsahuje databázi pro uchování dat, část pro zpracování dat a část pro zobrazení dat.

[1] Standardem je v dnešní době možnost tzv. CRUD operací nad obsahem, CRUD z anglického Create, Read, Update, Delete, označuje možnost základních operací vytvořit, přečíst / načíst, upravit, smazat. Na základě těchto operací mají možnost pověření uživatelé měnit obsah dle vlastní libosti a možností, které jim systém dává. Obsahem v případě redakčního systému, je například článek nebo komentář. Pro určení pověřeného uživatele pro danou operaci, se využívá systém rolí.

[1] Role v rámci CMS označuje skupinu uživatelů s definovanými právy. Pojem „práva“ označuje možnost provádět určité operace. Přidělení práv určité roli, umožňuje stejná práva jednoduše udělit celé skupině uživatelů. Tato skupina uživatelů je poté schopna na základě přidělené role provádět úkony, které jsou pro ni žádoucí. Zpravidla dochází k definování „nejvyšší“ role, kterou by mohl být například správce (anglicky

označován jako „admin“). Správce má následně maximální možnou kontrolu nad obsahem, včetně možnosti využít funkce specifické pro jeho roli.

Role	Článek			
	Vytvořit	Zobrazit	Upravit	Smazat
Správce	Ano	Ano	Ano	Ano
Přihlášený uživatel	Ne	Ano	Ne	Ne
Nepřihlášený uživatel	Ne	Ano	Ne	Ne
Role	Komentář			
	Vytvořit	Zobrazit	Upravit	Smazat
Správce	Ano	Ano	Vlastní	Všechny
Přihlášený uživatel	Ano	Ano	Vlastní	Vlastní
Nepřihlášený uživatel	Ne	Ano	Ne	Ne

Tabulka 1: Ukázka možných rolí redakčního systému
Zdroj: Autor

Výše uvedená tabulka, zobrazuje možné rozdělení rolí v rámci redakčního systému. Ve zmíněné tabulce je uvedena zvlášť role pro přihlášeného a nepřihlášeného uživatele. Role nejsou obecně definované, každý systém může mít různé role, či pouze odlišená práva pro každou z rolí. Role jsou zde popsány z pohledu celkového CMS, nejedná se o popis z pohledu registrovaných uživatelů systému. V takovém případě by nebyl odlišen přihlášený a nepřihlášený uživatel, ale pouze uživatel. Počet rolí není nijak omezen.

2.2 Existující redakční systémy

Redakční systémy se mohou lišit možnostmi, kterými je lze využít. Většinu z nich lze využít pro téměř jakýkoliv účel, což nemusí být vždy vhodné vzhledem ke skutečnosti, že nebyl pro potřebný účel primárně tvořen.

Nejpopulárnější CMS je WordPress [2] (viz. [3]), který umožňuje široké spektrum využití, například osobní blogy, firemní stránky, stránky online magazínů, ale také internetové obchody a další. Zároveň také umožňuje instalaci pluginů (rozšíření), které je možné získat z oficiálního repozitáře (úložiště) dostupné na: wordpress.org/plugins/. WordPress je tedy možné upravovat a od vývojářů třetích stran, doplnit funkce, které samotný WordPress neobsahuje. Právě skutečnost s pluginy nemusí být nutně pouze výhodou, pro mnoho funkcí je jejich instalace nutná. S tím je spojena náročnější údržba celku, včetně instalace a prvotních nastavení,

aktualizace pluginů, hledání kompatibilních pluginů. Pluginy třetích stran nebo také velké množství pluginů, mohou způsobit zpomalení načítání stránek.

Blogger [4] dříve blogpost, je CMS od společnosti Google, které se zaměřuje na tvorbu blogu. Systém umožňuje samotné vytvoření blogu, které je možné skrze několik motivů v různých barevných variantách upravovat. Je možná i úprava rozvržení stránek. Má zároveň připravené napojení na službu Google AdSense, která je určena k monetizaci blogu. Mezi funkce náleží například: zobrazení komentářů, jednoduchých statistik a kompletní správu příspěvků. Blogger disponuje několika nedostatky pro zapojení čtenářů. Dříve Blogger obsahoval funkci pro přidání anket, která byla odebrána. Jedinou aktuální možností zpětné vazby jsou komentáře. Ty je možné u svého blogu nastavit na otevřené všem. V tomto nastavení je možné komentovat články anonymně, nebo skrze účet Google, případně jednorázově zadaným jménem. Dalším nastavením komentářů, je možnost pouze pro uživatele s účtem od Google. Případně pouze pro členy konkrétního blogu, jež lze přidat v nastavení. Google skrze Blogger utváří komunitu uživatelů kolem samotné služby. Osoba s jedním uživatelským účtem (Google účet), spravuje vlastní blogy a komentuje příspěvky ostatních. Lze tedy říct, že Blogger formuje svou komunitu, nikoliv komunitu vašeho blogu. Zároveň nelze říct, že se jedná o vlastníka blogu, ale pouze jeho autora, blog stále náleží společnosti Google.

Dalšími zástupci jsou například Wix [5] nebo české Webnode [6]. Služby, které umožňují tvorbu vlastních blogovacích stránek díky připraveným šablonám. Tyto služby obsahují dělení do různých plánů. Plány se odvíjí podle měsíční platby na základě zvoleného balíčku, který se může lišit podle využití vytvořených stránek, od osobních blogů až po internetový obchod. Webnode nabízí tvorbu stránek i zdarma, jedná se však o stránky, u kterých jsou omezené funkce, kupříkladu doména je třetího řádu (například vasblog.webnode.cz), zároveň má také omezený prostor, limit na měsíční přenos dat a další omezení. Tyto služby, stejně jako Blogger, nemají možnost vlastního hostování CMS, jedná se tedy o stránky nabízející hostování stránek s využitím jejich CMS. V takovém případě záleží na preferencích konkrétního uživatele. Jako pozitivní lze brát, že odpadá starost s instalací a uvedením stránek do provozu. Negativum může být to, že uživatel nemá vlastní stránky plně pod svou kontrolou a platí službu, za jejichž cenu lze získat vlastní serverové prostředky, které lze zároveň využít pro jiný účel, nebo pro více takových stránek.

Název	Cena	Výhody	Nevýhody
WordPress	Zdarma (+ hosting, doména)	Jednoduchost, velká komunita, rozšíření, open-source, komplexita, úprava vizuálu	Kompatibilita rozšíření, prvotní nastavení požadovaných funkcí
Blogger	Zdarma (+ dobrovolná vlastní doména)	Bez nutnosti hostingu, napojení na Google služby	Obsah náleží Googlu, bez možnosti rozšíření, problémy se spamem, chybějící ankety, omezená možnost úprav
Wix	zdarma, poté od 11€ měsíčně	Úprava vizuálu, počet funkcí	Omezené funkce podle placeného balíčku, složitost nastavení
Webnode	Zdarma, poté od 165 Kč měsíčně	Vhodné pro nováčky, doména zdarma (u placených balíčků), úprava vizuálu, statistiky	Omezené úložiště a přenos dat dle balíčku, funkce omezené dle balíčku (vyhledávání, registrace uživatelů, reklama na Webnode)

Tabulka 2: Porovnání redakčních systémů
Zdroj: Autor

Tabulka výše zobrazuje základní přehled o ceně, a o výhodách a nevýhodách různých řešení. Jedná se o stručný přehled výše zmiňovaných systémů. Při pohledu na cenu je viditelné, že vítězem je Blogger od společnosti Google. Ten je totiž možné provozovat bez nutnosti investice jediné koruny, zároveň je jediným možným nákladem vlastní doména. Podle výhod systémů je viditelná častá možnost úpravy vizuálu, ta je však velmi různá v jednotlivých případech, ačkoliv je v každém nějakým způsobem možná. V souvislosti se systémem Wix a Webnode, zde vidíme stěžejní nevýhody v podobě nutnosti odemčení funkcionalit skrze dražší placený balíček. Cena pro odemčení všech funkcí by pravděpodobně přesahovala cenu základního hostingu a domény pro využití systému WordPress. Ten je vhodný, ale klíčovým aspektem je složitost prvotní instalace a nastavení. Další nevýhody mohou způsobit problémy s rozšířeními.

3 Průzkum

Průzkum proběhl mezi majiteli osobních blogů, kteří využívají různá dostupná řešení. Cílem bylo zjistit názor na nabízené možnosti redakčních systémů od lidí, které je používají či používali, včetně popularity jednotlivých možností. V rámci průzkumu byly zjištěny nedostatky, ale také oblíbené funkce, a zároveň návrhy na funkce, které systémům chybí.

3.1 Výzkumné otázky

- 1) Použitý blogovací systém?
- 2) Spokojenost s blogovacím systémem 1-5?
- 3) K jakému účelu využíváte blog?
- 4) Jaké je téma Vašeho blogu?
- 5) Cena provozu Vašeho blogu?
- 6) Možnost uvést odkaz na blog respondenta.
- 7) Zkušenosti s jiným systémem.
- 8) Důvod změny blogovacího systému?
- 9) Výběr blogovacího systému.
- 10) Které výhody má z Vašeho pohledu Vámi využívaný blogovací systém?
- 11) Které nevýhody má z Vašeho pohledu Vámi využívaný blogovací systém?
- 12) Co podle Vás aktuálním systémům chybí?
- 13) Pokud by existoval český systém pro blogování, který by obsahoval vše, co potřebujete a využíváte, případně něco navíc zvažovali byste změnu?
- 14) Vaše doplnění a sdělení.

3.2 Způsob dotazníkového šetření

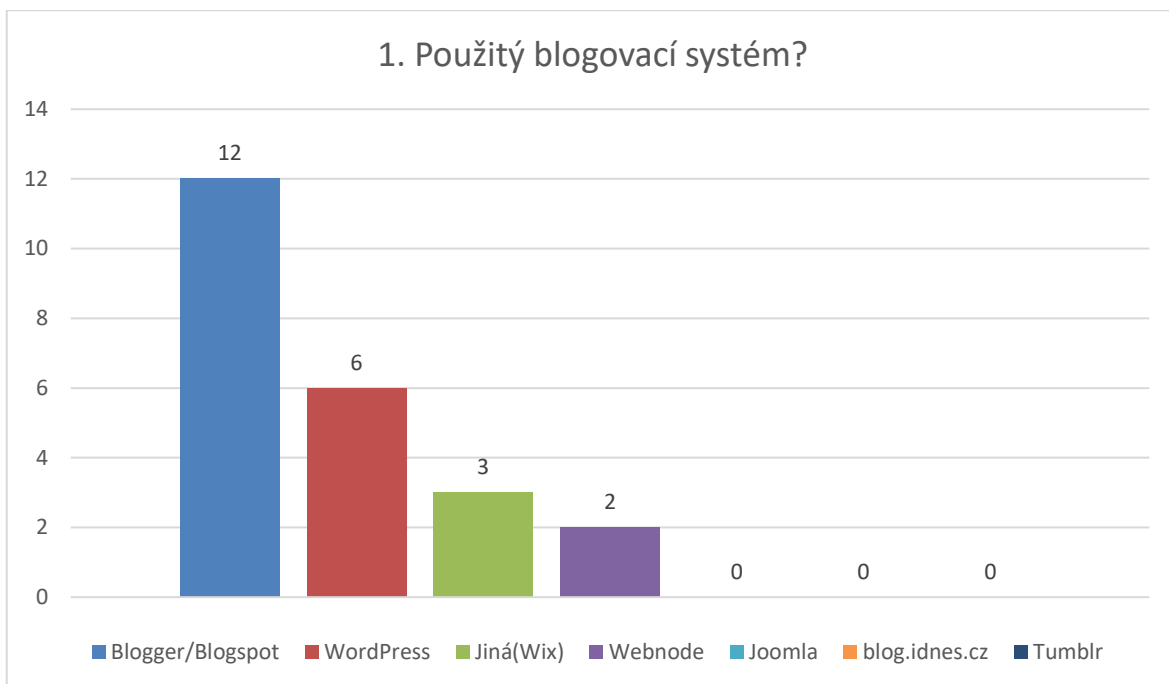
Dotazníkové šetření bylo uskutečněno online podobou. Dotazník byl tvořen pomocí online aplikace Survio [7] a sdílen skrze sociální síť Facebook ve skupině lidí, kteří se zaměřují na sdílení rad a zkušeností ohledně tvorby vlastních blogů. Dotazník byl rozdělen do čtyř stránek (celkem 14 otázek).

První otázka byla na používaný systém. Touto otázkou bylo možné přiřadit časté nedostatky pro konkrétní systémy a jejich pozitiva. Z odpovědí bylo možné při tvorbě systému čerpat. Na otázku dále navazuje otázka na spokojenost s aktuálně používaným systémem. Třetí otázka se táže, k jakému účelu je blog využíván, díky čemuž je možné určit, zda systémy pro osobní blogy nejsou využívány i pro jiné účely. Následující otázka je na téma stránek a ceny na provoz, což pomůže určit aktuální finanční náročnost pro majitele blogů a hlavní témata, která jsou dost populární. Poté měl dotazující možnost uvést odkaz na vlastní blog, díky čemuž je možné se podívat na některá řešení a v nich nalézt zmiňované chyby v dotazníku.

Následuje série otázek na důvod výběru systému a zkušenosti s jinými systémy. První otázka z této skupiny byla, zda má dotazující zkušenosti s jinými systémy, díky čemuž je možné zjistit, jak je respondent zkušený v této oblasti. Dále je otázka na důvod změny systému na aktuálně používaný, čímž je možné se dozvědět chyby a nedostatky, případně funkce u aktuálního systému, které respondenty zaujali. Další otázka je na důvody, proč si vybrali nový systém, což je možnost zjistit výhody, které systém má. Tato otázka je rozdělena do dvou, kde v první je výběr z připravených možností a druhá otázka nabízí volnou odpověď. Navazují dvě otázky s volnou odpovědí, první je na nevýhody systému a druhá na nedostatky, které systém má, respektive na chybějící funkce. Odpovědi umožňují získat přehled a předejít tomu, aby tvořený systém měl stejné nedostatky, případně chyby. Předposlední otázka je na téma, zda by uživatelé změnilo používaný systém, a to v případě, že by existovalo řešení, které by neobsahovalo jimi zmíněné nedostatky a chyby. Odpovědi u této otázky jsou vhodné k získání přehledu, jaký by byl zájem o tvořené řešení. Poslední otázka je s volnou odpovědí a dává respondentům možnost pro vlastní vyjádření.

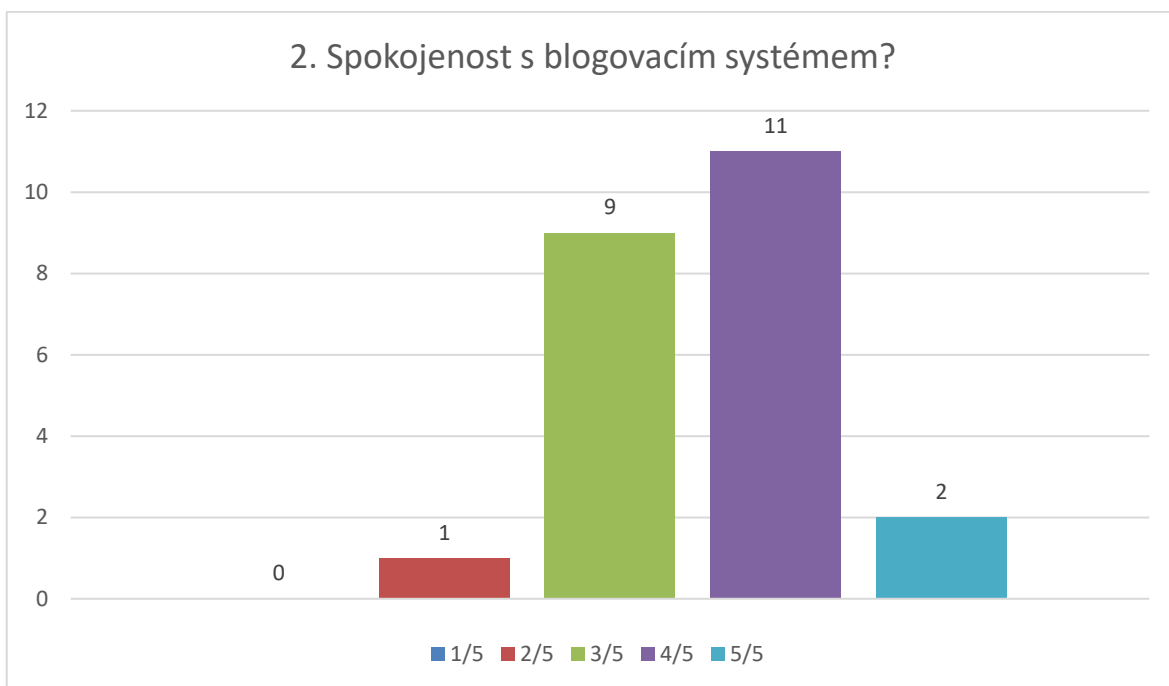
3.3 Výsledky dotazníku

Dotazník si zobrazilo 28 osob z nichž dotazník vyplnilo 23 respondentů, z čehož je zřejmá úspěšnost vyplnění 82,1 %.



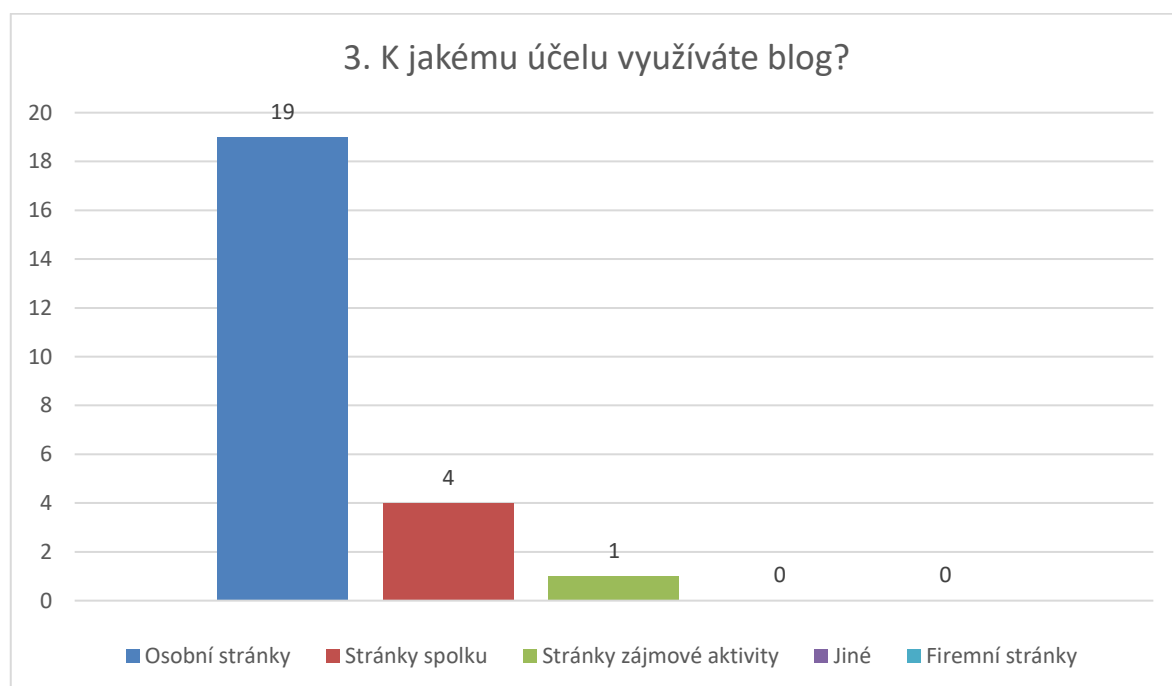
Graf 1: Výsledky otázky 1
Zdroj: Autor

Výše uvedený graf zobrazuje odpovědi k otázce na použitý blogovací systém respondenta. Lze vidět, že většina respondentů využívá blogovací systém Blogger. Respondenti, kteří využili možnost „Jiná“ odpověděli, že využívají službu Wix.



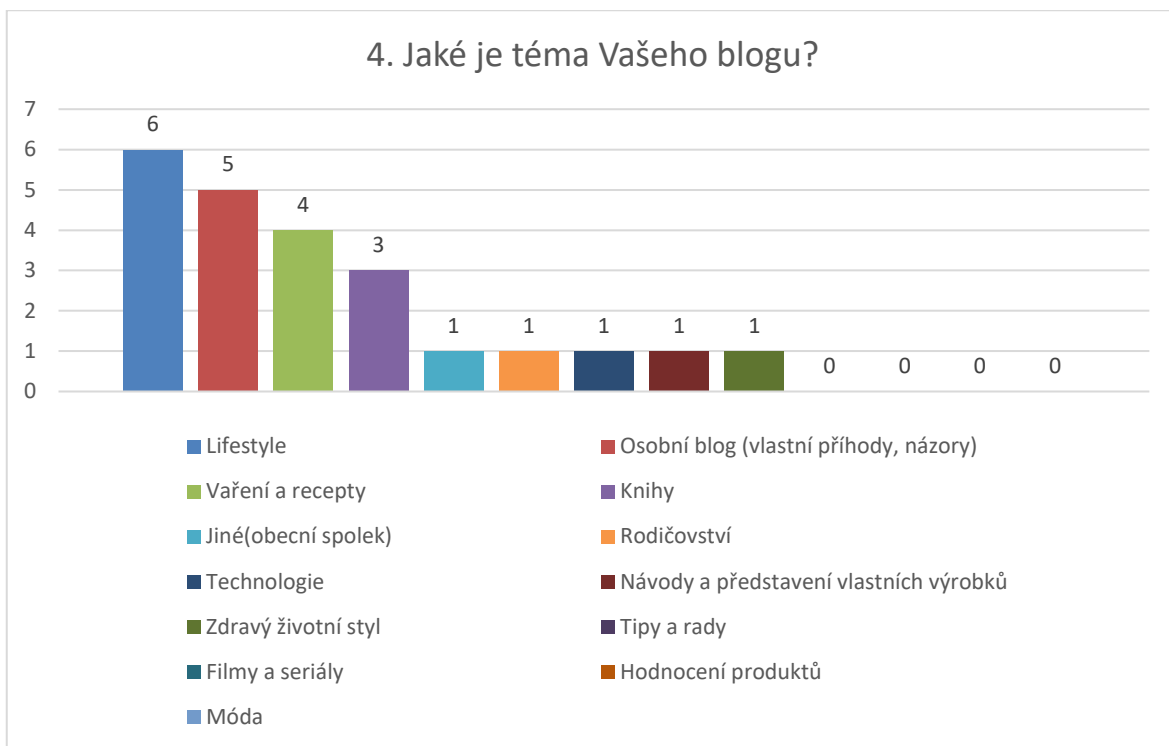
Graf 2: Výsledky otázky 2
Zdroj: Autor

Předchozí graf zobrazuje spokojenost respondentů s blogovacím systémem. Na základě výsledků lze určit, že blogovací systémy jsou kvalitní. Stále je zde však prostor pro zlepšení, což je zřejmé ze skutečnosti, že 9 (39,1 %) respondentů uvedlo blogovací systém jako průměrný.



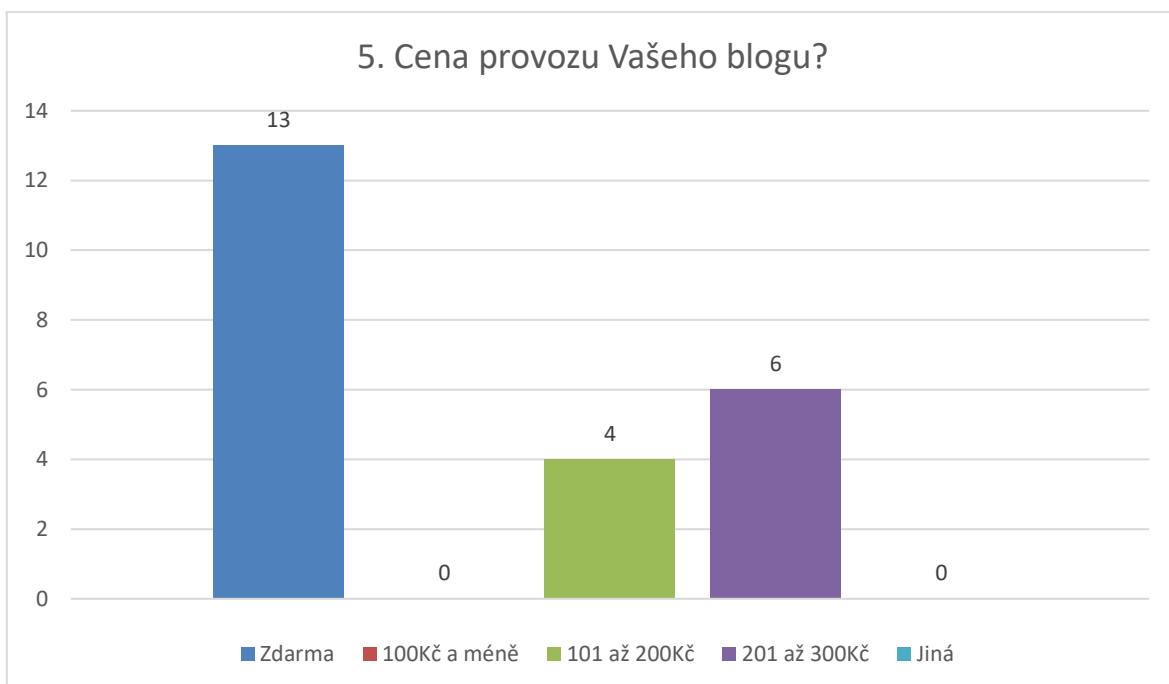
Graf 3: Výsledky otázky 3
Zdroj: Autor

Dle odpovědí viditelných na předešlém grafu, je zřejmé, že ačkoliv respondenti využívají blogovací systémy, jsou možnosti využití různé. Podle výsledku 5 respondentů využívá blogovací systém i za jiným účelem nežli za účelem vlastní osobní stránky.



Graf 4: Výsledky otázky 4
Zdroj: Autor

Odpovědi u otázky 4, viditelné na grafu výše, zobrazují informaci, že většina bloggerů sdílí svůj životní styl (lifestyle), či vlastní příhody. Poté má větší zastoupení ještě vaření a téma knihy. Ostatní témata jsou spíše ojedinělá.



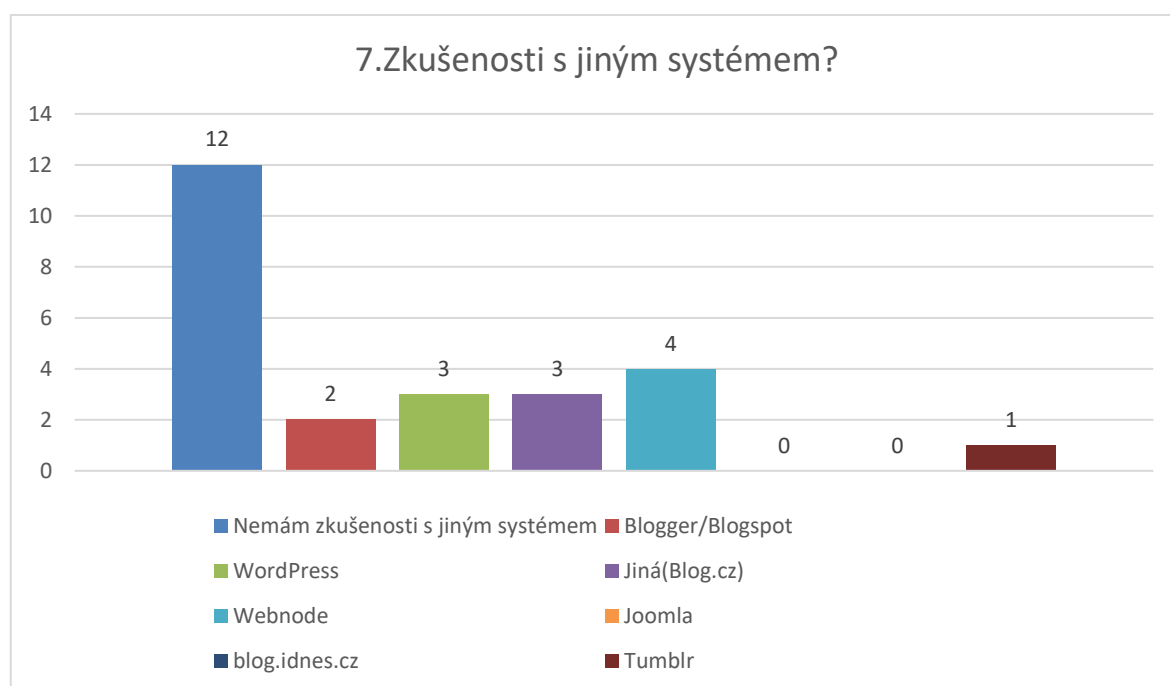
Graf 5: Výsledky otázky 5
Zdroj: Autor

Předešlý graf zobrazuje cenu provozu, kterou blogeři platí za provoz svých stránek. Většina respondentů uvedla, že provoz jejich blogu nevyžaduje finance. Pravděpodobnou příčinou je velké zastoupení služby Blogger, která je zdarma.

6. Zde můžete uvést odkaz na Váš blog.	
Odpoověď	Počet
Neuvedeno	19
www.malyvrabcak.cz	1
www.blogbyellie.com	1
www.barborah.com	1
https://popelciny-knizky.blogspot.com/	1

Tabulka 3: Výsledky otázky 6
Zdroj: Autor

Výše uvedená tabulka zobrazuje výsledky na otázku 6, která dala možnost uvést odkaz na blog respondenta. Možnost uvést odkaz na svůj blog využilo pouze 17,4 % respondentů.



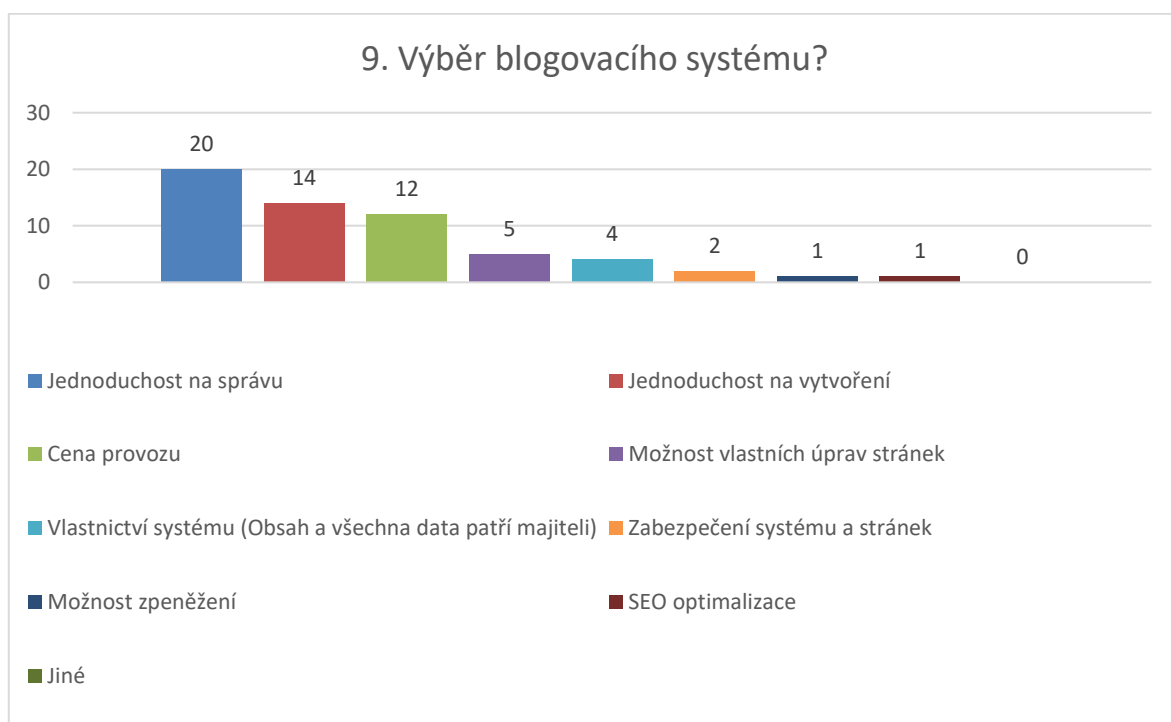
Graf 6: Výsledky otázky 7
Zdroj: Autor

Předešlý graf zobrazující výsledky otázky 7, poukazuje, že téměř polovina respondentů vyzkoušela více blogovacích systémů.

8. Důvod změny blogovacího systému?	
Neuvedena odpověď u 14 respondentů	WordPress mi přijde lepší, i designově
Více funkcí	Ukončení činnosti blog.cz
Blog.cz byl nejlepší blogovací platforma v ČR, bohužel skončila	Složitost s nastavením, časté chyby způsobené doplňky
Nevhodný systém pro udržení blogu větších rozměrů	Zbytečná složitost Webnode pro moje potřeby
Cena	Korporátní přístup Googlu

Tabulka 4: Výsledky otázky 8
Zdroj: Autor

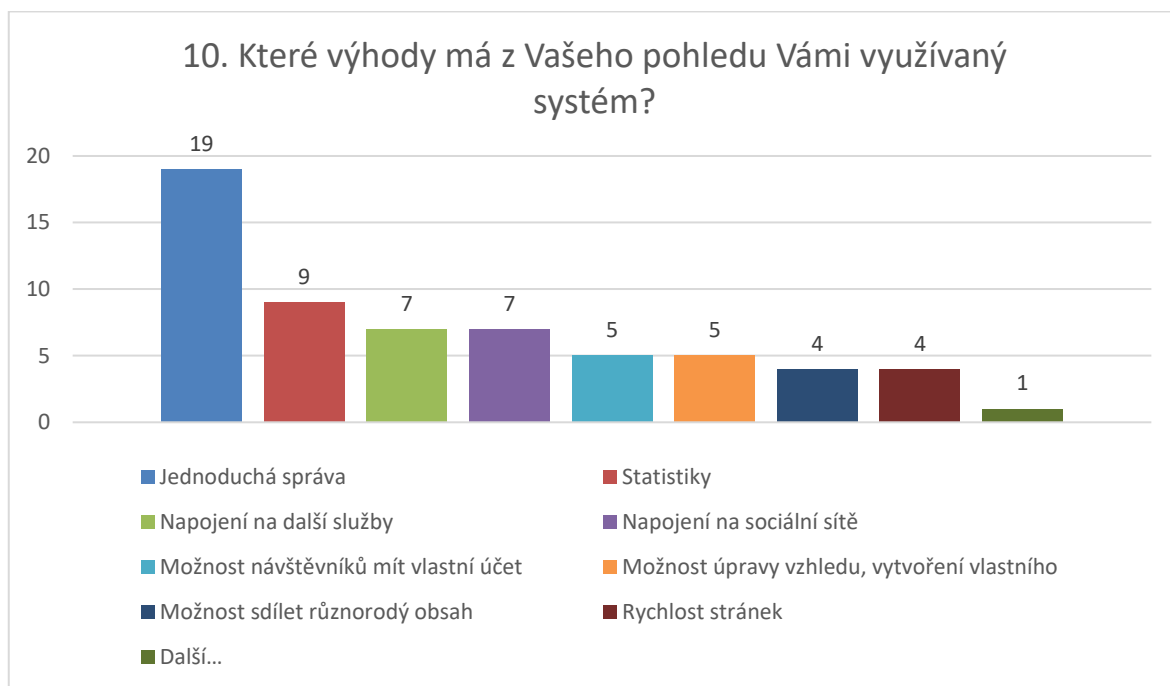
Výše uvedená tabulka vyjadřuje slovní odpovědi na otázku ohledně důvodu pro změnu blogovacího systému. Jak je viditelné 14 respondentů neuvedlo odpověď. Na základě dvou odpovědí lze vidět, že změna nebyla vždy dobrovolná. Konkrétně se jedná o ukončení činnosti služby blog.cz.



Graf 7: Výsledky otázky 9
Zdroj: Autor

Otázka 9, která je viditelná na grafu výše, odpovídá na důvody, proč si respondenti zvolili aktuálně používaný blogovací systém. Je zřejmé, že základem pro většinu respondentů je jednoduchost správy, instalace a cena provozu. Ostatní možnosti

disponují přibližně stejným počtem odpovědí. Na základě těchto odpovědí je možné říct, že klíčové je pro správce blogů prostředí, které umožňuje jednoduchou správu a je cenově přijatelné.



Graf 8: Výsledky otázky 10
Zdroj: Autor

Otázka 10, jejíž odpovědi jsou viditelné v předcházejícím grafu, zodpovídá přímé výhody používaného systému. Zároveň je lze vidět, že jednoduchá správa je velmi častá odpověď. Na základě ostatních odpovědí je viditelné, že oblíbenou výhodou jsou také statistiky, napojení na další služby včetně sociálních sítí. Tyto odpovědi jsou zřejmé, vzhledem k popularitě sociálních sítí v dnešní době. Poté následuje možnost vlastního účtu pro uživatele, skrze který mohou komentovat články, ukládat články či je hodnotit. Podobně je na tom možnost úprav vlastních stránek či případně vytvoření vlastního vzhledu. Tuto možnost nabízí většina blogovacích systémů. Možnosti úprav jsou často v podobě připravených šablon, či možnosti rozmístění komponentů výsledné stránky. V některých případech je také možné vytvořit vlastní vzhled s využitím HTML a CSS.

11. Které nevýhody má z Vašeho pohledu Vámi využívaný systém?	
Složitost instalace	2
Nevím / nesetkal/a jsem se s nevýhodou	4
Nutnost spojení s Google účtem	2
Spam nevhodných komentářů	3
V případě úprav grafiky to dá trochu práce a je nutná znalost alespoň základního kódování	V případě blogspotu korporátní přístup Googlu
Systém občas zlobí, při tvorbě článku občas nereaguje na změny editace textu, fotky si vkládá do článku vždy v jiném pořadí, někdy rozhodí text, jednoznačně problémy při editaci textu, občas nedojde k uložení mého komentáře u jiného blogu	Málo šablon a nepřehledné nastavení vzhledu, chybí mi například i možnost vytvořit pod článek anketu
Přístup k uživatelským účtům v rámci komentování	Pro někoho možná náročnější starání se o běh webu
Občasné problémy při psaní článků	Nutnost doplnit funkce pluginy
Místa primitivní nastavení, bez možnosti úprav dle vlastních představ	Malá možnost interakce s uživateli
Časté aktualizace	Chybí ankety, chybí interakce s uživateli, moc jednoduché nastavení, Google

Tabulka 5: Souhrn výsledků otázky 11

Zdroj: Autor

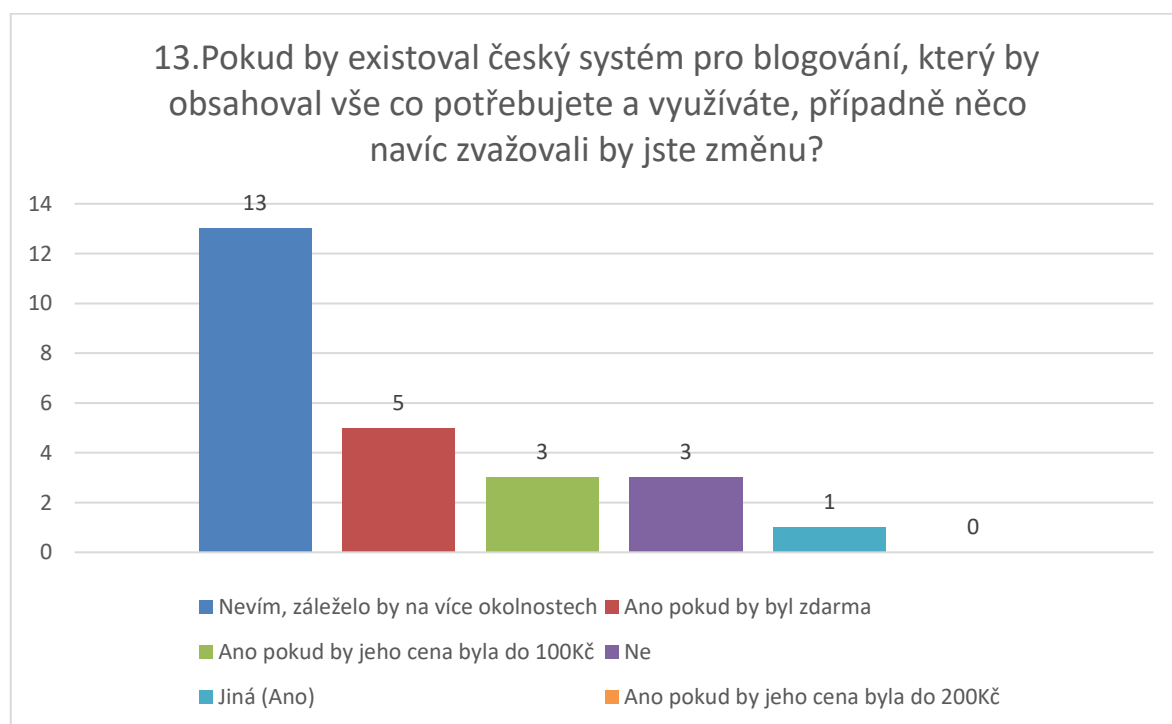
Výše uvedená tabulka zobrazující otázku 11, která byla v dotazníku vyžadována a zároveň je v podobě volné odpovědi obsahuje různorodé odpovědi. Časté odpovědi krátkého rozsahu byly sdruženy, a jejich počet je vyjádřen číslicí ve druhém sloupci. V odpovědích je viditelné, že 4 odpovědi vyjadřují, že respondent o nevýhodách neví. Další opakující se odpovědí je složitost instalace. Nedostatkem je také spam v komentářích u článků, případně falešné komentáře. Tato informace je zastoupena ve třech případech. V rámci komentářů jsou zde také zmíněné nedostatky týkající se možností zpětné vazby od čtenářů, případně interakce s nimi.

Odpovědi na otázku 12 viditelné v nadcházející tabulce, většina respondentů nevyužila možnost odpovědět. Podle respondentů, co odpověděli, lze říct, že chybějícím prvkem

jsou základní funkce, které je nutné doplnit skrze rozšíření v podobě doplňků či pluginů. Druhou chybějící funkcí je možnost vizuálních úprav stránek.

12. Co podle Vás aktuálním systémům chybí?	
Neuvedeno	16
Účet, rychlejší komunikace	Nezávislost
Potřebné možnosti v základu bez nutnosti dodatků v podobě pluginů	Na blog.cz si mohl i člověk neovládající HTML vytvořit celý design stránek podle sebe
Možnosti vlastního hostingu bez nutnosti složitého doplňování rozšíření pro požadované funkce	Jednoduchost se všemi funkcemi
Jednodušší změna designu (ne každý umí HTML/CSS) a než se to člověk dokáže naučit, hrozně to trvá, takže změnit jednoduchou věc je třeba na několik týdnů. Možnost soukromých zpráv, program na zmenšení velikosti fotek, kvůli načítání, detailnější přehledy o návštěvnosti blogu.	

Tabulka 6: Výsledky otázky 12
Zdroj: Autor



Graf 9: Výsledky otázky 13
Zdroj: Autor

Odpovědi na otázku 13 viditelné na předešlém grafu, Poukazují, zda jsou respondenti ochotní měnit svůj blogovací systém za nový. Podle odpovědí je zřejmé, že většina respondentů nedokáže přesně určit, zda ano nebo ne, dokud nemají bližší informace. Část respondentů by systém změnili, pokud by jeho provoz byl zdarma. V neposlední řadě, je zde rovnoměrný počet odpovědí u respondentů, co by byli ochotni zaplatit částku do 100 korun a těch, co by systém neměnili. Poslední otázka, která již není viditelná, byla možnost pro respondenty předat informaci, kterou uznají za vhodné, a pro kterou nebyla příležitost v rámci dotazníku. Tuto možnost žádný z respondentů nevyužil.

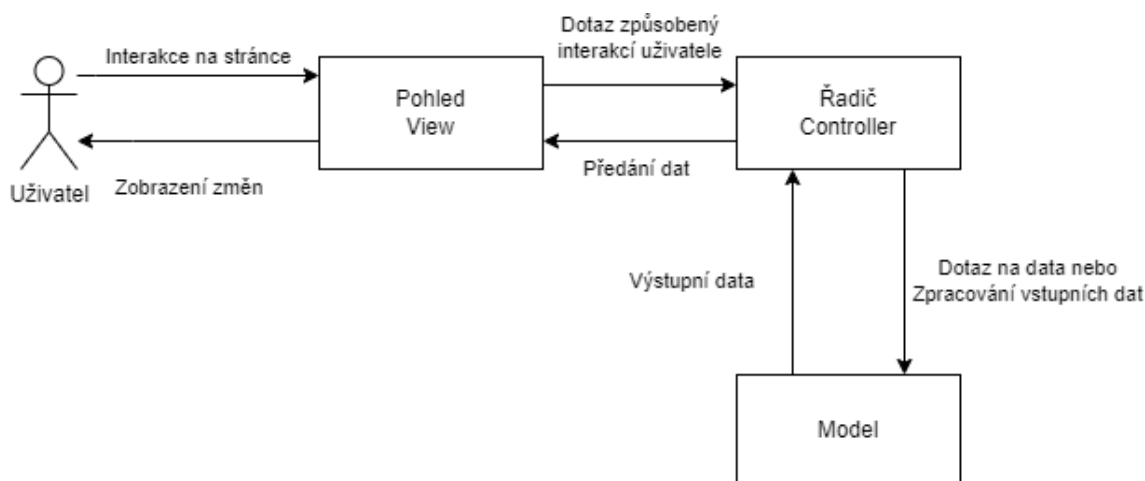
Na základě dotazníku lze říct, že pro respondenty, tudíž i pro osoby využívající blogovací systémy, je důležitá jednoduchá správa systému včetně prvotního vytvoření, možnost nastavení dle vlastních představ s možností napojení na sociální síť. Zároveň je také viditelné, že majitelům blogů chybí možnost využít více způsobů pro získání zpětné vazby od svých čtenářů. Dalšími funkcemi může být možnost upravit stránku z vizuálního pohledu, a prostředky k zamezení falešných komentářů.

4 Vývoj webových aplikací

Vývoj webové aplikace, kterou je i redakční systém, dává možnost využít různé architektury. V rámci této kapitoly jsou různé architektury a koncepty představeny. Závěrem kapitoly je pak stručný popis zvolených architektur či konceptů pro vývoj aplikace v rámci vlastního řešení.

4.1 Architektura MVC

[8] MVC neboli Model – View – Controller je architektura, jejíž hlavní myšlenkou je oddělení jednotlivých částí aplikace, do předem určených skupin. Díky tomu je celý projekt z vývojářského hlediska přehlednější a umožňuje snadnější vývoj složitějších aplikací.



Obr. 1: Diagram MVC architektury
Zdroj: Autor

[8][9] Část model je primárně určena pro logickou část aplikace. Do skupiny model tedy řadíme všechny komponenty, které jsou logického charakteru. Logickým charakterem jsou označeny operace nad daty. Může se jednat o úpravy před uložením, samotné uložení nebo naopak získání dat z databáze. Model nemá informace o původu vstupních dat, stejně jako nemá informace o způsobu jeho prezentace uživateli. V rámci modelu však vstupujeme do databáze, tudíž pokud model získá dotaz na data, která jsou uložena v databázi, provede tento požadavek a získaná data z databáze předá zpět. Pro model není důležitý výsledný formát, či způsob prezentace dat. Model je tedy izolovanou částí, která získává vstupní data z řadiče a vrací zpět výstupní data.

[8][9] Část view neboli pohled, je určena k prezentaci výstupu pro uživatele a zároveň způsob zadávání vstupních dat. V případě redakčního systému se jedná o webovou stránku, kterou vidí koneční uživatelé. Při interakci dojde ke vznešení požadavku, který dále zpracovávají další vrstvy aplikace.

[8][9] Část controller neboli řadič, jehož hlavní role je zprostředkování komunikace mezi pohledem a modelem. Lze tedy říct, že řadič udržuje celý systém pohromadě, jelikož bez jeho fungování by pohled nemohl získávat data a neměl by možnost komunikovat s modelem, tudíž by nemohlo docházet k logickým operacím, které aplikace umožňuje. Například pokud uživatel způsobí dotaz pomocí tlačítka v pohledu, dotaz přijde na řadič, který dále přesune práci na model, kde po vyřešení logické části úlohy, jsou výstupní data vrácena řadiči, který je zpět předá na pohled.

4.2 Rest API

[10] Rest API je architektura založená na principu příjmu dat klienta skrze API dotazy na server. Komunikace probíhá v rámci webové API, která využívá protokol http, případně zabezpečený https.

4.2.1 Komunikace skrze http

[10] Komunikace probíhá v rámci požadavků a odpovědí. HTTP request neboli požadavek, obsahuje několik informací. Informace obsažené v požadavku jsou metoda, adresa s parametry, hlavičky a tělo.

[11] Metod pro požadavky je několik. Metoda GET je určena pro získání dat a její použití by nemělo ovlivnit uložená data. POST je metoda pro vkládání dat, její využití je možné použít i pro úpravu dat. Metoda PUT je vhodnější na samotnou úpravu dat, nejvhodnější využití PUT je v situaci, kdy dochází ke změně všech atributů objektu. Pro úpravu dat je také vhodné využít PATCH, a to v případě, pokud nedochází ke změně u většiny atributů. Metoda sloužící pro mazání dat je DELETE.

[10] Adresa je odkaz na webové API, jedná se o URL adresu. Adresa může být s parametry nebo bez nich. V případě adresy bez parametrů se jedná o běžnou URL adresu. Pokud adresa obsahuje parametry, nese v nich data, která jsou součástí dotazu.

[11] Hlavičky pro uchování a předání doplňkových informací o dotazu. Jedná se o další parametry, ale oproti parametrům předaným v adrese, jsou tyto na základě

standardu, který se v http protokolu používá. Parametry v hlavičce jsou například *Date*, který je určen k uchování času, kdy došlo k odeslání, *Content-Type*, který obsahuje typ, ve kterém je posílán obsah v těle dotazu, aby bylo možné správné zpracování. Na základě toho je možné určit v jakém formátu budou předána data, je možné využít JSON, XML, volný text a další. Parametrů, které hlavičky obsahují je mnoho, ale dnešní řešení pro tvorbu komunikace skrze http protokol, si zajišťuje většinu nastavení automaticky, případně jejich úprava není důležitá pro předání dat v komunikaci, ale jedná se o data, díky kterým je komunikace možná a nesou informativní charakter, například operační systém, ze kterého byl dotaz odeslán.

[11] Odpověď při http požadavku neboli http odpověď, obsahuje velmi podobné informace jako dotaz. Opět se zde nachází hlavičky, tělo, které jsou stejné jako u požadavku.

[11] Další informací je však http status code. Jedná se o číselný kód, který předává informaci o výsledku dotazu. Číselné kódy lze základně rozdělit do pěti skupin, které se různí na základně první ze tří cifer kódu. Pokud je ve formátu 1xx, jedná se o informativní odpověď, že byl požadavek přijat, ale jeho zpracování stále probíhá. V případě formátu 2xx, se jedná o informaci, že byl požadavek úspěšně přijat a zpracován. Formát 3xx označuje, že je potřeba zaslat další požadavek jinam aby bylo možné získat odpověď. Zbylé dva formáty jsou chybové. 4xx označuje špatně položený dotaz. Jedná se o chybu na straně klienta, kdy dotaz mohl být špatně položen nebo mohl obsahovat špatná data. Poslední formát 5xx informuje o chybě na straně serveru, kdy webové API nebylo schopné odpovědět. Těchto kódu je vícero v každé skupině, ale na základě první cifry je možné určit skupinu o kterou se jedná. V rámci status kódu je i krátké slovní vyjádření chyby. Například pro úspěch je číselný kód 200 a vyjádření OK.

4.2.2 Koncept REST

[10] Koncept REST, také REST API je označení konceptu, který popisuje zpřístupnění služeb či zdrojů dat. REST využívá komunikaci skrze http. Využití http, včetně metod, které nabízí, dává možnost tvořit jednotné rozhraní z pohledu API s širokou možností využití. Při tvorbě API na základě konceptu REST je vhodné dodržení několika pravidel.

Jednotné rozhraní, kterým je dosažena podobnosti všech URL pro dané API. Jednotnou URL je zároveň vhodné využít pro veškeré požadované operace nad danou entitou, a to skrze využití různých http metod. Například adresa „/users“ je vhodná pro

získání všech uživatelů a vložení nového uživatele. Pro operace s konkrétním uživatelem je vhodné doplnit identifikátor „/users/{id}“, čímž lze využít různé http metody, pro dosažení všech potřebných operací. Odkazy je také vhodné tvořit v podobě množin, kde s každým lomítkem jdeme o stupeň níže. Například „/department/1/users/2“ pro získání uživatele s identifikátorem 2 z oddělení, které je označeno identifikátorem 1. Tímto způsobem je dosaženo CRUD operací nad daty.

[12] Oddělení klienta od serveru. Dalším pravidlem je nezávislost poskytovatele v podobě serveru s REST API a klientem, který bude dané API využívat. Není tedy vhodné, aby došlo k interakci jiným způsobem nežli komunikací skrze http.

Nestavový přístup (Stateless) je pravidlo, které udává, že server by neměl udržovat session pro daného klienta. Tudíž každý příchozí http dotaz, musí obsahovat všechna data potřebná ke zpracování. Server tedy neobsahuje žádná data o klientovi vyjma dat, která jsou uložena v databázi.

Ukládání dat do cache, by mělo být využito kdykoliv, kdy je to možné. Toto pravidlo platí pro stranu klienta i serveru. Důvodem je zvýšení výkonu na straně klienta. Každý dotaz by měl zároveň obsahovat informaci, zda jsou daná data vhodná pro uložení do cache.

Rozdělení do vrstev, označuje, že nelze předpokládat komunikaci klienta se serverem přímo mezi sebou. Předpokladem je, že daná komunikace projde skrze prostředníky a jejich počet není jistý. Klient i server tedy nemohou předpokládat, že se bude jednat o přímou komunikaci. Toto pravidlo je převážně tvořeno s předpokladem, že REST API, je využíváno převážně v architektuře mikroslužeb.

Zaslání spustitelného kódu, je pravidlo vedené jako volitelné. Běžně API odesílá statická data v podobě JSON souboru nebo v jiných formátech. V určitých případech je však možnost využít, spustitelného kódu. V těchto případech by se mělo jednat o kód na vyžádání.

4.3 Backend

[13] Backend je název pro část aplikace na serveru. Tato část aplikace, která může být i samostatnou aplikací nikoliv pouze částí, obstarává komunikaci s databází a manipulaci s daty. K porovnání s dříve popsanou architekturou MVC, jednalo by se o vrstvy model a controller. Služby backendu jsou zpravidla dostupné skrze http

požadavky. Pro tyto požadavky je velmi časté využití konceptu REST API. Backend je také využíván pro jiné účely, kupříkladu paralelní zpracování.

Backend lze tvořit za využití velkého množství programovacích jazyků, pro které jsou dostupné frameworky, které se zaměřují na tvorbu aplikací s tímto využitím. Mezi populární frameworky se řadí Spring pro Javu, ASP.NET pro C#, Node.js a Express pro JavaScript, Laravel pro PHP a další.

4.4 Frontend

[13] Frontend označuje část aplikace, která zajišťuje viditelnou část webu pro uživatele skrze webový prohlížeč. Frontend je napojen na backend aplikaci, čímž je zajištěna dostupnost požadovaných dat. Při porovnání s architekturou MVC, by se jednalo o vrstvu view. Samotný vzhled je velmi důležitý, jelikož nepřehledný web může odradit uživatele. Ke vzhledu webu se váže pojem user interface, který se zabývá návrhem a designem aplikace a zajišťuje celkovou vizuální podobu webu.

Frontend je tvořen s využitím jazyků HTML a CSS. Neméně důležitou částí je také JavaScript, který je vhodný pro napojení na backend a pro práci s daty. V rámci vývoje se často využívá frameworků nebo knihoven pro JavaScript. Nejpopulárnější knihovnou je React. Existuje jich však mnoho, mezi další náleží například Vue, Angular, Svelte a další.

4.5 Single-page aplikace

[14] Single-page aplikace (SPA) je webová aplikace, jejíž veškerý obsah je tvořený jedinou stránkou. SPA zastává funkci frontendu aplikace. Tímto způsobem dochází k oddělení prezenční vrstvy aplikace na jedinou stránku, která po načtení neaktualizuje celkovou stránku, ale pouze obsah částí případně vlastní komponenty, ve kterých je změna požadována.

SPA zpracovává vstupy od uživatele, které následně zpracuje prezenční logika. Prezenční logika označuje postup pro přípravu dat a samotnou komunikaci prezenční vrstvy se serverem. Komunikace bývá zpravidla skrze API rozhraní na serveru, není to však jediný způsob komunikace, další možnou komunikací jsou například webové sokety. Logika obsažená v SPA umožní zpracování obdržení dat na základě požadavku

a aktualizaci pouze žádoucího obsahu. Tento způsob odstraňuje odesílání celých HTML stránek, čímž je zmenšena náročnost síťového provozu, a také rychlost aktualizace dat.

Vzhledem k tomu nemají single-page aplikace nutnost obnovy v prohlížeči, jelikož nepřesměrovávají na novou stranu, ale pouze dochází k aktualizaci jejich obsahu. Ačkoliv k obnově stránek často dochází i u aplikací, které jsou způsobem SPA tvořeny, nejedná se totiž o striktně předepsané pravidlo, ale o možnost, která se nabízí. Zároveň na první pohled koncový uživatel, nemusí mít jistotu, zda došlo či nedošlo k obnově. Existují totiž způsoby, jak změnit část odkazu za lomítkem bez nutnosti obnovy stránky.

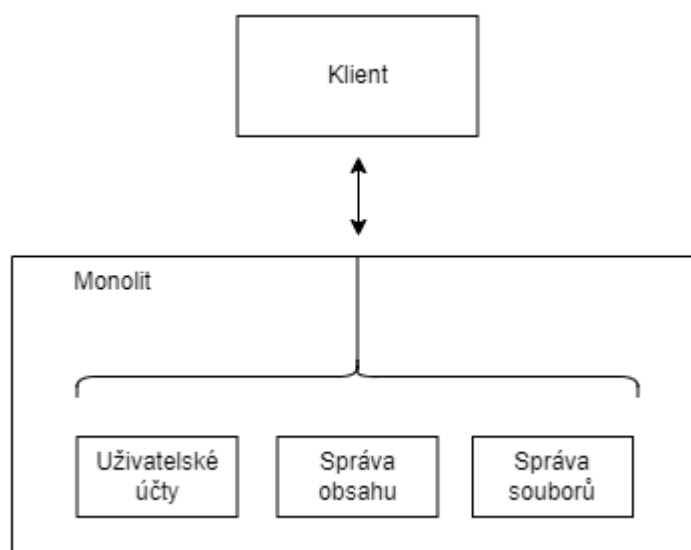
Hlavním prvkem při tvorbě SPA je využití skriptovacího jazyka JavaScript s technologií s názvem AJAX (Asynchronous JavaScript And XML), která zajišťuje výměnu dat s webovým serverem. AJAX je ovšem pojem, který označuje princip aktualizace části obsahu bez nutnosti obnovy stránky, technologie zároveň využívá další technologie, například samotné HTML a CSS pro zobrazení dat, DOM ve spojitosti s JavaScriptem pro jejich samotnou aktualizaci. DOM (Document Object Model) označuje API pro práci s HTML či XML dokumenty a prvky daného dokumentu, skrze toto rozhraní je možné aktualizovat jednotlivé elementy dokumentu například právě s využitím skriptovacího jazyka JavaScript. Samotné dotazy skrze XMLHttpRequest, který se využívá pro tvorbu a zpracování http požadavků. Jeho využití však nezávisí pouze na formátu XML, ale je možné využít jiné formáty, například JSON, samotný text a další. Aktuálně se jedná o technologii, která se sice využívá, ale zpravidla je využita jako součást knihovny či frameworku různého programovacího jazyka.

4.6 Architektura monolitu

[15] Architektura monolitu popisuje vývoj softwaru jako jednotného celku. Monolitem označíme software, který je vytvořen jako samostatný celek, obsahující veškeré požadované prvky.

Vývoj na základě monolitické architektury je možný v různých částech webové aplikace. Jako absolutní monolit bychom nazvali tzv. serverově vykreslované stránky, které fungují na principu generování uživatelského rozhraní na straně serveru a prohlížeč poté dostane samotnou HTML stránku, která se má uživateli zobrazit. Je také možné nahlížet na vývoj monolitu pouze v určité části celkového systému, příkladem může být rozdělení na backend a frontend. V takovém případě se nejedná o jeden

spustitelný software, který by vykonával veškeré potřeby, ale i přes to, je možné využít monolitickou architekturu pouze v případě backendu, tudíž kompletní backend je jeden software, který plní funkci několika služeb.



Obr. 2: Diagram architektury monolitu
Zdroj: Autor

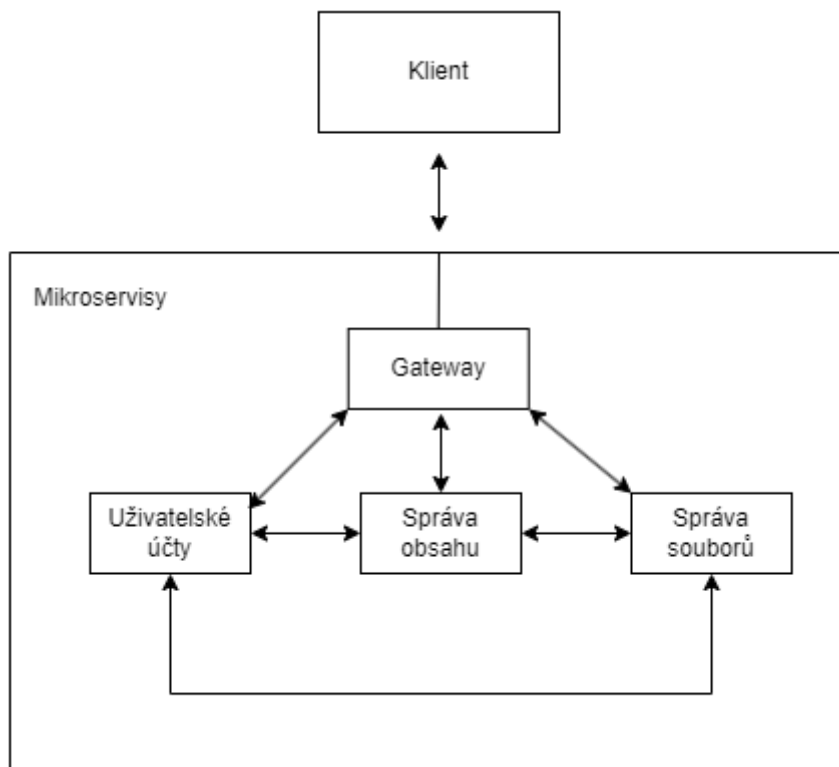
Monolitická architektura má řadu výhod. Výhodou je jednoduchost nasazení softwaru, což v tomto případě je spuštění jediné aplikace, která obsahuje kompletní funkcionalitu, která se od daného systému očekává. Další výhody jsou převážně z pohledu vývojáře, vzhledem k jednotnému systému je méně náročné takový systém podrobit testování a krokování. Zároveň umožňuje jednodušší opakované využití již vytvořeného kódu. V ohledu samotného programování může být výhodný či neutrální v případě jediného vývojáře nebo malého týmu, ale v případě velkého vývojového týmu, či několika menších skupin vývojářů pracujících na jediném systému dochází ke komplikacím například s tvorbou duplicitního kódu, který se musí následně vyřešit.

Monolit disponuje také nevýhodami. Jako jednou z hlavních nevýhod je riziko, že pokud bude jediná část monolitu obsahovat chybu, dojde k vypnutí celého systému. Další nevýhodou je také nutnost vypnutí a aktualizace celého softwaru, a to i v případě, že provedená změna byla minimální.

Monolit má i přes své nevýhody stále své uplatnění, v případě vývoje malým týmem či jednotlivcem, a za předpokladu, že systém nebude mít veliké vytížení je tato architektura vhodná.

4.7 Architektura mikroservis

[15] Architektura mikroservis, též mikroslužeb, popisuje vývoj systému rozděleného do jednotlivých částí, kde každá část může být jedna nebo více instancí.



Obr. 3: Diagram architektury mikroservis
Zdroj: Autor

Mikroservisy lze nazvat jako jednotlivé aplikace, každá z mikroservis obsahuje vlastní logiku, připojení na databázi, přístupové body a další. Při nasazení této architektury na server, je nutné projít kompletním procesem nasazení aplikace u každé mikroservisy, to samé platí při provedení malé změny. V případě dvou mikroservis, je nutná aktualizace obou mikroservis, zároveň v případě velké změny či kompletního přepsání jedné mikroservisy, dochází k aktualizaci právě této jedné, čímž není ochromena dostupnost ostatních prvků systému. Zpravidla dochází k použití brány (gateway), pro rozdělení komunikace, na konkrétní služby.

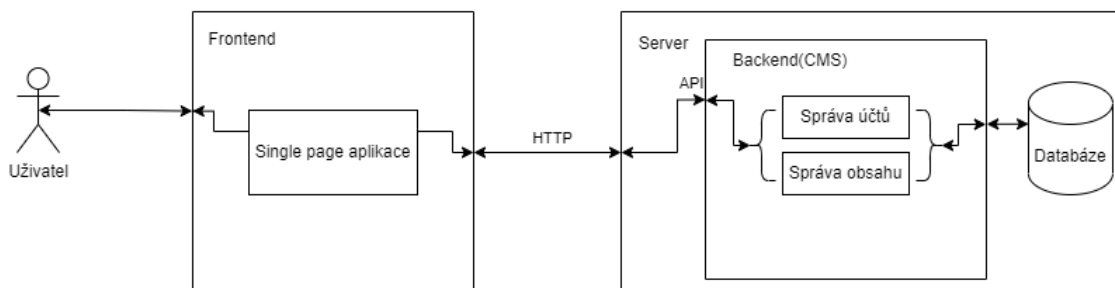
Výhod využití architektury mikroservis je několik, mezi zásadní náleží možnost škálování jednotlivých částí. Pokud je tedy jediná část přetěžována velikým provozem, je možné přidat další instanci té samé mikroservisy. Další výhodou je možnost aktualizace částí, bez nutnosti vypnutí celé aplikace. Z pohledu vývojáře dochází k nejvíce výhodám v moment, kdy se na vývoji podílí více skupin vývojářů. Pro tento

případ, dochází v praxi k přidělení konkrétní mikroservisy pro daný tým, případně týmy. Díky tomu dochází k rychlému vývoji konkrétní mikroservisy.

Nevýhod s architekturou mikroservis může být náročnost na provoz daného systému. Jelikož jednotlivé části mezi sebou mohou komunikovat, což se zpravidla děje při výměně dat, má tedy, oproti monolitické architektuře, další síťový provoz. Vzniká také náročnost na testování a odhalování chyb v celkovém systému. Vzrůstá náročnost na vedení celkového vývoje a komunikace mezi jednotlivými skupinami vývojářů při tvorbě API, jelikož mezi sebou různé částí mohou komunikovat. A také náročnost na samotné nasazení a spuštění systému.

4.8 Architektury použité pro vlastní řešení

Bylo představeno několik architektur a vývojových konceptů, které lze použít pro tvorbu různých systémů, včetně redakčního systému, na jehož tvorbu se práce zaměřuje. V rámci vlastního řešení, kterým je vývoj redakčního systému, bude využito rozdělení na backend a frontend. Tudíž nedojde k přímému použití architektury MVC, která v původní myšlence předpokládá server-side vykreslování stránek, tudíž i samotný server zajišťuje vytvoření výsledné stránky většinou s využitím šablon.



Obr. 4: Diagram architektury vlastního řešení
Zdroj: Autor

Frontend bude tvořen v podobě single-page aplikace, bude se tedy jednat o jedinou stránku, jejíž obsah bude aktualizován na základně podnětů od uživatele, což spustí dotaz na backend, odkud jsou poté obdržena data, pomocí kterých se obsah aktualizuje. Backend bude tvořen na základě monolitické architektury. Tato architektura byla zvolena vzhledem k jednoduchosti nasazení, vývoje a testování. Komunikace mezi backend a frontend částí aplikace bude tvořena skrze Rest API koncept, který využívá ke komunikaci http požadavky.

5 Technologie

Tato kapitola popisuje technologie, které byly zvoleny pro vývoj vlastního řešení. Obsahem jsou nejen popisy samotných technologií, které jsou využity k implementaci. Část kapitoly, je také věnována popisu vývojářských nástrojů, použitých v rámci řešení.

5.1 Backend – Spring

Tato část se věnuje popisu technologií použitých pro vývoj backend části aplikace. Konkrétně se jedná o programovací jazyk Java s využitím frameworku Spring.

5.1.1 Java

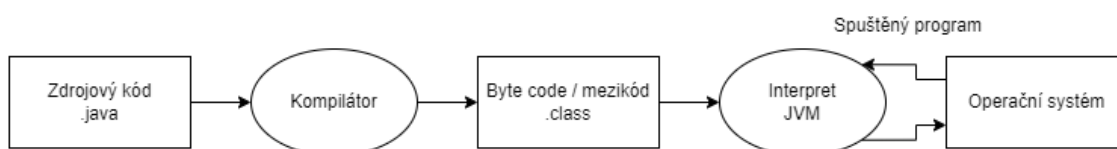
[16] Java jako programovací jazyk, je objektově orientovaný programovací jazyk. Cílem bylo vytvořit programovací jazyk, který má syntaxi podobnou jazyku C. Také bylo cílem vytvořit jazyk, který nabízí možnost tvořit různorodé aplikace do různých prostředí. Výsledkem bylo vytvoření jazyka, který je multiplatformní a nabízí možnost vytvářet programy s různým zaměřením. Java je často spojována s pojmem „Write Once, Run Anywhere“, který lze přeložit jako „Napiš jednou, spust' všude“. Tento pojem označuje, že díky virtuálnímu stroji, skrze který samotné aplikace fungují, je možné jednu vytvořenou aplikaci spustit napříč operačními systémy.

[16] *Java Virtual Machine* (JVM) pracuje jako virtuální stroj, skrze který je možnost spouštět aplikace tvořené programovacím jazykem Java. JVM napomáhá tomu, aby se jednalo o multiplatformní jazyk. Využití aplikace je tedy možné na zařízení, které obsahuje tento virtuální stroj, jež zajišťuje komunikaci s operačním systémem.

[16] Java je distribuována ve třech edicích, které jsou označovány jako Java Runtime Edition (JRE), jež je určena pro spuštění aplikací a obsahuje JVM, které je potřeba pro spuštění. Druhou edicí je Java Development Kit (JDK), které již podle názvu umožňuje vývoj v jazyce Java. Vývojářská edice obsahuje kromě JRE, také kompilátor, debugger, nástroje pro dokumentaci kódu, zobrazení statistik ohledně spuštěných programů, které jsou spuštěny skrze JVM a další. Poslední a rozsahem největší edicí je Software Development Kit (SDK). Jedná se o rozšířenější variantu JDK, jež navíc

obsahuje další software. Mezi tento software náleží například aplikační servery, debuggery.

[17] Java se také dělí do čtyř platforem na základě primárního využití a podle knihoven, které obsahuje. První edicí je Java SE, též Java Standard Edition. Tato standardní edice obsahuje veškerou základní vybavenost pro tvorbu aplikací. Disponuje knihovnamy pro síťové aplikace, tvorbu grafického rozhraní, zabezpečení, databázový přístup a další. Poskytuje jádro programovacího jazyka Java. Primárně se využívá pro tvorbu desktop aplikací. Druhá edice označená Java EE neboli Java Enterprise Edition, je nadřazena standardní edici. Poskytuje API pro tvorbu síťových aplikací, zpřístupňuje prostředky pro škálování aplikací. Její primární využití je při tvorbě webových aplikací. Třetí edicí je Java Micro Edition (ME). Poskytované API je podmnožinou Javy SE. Obsahuje však knihovny určené pro vývoj aplikací na mobilní telefony, což je také primární využitím. Poslední edicí je JavaFX, která disponuje prostředky pro tvorbu atraktivnějšího grafického rozhraní. Obsahuje knihovny, které pomáhají lépe zacházet s medii oproti například Javě SE. Jejím primárním využitím je tvorba klientských aplikací pro služby vytvořené s Javou EE.



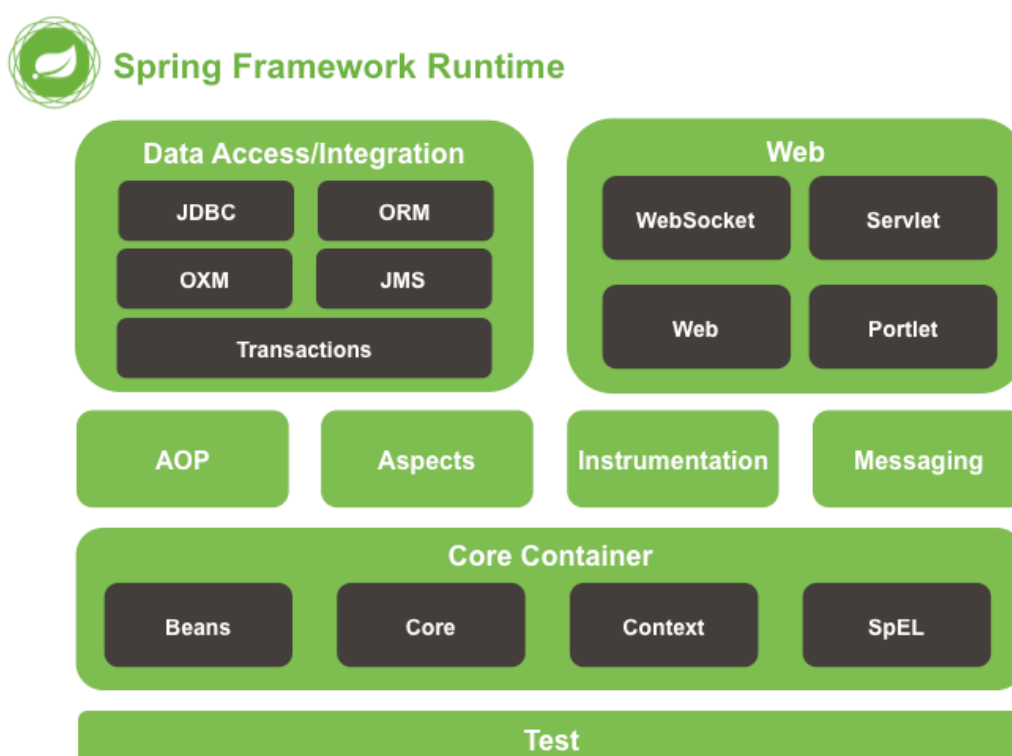
Obr. 5: Architektura jazyku Java a JVM
Zdroj: Autor

[18] Na výše uvedeném obrázku, lze vidět celou architekturu od vzniku zdrojového kódu až po samotné spuštění. Prvním krokem je vytvoření zdrojového kódu. Zdrojový kód programovacího jazyka Java se píše do souboru s příponou „.java“. Následujícím krokem je kompilátor, který již zobrazí chyby, které by mohl výsledný program obsahovat. Případně na ně upozorní a programátor má možnost tyto chyby odstranit. Po kompilaci zdrojových kódů je získán byte kód, též označovaný jako mezikód. Mezikód tvořený z jednoho zdrojového souboru obsahuje příponu „.class“. Pokud se kompiluje celá aplikace, která má být spustitelná a jedná se o více souborů, zpravidla dochází k zapouzdření do souboru s příponou „.jar“. Tento soubor je následně možné spustit, pokud zařízení disponuje JRE. Finálním krokem je samotné spuštění skrze JVM, které zajistí kompletní komunikaci s operačním systémem.

5.1.2 Spring

[8][19] Spring je framework (aplikační rámeček) pro tvorbu primárně webových aplikací. Pro automatizaci a správu samotného projektu využívá nástroje Maven nebo Gradle, skrze které je možné přidávání nových závislostí na knihovny. Zároveň obsahuje aplikační server Tomcat, který je určen pro spuštění webové aplikace. Tento server je vložený přímo ve Sprigu, díky čemuž je možné výsledné aplikace spouštět bez nutnosti složitějších konfigurací a spuštění na aplikačním serveru externě.

[19] Klíčovým aspektem pro celý framework je Dependency Injection (DI) a Inversion of Control (IoC). Tímto odpadá nutnost přímého vytváření vazeb mezi konkrétními třídami, ale skrze anotace, případně vynucení získání instance skrze konstruktory, které Spring automaticky při spuštění doplní instancí vhodné třídy. Celý framework velmi silně pracuje s anotacemi, pokud tedy chceme danou instanci doplnit je nutné, aby měla jednu z dostupných anotací.



Obr. 6: Spring architektura
Zdroj: [20]

Spring je dělen do několika modulů, díky čemuž je možné využití pouze částí, které jsou opravdu potřebné pro daný projekt. Vzhledem ke komplexitě celého jsou

samotné moduly ještě děleny do několika kategorií. Někdy je užito označení, že se jedná o několik modulů se submoduly. Toto rozdělení lze vidět na výše uvedeném obrázku.

Modul s názvem Core je automaticky obsažen v každém projektu, kde je framework využit. Tento modul zajišťuje doplnění skrze anotace, sledování kontextu (context), tvorbu a správu Bean. Bean je pojem, který dovoluje využít IoC a DI, jelikož se jedná o označení třídy, kterou je možné skrze DI doplnit do míst, kde je vyžadována.

[19] Dalším větším modulem jsou Data Access / Integration. Tento modul umožňuje práci s daty. Disponuje JDBC, díky čemuž máme možnost využít napojení na databáze a získávat či ukládat do nich data. ORM neboli Object Relational Mapping dostupný skrze některou z knihoven, například Hibernate, umožňuje objektově relační mapování. Jedná se o způsob popisu databázové entity jako obyčejné třídy, která je doplněná o anotace, díky čemuž je možné získávat z databáze přímo daný objekt, bez nutnosti ručního získávání dat z odpovědi na SQL dotaz. Modul také nabízí možnost využití JPA včetně repositářů, což je možnost, jak komunikovat s databází skrze jednoduchý interface, který je rozšířením již definovaného repositáře. Tímto jednoduchým způsobem máme veliké spektrum možností, jak poskládat vlastní dotazy na databázi, bez nutnosti tvořit samotný dotaz. JPA pouze pomocí názvu dokáže dotaz poskládat a provést.

5.2 Frontend – React

[21] ReactJS běžně jako React, je knihovna, která umožňuje pro vývoj použít například JavaScript nebo TypeScript, se zaměřením na tvorbu uživatelského rozhraní. Jedná se o open-source knihovnu, kterou vytvořila společnost Meta [22] (dříve Facebook).

[21] React je možné využít pro tvorbu single-page aplikací (SPA), další možností je nativní aplikace pro mobilní zařízení s využitím React Native. React také umožňuje server-side vykreslování webových stránek.

React umožňuje tvorbu tzv. komponent, které je následně možné využít jako prvek v rámci stránky. Tento princip poskytuje možnost opětovného využití jednotlivých komponent na více místech, bez nutnosti tvořit duplicitní kód. Vzhledem ke konceptu SPA (popsáno v kapitole 4.5 Single-page aplikace) je možné změnit aktuálně zobrazenou komponentu, případně pouze změnit obsah, který zobrazuje.

[23] Pro vytvoření komplexnější webové aplikace, je vhodné React obohatit o další knihovny. Jedná se například o knihovnu Axios [23]. Jedná se o open-source

knihovnu, která zastupuje http klienta při tvorbě požadavků. Axios disponuje možností konfigurace dotazů, rozděluje výsledek požadavku podle úspěšného provedení (http kód ve formátu 2XX) a neúspěšného. Práce s odpovědí na dotaz, je tedy přehlednější. Mezi další funkce, co Axios nabízí je konfigurace http headers, automatické převedení obsahu na JSON, pokud není definováno jinak, rozdělení jednotlivých http metod, které jsou dostupné skrze volání konkrétní metody.

[21] React využívá velká komunita, která zhotovuje open-source knihovny, které jsou využitelné společně s Reactem. V rámci vývoje webové stránky, je rozšířená knihovna Bootstrap [24]. Pro jednodušší využití možností této knihovny, vznikla knihovna React Bootstrap [25], která umožňuje využití prvků Bootstrapu jako komponent pro React. Podobných knihoven, které vychází z původní samostatné knihovny a následně jsou přepsány pro snadné využití v rámci Reactu, je několik, další využitou v projektu je React Charts [26].

5.3 Databáze – MySQL

[27] Databáze je určena k organizovanému uchování dat. Databáze je zpravidla řízena DBMS (Database Management System), česky jako systém řízení báze dat. Jedná se o rozhraní mezi uloženými daty a aplikací nebo uživatelem, umožňuje provádět operace s daty, ale také definici jejich uložení.

[28] Relační databáze reprezentuje data jako řádky v rámci tabulky, sloupec definuje konkrétní atribut. Relace je napojení jednotlivých řádků na řádky jiné, toto napojení je možné napříč tabulkami. Pro identifikaci jednotlivých řádků je potřeba primární klíč, ten je zpravidla označen jako ID (identifikátor). Musí se jednat o unikátní hodnotu pro každý záznam v rámci tabulky. Samotná relace je poté definována pomocí primárního klíče na jedné straně a cizího klíče, který je identifikací konkrétního záznamu v jiné tabulce.

[29] MySQL je open-source relační systém řízení báze dat. Jedná se o jeden z nejpoužívanějších databázových systémů na světě. Pro manipulaci s daty využívá dotazovací jazyk SQL. Samotnou databázi je možné ovládat z příkazové řádky pomocí textových příkazů. K dispozici je ale také několik nástrojů, které zpřístupňují databázi skrze uživatelské rozhraní. Mezi uživatelské rozhraní, které je možné využít náleží například phpMyAdmin [30]. Jedná se o webový open-source nástroj, vytvořený

v jazyce PHP. Mezi desktopové aplikace zpřístupňující uživatelské rozhraní lze zařadit například MySQL Workbench [31].

5.4 Vývojářské nástroje

Tato kapitola se věnuje popisu nástrojů, které byly využity při vývoji aplikace.

5.4.1 Vývojové prostředí

[32] Vývojové prostředí (IDE) je software určený pro programátory. Vývojové prostředí obsahuje zpravidla několik částí, kterými jsou například editor, kompilátor, debugger. V dnešní době však nástroje obsahují zpravidla více funkcí, kterými může být možnost napojení na databázi. Celkově je určeno pro usnadnění práce programátora. Jednotlivé vývojové prostředí, mají zpravidla seznam podporovaných programovacích jazyků. Některá prostředí jsou primárně určena pouze na několik, či pouze jediný jazyk. Existují ale také nástroje, které podporují velké množství jazyků současně.

[32] Editory ve vývojových prostředích dnes obsahují „našeptávač“, který po napsání začátku požadovaného příkazu nabídne doplnění na základě oficiální dokumentace daného programovacího jazyka, případně na základě již vytvořeného kódu. Při zadání dané třídy či metody, kterou „našeptávač“ ukáže, se zobrazí také její dokumentace, díky které lze jednoduše zjistit jak metodu či třídu správně použít a případně jaké jsou její atributy a mnoho dalších informací. Editory také obsahují zobrazovač syntaxe, který podle barev formuje vizuální rozdělení napsaného kódu podle syntaxe daného jazyka. V rámci toho, je také častá kontrola na základě syntaxe a zvýraznění chyb již při psaní kódu, díky čemuž je možné se vyvarovat řadě chyb, které by mohly být způsobeny zápisem nikoliv, však špatně myšlenou logikou. V rámci logiky, jsou nástroje také schopné předpokládat případné chyby a upozornit na ně, nelze však spoléhat na odhalení všech chyb v logice, ale například pokud by mělo u kódu dojít k zacyklení, či požadujeme operace ve fázi, do které není možné se v rámci programu dostat, vývojové prostředí zobrazí upozornění.

Debugger je nástroj, který je určen k testování a hledání chyb v kódu. Disponuje však zpravidla více funkcemi, jako je krokování aplikací při jejich spuštění podle jednotlivých bodů, nahlížení do proměnných a zobrazení jejich hodnot v daném kroku,

zobrazení systémových prostředků a jejich využití. Díky těmto funkcím je možné odladit aplikace a zbavit je chyb, případně snížit jejich systémové nároky.

Jedná se však o více komplikovaný software, který umožňuje velké množství přizpůsobení, včetně klávesových zkratk, které jsou dnes v nástrojích k dispozici. Ale nejen s využitím klávesových zkratk je běžná možnost automatického generování částí kódů, které neobsahují téměř žádnou logiku, ale dochází k jejich opakovanému využití, například metody tříd na získání hodnot proměnných či nastavení hodnot proměnných. Funkcí, které usnadňují či zrychlují práci, je skutečně mnoho, může mezi ně patřit například možnost úpravy více řádků, možnost přejmenování proměnné včetně všech míst, kde je použita, vyhledávání tříd, či možnost přesměrování se na třídu, které je použita, generátor předem upravených šablon a další. Nedílnou součástí je také možnost rozšíření celého IDE o vlastní doplňky či o doplňky, které již někdo vytvořil. Díky tomu je možné si celé prostředí ještě více upravit a doplnit o funkce, které nejsou integrovány v základní verzi od tvůrců.

Na trhu je mnoho různých vývojových prostředí, pro jazyk Java je to například IntelliJ Idea [33] od české společnosti JetBrains, Netbeans [34], Eclipse [35] a další.

5.4.2 Verzovací systém

[36] Během vývoje aplikace je verzování nezbytnou součástí. Jedním z nejznámějších a nejvíce používaných nástrojů pro verzování je Git [37], který je určen k průběžnému ukládání změn. Změny se z pravidla týkají zdrojových kódů. Sledování změn je však možné využít pro téměř jakýkoliv druh souboru. Git nabízí tvorbu vzdáleného repozitáře, na které je možné nahrávat změny z lokálního repozitáře a provádět synchronizaci oběma směry. Pro zaznamenání změn je použit příkaz a označení „commit“. Tomu lze nastavit několik parametrů, například autor a textový popis změn. Změny v souborech, které jsou již obsažené v commitu je možné nahrávat na vzdálený repozitář, tomuto procesu se říká „push“. Ten je možné provést v moment, kdy je lokální i vzdálený repozitář úspěšně synchronizovaný. Tyto změny je možné nahrát do tzv větve. Git umožňuje tvorbu větví, díky tomu je možné pracovat paralelně na jednom projektu ve více lidech. Tyto větve je poté možné sloučit, pro proces slučování se používá výraz „merge“. Nedílnou součástí je také možnost návratu k předchozím verzím, případně odebrání změny (commitu) z historie, případně návrat ke přímo zvolenému stavu. Vracení k velmi staré změně, může být problematické kvůli

vypořádání s dalšími změnami. Git je však nástrojem složitějším, obsahuje více funkcí, zároveň je většina z nich parametrická, tudíž je jeho použití velmi obsáhlé, avšak dnešní vývojová prostředí zpravidla obsahují Git, tudíž je možné provádět velké množství příkazů skrze grafické rozhraní. Služby nabízející vzdálené repozitáře jsou dnes například GitHub [38], GitLab [39]. Služby nabízejí zdarma hostování vzdálených repozitářů. Tyto repozitáře jsou veřejné nebo soukromé. Služby často nabízejí další služby či funkce související s prací v rámci repozitářů, některé z těchto funkcí mohou být zpoplatněny.

5.4.3 Nástroje pro Api testování

[40] Nástroj pro API testování funguje převážně k tvorbě a zasílání http požadavků na server. Tento nástroj lze využít k testování backend serveru. Těmito nástroji jsou například Postman [41], Insomnia [42], které jsou primárně zaměřené na tento typ testování. Klíčové funkce pro testování API, obsahují i některá vývojová prostředí, využití těchto nástrojů tedy není nutné. Jedná se pouze o možný doplněk k vývojovým nástrojům. Umožňují vytvářet dotazy, které je možné modifikovat a upravovat podle potřeby. Skrze tyto úpravy je možné například posílat dotazy včetně ověření, obsahu dotazu, vlastních hlaviček nebo URL parametrů. Umožňují zasílání například obsahu (body) v různých podobách jako je JSON, Multipart Form, XML, binární soubor nebo například prostý text. Stejně je to u nabídek při zasílání ověření, je možné využít základní Basic Auth, OAuth, tokeny a další. Nástroj po položení dotazu zobrazí přehledně návrat včetně kódu odpovědi a všech obsažených informací. Tyto nástroje umožňují tvorbu více dotazů a jejich uchování pro opětovné volání.

6 Návrh a vývoj aplikace

Tato část se věnuje vytvořené aplikaci, včetně implementace vybraných problematik tvořeného redakčního systému. Zahrnuje pohled na principy fungování a struktury frontend a backend části.

6.1 Představení aplikace

Aplikace je vytvořena pomocí architektur popsaných v předchozí kapitole. K vývoji posloužili technologie popsané v předešlé kapitole. Samotná aplikace je přiložena v rámci elektronické přílohy.

Aplikací je označen redakční systém, který obsahuje základní potřebné funkce. Mezi tyto funkce se řadí správa obsahu administrátorem, uživatelské účty, komentáře. Tyto funkce jsou více popsány dále v práci. Obsahuje také možnost anket, u této funkce, se jedná o funkci, kterou si majitelé blogů, vyžadují v dotazníku, který je součástí práce.

Spuštění aplikace je popsáno v přiložených souborech. Veškeré části je možné snadno spustit skrze Docker [43]. Vzhledem k tomu, je aplikaci možné skrze virtualizaci spustit ve všech operačních systémech, které Docker podporují.

Aplikace je vhodná pro osoby, které hledají nástroj pro správu osobního blogu. Aplikace obsahuje funkce, které byly přidány na základě dotazníku.

Nedostatky dle dotazníku	Výsledná aplikace
Úprava designu bez programování	Funkce není obsažena
Problém s editorem článků	Editor článků je plně funkční
Spam komentářů	Odstraněno, vzhledem k nutnosti mít uživatelský účet
Ankety	Přidány v rámci článku
Malá možnost zpětné vazby	Hodnocení, ankety, komentáře
Výhody dle dotazníku	Výsledná aplikace
Jednoduchost na správu	Přehledná část pro správce, včetně nastavení
Statistiky	Pro správce, o článku, uživateli, či obecné
Napojení na další služby	Není
Napojení na sociální síť	Není
Možnost vlastního účtu pro návštěvníky	Ano, včetně nastavení, obrázku a dalších informací
Možnost úpravy vzhledu, či vytvoření vlastního	Minimální, případně možnost vytvořit vlastní frontend
Možnost sdílet různorodý obsah	Ne, pouze obrázky v galerii, či článků
Rychlost	Ano, vzhledem k SPA konceptu

Tabulka 7: Funkce dle dotazníku a výsledné aplikace

Zdroj: Autor

Na výše uvedené tabulce jsou viditelné funkce z pohledu dotazníku. V první části jsou nedostatky dotazníku oproti výsledné implementaci ve výsledné aplikaci. Druhá část tabulky zobrazuje výhody dle dotazníku a oproti tomu, zda jsou ve výsledné aplikaci.

6.2 Popis aplikace

Vytvořený redakční systém disponuje funkcemi pro fungování osobního blogu. Zároveň obsahuje také některé funkce, které dle dotazníkového šetření stávajícím systémům chybí, či nejsou vhodně implementovány. V příloze bakalářské práce je fotodokumentace vybraných částí finálního systému, vzhledem k množství pohledů.

Výsledná aplikace dělí uživatele do tří skupin. Dělení je na nepřihlášeného uživatele dále jako návštěvník, přihlášeného uživatele dále jako uživatel a na správce. Uživatel je opatřen rolí „USER“. Správce je také uživatel, ale navíc disponuje rolí „ADMIN“. Správce má tedy stejné možnosti jako uživatel, a zároveň další práva díky roli admina. Podle tohoto rozdělení je vytvořena následující tabulka.

Role	Článek + Anketa ¹			
	Zobrazit	Vytvořit	Změnit	Odebrat
Návštěvník	Ano	Ne	Ne	Ne
Uživatel	Ano	Ne	Ne	Ne
Správce	Ano	Ano	Ano	Ano
Role	Hlas v anketě			
	Zobrazit	Vytvořit	Změnit	Odebrat
Návštěvník	Dle nastavení	Ne	Ne	Ne
Uživatel	Ano ²	Ano	Ano	Ne
Správce	Ano	Ano	Ano	Ano
Role	Komentář			
	Zobrazit	Vytvořit	Změnit	Odebrat
Návštěvník	Ano	Ne	Ne	Ne
Uživatel	Ano	Ano	Ne	Vlastní
Správce	Ano	Ano	Ano ³	Vlastní
Role	Hodnocení			
	Zobrazit	Vytvořit	Změnit	Odebrat
Návštěvník	Dle nastavení	Ne	Ne	Ne
Uživatel	Dle nastavení	Ano	Ano	Ano
Správce	Ano	Ano	Ano	Ano

Tabulka 8: Možnosti z pohledu rolí ve výsledné aplikaci
Zdroj: Autor

Tabulka výše popisuje dostupné operace nad základním obsahem vytvořené aplikace z pohledu rolí. Definice rolí vychází z popisu, před tabulkou. K poznámce 1 z tabulky, anketa nemusí být součástí článku, článek je možné vytvořit i bez ankety. Důvodem společného popisu je fakt, že operace na základě role jsou totožné. Poznámka 2, u uživatele v rámci zobrazení hlasu v anketě, vlastní hlas je možné zobrazit vždy. Celkové výsledky ankety je možné zobrazit na základě nastavení dané ankety. Správce

má možnost zobrazit i celkové hlasy kdykoliv skrze statistiky. Poznámka 3, u role správce v části se změnou komentáře, správce nemá možnost měnit vlastní komentář ani jiného uživatele. Správce ale může označit komentář jako nevhodný, ten bude dále dostupný a viditelný pro všechny skupiny rolí. Pro zobrazení nevhodného komentáře musí být nejprve potvrzeno, že si chce návštěvník či uživatel zobrazit nevhodný komentář. Každá skupina rolí má také k dispozici vyhledávání článků.

Uživatel dle dřívějšího popisu (registrovaný a přihlášený) má viditelný profil. Na profilu uživatele je možné zobrazit informace o uživateli, například datum narození, datum a čas registrace, email, popis, hodnocení u článků, komentáře u článků a pohlaví. Některé z těchto informací je možné měnit, převážně doplňkové informace (datum narození, pohlaví ...). U všech informací, nejen doplňkových, je možné nastavit úroveň soukromí. Každý z uživatelů má tedy možnost si nastavit, které informace budou veřejné (bez ohledu na roli), soukromé (viditelné pouze vlastníkem účtu a správcem) a pouze pro přihlášené. Uživatelský profil také obsahuje profilový obrázek, který je možné zadat již při registraci. Veškeré tyto informace, včetně hesla, má uživatel možnost změnit v nastavení svého profilu.

Možnosti operací s článkem dle rolí, již byly představeny. Při tvorbě článku má správce možnost využít hned několika funkcí. V rámci tvoření či úpravy stávajícího článku může vyplnit následující informace: nadpis, stručný popis, viditelnost (zobrazení), obrázek, text, případně anketu. Stručný popis má možnost vytvořit ručně, případně využít automatický popis. Ten je tvořen ze začátku textu článku. Obrázek je jako miniatura pro článek. Správce má skrze editor zpřístupněné možnosti, jak vložit obrázky i do textu samotného článku. Editor zároveň zpřístupňuje základní formátování, nadpisy, seznamy, vložení odkazu, odsazení a další. Pokud je k článku vytvořena také anketa, správce definuje počet a text jednotlivých odpovědí a povolí či zakáže zobrazení výsledků u ankety. Pokud by například došlo k překlepu při psaní odpovědi, a již by anketa měla platné hlasy pro danou odpověď, má správce stále možnost změnit text, či odebrat jiné odpovědi. Každá možnost odpovědi má unikátní identifikátor, díky čemuž je rozpoznatelná bez ohledu na text. Možnost článek nechat skrytý lze využít k sepsání více článků v jeden den, či denní dobu a následně články zveřejňovat. Články je možné nastavit více způsoby, ty jsou však přístupné pro všechny články společně, a to skrze nastavení blogu.

Správce má vůči uživatelům možnost měnit jejich role. Zde je pojmem role myšleno „USER“ a „ADMIN“. Má tedy možnost dalšímu uživateli udělit stejná práva jako

má on. Stejně je to s odebráním role správce, ta je však nepřístupná v souvislosti s přihlášeným správcem. Toto omezení je z důvodu možného odebrání vlastní role správce vlivem nepozornosti.

Nastavení v části pro správce, je rozděleno do tří skupin. První skupinou je nastavení celého blogu. Prvky z celkového nastavení jsou využity napříč celým blogem. Druhou skupinou je nastavení galerie. Třetí skupinou je nastavení stránky o tvůrci blogu.

Obecné nastavení blogu dává možnost definovat název blogu. Název blogu je následně využit v názvu záložky prohlížeče a také je zobrazen na samotné stránce v menu. Dalším prvkem je možnost změny ikony (viditelné u záložky v prohlížeči). Poté je k dispozici obecné nastavení článků. Je možné nastavit, zda má být viditelný počet zobrazení u článku a počet hodnocení. Dále je nastavení času, díky editoru a nápovědě, je možné nastavit formát, kterým se bude zobrazovat čas včetně datumu napříč celým blogem. S nastavením času také souvisí zobrazení data a času u článků. Správce má možnost zvolit, zda se bude zobrazovat datum a čas vytvoření, poslední úpravy, či oba současně. Správce má také možnost nastavit podrobnosti o patičce stránky. Nejen text, ale také barvu pozadí a barvu textu. Poslední částí celkového nastavení je věnována stránkám o tvůrci a galerii. Zde je možné definovat text v nabídce, který odkazuje na tyto stránky a zároveň zda mají být přístupné či nikoliv.

Nastavení stránky o tvůrci dává možnost využít editor, který je totožný s editorem článků. Výhodou je stejné ovládání a stejné funkce. Skrze tento editor si může správce vytvořit stránku, kde o sobě jako o tvůrci předá veškeré informace. Poté je možnost také přidat sociální síť tvůrce. Nechybí ani možnost ukázky, jak bude výsledná stránka vypadat.

Posledním nastavením je galerie. Jedná se o přehledné a jednoduché nastavení, kde má správce možnost nahrát nové fotografie, odebrat současné, či si je zobrazit.

Správce má také k dispozici statistiky týkající se obsahu. Statistiky využívají různé grafy, případně dodatečné informace v podobě tabulky. Veškeré grafy jsou interaktivní a je možné zobrazit či skrýt jednotlivé informace v grafu. U každého zobrazení je také možné zvolit časové rozmezí, které definují období pro zobrazená data. Statistiky pro konkrétní uživatele či články je možné vyhledat, či zobrazit ze seznamů, které má správce k dispozici. Zároveň správce u zobrazení profilu či článku má možnost zobrazení statistik pro daný obsah skrze nabídku možností pro správce.

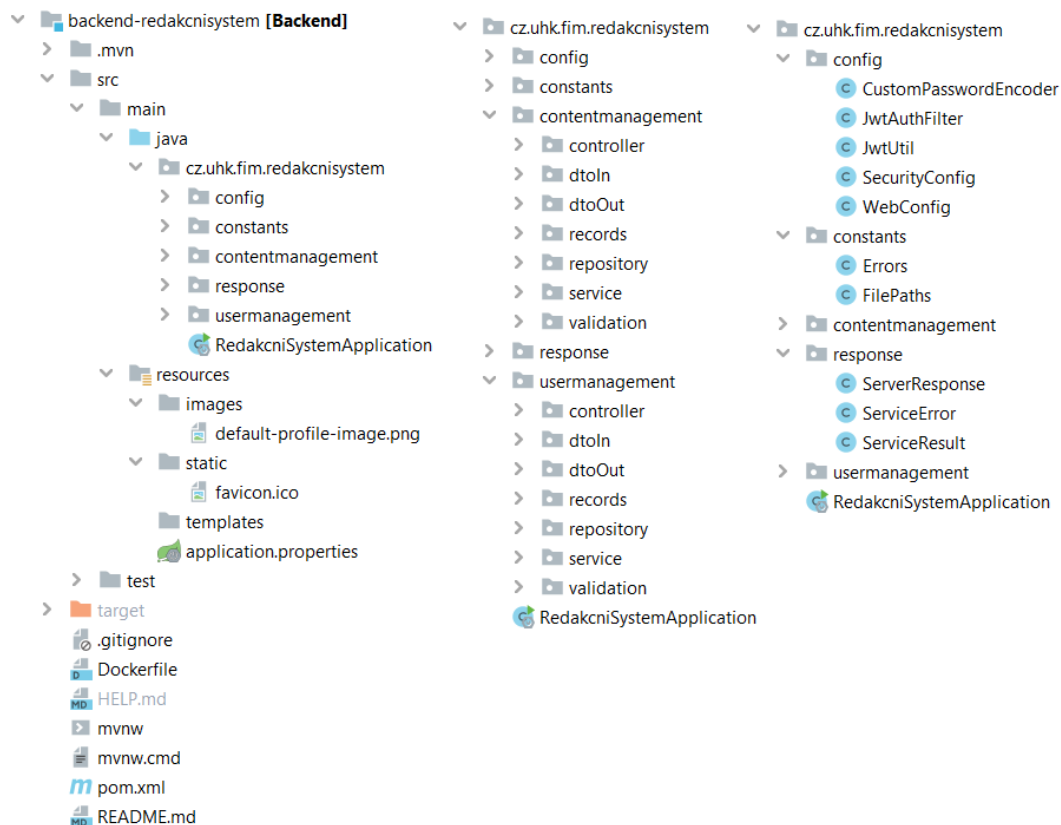
Jednou z možností je zobrazit obecné statistiky celého blogu. Obecné statistiky obsahují dvě části, jednu pro články a druhou pro uživatele. Část s články zobrazuje počty nových hodnocení, komentářů a zobrazení celkem u všech článků za daný den. U uživatelů je možné zobrazit počet unikátních přihlášení, počet registrací a počet přihlášení za daný den.

Poté následují statistiky pro konkrétní článek či uživatele. V souvislosti s článkem jsou viditelné celkové údaje k danému dni ohledně počtu komentářů, hodnocení a zobrazení. Tyto informace jsou zároveň na druhém grafu viditelné na základě denního přírůstku. Dále je viditelný graf s počtem hodnocení podle hodnoty. Ostatní informace jsou zobrazené v tabulce. Ta obsahuje základní informace jako je název, datum a čas přidání a poslední úpravy, aktuální hodnocení, počty komentářů a hodnocení a kolikrát je aktuálně článek uložen v seznamech uživatelů. Pokud zobrazený článek disponuje anketou jsou v tabulce viditelné údaje o anketě. Tyto údaje jsou název, počet hlasů, jednotlivé možnosti včetně počtu hlasů a přepočtu na procenta. Výsledek hlasování je také zobrazen grafem. Ze statistik je také možnost rychlého přesunu díky odkazu na úpravu článku, či jeho zobrazení.

Statistiky týkající se uživatelů zobrazují skrze graf, nové komentáře, hodnocení, zobrazení článků a unikátní přihlášení. Dalším grafem je graf s celkovým počtem komentářů a hodnocení. V rámci dalších informací zobrazených v tabulce, je možné vidět věk uživatele, datum narození, průměrné hodnocení v rámci všech článků, role, email, datum a čas registrace, pohlaví a počet uložených článků. Opět je zde možnost přejít na profil uživatele.

6.3 Backend

Pro vývoj backend části, byl využit Spring framework, samotný framework je popsán v kapitole 5.1 Backend – Spring.



Obr. 7: Struktura backend části projektu
Zdroj: Autor

Podle obrázku výše je viditelné rozložení backend části. Rozdělený je na pět hlavních balíčků. Tyto balíčky lze rozdělit do tří skupin, obsluha a zajištění práce s obsahem, uživateli a podpůrné balíčky potřebné napříč celým systémem.

Balíčky usermanagement a contentmanagement, mají podobnou další strukturu, která odděluje jednotlivé části. Popis jednotlivých balíčků:

- controller – obsahuje kontroléry, které jsou určeny ke zpracování požadavku, více dále v textu
- dtoIn, dtoOut
 - DTO (Data Transfer Object) fungují pro definici vyžadovaných dat v dotazu, případně odpovědi
 - oddělení in a out, rozděluje na vstupní a výstupní
 - zpravidla se jedná o záznamy z databáze, bez vybraných atributů, či složení různých atributů z různých záznamů
- records – obsahuje objekty, které slouží k definici záznamů v databázi
- repository – obsahuje interface, pro každou tabulku v databázi
 - v rámci rozhraní, je možné definovat různé požadavky

- implementaci zajišťuje Hibernate
- service – obsahuje interface a následně implementaci, potřebných služeb.
 - hlavní logika požadované operace
 - pracuje se všemi ostatními balíčky
- validation – obsahuje třídy pro validaci vstupních dat

Podpůrné balíčky, mezi které náleží config, constants a response mají využití na různých místech. V pravé části výše uvedeného obrázku je viditelný obsah jednotlivých balíčků. Config je určen k nastavení, konkrétně třída SecurityConfig funguje pro nastavení tzv. SecurityFilterChain, jedná se o nastavení zabezpečení pro příchozí dotaz. Constants obsahuje třídu s opakovaně používanými chybami při vykonávání dotazu, FilePaths pro změnu cesty k souborům v rámci uložení. Balíček response definuje třídy využívané při odpovědích na http požadavky, více popsáno dále v textu.

Důležitou částí je také resources, která obsahuje v rámci složky images, výchozí obrázek pro uživatelský profil, výchozí ikonu v rámci složky static. Dále také konfigurační soubor application.properties, který definuje nastavení pro komunikaci s databází, jako je jméno, heslo, URL k databázi, ale také maximální velikost požadavku a další.

```

@Bean
public SecurityFilterChain securityFilterChain(HttpSecurity httpSecurity) throws
Exception {
    httpSecurity
        .cors().and()
        .csrf().disable()
        .authorizeRequests()
        .antMatchers(HttpMethod.POST, publicAuthEndpoints).permitAll()
        .antMatchers(HttpMethod.GET, publicEndpointsGet).permitAll()
        .anyRequest().authenticated()
        .and()
        .sessionManagement()
        .sessionCreationPolicy(SessionCreationPolicy.STATELESS)
        .and()
        .addFilterBefore(jwtAuthFilter, UsernamePasswordAuthenticationFilter.class);
    return httpSecurity.build();
}

```

Zdrojový kód 1: Nastavení řetězce filtrů pro příchozí dotazy
Zdroj: Autor

Při pohledu na předešlý zdrojový kód, je vidět vytvoření zmíněné SecurityFilterChain, jedná se o způsob zabezpečení, skrze který je možné nastavit

dotazy z pole `publicAuthEndpoints` pro metody `POST` a `publicEndpointsGet` pro metodu `GET`. Toto zajistí, že pouze dotazy podle prvků v těchto polích a dané metody jsou přístupné bez autentizace. Zároveň také přidává další filtr `jwtAuthFilter`, který je vytvořen ve třídě `JwtAuthFilter`, jedná se o metodu, která z hlavičky `http` dotazu získá `JWT Token`, který následně provede ověření a zjistí, zda uživatel, kterému byl klíč vydán existuje. Pokud by token vypršel, nebo uživatel neexistoval, případně by byla autentizace vyžadována, ale hlavička neobsahovala token, dotaz dále nepokračuje, ale vrací příslušnou chybu.

Pokud je autentizace úspěšná, nebo není potřebná, pokračuje dotaz podle tzv mapování (mapping), ke kontroléru. Příkladem kontroléru je `AuthController` z balíčku `usermanagement`, který je určen k registraci, přihlášení a pro tokeny.

```

@RestController
@RequiredArgsConstructor
@RequestMapping("api/v1/auth")
public class AuthController {

    private final AuthService authService;

    @PostMapping(
        value = "/registration",
        consumes = MediaType.MULTIPART_FORM_DATA_VALUE)
    public ResponseEntity<ServerResponse<String>> registration(
        @RequestPart(name = "userData")
            UserRegistrationIn userRegistrationIn,
        @RequestPart(name = "image", required = false)
            MultipartFile image) {

        return ServerResponse.createResponseEntity(
            authService.registration(
                userRegistrationIn,
                image));
    }

    @Secured({"ROLE_USER"})
    @GetMapping("/token")
    public ResponseEntity<ServerResponse<UserTokenOut>> token(
        @RequestHeader(HttpHeaders.AUTHORIZATION)
            String tokenHeader) {

        UserTokenOut userTokenOut = UserTokenOut.builder()
            .token(JwtUtil
                .refreshTokenFromHeader(tokenHeader))
            .username(JwtUtil
                .extractUsernameFromHeader(tokenHeader))
            .build();
        return ServerResponse.createResponseEntity(
            ServiceResult.createServiceResult(
                userTokenOut,
                HttpStatus.OK));
    }
}

```

Zdrojový kód 2: Příklad kontroléru, AuthController
Zdroj: Autor

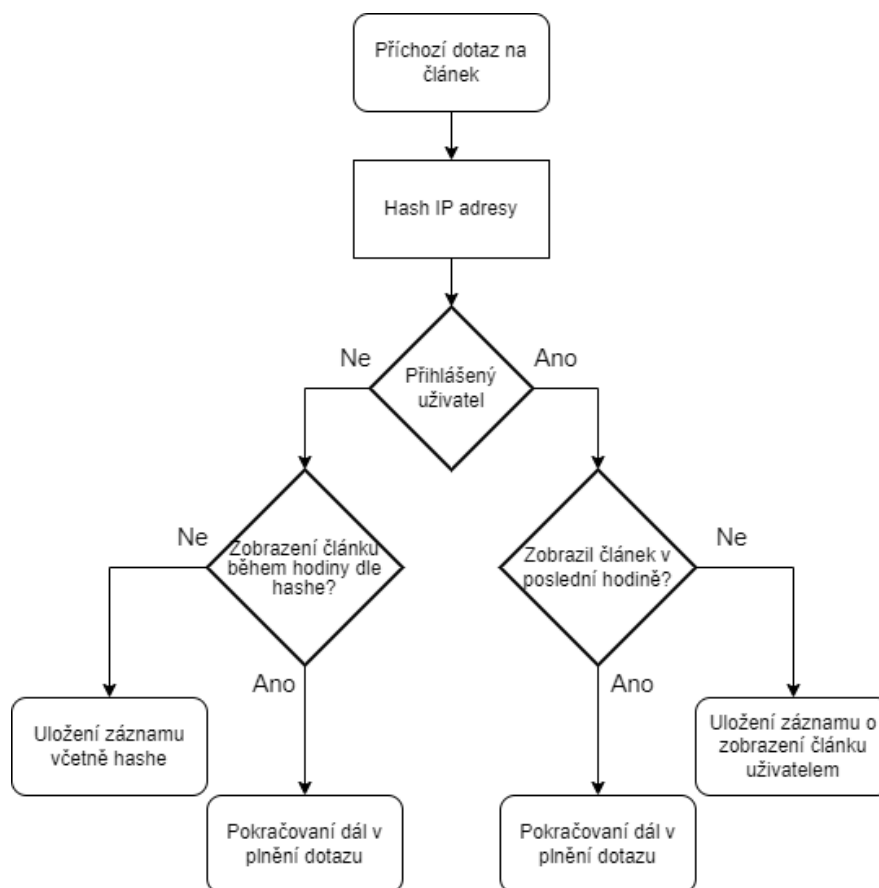
Výše uvedený zdrojový kód ukazuje princip definice URL cesty pro dotaz skrze anotace. Využití anotace nad definicí třídy zajistí základní cestu pro celý kontrolér, dále zde je viditelná anotace `@RestController`, která třídu přímo definuje jako kontrolér a anotace `@RequiredArgsConstructor` jež je z knihovny Lombok. Tato anotace automaticky vygeneruje konstruktor pro objekty, které jsou final. Knihovna Lombok

však nabízí i další užitečné anotace. Většina těchto anotací je pouze náhradou za základní metody, jako jsou konstruktory, přístupové metody.

U jednotlivých dotazů lze vidět anotaci `@xxxMapping`, tato anotace funguje k mapování metody pro danou cestu, a podle názvu anotace také pro danou http metodu. Metoda obsahuje návratovou hodnotu, zde lze vidět `ResponseEntity`, která je generická, parametr je tělo odpovědi na http požadavek. Tím je v případě aplikace `ServerResponse`, který má parametr tělo podle své generičnosti a seznam chyb jako `List ServiceErrorů`. Tělo je zpravidla třída `ServiceResult`, která obsahuje buďto generický parametr, případně zmiňovaný seznam chyb. Tento postup je volen ve většině aplikace až na výjimky, jako je například návrat obrázku. Další anotací využívanou pro dotazy je `@Secured`. V rámci této anotace je definována minimálně role, kterou musí podle tokenu uživatel mít. Pokud je vyžadována role `USER`, je tento dotaz oprávněn využít i `ADMIN`. V rámci atributů metody je možné získat tělo http požadavku, jeho část, či další atributy z http požadavku například hlavičku.

Dotaz poté zpravidla pokračuje do požadované služby (`Service`), která je definována jako interface a následně implementována. Zde dochází k samotné logice a následně je vrácen `ServiceResult`. Ten je vrácen i dříve, nežli dojde ke splnění dané akce, například pokud daný obsah neexistuje, vstupní objekt obsahuje nevalidní data, či jiný případ, kdy je žádoucí vrátit chybu, či více chyb. V rámci služby dochází k využití repository, pro práci s daty v databázi, případně s třídami z balíčku `validation` pro validaci dat.

Jedna z řešených problematik byla ohledně počítání zobrazení článku. Prvotní myšlenka započítat každý dotaz, by mohla způsobit záměrné dotazování, či obnovování stránky, které by se počítalo jako přečtení článku. Aktuálně tedy systém pro započtení nového zobrazení sleduje několik podmínek.



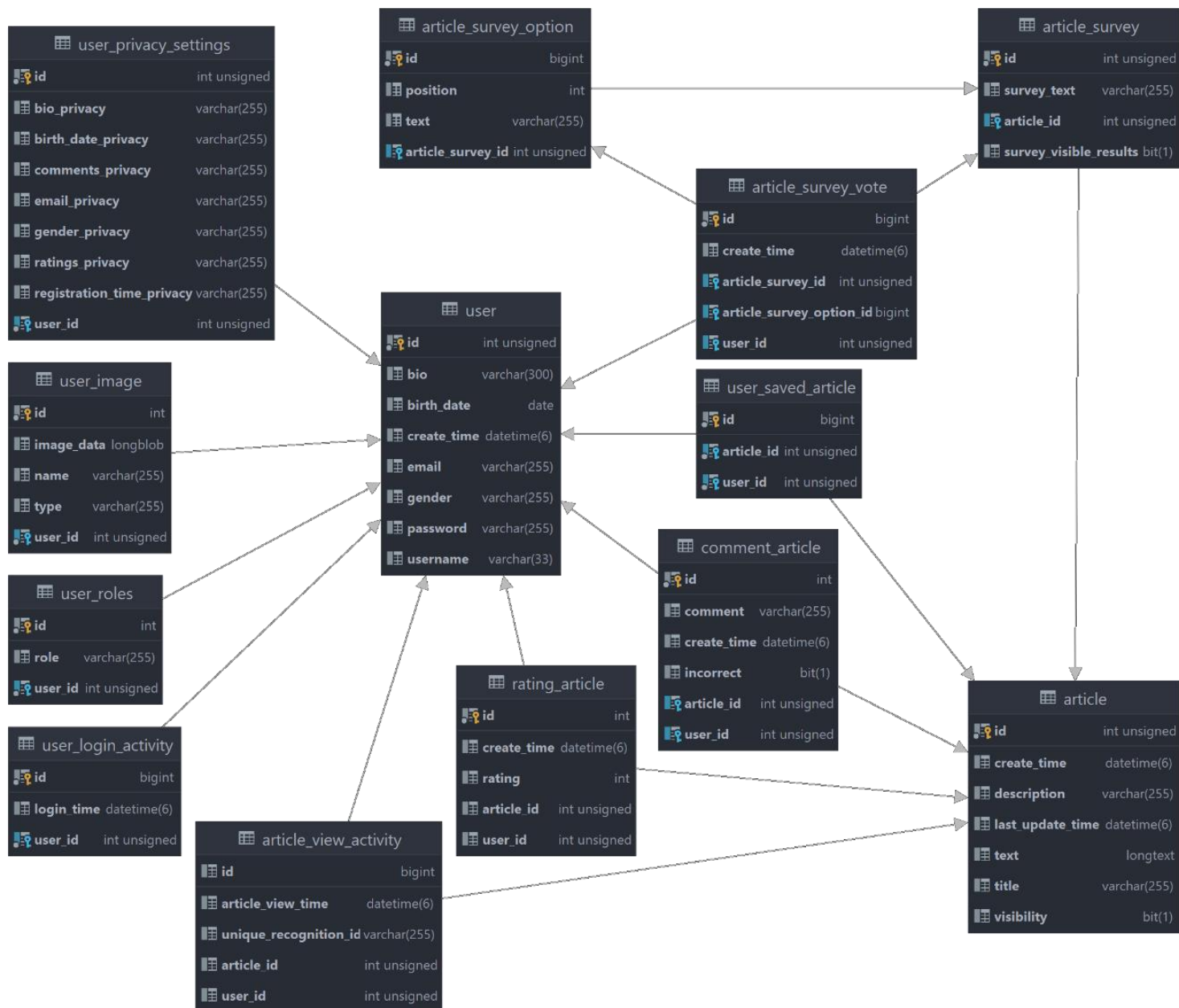
Obr. 8: Vývojový diagram pro počítání zobrazení článku

Zdroj: Autor

Na výše uvedeném obrázku je diagram, který demonstruje podmínky, za kterých je uložen záznam o novém zobrazení článku. Jako časové omezení pro nové zobrazení byla zvolena jedna hodina. Pokud je tedy uživatel přihlášen, jeho adresa se nebere jako podmínka. Důvodem je možnost přihlášení z různých zařízení, či míst. V případě, že by uživatel článek první zobrazil jako nepřihlášený a následně se přihlásil, budou započítány obě přečtení. V rámci práce se předpokládá jako možná situace sdílení jednoho zařízení například v domácnosti. V takovém případě by mohla být obě přečtení skutečná. Zároveň se nepředpokládá fakt, že by někdo vytvářel velké množství účtů pro umělé ovlivnění počtu zobrazení u článku. Z čehož plyne, že i tento způsob počítání je možné zneužít a ovlivnit. V rámci této práce, se tato skutečnost nepředpokládá.

Backend komunikuje s databází. Samotný obsah databáze je vytvořen automaticky backendem, který podle definovaných entit vygeneruje příhodné tabulky. Jednotlivé tabulky jsou spojeny skrze relace s využitím primárního a cizího klíče. Databáze obsahuje 13 tabulek.

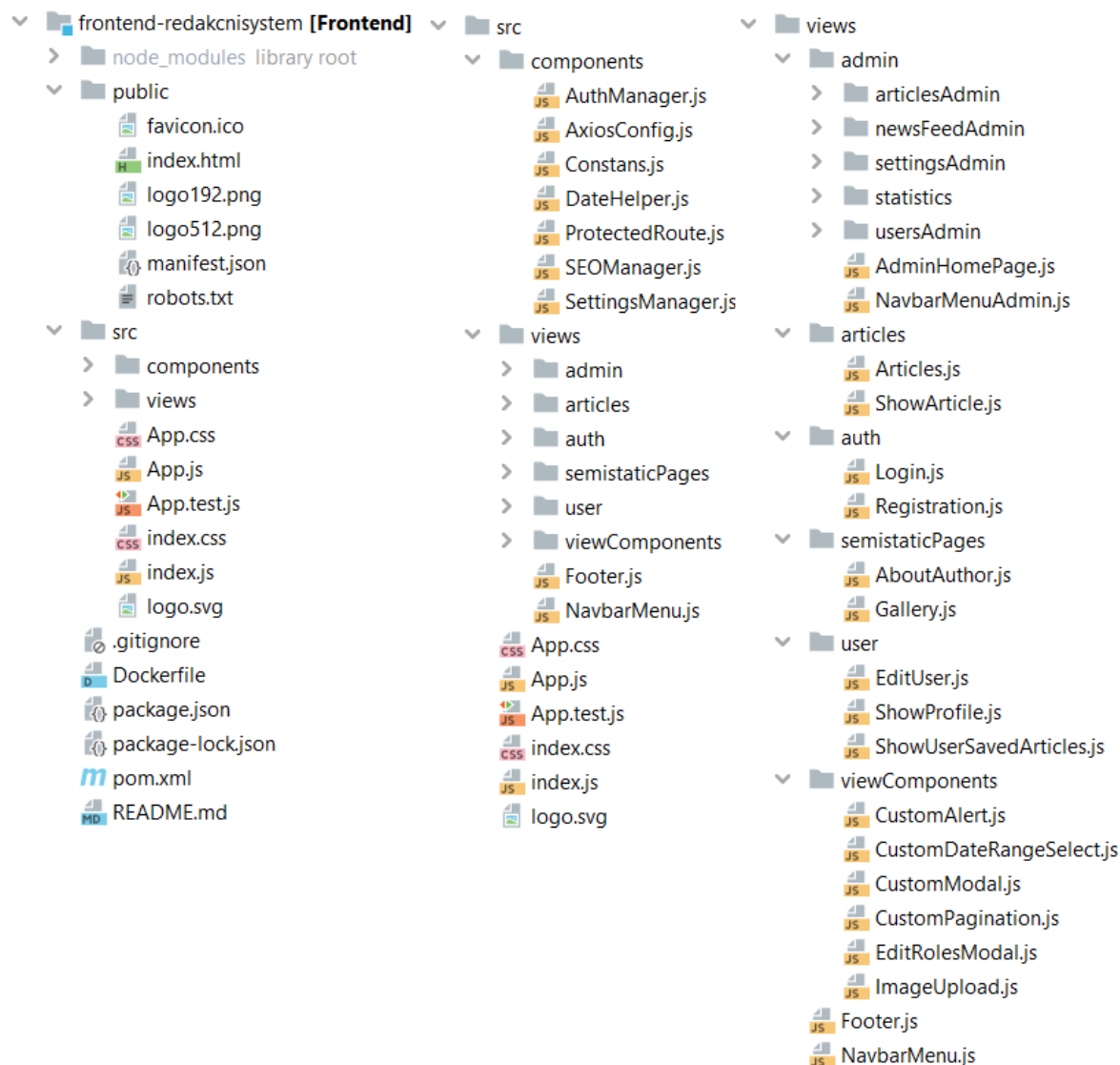
Následující obrázek je exportované schéma databáze z vývojářského prostředí. Na obrázku jsou viditelné tabulky včetně relací, lze zde také vidět rozdělení a uložení jednotlivých dat a jejich vzájemné souvislosti.



Obr. 9: Schéma databáze
Zdroj: Autor

6.4 Frontend

Pro vývoj frontend části, byla využita technologie ReactJS popsána v kapitole 5.2 Frontend – React.



Obr. 10: Struktura frontend části projektu
Zdroj: Autor

Na obrázku výše je viditelná struktura pro frontend část projektu. V levé části je viditelné základní rozdělení. Složka `node_modules`, obsahuje knihovny a balíčky, a je vytvořena po instalaci závislostí, které jsou definované v `package.json`. Část `public` je statická, obsahuje základní HTML předpis a doplňkové soubory, loga, ikony, výchozí nastavení pro vyhledávače. Samotné zdrojové kódy jsou k nalezení ve složce `src`, která má své další dělení. Ostatní soubory jsou konfigurační a pomocné.

Při pohledu na složku se zdrojovými kódy je viditelné rozdělení na komponenty (components) a pohledy, či viditelné komponenty (views). Toto rozdělení jasně odděluje zdrojové kódy podpůrných částí, které nejsou určené pro zobrazení a na části či celky, které jsou viditelné na samotné stránce. Komponenty určené pro uchování funkcí, které jsou využity v různých částech programu a jejich umístění do vlastních tříd předchází duplicitě. Další využití komponent je například v případě SEOManager a SettingsManager, tyto komponenty obsahují nastavení kontextu, který je následně využíván.

Balíček s viditelnými komponentami nebo jejich částmi, je následně dále strukturován pro větší přehlednost. Balíček admin je určen pro oddělení pohledů, které jsou využitelné pro správce stránky. Složka viewComponents odděluje vytvořené vlastní komponenty, které obsahují pouze část generované stránky. Ostatní komponenty jsou ve většině pro celou stránku. Pojmeme pro celou stránku se myslí obsahová stránka, ke každé takové stránce je přidána komponenta s nabídkou.

Frontend využívá router, který umožňuje zobrazení konkrétních komponent podle URL adresy. Router je obsažen v app.js, kde jsou nastavené veškeré cesty.

```
{!window.location.pathname.match(/\/admin.*\/g) ? <Navbar/> :  
<NavbarAdmin/>}  
<Routes>  
  {/* Admin routes */}  
  <Route path='/admin/pridat-clanek' element={  
    <ProtectedRoute childrenRoute=<ArticleEditor/>  
      mustBeLogged={true}  
      alternativePath={"/"}  
      roleList=["ADMIN"]}/>/>  
  {/*Ostatní admin cesty...*/}  
  {/*Default routes*/}  
  <Route path='/clanky' element=<Articles/>}/>  
  {/*Další veřejné cesty...*/}  
  {/*Pouze pro nepřihlášené */}  
  <Route path='/prihlaseni' element={  
    <ProtectedRoute childrenRoute=<Login/> mustBeLogged={false}  
    alternativePath={"/"}}/>/>  
  {/*Pouze pro přihlášené */}  
  <Route path='/upravit-profil' element={  
    <ProtectedRoute childrenRoute=<EditUser/> mustBeLogged={true}  
      alternativePath={"/prihlaseni"} roleList=["USER", "ADMIN"]}/>/>  
</Routes>  
<Footer/>
```

Zdrojový kód 3: Příklad routeru

Zdroj: Autor

Výše uvedený zdrojový kód je zkrácenou verzí dříve zmiňovaného routeru. Na začátku kódu je viditelná podmínka, která rozhoduje, zda se má zobrazit menu pro část stránek určené správci, či menu pro veřejnou část stránek. První cesta je do části správce, kde dochází k využití vytvořené komponenty ProtectedRoute, ta umožňuje nastavení podmínek, za kterých je možný vstup a definici cesty, kam bude uživatel přesměrován, pokud nejsou podmínky splněny. Následuje cesta přístupná všem, která již nevyužívá zmiňovanou komponentu. Ostatní routy jsou pro ukázkou, různých možností nastavení. Na konci je viditelná komponenta Footer, pro zobrazení patičky.

```
export const ProtectedRoute = ({childrenRoute, mustBeLogged, alternativePath,
roleList}) => {
  // získání id uživatele z cookies a ověření zda obsahuje validní hodnotu
  const [cookies] = useCookies('userId');
  let userLogged = Number.isInteger(Number(cookies.userId));
  let userRoles = userRolesExists() ? getUserRoles() : undefined;
  alternativePath = alternativePath ? alternativePath : "";
  // pokud uživatel musí být přihlášen ale není dojde k přesměrování na alternativní
  cestu
  if(mustBeLogged && !userLogged){
    return <Navigate to={alternativePath}/>
  }
  // pokud uživatel nesmí být přihlášen ale je dojde k přesměrování na alternativní
  cestu
  if(!mustBeLogged && userLogged){
    return <Navigate to={alternativePath}/>
  }
  // musí být přihlášen ale nemá požadovanou roli
  if(mustBeLogged && !roleList.some(role => userRoles.includes(role))){
    return <Navigate to={alternativePath}/>
  }
  // přesměrování na zabezpečenou komponentu
  return childrenRoute;
};
```

Zdrojový kód 4: Hlavní obsah komponenty ProtectedRoute

Zdroj: Autor

Zdrojový kód uvedený výše, je hlavní logika obsažená v ProtectedRoute, jenž byla zmíněna dříve. Dochází zde podle podmínek k přesměrování na cílovou komponentu (childrenRoute), či na alternativní cestu.

7 Shrnutí výsledků

Pro implementaci výsledného řešení byl použit programovací jazyk Java [18] s využitím frameworku Spring [19] pro backend aplikace. Frontend část byla tvořena v jazyce Javascript s knihovnou React [21]. Pro celý vývoj bylo využito vývojářské prostředí IntelliJ IDEA [33].

Vytvořená aplikace naplnila hlavní cíl práce, kterým bylo vytvořit alternativní redakční systém se zaměřením na osobní blogy. Výsledná aplikace nabízí základní požadované funkce. Stránky jsou responzivní. S využitím dotazníkového šetření byla získána data, která poukazují na nedostatky u aktuálních systémů. Na základě těchto dat byly implementovány funkce, které odstraňují určité nedostatky. Konkrétními příklady nedostatků je možnost vytvářet anketu pro jednotlivé články a vlastní uživatelský účet pro návštěvníky. Dotazníkové šetření také poskytuje seznam výhod, například komentáře nebo statistiky. Tyto funkce jsou také implementovány.

Funkce, které aplikace obsahuje je rozdělení uživatelů do rolí. Potřebné operace s články, komentáři, hodnocením, anketami dle přidělené role. Možnost uživatelských profilů včetně podrobného nastavení vlastního profilu. Dále pak funkce specifické pro roli správce. Správce má možnost využít zobrazení statistik, provést různá nastavení napříč celým systémem. V rámci statistik systém ukládá zobrazení jednotlivých článků, záznamy o přihlášení uživatelů.

Kvůli rozsahu a náročnosti implementace, nebyly některé funkce do aplikace přidány. Konkrétně se jedná o možnost úpravy vzhledu stránek bez znalostí programování.

Testování aplikace probíhalo technikou lokálního spuštění a následným používáním. Pro testování samotné backend části, byl využit nástroj Insomnia [42] pro API testování. Vhodnějším způsobem by bylo vytvoření automatizovaných testů pro každou z částí. Součástí testování bylo také spuštění pomocí popisu, který je obsažen v projektu. Toto spuštění bylo provedeno na dvou různých počítačích. Spuštění proběhlo bez obtíží.

Pro nasazení aplikace do provozu pro tvorbu blogu, by bylo dobré otestování funkčnosti na produkčním prostředí. Nasazení výsledné aplikace, by zároveň mohlo vyžadovat úpravy nastavení v obou částí z pohledu nastavení bezpečnosti, převážně CORS policy.

8 Závěry

Práci lze rozdělit do pěti částí, první část popisuje představení pojmu redakční systém, včetně konkrétních příkladů a jejich porovnání. Druhá část pojednává o provedení dotazníků v souvislosti s využitím redakčního systému pro účely osobního blogu, součástí jsou výsledky jednotlivých otázek dotazníků. Třetí část popisuje možnosti vývoje webové aplikace, mezi které se řadí i redakční systém z pohledu architektur a konceptů. Tato část poukazuje na různé možnosti a odlišnosti jednotlivých způsobů. Závěrem této části je představení architektur, které byly zvoleny pro tvorbu výsledného řešení, včetně důvodů k tomuto rozhodnutí. Část čtvrtá popisuje konkrétní technologie využité v rámci implementace. Součástí je také představení vývojářských nástrojů. Poslední část popisuje výslednou aplikaci, která byla v rámci práce vytvořena. Součástí tohoto popisu, jsou popisy jednotlivých částí systémů, včetně vybraných problematik.

Vhodným pokračováním ve vývoji aplikace by bylo doplnění dalších funkcí, které jsou žádoucí na základě dotazníkového šetření. Vhodná funkce může být možnost úpravy designu stránek pomocí uživatelského rozhraní bez nutnosti psaní kódu. Mohlo by se jednat o editaci umístění jednotlivých prvků skrze tzv. drag and drop. Tuto funkci se vzhledem k rozsahu a náročnosti nepovedlo implementovat.

Užitečným rozšířením existující funkce by bylo přidání více možností ohledně statistik. Aplikace zobrazuje většinu dostupných dat, ale zobrazení je ve většině případů koncipováno, jako zobrazení počtu za den, s vybraným rozsahem dnů. Není tedy možné zobrazení údajů s přesností na hodinu v daný den, či naopak sloučením denních údajů jako celek za týden s rozsahem měsíců.

9 Seznam obrázků

Obr. 1: Diagram MVC architektury	17
Obr. 2: Diagram architektury monolitu	23
Obr. 3: Diagram architektury mikroservis	24
Obr. 4: Diagram architektury vlastního řešení	25
Obr. 5: Architektura jazyku Java a JVM	27
Obr. 6: Spring architektura	28
Obr. 7: Struktura backend části projektu.....	40
Obr. 8: Vývojový diagram pro počítání zobrazení článku	45
Obr. 9: Schéma databáze	46
Obr. 10: Struktura frontend části projektu	47
Obr. 11: Příloha 2.1(a) Hlavní stránka aplikace	1
Obr. 12: Příloha 2.1(b) Hlavní stránka aplikace bez přihlášení	2
Obr. 13: Příloha 2.2 Zobrazení článku, včetně komentářů.....	3
Obr. 14: Příloha 2.3 Stránka o autorovi	4
Obr. 15: Příloha 2.4 Uživatelský profil	5
Obr. 16: Příloha 2.5 Nastavení uživatelského profilu	6
Obr. 17: Příloha 2.6 Hlavní stránka admin sekce.....	7
Obr. 18: Příloha 2.7 Editor článků	8
Obr. 19: Příloha 2.8 Zobrazení seznamu článků.....	9
Obr. 20: Příloha 2.9 Zobrazení seznamu uživatelů	10
Obr. 21: Příloha 2.10 Obecné nastavení blogu	10
Obr. 22: Příloha 2.11 Nastavení galerie	11
Obr. 23: Příloha 2.12 Nastavení stránky o autorovi	12
Obr. 24: Příloha 2.13 Zobrazení obecných statistik.....	13
Obr. 25: Příloha 2.14 Zobrazení statistik o článku	14
Obr. 26: Příloha 2.15 Zobrazení statistik o uživateli	15
Obr. 27: Příloha 2.16 Ukázka responzivity.....	16

10 Seznam tabulek

Tabulka 1: Ukázka možných rolí redakčního systému	3
Tabulka 2: Porovnání redakčních systémů	5
Tabulka 3: Výsledky otázky 6	11
Tabulka 4: Výsledky otázky 8	12
Tabulka 5: Souhrn výsledků otázky 11	14
Tabulka 6: Výsledky otázky 12	15
Tabulka 7: Funkce dle dotazníku a výsledné aplikace	35
Tabulka 8: Možnosti z pohledu rolí ve výsledné aplikaci	36

11 Seznam grafů

Graf 1: Výsledky otázky 1.....	8
Graf 2: Výsledky otázky 2.....	8
Graf 3: Výsledky otázky 3.....	9
Graf 4: Výsledky otázky 4.....	10
Graf 5: Výsledky otázky 5.....	10
Graf 6: Výsledky otázky 7.....	11
Graf 7: Výsledky otázky 9.....	12
Graf 8: Výsledky otázky 10.....	13
Graf 9: Výsledky otázky 13.....	15

12 Seznam zdrojových kódů

Zdrojový kód 1: Nastavení řetězce filtrů pro příchozí dotazy	41
Zdrojový kód 2: Příklad kontroléru, AuthController.....	43
Zdrojový kód 3: Příklad routeru.....	48
Zdrojový kód 4: Hlavní obsah komponenty ProtectedRoute	49

13 Seznam použité literatury

- [1] BARKER, Deane. Web Content Management: Systems, Features, and Best Practices. United States of America: O'Reilly Media, 2016. ISBN 978-1-491-90812-9.
- [2] WordPress [online]. [cit. 2023-08-29]. Dostupné z: <https://wordpress.org/>
- [3] Blog Usage Distribution [online]. [cit. 2024-03-22]. Dostupné z: <https://trends.builtwith.com/cms/blog>
- [4] Blogger [online]. [cit. 2023-08-29]. Dostupné z: <https://www.blogger.com/>
- [5] Wix [online]. [cit. 2023-08-29]. Dostupné z: <https://www.wix.com/>
- [6] Webnode [online]. [cit. 2023-08-29]. Dostupné z: <https://www.webnode.com/cs/>
- [7] Survio [online]. [cit. 2024-04-05]. Dostupné z: <https://www.survio.com/>
- [8] GANESHAN, Amuthan. Spring MVC Beginner's Guide. Second Edition. Birmingham B3 2PB, UK.: Packt Publishing, 2016. ISBN 978-1-78588-063-6.
- [9] ČÁPKA, Daniel. MVC architektura. In: Itnetwork [online]. [cit. 2023-08-20]. Dostupné z: <https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor>
- [10] MASSÉ, Mark. REST API Design Rulebook. United States of America: O'Reilly Media, 2012. ISBN 978-1-449-31050-9.
- [11] MDN Web Docs [online]. [cit. 2023-08-29]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP>
- [12] What is REST API? [online]. IBM [cit. 2023-08-29]. Dostupné z: <https://www.ibm.com/topics/rest-apis>
- [13] SMITH, Peter. Professional Website Performance: Optimizing the front end and the back end. Canada: John Wiley, 2013. ISBN 978-1-118-48752-5.
- [14] SCOTT, Emmit. SPA Design and Architecture: Understanding single-page web applications. United States of America: Manning Publications Co., 2016. ISBN 978-1-617-29243-9.
- [15] NEWMAN, Sam. Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith. United States of America: O'Reilly Media, 2019. ISBN 978-1-492-07554-7.
- [16] GeeksforGeeks. Introduction to Java [online]. [cit. 2023-08-29]. Dostupné z: <https://www.geeksforgeeks.org/introduction-to-java/>
- [17] Oracle. Differences between Java EE and Java SE [online]. [cit. 2023-08-29]. Dostupné z: <https://docs.oracle.com/javaee/6/firstcup/doc/gkhoy.html>
- [18] FLANAGAN, David. Java in a nutshell. Fourth Edition. United States of America: O'Reilly Media, 2002. ISBN 0-596-00283-1.

- [19] WALLS, Craig. Spring in Action. Fifth Edition. United States of America: Manning Publications Co., 2019. ISBN 978-1-617-29494-5.
- [20] Overview of Spring Framework: Figure 1. In: Spring Documentation [online]. 2017 [cit. 2023-08-20]. Dostupné z: <https://docs.spring.io/spring-framework/docs/5.0.0.RC3/spring-framework-reference/overview.html>
- [21] META. ReactJS [online]. [cit. 2024-03-22]. Dostupné z: <https://react.dev/>
- [22] Meta [online]. [cit. 2024-03-22]. Dostupné z: <https://www.meta.com/>
- [23] Axios [online]. [cit. 2024-03-22]. Dostupné z: <https://axios-http.com/>
- [24] Bootstrap [online]. [cit. 2024-03-22]. Dostupné z: <https://getbootstrap.com/>
- [25] React Bootstrap [online]. [cit. 2024-03-22]. Dostupné z: <https://react-bootstrap.netlify.app/>
- [26] React ChartsJS [online]. [cit. 2024-03-22]. Dostupné z: <https://react-chartjs-2.js.org/>
- [27] What Is a Database? [online]. ORACLE. [cit. 2024-03-22]. Dostupné z: <https://www.oracle.com/database/what-is-database/>
- [28] What is a Relational Database (RDBMS)? [online]. ORACLE. [cit. 2024-03-22]. Dostupné z: <https://www.oracle.com/database/what-is-a-relational-database/>
- [29] DYER J. T., Russel. MySQL in a Nutshell, Second Edition. 2. O'Reilly Media, 2008. ISBN 978-0-596-51433-4.
- [30] PhpMyAdmin [online]. [cit. 2024-03-22]. Dostupné z: <https://www.phpmyadmin.net/>
- [31] MySQL Workbench [online]. [cit. 2024-03-22]. Dostupné z: <https://www.mysql.com/products/workbench/>
- [32] AWS Amazon. What is an IDE? [online]. [cit. 2023-08-29]. Dostupné z: <https://aws.amazon.com/what-is/ide/>
- [33] IntelliJ IDEA [online]. [cit. 2023-08-29]. Dostupné z: <https://www.jetbrains.com/idea/>
- [34] NetBeans [online]. [cit. 2023-08-29]. Dostupné z: <https://netbeans.apache.org/>
- [35] Eclipse IDE [online]. [cit. 2023-08-29]. Dostupné z: <https://eclipseide.org/>
- [36] LOELIGER, Joe a Matthew MCCULLOUGH. Version Control with Git. Second Edition. United States of America: O'Reilly Media, 2012. ISBN 978-1-449-31638-9.
- [37] Git [online]. [cit. 2023-08-29]. Dostupné z: <https://git-scm.com/>
- [38] GitHub [online]. [cit. 2023-08-29]. Dostupné z: <https://github.com/>

- [39] GitLab [online]. [cit. 2023-08-29]. Dostupné z: <https://gitlab.com/>
- [40] TechTarget. What is API Testing? [online]. [cit. 2023-08-29]. Dostupné z: <https://www.techtarget.com/searcharchitecture/definition/API-testing>
- [41] Postman [online]. [cit. 2023-08-29]. Dostupné z: <https://www.postman.com/>
- [42] Insomnia [online]. [cit. 2023-08-29]. Dostupné z: <https://insomnia.rest/>
- [43] Docker [online]. [cit. 2024-04-12]. Dostupné z: <https://www.docker.com/>

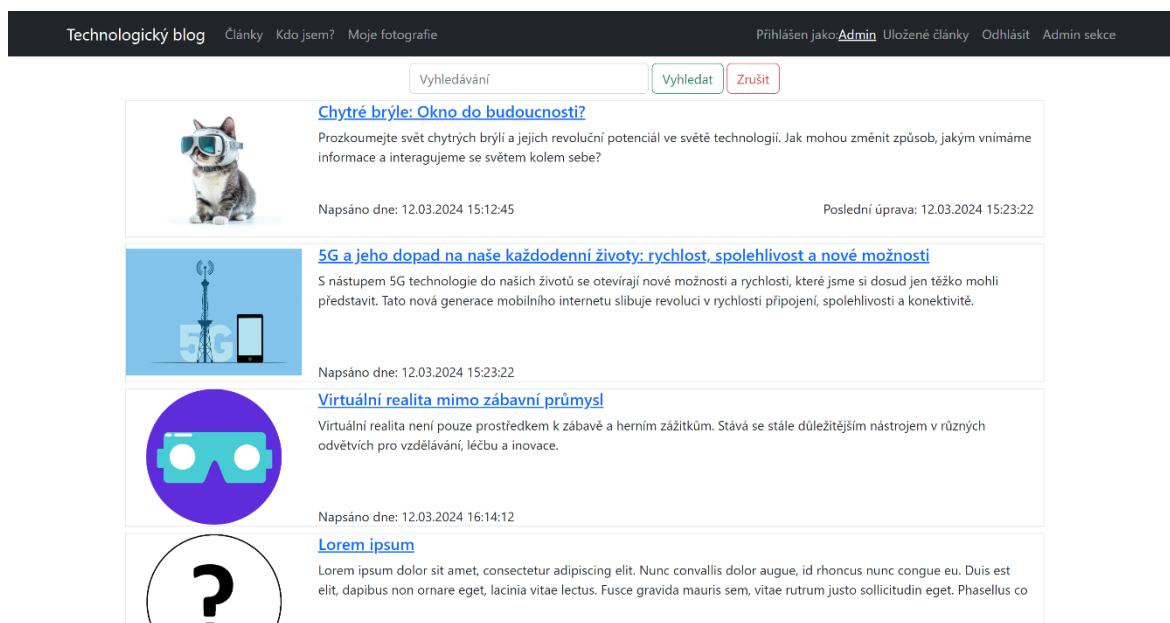
14 Přílohy

14.1 Příloha 1

Součástí elektronického odevzdání jsou kompletní zdrojové kódy včetně popisu různých možností spuštění aplikace.

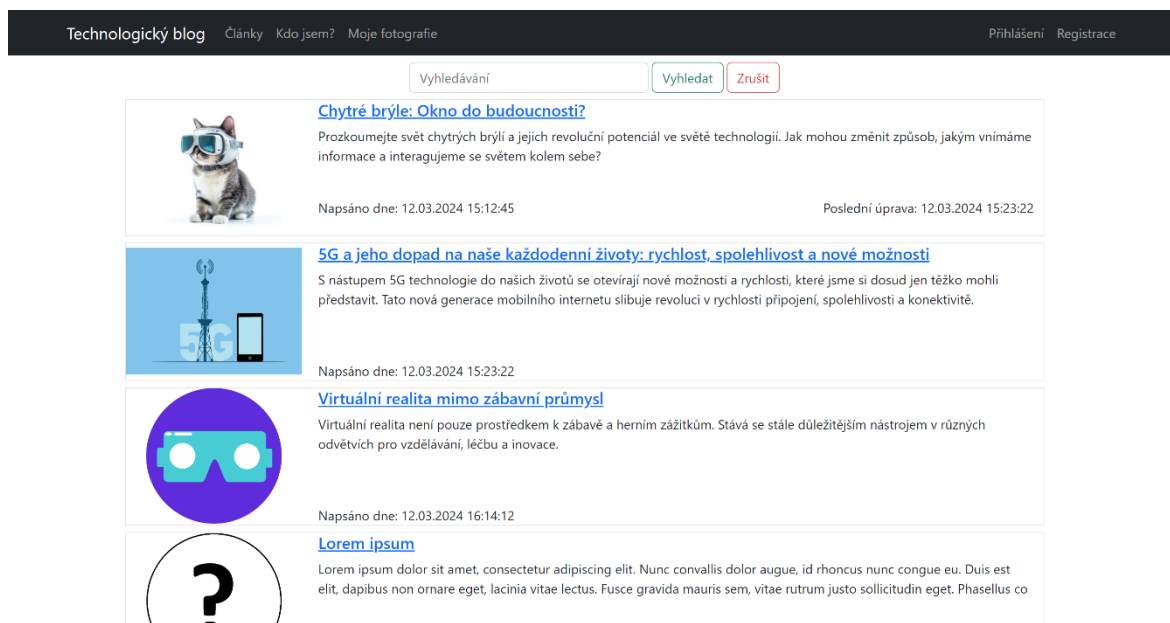
14.2 Příloha 2 – Fotodokumentace aplikace

Následující obrázky, které se týkají fotodokumentace výsledné aplikace, jsou také součástí elektronické přílohy. Je tedy možné si je zobrazit ve větší kvalitě. Veškeré snímky jsou tvořeny při šířce zařízení 1440 pixelů. Pokud se jedná o jinou šířku, je uvedena v textu u obrázku.



Obr. 11: Příloha 2.1(a) Hlavní stránka aplikace
Zdroj: Autor

Na výše uvedeném obrázku je hlavní stránka aplikace. Každá stránka obsahuje navigační menu viditelné nahoře. U následujících snímků je přihlášen uživatel s rolí správce. Pokud by přihlášený uživatel nebyl správcem, není zde viditelná nabídka admin sekce nahoře v pravé části. Situace s nepřihlášeným uživatelem, je viditelná na následujícím obrázku. Navigační menu obsahuje odkazy na seznam článků, na stránku o tvůrci a na galerii. Dále v pravé části jsou informace o uživateli. Možnost zobrazit si uložené články, odhlásit se, či v případě, že se jedná o správce, tak přechod do admin sekce. Samotný obsah je tvořen seznamem článků, rozděleným do stránek a s možností vyhledávat.



Obr. 12: Příloha 2.1(b) Hlavní stránka aplikace bez přihlášení
Zdroj: Autor

Výše uvedený obrázek zobrazuje hlavní stránku aplikace jako obrázek předchozí. Rozdílem je nepřihlášený uživatel. Na tomto obrázku je viditelné rozložení navigačního menu, pokud není uživatel přihlášen. Lze vidět, že menu je v levé části stejné, ale pravá část místo informací a možností pro uživatele, zobrazuje možnost přejít na přihlášení či registraci.

Technologický blog
Přihlášen jako **Admin** | Uložené články | Odhlásit | Admin sekce

Chytré brýle: Okno do budoucnosti?

Uložit do seznamu
Možnosti admina ▾

Prozkoumejte svět chytrých brýlí a jejich revoluční potenciál ve světě technologií. Jak mohou změnit způsob, jakým vnímáme informace a interagujeme se světem kolem sebe?

V dnešním světě, kde technologie neustále posouvá hranice možnosti, se můžeme ptát: Co je další velká věc? Mezi předměty, které se rychle stávají středem pozornosti, jsou chytré brýle. Tyto zařízení nám slibují nový způsob, jak vnímáme a interagujeme se světem. Chytré brýle, jako [Google Glass](#) nebo Apple Glass, nejsou jenom o klasickém zobrazování informací před očima. Jsou to nástroje pro rozšířenou realitu (AR), které mohou měnit náš pohled na okolní prostředí. Můžeme si představit, že při procházce městem získáváme informace o historických památkách, recenze restaurací nebo navigaci bez nutnosti sledovat displej telefonu.

V oblasti práce mohou chytré brýle změnit způsob, jakým plníme úkoly. Technici v terénu mohou mít přímý pohled na návody k opravě, lékaři mohou během operace mít okamžitý přístup k zdravotní historii pacienta. To všechno může vést k efektivnější práci a zvýšení produktivity. Ale nejenom práce, i zábava se může stát zcela novým zážitkem. S chytrými brýlemi můžete hrát AR hry, sledovat filmy nebo dokonce virtuálně setkávat se s přáteli přes aplikace pro videohovory.

Samozřejmě, jako u všech nových technologií, jsou zde otázky související s soukromím a bezpečností dat. Jak bude naše osobní data chráněna, když chytré brýle sledují každý pohyb našich očí?

V každém případě jsou chytré brýle fascinujícím krokem směrem k ještě propojenějšímu a efektivnějšímu světu. Jsou to okna do budoucnosti, která nám mohou otevřít nové možnosti a zážitky.

Zdá se chytré brýle stanou běžnou součástí našich životů, to čas ukáže. Ale jedno je jisté: jsou to zařízení, která stojí za pozornost a která mohou mít obrovský dopad na způsob, jakým žijeme a pracujeme.

Budoucnost je před námi. A ta budoucnost může být připoutána ke konci našich nosů.

Tak co si myslíte o chytrých brýlích? Jsou to inovativní technologie, které nás povedou do nového věku, nebo jenom krátkodobý trend? Podělte se o své názory a zkušenosti v komentářích!

Máte zájem o chytré brýle?

Ano (Počet hlasů: 1 | 100%)

Ne (Počet hlasů: 0 | 0%)

Uvidím podle ceny (Počet hlasů: 0 | 0%)

Hlasovat

Vaše hodnocení: 3

Ze dne: 05.04.2024 15:09

Počet hodnocení: 2

Smazat hodnocení

Napsáno dne: 12.03.2024 15:12

Počet zobrazení: 8

Upraveno: 12.03.2024 15:23

Přidat komentář

Zde napište komentář

Přidat komentář

Komentáře Od nejnovějších ▾

Tomáš komentoval:

05.04.2024 15:10

Vhodný

Komentář je správcem označen jako nevhodný. [Zobrazit](#)

Admin komentoval:

05.04.2024 15:10

Smazat
Nevhodný

Děkuji, ano mám v plánu podobné články. Napiš prosím další komentář, který označím jako nevhodný.

Tomáš komentoval:

05.04.2024 15:07

Nevhodný

Opravdu zajímavý článek. Bude více podobných článků?

Děkuji za návštěvu na mém blogu.

© 2024 Copyright

Obr. 13: Příloha 2.2 Zobrazení článku, včetně komentářů
Zdroj: Autor

Výše uvedený obrázek je zobrazení konkrétního článku. Na tuto stránku je možné se dostat rozkliknutím nadpisu, některého z článků v seznamu na hlavní stránce. Možností, jak článek zobrazit je více, dále je to zobrazení z uloženého seznamu, z profilu uživatele. Při popisu shora je viditelná nadpis, v pravé části u nadpisu je možnost uložení článku (díky přihlášenému uživateli) a možnosti admina (díky roli správce). Možnosti správce skrývají nabídku přejít na úpravu článku a na statistiky článku. Následuje obrázek článku, vedle kterého je krátký popis článku. Pokračuje samotný obsah článku. Pod článkem je anketa, pokud jí článek obsahuje. Dále je zde hodnocení, v případě, že uživatel hodnotil možnost odebrat hlasování. Následují

informace o počtu zobrazení, vytvoření a úpravě článku. Tyto informace záleží na nastavení blogu. Následuje část s komentáři, vzhledem k přihlášenému uživateli je možnost přidat komentář, či odebrat vlastní. Označení vhodného a nevhodného komentáře je možné vzhledem k roli správce u přihlášeného uživatele.

Technologický blog [Články](#) [Kdo jsem?](#) [Moje fotografie](#) Přihlášen jako **Admin** [Uložené články](#) [Odhlásit](#) [Admin sekce](#)

O autorovi


Vítejte na mé osobní stránce! Jsem Jakub Novák, a rád bych vám trochu přiblížil, kdo jsem a co vás zde může čekat. Rád bych Vás také upozornil, že jsem smyšlený a celý tento obsah byl vytvořen díky ChatGPT. Obsah byl následně jen vzhledově upraven, aby byly viditelné možnosti nastavení.

Kdo jsem?

Jsem váš průvodce a tvůrce obsahu na této stránce. Mé jméno je Jakub Novák, a jsem nadšeným spisovatelem, výzkumníkem a vášnivým průzkumníkem světa kolem sebe. Věřím v sílu slova a jsem přesvědčen, že sdílení zkušeností a příběhů může být inspirativní a obohacující pro všechny zúčastněné.

Co můžete očekávat?

Na této stránce se můžete těšit na různorodý obsah, který odráží mé zájmy, zkušenosti a zamyšlení. Najdete zde články a eseje, které se týkají mého oboru, ale také osobní příběhy, tipy a návody, které se mohou týkat vašeho každodenního života. Mým cílem je nabídnout inspiraci, povzbuzení a možná i trochu zábavy prostřednictvím mého psaní.



Proč tento blog?

Rozhodl jsem se vytvořit tento blog jako prostor pro sdílení svých myšlenek, zkušeností a objevů. Věřím, že každý z nás má svůj jedinečný příběh a perspektivu, kterou může přinést do světa. Tento blog je pro mě platformou, abych mohl sdílet své příběhy a zážitky s vámi a možná vás inspirovat k objevení vašich vlastních příběhů.

Jak mě můžete kontaktovat?

Chci, aby tento blog byl interaktivním prostorem, kde můžeme společně komunikovat a sdílet naše myšlenky. Pokud máte otázky, náměty nebo jednoduše chcete podělit se svými názory, neváhejte mě kontaktovat. Stačí mi napsat e-mail na jakubnovak@example.com nebo mě sledovat na sociálních sítích [@jakubnovak](#).
Děkuji, že jste si našli čas k prozkoumání mého blogu. Doufám, že vás můj obsah osloví a že se společně budeme moci podělit o inspirativní příběhy a zážitky.
S úctou,
Jakub Novák

Místa kde trávím až moc času


[Facebook](#)
[Instagram](#)

Děkuji za návštěvu na mém blogu.
© 2024 Copyright

Obr. 14: Příloha 2.3 Stránka o autorovi
Zdroj: Autor

Na výše uvedeném obrázku je viditelná stránka o autorovi. Na tuto stránku je možné se dostat skrze navigační menu. Zde je díky nastavení název položky v menu nazván jako „Kdo jsem?“.

Technologický blog [Články](#) [Kdo jsem?](#) [Moje fotografie](#) Přihlášen jako **Admin** [Uložené články](#) [Odhlásit](#) [Admin sekce](#)





Tomáš
Email: Tento údaj je soukromý.
Registrace: 05.04.2024 14:53
Datum narození: 01.01.1990
Pohlaví: Muž

Ahoj, jsem Tomáš a jsem zde pro přidání ukázkového obsahu.


[Možnosti admina](#)

Hodnocení uživatele


 [Tomáš](#) hodnotil: ★★★★★ 05.04.2024 15:09


 [Chytré brýle: Okno do budoucnosti?](#)
Prozkoumejte svět chytrých brýlí a jejich revoluční potenciál ve světě technologií. Jak mohou změnit způsob, jakým vnímáme informace a interagujeme se světem kolem sebe?

Komentáře uživatele


 [Tomáš](#) komentoval: 05.04.2024 15:07 Nevhodný

Opravdu zajímavý článek. Bude více podobných článků?

 [Chytré brýle: Okno do budoucnosti?](#)
Prozkoumejte svět chytrých brýlí a jejich revoluční potenciál ve světě technologií. Jak mohou změnit způsob, jakým vnímáme informace a interagujeme se světem kolem sebe?

 [Tomáš](#) komentoval: 05.04.2024 15:10 Vhodný

Komentář je správcem označen jako nevhodný. [Zobrazit](#)

 [Chytré brýle: Okno do budoucnosti?](#)
Prozkoumejte svět chytrých brýlí a jejich revoluční potenciál ve světě technologií. Jak mohou změnit způsob, jakým vnímáme informace a interagujeme se světem kolem sebe?

Děkuji za návštěvu na mém blogu.
© 2024 Copyright

Obr. 15: Příloha 2.4 Uživatelský profil
Zdroj: Autor

Obrázek uvedený výše zobrazuje uživatelský profil. Zde je viditelný obrázek uživatele, vedle obrázku jsou viditelné údaje, jméno, email, datum a čas registrace, datum narození a pohlaví. Vedle dále je pak popis uživatele. Tyto údaje je možné měnit v nastavení, kromě jména, emailu a datumu registrace. Zároveň je u všech údajů krom jména možné nastavit úroveň soukromí. Toto nastavení je viditelné na následujícím obrázku. Pod informacemi o uživateli jsou hodnocení uživatele a komentáře. Možnost admina je opět viditelná díky roli správce. Pokud by se jednalo o profil přihlášeného uživatele v těchto místech je tlačítko, které vede do nastavení profilu. Možnosti admina zobrazují možnost přejít na statistiky uživatele.

Technologický blog [Články](#) [Kdo jsem?](#) [Moje fotografie](#) Přihlášen jako **Admin** [Uložené články](#) [Odhlásit](#) [Admin sekce](#)

[Zpět na profil](#) [Smazat profil](#)

Změna hesla

Změna profilového obrázku

Nastavení soukromí a osobních údajů

Nastavení soukromí

Email: admin@admin.cz	Zobrazení na profilu:	Soukromé <input type="checkbox"/>
Registrace: 05.04.2024 15:42	Zobrazení na profilu:	Pro přihlášené <input type="checkbox"/>
Komentáře článků	Zobrazení na profilu:	Veřejné <input type="checkbox"/>
Hodnocení článků	Zobrazení na profilu:	Veřejné <input type="checkbox"/>

Osobní údaje

Popis	<input type="text"/>	Veřejné <input type="checkbox"/>
0/300		
Datum narození:	<input type="text" value="dd.mm.rrrr"/>	Veřejné <input type="checkbox"/>
Pohlaví:	Jiné <input type="checkbox"/>	Veřejné <input type="checkbox"/>

Děkují za návštěvu na mém blogu.
© 2024 Copyright

Obr. 16: Příloha 2.5 Nastavení uživatelského profilu
Zdroj: Autor

Výše uvedený obrázek zobrazuje nastavení uživatelské účtu. V horní části jsou vidět možnosti přejít zpět na profil a smazat profil. Dále jsou viditelné zavřené možnosti změny hesla a změny profilového obrázku. Změna hesla obsahuje pouze formulář, kde uživatel zadá stávající heslo, nové heslo a potvrdí nové heslo a uložení této změny. Změna profilového obrázku zobrazuje aktuální obrázek a možnost nahrát nový a uložit změnu. Poslední část je nastavení soukromí a nastavení osobních údajů. Zde je možné nastavit soukromí u údajů, které uživatel nemůže měnit. Následně pak přidat dobrovolné údaje a nastavit jejich soukromí. Ukládání probíhá odděleně, na uložení soukromí a uložení osobních údajů. Na tuto stránku je možné se dostat po přihlášení z vlastního uživatelského profilu.

Admin sekce [Přidat článek](#) [Články](#) [Uživatelé](#) [Statistiky](#) [Nastavení](#)
Přihlášen jako: **Admin** [Uživatelská sekce](#)

Přehled aktivity

Začátek Konec

#	Uživatel	Článek	Komentář	Čas	#	Uživatel	Článek	Hodnocení	Čas
1	Tomáš	Chytré brýle: Okno do budoucnosti?	Opravdu zajímavý článek. Bude více podobných článků?	05.04.2024 15:07	1	Admin	Chytré brýle: Okno do budoucnosti?	★★★★☆	05.04.2024 15:09
2	Admin	Chytré brýle: Okno do budoucnosti?	Děkuji, ano mám v plánu podobné články. Napiš prosím další komentář, který označím jako nevhodný.	05.04.2024 15:10	2	Tomáš	Chytré brýle: Okno do budoucnosti?	★★★★★	05.04.2024 15:09
3	Tomáš	Chytré brýle: Okno do budoucnosti?	Nevhodný komentář	05.04.2024 15:10					

Děkuji za návštěvu na mém blogu.
 © 2024 Copyright

Obr. 17: Příloha 2.6 Hlavní stránka admin sekce
Zdroj: Autor

Následující obrázky včetně výše uvedeného jsou z admin sekce (část pro správce). Navigační menu je odlišné, aktuálně jsou v pravé části viditelné jednotlivé stránky správce, v levé části přechod na uživatelskou část. Na hlavní stránce, je viditelný přehled nové aktivity, zde jsou viditelné nové komentáře a hodnocení. Výchozí čas pro novou aktivitu je posledních 24 hodin.

Upravit článek

Editor článku ↑


Název článku

Chytré brýle: Okno do budoucnosti?

Vyberte způsob tvorby stručného popisu: Viditelnost článku: Nahrajte obrázek.

Vlastní popis článku Zobrazit článek Byl vybrán:

Aktuální obrázek:



Stručný popis článku. Počet znaků: 169/250

Prozkoumejte svět chytrých brýlí a jejich revoluční potenciál ve světě technologií. Jak mohou změnit způsob, jakým vnímáme informace a interagujeme se světem kolem sebe?

Text článku

Normal **B** **I** **U** **S** **”** **≡** **≡** **≡** **≡** **≡** **≡** **A**

V dnešním světě, kde technologie neustále posouvá hranice možností, se můžeme ptát: Co je další velká věc? Mezi předměty, které se rychle stávají středem pozornosti, jsou chytré brýle. Tyto zařízení nám slibují nový způsob, jak vnímáme a interagujeme se světem. Chytré brýle, jako [Google Glass](#) nebo [Apple Glass](#), nejsou jenom o klasickém zobrazování informací před očima. Jsou to nástroje pro rozšířenou realitu (AR), které mohou měnit náš pohled na okolní prostředí. Můžeme si představit, že při procházce městem získáváme informace o historických památkách, recenze restaurací nebo navigaci bez nutnosti sledovat displej telefonu.

V oblasti práce mohou chytré brýle změnit způsob, jakým plníme úkoly. Technici v terénu mohou mít přímý pohled na návody k opravě, lékaři mohou během operace mít okamžitý přístup k zdravotní historii pacienta. To všechno může vést k efektivnější práci a zvýšení produktivity. Ale nejenom práce, i zábava se může stát zcela novým zážitkem. S chytrými brýlemi můžete hrát AR hry, sledovat filmy nebo dokonce virtuálně setkat se s přáteli přes aplikace pro videohovory.

Samozřejmě, jako u všech nových technologií, jsou zde otázky související s soukromím a bezpečností dat. Jak bude naše osobní data chráněna, když chytré brýle sledují každý pohyb našich očí?

V každém případě jsou chytré brýle fascinujícím krokem směrem k ještě propojenějšímu a efektivnějšímu světu. Jsou to okna do budoucnosti, která nám mohou otevřít nové možnosti a zážitky.

Zda se chytré brýle stanou běžnou součástí našich životů, to čas ukáže. Ale jedno je jisté: Jsou to zařízení, která stojí za pozornost a která mohou mít obrovský dopad na způsob, jakým žijeme a pracujeme.

Budoucnost je před námi. A ta budoucnost může být připoutána ke konci našich nosů.

Tak co si myslíte o chytrých brýlích? Jsou to inovativní technologie, které nás povedou do nového věku, nebo jenom krátkodobý trend? Podělte se o své názory a zkušenosti v komentářích!

Anketa k článku

Má tento článek obsahovat anketu.

Zobrazit výsledky anket pro čtenáře?

Text k anketě:

Máte zájem o chytré brýle?

[Přidat další možnost](#)

Pozice: 1	Odpověď:	<input type="text" value="Ano"/>	Odebrat možnost
Pozice: 2	Odpověď:	<input type="text" value="Ne"/>	Odebrat možnost
Pozice: 5	Odpověď:	<input type="text" value="Uvidím podle ceny"/>	Odebrat možnost

[Uložit](#)

Ukázka článku ↓

Děkuji za návštěvu na mém blogu.

© 2024 Copyright

Obr. 18: Příloha 2.7 Editor článků
Zdroj: Autor

Výše viditelný obrázek zobrazuje editor článku. Na obrázku je viditelný již vytvořený článek. Editor je stejný pro nový článek, či úpravu stávajícího. V horní části je viditelné rozdělení na editor a ukázkou. Editor viditelný na obrázku umožňuje nastavit všechny atributy pro článek. Po hlavním obsahu článku, je viditelné nastavení k anketě, či

případně samotné možnosti. Na tuto stránku je možné se dostat v případě nového článku z navigačního menu nebo ze seznamu článku. V případě úpravy ze zobrazení článku skrze možnosti admina nebo ze seznamu článků v admin sekci.

Admin sekce [Přidat článek](#) [Články](#) [Uživatelé](#) [Statistiky](#) [Nastavení](#)
Přihlášen jako: Admin [Uživatelská sekce](#)

[Přidat článek +](#)
[Obnovit ↻](#)

Zobrazení na stránce 10

#	ID	Titulek	Datum přidání	Datum úpravy	Počet zobrazení	Viditelný	Statistiky	Upravit	Zobrazení	Odebrat
1	10	Desátý článek	12.03.2024 21:39	12.03.2024 21:43	1	Ano	Zobrazit	Upravit	Skrýt	Smazat
2	7	titulek	12.03.2024 21:39	12.03.2024 21:39	0	Ano	Zobrazit	Upravit	Skrýt	Smazat
3	6	titulek	12.03.2024 21:38	12.03.2024 21:38	0	Ano	Zobrazit	Upravit	Skrýt	Smazat
4	4	Lorem ipsum	12.03.2024 16:28	12.03.2024 16:28	2	Ano	Zobrazit	Upravit	Skrýt	Smazat
5	3	Virtuální realita mimo zábavní průmysl	12.03.2024 16:14	12.03.2024 16:14	3	Ano	Zobrazit	Upravit	Skrýt	Smazat
6	2	5G a jeho dopad na naše každodenní životy: rychlost, spolehlivost a nové možnosti	12.03.2024 15:23	12.03.2024 15:23	4	Ano	Zobrazit	Upravit	Skrýt	Smazat
7	1	Chytré brýle: Okno do budoucnosti?	12.03.2024 15:12	12.03.2024 15:23	8	Ano	Zobrazit	Upravit	Skrýt	Smazat

Děkuji za návštěvu na mém blogu.
© 2024 Copyright

Obr. 19: Příloha 2.8 Zobrazení seznamu článků

Zdroj: Autor

Výše je na obrázku viditelná stránka se seznamem článku. Zde má správce možnost přejít na vytvoření nového článku, obnovit data, vyhledat článek, či nastavit počet článků viditelných na jedné stránce. Následuje samotná tabulka, zde jsou viditelné základní informace o článku, v pravé části poté operace, které může správce s článkem provádět. Na stránku je možné se dostat skrze navigační menu.

Na následujícím obrázku je viditelný seznam uživatelů. Stránka je velmi podobná stránce na předchozím obrázku. Opět je zde možnost obnovit data, vyhledávat, či nastavit počet zobrazení na stránce. Samotná tabulka obsahuje také základní data a v pravé části možnosti pro správce. Správce zde má možnost změnit role uživatele. Zajímavostí je, že uživatel nemůže odebrat roli správce u aktuálně přihlášeného účtu. Jedná se o prevenci proti odebrání jediného správce. Zároveň pokud není registrován žádný uživatel, tak první registrovaný získá automaticky roli správce. Na stránku je možné se dostat skrze položku uživatelé v navigačním menu.

#	ID	Jméno	Email	Registrován	Poslední přihlášení	Počet hodnocení	Počet komentářů	Role	Statistiky
1	3	Admin	admin@admin.cz	10.03.2024 20:40	05.04.2024 15:06	3	2	ADMIN,USER	Zobrazit
2	4	Pepa	testfail@random.cz	05.04.2024 14:49	Nebyl přihlášen	0	0	USER	Zobrazit
3	5	Tomáš	random1@random.cz	05.04.2024 14:53	05.04.2024 14:53	1	2	USER	Zobrazit

Děkuji za návštěvu na mém blogu.

© 2024 Copyright

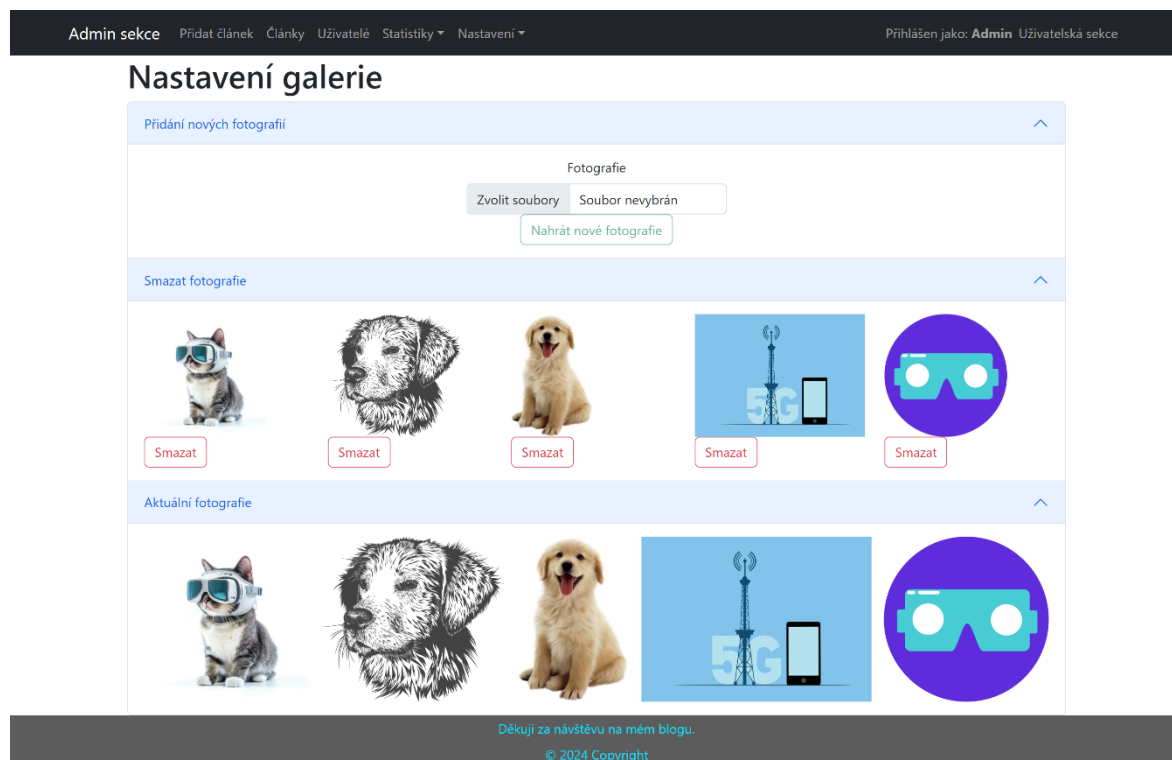
Obr. 20: Příloha 2.9 Zobrazení seznamu uživatelů
Zdroj: Autor

Admin sekce		Přidat článek		Články		Uživatelé		Statistiky		Nastavení	
Přihlášen jako: Admin											Uživatelská sekce
Obecné nastavení											
Název blogu:			Technologický blog								
Ikona		Náhled:		Vybrat soubor		So...rán		Uložit ikonu			
Nastavení článků											
Zobrazit počet hodnocení článků:			<input checked="" type="checkbox"/>		Zobrazit počet zobrazení článku:			<input checked="" type="checkbox"/>			
Nastavení času											
Zobrazit čas u článku:			Vytvoření a úprava								
Formát času a data:		Náhled: 05.04.2024 15:38		DD.MM.YYYY HH:mm							
Patička											
První řádek:			Děkuji za návštěvu na mém blogu.								
Druhý řádek:			© 2024 Copyright								
Barva pozadí:			<input type="checkbox"/>		Barva textu:			<input checked="" type="checkbox"/>			
Stránky											
Stránka autora:			<input checked="" type="checkbox"/> Přístupná veřejně		Text v nabídce odkazující na stránku:			Kdo jsem?			
Galerie:			<input checked="" type="checkbox"/> Přístupná veřejně		Text v nabídce odkazující na stránku:			Moje fotografie			
Uložit											
Děkuji za návštěvu na mém blogu.											
© 2024 Copyright											

Obr. 21: Příloha 2.10 Obecné nastavení blogu
Zdroj: Autor

Na obrázku výše, lze vidět obecné nastavení. Jedná se o jednu ze stránek z položky nastavení v navigačním menu. Zde je možné nastavit obecné informace pro celý blog,

jako je název blogu, ikona, nastavení článků. Mezi nastavení článků náleží zobrazení počtu hodnocení a počtu zobrazení. Dále je zde možné nastavit jaký datum a čas má být u článků viditelný a v jakém formátu. Při editaci formátu se zobrazí nápověda, která vysvětluje způsob nastavení. Následuje nastavení patičky a jednotlivých stránek.



Obr. 22: Příloha 2.11 Nastavení galerie

Zdroj: Autor

Obrázek výše zobrazuje stránku, která je jako další v navigačním menu součástí položky nastavení. Jedná se o nastavení galerie. Zde je viditelné rozdělení do tří částí. První část poskytuje možnost přidat nové obrázky či fotografie. Další část je určena k odebrání stávajících fotografií. Část poslední je koncipována jako ukázka či zobrazení aktuálních fotografií. Zde je možné otestovat po rozkliknutí obrázku styl zobrazení obrázků.

Admin sekce Přidat článek Články Uživatelé Statistika Nastavení
Přihlášen jako: Admin Uživatelská sekce

Nastavení o tvůrci

Sociální sítě

Nadpis k sociálním sítím:

Místa kde trávím až moc času

Facebook	Instagram
odkaz na facebook	odkaz na instagram
X / Twitter	LinkedIn
<input type="text"/>	<input type="text"/>

Hlavní obsah stránky:

Normal **B** *I* U ~~Q~~ **”** **≡** **≡** **≡** **≡** **≡** **≡** **A**

O autorovi


Vítejte na mé osobní stránce! Jsem Jakub Novák, a rád bych vám trochu přiblížil, kdo jsem a co vás zde může čekat. Rád bych Vás také upozornil, že jsem smyšlený a celý tento obsah byl vytvořen díky ChatGPT. Obsah byl následně jen vzhledově upraven, aby byly viditelné možnosti nastavení.

Kdo jsem?

Jsem váš průvodce a tvůrce obsahu na této stránce. Mé jméno je Jakub Novák, a jsem nadšeným spisovatelem, výzkumníkem a vášnivým průzkumníkem světa kolem sebe. Věřím v sílu slova a jsem přesvědčen, že sdílení zkušeností a příběhů může být inspirativní a obohacující pro všechny zúčastněné.

Co můžete očekávat?

Na této stránce se můžete těšit na různorodý obsah, který odráží mé zájmy, zkušenosti a zamyšlení. Najdete zde články a eseje, které se týkají mého oboru, ale také osobní příběhy, tipy a návody, které se mohou týkat vašeho každodenního života. Mým cílem je nabídnout inspiraci, povzbuzení a možná i trochu zábavy prostřednictvím mého psaní.



Proč tento blog?

Rozhodl jsem se vytvořit tento blog jako prostor pro sdílení svých myšlenek, zkušeností a objevů. Věřím, že každý z nás má svůj jedinečný příběh a perspektivu, kterou může přinést do světa. Tento blog je pro mě platformou, abych mohl sdílet své příběhy a zážitky s vámi a možná vás inspirovat k objevení vašich vlastních příběhů.

Jak mě můžete kontaktovat?

Chci, aby tento blog byl interaktivním prostorem, kde můžeme společně komunikovat a sdílet naše myšlenky. Pokud máte otázky, náměty nebo jednoduše chcete podělit se svými názory, neváhejte mě kontaktovat. Stačí mi napsat e-mail na jakubnovak@example.com nebo mě sledovat na sociálních sítích @jakubnovak. Děkuji, že jste si našli čas k prozkoumání mého blogu. Doufám, že vás můj obsah osloví a že se společně budeme moci podělit o inspirativní příběhy a zážitky. S úctou, Jakub Novák

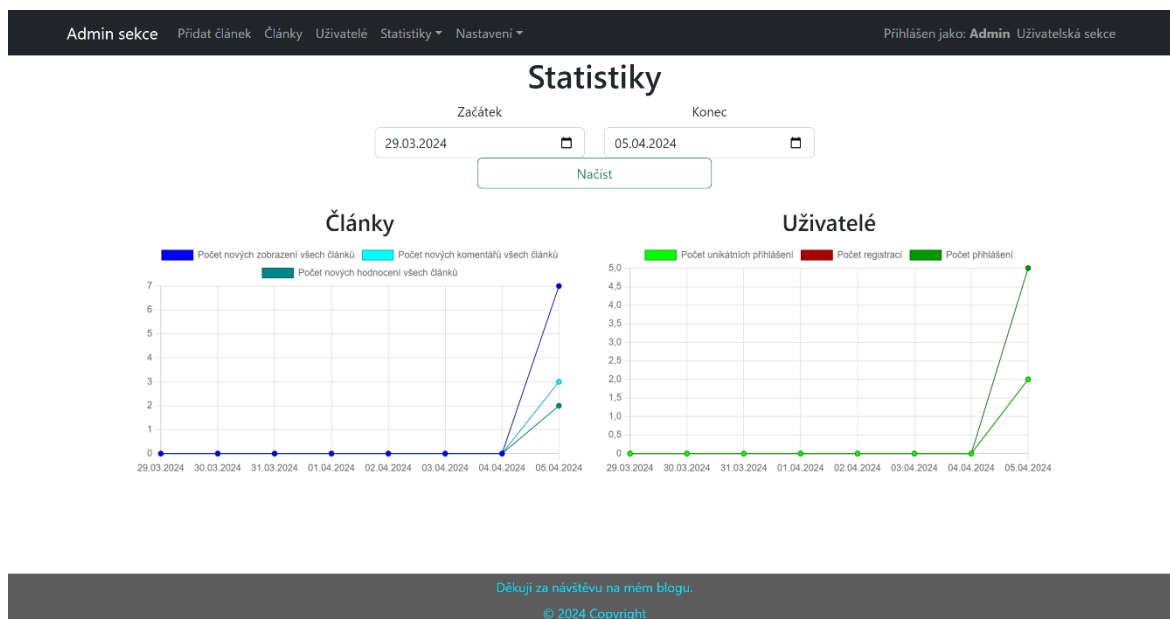
Ukázka

Zobrazení ukázky

Děkuji za návštěvu na mém blogu.
© 2024 Copyright

Obr. 23: Příloha 2.12 Nastavení stránky o autorovi
Zdroj: Autor

Poslední položka v nastavení, které je součástí navigačního menu, je nastavení o tvůrci. Toto nastavení je viditelné na obrázku výše. Zde je viditelné jednoduché nastavení sociálních sítí. Následuje editor stejný jako u článků, skrze ten je možné vytvořit celý obsah stránky. Zároveň je zde možnost zobrazit ukázkou, jak bude výsledná stránka vypadat.



Obr. 24: Příloha 2.13 Zobrazení obecných statistik
Zdroj: Autor

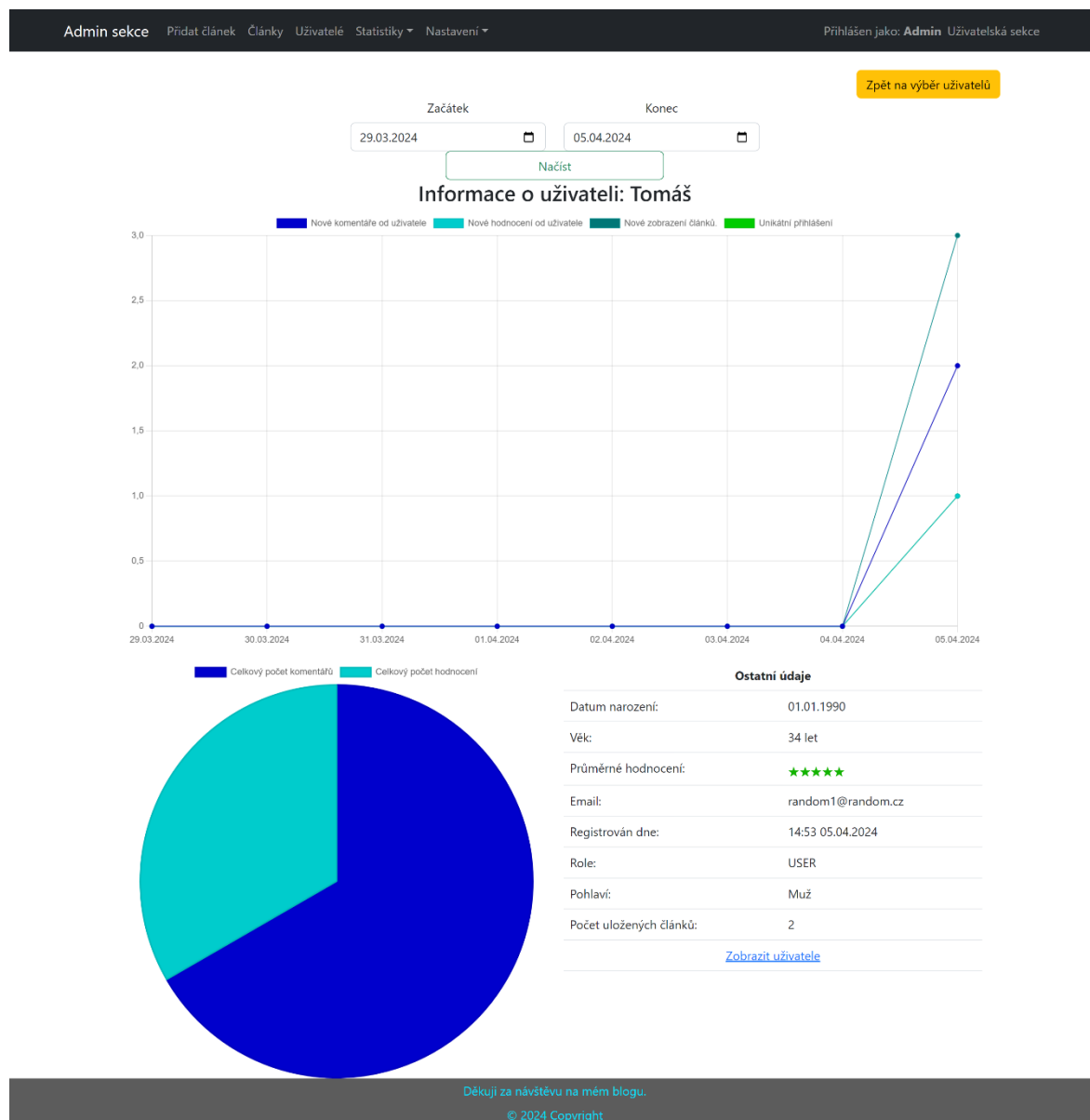
Obrázek výše zobrazuje první stránku ze tří, které jsou součástí položky statistiky v navigačním menu. Jedná se konkrétně o obecné statistiky, Zde je viditelné nastavení časového rozmezí pro grafy a možnost načíst data. Dále je pak viditelné rozdělení na články a uživatele. Graf pro články zobrazuje počty nových údajů. Konkrétně se jedná o nová zobrazení, komentáře a hodnocení. Jedná se o sumu u všech článků. Graf v pravé části, zobrazuje počet unikátních přihlášení, registrací a přihlášení. Jedná se o sumu všech uživatelů za daný den. Unikátní přihlášení znamená, že se přihlášení uživatele počítá maximálně jednou za den. Pokud se tedy uživatel za den přihlásí několikrát, počet přihlášený je vyšší, ale počet unikátních přihlášení je stále jedna. Všechny prvky v grafech je možné vypnout či zapnout dle potřeby. To je možné skrze kliknutí na příslušnou barvu v legendě.



Obr. 25: Příloha 2.14 Zobrazení statistik o článku
Zdroj: Autor

Obrázek výše zobrazuje statistiky o článku. Na statistiky o článku je možné se dostat ze seznamu článků, samotného článku či skrze navigační menu v admin sekci. Jedná se konkrétně o položku statistiky a následně statistika o článku, zde je možné články vyhledat a následně zobrazit tuto stránku pro vybraný článek. Na začátku stránky je viditelné časové rozpětí, zde je možné zadat rozmezí pro zobrazená data. Rozmezí reaguje pouze na první dva grafy, které jsou pod samotným nastavením rozmezí. Tyto dva grafy, které následují zobrazují nalevo celkové údaje a na pravé straně nové údaje. Zobrazené údaje jsou počet komentářů, hodnocení, zobrazení. Lze se tedy přesně podívat na denní přírůstky, či sledovat dlouhodobý růst při pohledu na celek. Následuje graf se zobrazením počtu hodnocení podle hodnoty hodnocení. Další je tabulka, ta je

určena pro zobrazení dalších údajů, které se týkají článku. Pokud článek obsahuje anketu, následuje část ohledně ankety. Zde je viditelný graf, který zobrazuje podíl jednotlivých odpovědí. Vedle grafu je tabulka, která zobrazuje stejné údaje, jen v textové podobě.



Obr. 26: Příloha 2.15 Zobrazení statistik o uživateli
Zdroj: Autor

Obrázek výše zobrazuje statistiky o uživateli. Stránku je možné navštívit přes navigační menu, jedná se o položku statistika o uživateli při otevření položky statistika. Na této stránce je možnost vyhledat daného uživatele. Další způsob je skrze možnosti admina přímo z uživatelského profilu, či ze seznamu uživatelů. Stejně jako u předchozích stránek ohledně statistik, je zde viditelná možnost nastavit rozmezí pro data v grafu.

Tento hlavní graf zobrazuje nové komentáře, hodnocení a zobrazení článků za daný den a počet unikátních přihlášení. Následuje graf, který ukazuje poměr hodnocení s komentáři. Poslední částí je tabulka zobrazující doplňující údaje a data o uživateli.


Technologický blog

- Články
- Kdo jsem?
- Moje fotografie
- Přihlášen jako: Admin
- Uložené články
- Odhlásit
- Admin sekce

Chytré brýle: Okno do budoucnosti?

[Uložit do seznamu](#)

[Možnosti admina ▾](#)



Prozkoumejte svět chytrých brýlí a jejich revoluční potenciál ve světě technologií. Jak mohou změnit způsob, jakým vnímáme informace a interagujeme se světem kolem sebe?

Obr. 27: Příloha 2.16 Ukázka responzivity
Zdroj: Autor

Na výše uvedeném obrázku je viditelná responzivita. Jedná se o pohled na stránku s maximální šířkou 425 pixelů. Na obrázku je viditelné navigační menu a kousek článku. Jedná se pouze o představu ohledně responzivity. Aplikace je responzivní celá.

15 Zadání práce



Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání bakalářské práce

Autor:	Matěj Váňa
Studium:	I2000433
Studijní program:	B1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název bakalářské práce:	Redakční systém pro tvorbu osobních blogů
Název bakalářské práce AJ:	Content management system for creating personal blogs

Cíl, metody, literatura, předpoklady:

Cíl

Cílem práce je navrhnout a implementovat redakční systém zaměřený pro vývoj osobních blogů.

Osnova

1. Úvod
2. Analýza a návrh
3. Implementace
4. Shrnutí výsledků
5. Závěr
6. Literatura

Zadávací pracoviště:	Katedra informatiky a kvantitativních metod, Fakulta informatiky a managementu
Vedoucí práce:	Ing. Jan Krunčík
Datum zadání závěrečné práce:	26.1.2021